COMMUNITY ANALYSIS IN LARGE NETWORKS:

METHODS AND APPLICATIONS

Tanmoy Chakraborty

COMMUNITY ANALYSIS IN LARGE NETWORKS:

METHODS AND APPLICATIONS

Thesis submitted to the
Indian Institute of Technology, Kharagpur
for award of the degree

of

Doctor of Philosophy

by

Tanmoy Chakraborty

Under the supervision of

Dr. Animesh Mukherjee
and
Prof. Niloy Ganguly

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

September 2015

APPROVAL OF THE VIVA-VOCE BOARD

Date:

Certified that the thesis entitled **“Community Analysis in Large Networks: Methods and Applications”** submitted by Tanmoy Chakraborty to the Indian Institute of Technology, Kharagpur, India for the award of the degree of Doctor of Philosophy has been accepted by the external examiners and that the student has successfully defended the thesis in the viva-voce examination held today.

(Prof. Sudeshna Sarkar)
(Chairman, DSC)

(Prof. Pabitra Mitra)
(Member of DSC)

(Prof. Jayanta Mukhopadhyay)
(Member of DSC)

(Prof. Anirban Mukherjee)
(Member of DSC)

(Prof. Rajib Mall)
(HoD, CSE)

(Prof. Animesh Mukherjee)
(Supervisor 1)

(Prof. Niloy Ganguly)
(Supervisor 2)

(Prof. Y. Narahari)
(External Examiner)

CERTIFICATE

This is to certify that the thesis entitled **"Community Analysis in Large Networks: Methods and Applications"***, submitted by* **Tanmoy Chakraborty** *to the Indian Institute of Technology, Kharagpur, for the award of the degree of Doctor of Philosophy, is a record of bona fide research work carried out by him under my supervision and guidance. The thesis, in my opinion, is worthy of consideration for the award of the degree of Doctor of Philosophy of the Institute. To the best of my knowledge, the results embodied in this thesis have not been submitted to any other University or Institute for the award of any other Degree or Diploma.*

Animesh Mukherjee
Assistant Professor
CSE, IIT Kharagpur

Niloy Ganguly
Professor
CSE, IIT Kharagpur

Date:

DECLARATION

I certify that

a. The work contained in this thesis is original and has been done by myself under the general supervision of my supervisors.

b. The work has not been submitted to any other Institute for any degree or diploma.

c. I have followed the guidelines provided by the Institute in writing the thesis.

d. I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

e. Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references.

f. Whenever I have quoted written materials from other sources, I have put them under quotation marks and given due credit to the sources by citing them and giving required details in the references.

Tanmoy Chakraborty

ACKNOWLEDGMENTS

I would like to take the opportunity to thank everybody who inspired me, helped me and contributed directly or indirectly in realizing this thesis. I would wish to thank all my family members who continuously supported me throughout my life. I would like to thank my father who continues to be a great inspiration for me. I would like to thank my mother, who was also my childhood teacher and is always there to stands by my side. My parents did a lot of hard work to support the logistics for my studies and always encouraged me in every decision that I took. I would like to specially thank my uncle, aunt and my little cute brother for their continuous support. The other persons who made a huge contribution and without whom the thesis would not have been possible are my supervisors, Dr. Animesh Mukherjee and Prof. Niloy Ganguly. The long discussions with them not only helped me in culminating to the problems of this thesis, but also helped me to understand the nuances of doing research. I would like to say thank for all their support from the core of my heart. I would also like to thank Dr. Sanjukta Bhowmick (University of Nebraska, Omaha) and Dr. Pawan Goyal (IIT Kgaragpur, India), two of my collaborators, who have not only helped me with necessary technical guidance on my research, but more importantly, have taught me several ways of becoming a better researcher. I express my gratitude to Dr. Ramasuri Narayanam (IBM Research Lab, India) for his enormous support and encouragement during my internship. I am very much thankful to all the co-authors of my different publications who directly contributed in this thesis. I wish to specially convey my gratitude to Suhansanu

Kumar and Sandipan Sikdar who have remained good friends and co-authors to various papers realizing this thesis. I always consider myself fortunate to be a part of the Complex Network Research Group which includes several good friends and excellent researchers. I would wish to extend my gratitude towards all my CNeRG lab mates Dr. Rajib Maiti, Dr. Saptarshi Ghosh, Dr. Rishiraj Saha Roy, Dr. Sudipta Saha, Kaustav Rudra, Parantapa Bhattacharya, Animesh Srivastava, Abir De, Sourav Dandapat, Sankarshan Mridha, Abhijnan Chakraborty, Madhumita Mallick, Souvik Sur, Suman Maity, Soumajit Pramanik and Mayank Singh. I express my sincere thanks to the faculty members and staffs of the department. I would specially like to thank the members of my Doctoral Scrutiny Committee, Prof. S. Sarkar, Dr. P. Mitra, Prof. J. Mukherjee, Dr. A. Mukherjee, and our Head of the Department, Prof. R. Mall for their invaluable suggestions regarding the thesis. I would also like to thank Durga Prasad Sahoo whose company made my life easy at Kharagpur. I was supported by Google India PhD Fellowship Award. I am thankful to several organizations such as Google India, ACM IARCS, Yahoo India and to the organizers of conferences such as ACM SIGKDD, ACM CODS, COMAD for financial assistance for attending the conferences.

I would like to thank all my teachers throughout my life. I would like to specially thank Mr. Dipankar Sannigrahi and Mr. Asish Kar who have been sources of continuous encouragement and confidence for me. Finally, I am grateful to God for his blessings and for giving me the strength to persevere throughout the long and arduous journey towards a Ph.D.

Tanmoy Chakraborty

Kharagpur, India

Author's Biography

Tanmoy Chakraborty has received his B.Tech degree in 2009 from Computer Science and Engineering department, Kalyani Govt. Engg. College (West Bengal University of of Technology) and M.E. degree in 2011 from Jadavpur University, Computer Science and Engineering department. He has been pursuing Ph.D. at the Department of Computer Science and Engineering, IIT Kharagpur, since December 2011. He was awarded the Google India Ph.D. Fellowship in 2012. He is a part of the Complex Network Research Group (CNeRG) at IIT Kharagpur. His primary research interests include Complex Networks, Social Media, Machine Learning, Text Mining and Natural Language Processing.

Publications from the Thesis (Listed in reverse chronological order)

Journals:

1. **T. Chakraborty**, N. Ganguly, A. Mukherjee, S. Bhowmick. "GenPerm: A Unified Framework for Non-overlapping and Overlapping Communities" (communicated to *IEEE TKDE*).

2. **T. Chakraborty**, A. Krishna, M. Singh, N. Ganguly, P. Goyal, A. Mukherjee. "FeRoSA: A Faceted Recommendation System for Scientific Articles" (communicated to *ACM TIST*).

3. **T. Chakraborty**, N. Ganguly, A. Mukherjee, S. Bhowmick. "Permanence and Community Analysis in Complex Networks" (communicated to *ACM TKDD*).

4. **T. Chakraborty**, S. Kumar, P. Goyal, N. Ganguly, A. Mukherjee. "On the Categorization of Scientific Citation Profiles in Computer Sciences", *Communications of the ACM* (*CACM*), 58:9, pp. 82-90, 2015.

5. **T. Chakraborty**, V. Tammana, N. Ganguly, A. Mukherjee. “Understanding and Modeling Diverse Scientific Careers of Researchers”, *Journal of Informetrics* (*JOI*), 9:1, ISSN 1751-1577, pp. 69-78, Jan 2015.

6. **T. Chakraborty**, S. Sikdar, N. Ganguly and A. Mukherjee. “Citation Interactions among Computer Science Fields: A Quantitative Route to the Rise and Fall of scientific Research”, *Social Network Analysis and Mining* (*SNAM*), 4:1, Springer Vienna, ISSN 1869-5450, pp. 1-18, 2014.

7. **T. Chakraborty**, S. Srinivasan, N. Ganguly, S. Bhowmick, A. Mukherjee. “Constant Communities in Complex Networks”, *Nature Scientific Reports*, 3, 1825, ISSN 2045-2322, 2013.

Conferences:

1. **T. Chakraborty**, S. Patranabis, P. Goyal, A. Mukherjee. “On the formation circles in co-authorship networks”. In *21th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Sydney, August 10 - 13, 2015, pp. 109-118.

2. **T. Chakraborty**, N. Modani, R. Narayanam, S. Nagar. “DiSCern: A Diversified Citation Recommendation System for Scientific Queries”, In *31st IEEE International Conference on Data Engineering* (*ICDE*), Seoul, Korea, April 13-17, 2015, pp. 555-566.

3. **T. Chakraborty**, S. Kumar, P. Goyal, N. Ganguly, A. Mukherjee. “Towards a Stratified Learning Approach to Predict Future Citation Counts”, In *ACM/IEEE Digital Libraries* (jointly with *JCDL* and *TPDL*), London, United Kingdom, September 8-12, 2014, pp. 351-360.

4. **T. Chakraborty**, S. Srinivasan, N. Ganguly, A. Mukherjee, S. Bhowmick. “On the Permanence of Vertices in Network Communities”, In *20th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, New York City, August 24 - 27, 2014, pp. 1396-1405.

5. **T. Chakraborty**, N. Ganguly, A. Mukherjee. “Rising Popularity of Interdisciplinary Research - an Analysis of Citation Networks”, *Workshop on Science and Engineering of Social Networks, 6th International Conference on Communication System and Networks* (*COMSNETS*), Bangalore, India, January 10, 2014. **(Best presentation award)**

6. **T. Chakraborty**, S. Kumar, M. D. Reddy, S. Kumar, N. Ganguly, A. Mukherjee. “Automatic Classification and Analysis of Interdisciplinary Fields in Computer Sciences”, In *2013 ASE/IEEE International Conference on Social Computing* (*SocialCom*), Washington D.C., USA, September 8- 14, 2013, pp. 180 - 187.

7. **T. Chakraborty**, S. Sikdar, V. Tammana, N. Ganguly, A. Mukherjee. “Computer Science Fields as Ground-truth Communities: Their Impact, Rise and Fall”, In *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining* (*ASONAM*), Niagara Falls, Canada, August 25-28, 2013, pp. 426-433. **(Best paper nomination)**

ABSTRACT

Real world complex networks such as social networks, biological networks usually exhibit inhomogeneity, resulting in densely interconnected nodes, *communities*, which play an important functional role in the original system. Analyzing such communities in large networks has rapidly become one of the major topics in complex networks. In this thesis, we study four different aspects related to communities: (i) analyzing dependency of existing community detection algorithms on vertex ordering, (ii) quantifying the extent of belongingness of nodes in a community; (iii) unfolding the evolution dynamics of communities in a real-world network; (iv) designing different community-based applications.

While observing the variability in the outputs obtained from community finding algorithms, we notice that some groups of vertices always remain together despite any vertex ordering. We call these groups *constant communities*. We characterize constant communities and show that prior detection of such constant communities improves the performance of a community detection algorithm and reduces the variability of the output.

Then we quantify the membership of a vertex within a community by formulating two vertex-centric metrics: *permanence* (*Perm*) for non-overlapping communities and *overlapping permanence* (*OPerm*) for overlapping communities. We show the effectiveness of these metrics by comparing the results with the ground-truth community structure. We also design two algorithms, *MaxPerm* and *MaxOPerm*, to detect non-overlapping and overlapping communities respectively.

We crawl a massive publication dataset of computer science domain constituting more that 1.5 million scientific articles. We tag each paper by its related research field(s) that act as ground-truth communities. Then we study the temporal interactions of these communities through citations over the last fifty years and unfold the landscape of scientific paradigms. Moreover, we quantify the degree of interdisciplinarity of each field and describe the evolutionary landscape of the interdisciplinary fields over the years.

Finally, we study the citation growth of a paper after publication and discover six distinct categories of citation profile. This observation leads us to adopt *stratified learning* approach in a prediction task, whereby, we propose a two-stage model to predict the future citation count of a paper after a certain time period of its publication. We also design *FeRoSA*, a framework of faceted recommendation for scientific articles that apart from ensuring quality retrieval also efficiently arranges the recommended papers into different facets (categories) that indeed show how these recommendations are related to the query paper.

Keywords: Community analysis, Permanence, Community detection algorithms, Community evolution, Citation networks, Faceted recommendation system

Contents

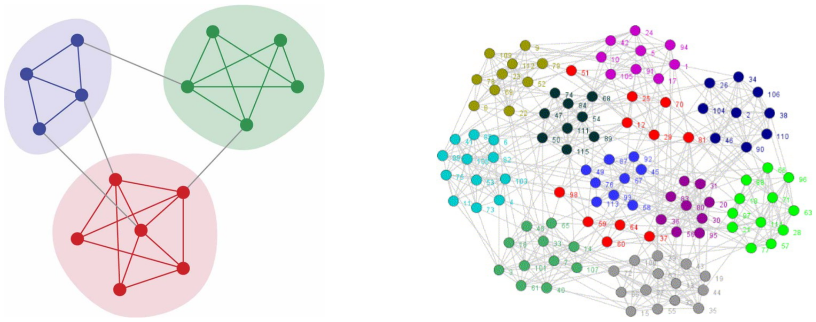
List of Figures

List of Tables

Chapter 1

Introduction

A complex network is a graph-based representation of the interactions amongst entities that take place in the real world. Examples include social networks such as acquaintance networks [6], collaboration networks [157], technological networks such as the Internet [63] and the World Wide Web [4], and biological networks such as neural networks [220], and metabolic networks [107]. Real networks are not random and they usually exhibit *inhomogeneity* [13], indicating the coexistence of order and organization. Furthermore, the distribution of links also shows inhomogeneity, both globally and locally, describing the phenomenon that nodes naturally cluster into groups and links are more likely to connect nodes within the same group. This phenomenon tells us that the organization of such complex network is modular. Network scientists call this organization as the *community structure* of networks. Though there is a lack of consensus in the definition of communities, most popular and well-accepted definition suggests that: communities are the subsets of vertices within which vertex-vertex connections are dense, but between which connections are less dense [78]. A figurative sketch and a real-world community structure are shown in Figure 1.1. Analysis of such communities is essential to understand the structural and the functional organizations of the network.



(a) (b)

Figure 1.1: (a) A schematic representation of a network with community structure. (b) Real-world community structure of American collage football team network [78]. The communities are represented by different colors.

1.1 Major Challenges

Detecting communities is of prime importance in sociology, biology and computer science disciplines where systems are often represented as graphs. This problem is very hard and not yet satisfactorily solved, despite the huge effort of a large interdisciplinary community of scientists working on it over the past one and half decades (see [66] for the reviews). Besides this, several other challenges have been encountered during the analysis of community structure in large networks, some of which are as follows:

- Most community detection algorithms are based on optimizing a combinatorial parameter (for example, modularity [26, 163]). This optimization is generally non-deterministic [31], thus merely changing the vertex order can alter the vertex-to-community assignments. Therefore, a crucial question about the variance of results in community assignment remains unanswered – what does the invariance of the results tell us about the network structure?
- The goodness of community detection algorithms (see [69] for a review) is often objectively measured according to how well they achieve the optimization. Modularity [163] is a widely accepted metric for measuring the quality of community structure identified by various community detection algorithms. However, a growing

body of research have begun to explore the limitations of maximizing modularity for community identification and evaluation; three such limitations include – resolution limit [81], degeneracy of solutions and asymptotic growth of the modularity value. Therefore, a new goodness measurement metric needs to be formulated that can overcome (or minimize) such limitations.

- Due to the limitations of the goodness measures (such as modularity) described above, researchers often rely on manual inspection in order to evaluate the detected communities. For each detected community an effort is made to interpret it as a "real" community by identifying a common property or external attribute shared by all the members of the community. Such anecdotal evaluation procedures require extensive manual effort; therefore these are non-comprehensive and are limited to small networks. Therefore, a possible solution would be to find a reliable definition of explicitly labeled ground-truth communities.

- Although there is a large volume of research on community detection, systematic post-hoc analysis of the communities, which can unfold interesting characteristic properties of various real systems, is missing in the literature. For instance, temporal community interactions on a longitudinal scale (i.e, with the progress of time) often unveil the opportunity to analyze the rise and fall of dominant clusters in different time points. This analysis might be helpful in detecting the trending topics in Twitter, identifying major research fields in different scientific domains, information diffusion among scientific communities [199] etc.

Given this scenario, it is clear that we need to develop a better understanding of community structure in various types of large networks. The goal of our research is to study different aspects of community analysis in complex networks that mainly focus on two major directions – (i) identification of realistic communities in different large networks and (ii) leveraging such community structure for developing various applications.

1.2 Objectives

To deal with all the challenges mentioned above, we identify four major issues mentioned below that contribute to different chapters of the thesis.

(i) Investigating the dependence of community detection algorithms on vertex ordering: Here we intend to study the variation of results produced by the algorithms due to different vertex orderings. Moreover, we posit that despite any vertex ordering, there exist some invariant groups in each network whose constituent vertices always remain together. In particular, we ask the following questions – what does the invariance of the results tell us about the network structure? what is the significance of these invariant substructures in a network? how are they related with the actual community structure of a network?

(ii) Formulating a new metric for community analysis: Most of the community scoring functions are global, thus do not imply anything about the vertices of a network. We believe that the individual constituent vertices in a community do not belong to the community with equal strength. Further, there is a lack of a proper quantitative indicator that would entail the true modular structure of a network. For instance, the highest modularity in the Jazz network is 0.45 and that of the Western USA power grid is 0.98 [156]. However, it has been observed that Jazz has a much stronger community structure than the power grid [156]. Therefore, formulation of a vertex-centric measure for community analysis that correctly indicates the presence of community structure in a network is needed.

Here we intend to ask few fundamental questions pertaining to the community analysis of a network – is the membership of vertices in a community homogeneous (which has been the common consensus so far)? do we need to check the eligibility of a network for community detection prior to running the community detection algorithm? can one formulate a metric that suitably reduces the limitations of the existing metrics for community detection?

(iii) Analyzing real-world community structure: Several works on detecting and tracking communities in a temporal environment have been conducted [69]. However, the interactive patterns of detected communities over a temporal scale still remain unexplored mainly due to the lack of standard ground-truth community structure of a network. The availability of ground-truth communities allows to explore a range of interesting characteristics of a time-varying systems. For example, deep understanding of the connectivity structure in and across ground-truth communities could lead to realistic community detection methods. Here, we focus on a typical real network, *citation network*, whose nodes correspond to scientific articles and links correspond to the citations from citing papers to cited papers. We aim at investigating different aspects of this network such as – how do the communities form in this network? what do the topological features of citation network tell us? what can we learn from them? what kind of trends are observed over-time in these networks? how often do authors publish and collaborate?

(iv) Developing community based applications: Once the community structure is detected from a network, an immediate question might arise that how this information can help us in building real applications. Citation profiles over time can be shown to group in different communities, which can be further used to develop more accurate citation prediction models. Further, it is possible to arrange citations into semantic communities which can facilitate developing a full-fledged faceted recommendation system of scientific articles.

1.3 Constant Communities in Networks

An automatic way of detecting the communities from networks has attracted much attention in recent years and many community detection algorithms have been proposed. Most of these algorithms are based on the maximization of a quality function known as modularity, which measures difference between the fraction of edges in the network that connect vertices of the same type (within community type) and the expected value of the same quantity in a network with the similar community divisions but random connection between vertices (see Section 2.1.1). Modularity maximization is an NP-hard problem [31], and most algorithms use heuristics. For several reasons related to the modularity, as well as the non-determinism of the algorithms or randomness in initial configuration, such algorithms often produce different partitions of similar quality, and there is no reason to prefer one above another. Besides, such approaches may produce communities with a high modularity in networks which have no community structure, e.g., random networks. This is related to the instability of algorithms: small perturbations of the input graph can significantly influence the output.

Here, we investigate the effect of input ordering on two non-deterministic agglomerative methods for modularity maximization – (i) CNM algorithm [48] and (ii) Louvain method [26]. Both these methods are based on combining appropriate pairs of vertices to increase modularity. Based on these results, we posit that the permutation of the vertices is a key point for obtaining high modularity. A bad permutation can lead to sub-optimal combination of vertex pairs that in turn can affect the communities obtained. The notion of stability is governed by the inherent compartmental structure of the nodes in a network. Our intuition is based on the fact that some vertices always persist within same communities despite any combinatorial ordering of input edge sequence. Those vertices may have some intrinsic connectivity property that forces them not to share other communities under any circumstance. We call such groups of vertices as *constant communities* and the constituent vertices as *constant vertices*. We observe that if these constant vertices are grouped together in the pre-processing step, it significantly improves the accuracy of hierarchical clustering technique by increasing the modularity. We further analyze the properties of constant communities in order to identify the characteristic that keep them together independent of the order of the vertices in which the community detection algorithm is fed in. In particular, we observe that constant vertices experience minimum "pull" from external nodes in the network. Further, we present a case study on phoneme network and illustrate that constant communities, quite strikingly, form the core functional units of the larger communities.

1.4 Permanence and Network Communities

Community detection algorithms primarily deal with identifying densely-connected units from within large networks. So far, the common consensus in the analysis of the community structure is that the community membership is homogeneous, i.e., each node belongs to one or more communities with equal extent. Therefore, less attention has been paid in analyzing individual vertices in a community, and a community is mostly considered as a whole. Here we argue that the community membership of vertices is *heterogeneous*; where few vertices have more involvement into the community and others have less. To quantify the membership of a vertex, we need a proper local vertex-based metric. Modularity is a widely accepted global metric for measuring the quality of community structure identified by various community detection algorithms. However, a growing body of research have begun to explore the limitations of maximizing modularity for community identification and evaluation; three such limitations include – resolution limit [67], degeneracy of solutions [81] and asymptotic growth of the modularity value [81].

To address these issues, we here propose a novel vertex-level metric called *permanence* (Perm) for analyzing disjoint communities which is built on the notion of relative pull experienced by a vertex from its neighbors that lie external to its own community. The value of permanence indicates the extent to which a vertex belongs to a community. We show that this metric as compared to other standard measures, namely modularity, conductance and cut-ratio qualifies as a better community scoring function for evaluating the detected community structures from both synthetic and real-world networks. We demonstrate that the process of maximizing permanence produces communities that concur with the ground-truth structure of the networks more accurately than the modularity based and other approaches. Finally, we show that maximizing permanence (named as MaxPerm) can effectively reduce the limitations associated with modularity maximization as well as can indirectly help in inferring the community quality of a network.

Further, we formulate a generalized version of this metric called *overlapping permanence* (abbreviated as OPerm) that, although is developed for overlapping community, translates to the non-overlapping case under special boundary conditions. Note that this is one of the rarest formulations which can be useful for both non-overlapping and overlapping community analysis. Since every vertex gets scored by this metric, it can be used to rank the vertices within a community as well as can give an overview of the belongingness of nodes in the community. Detailed experimentation demonstrates OPerm's superiority over other state-of-the-art scoring metrics in terms of performance as well as its resilience to minor perturbations. We also present an algorithm, MaxOPerm, to detect communities based on maximizing OPerm. Over a test suite of synthetic and six large real-world networks we show that MaxOPerm outperforms six state-of-the-art algorithms in terms of accurately predicting the ground-truth labels. We also demonstrate that MaxOPerm is resistant to degeneracy of solution. Further, we introduce the resolution limit problem in the context of overlapping communities and show that an algorithm which can maximize OPerm can effectively tackle the problem.

1.5 Analyzing Ground-truth Communities

Most of the existing works on community analysis have concentrated on developing and improving the algorithms for discovering communities. Evaluating the performance of such algorithms is incomplete without comparing the detected output with the actual ground-truth community structure of the network under investigation. However, such ground-truth community structure is limited in number. Moreover, availability of such community structure of a labeled network would unveil the opportunity to investigate its characteristics and functionality thoroughly. To this purpose, we particularly focus on a scientific network, called *citation network*, whose nodes indicate scientific articles and links correspond to the citations. We gather all the papers in computer science domain published in the last fifty years and indexed by Microsoft Academic Search[1]. Each paper comes along with various bibliographic information – the title of the paper, a unique index number, its author(s) etc. Each individual community in a citation network is naturally defined by a research field – i.e., acting as ground-truth. Then we study the interactions among these communities through citations in real time which unfold the landscape of dynamic research trends in the computer science domain over the last fifty years. We quantify the interaction in terms of a metric called *inwardness* that captures the effect of local citations to express the degree of *authoritativeness* of a community (research field) at a particular time instance. Several arguments to unfold the reasons behind the temporal changes of inwardness of different communities are put forward using exhaustive statistical analysis. The measurements (importance of field) are compared with the project funding statistics of NSF and it is found that the two agree to a considerable extent.

As a second step we quantify the interdisciplinarity of a research field through four indicative measures. Three of the indicators, namely *Reference Diversity Index* (*RDI*), *Citation Diversity Index* (*CDI*) and *Membership Diversity Index* (*MDI*) are directly related to the topological structure of the citation network. The last feature called the *Attraction Index* of a field is based on the propensity of the new researchers to start research in a particular field. Further, to check the significance of these features in characterizing interdisciplinarity, we rank the fields based on the value of each of the features separately. Next, we propose an unsupervised classification model that can efficiently cluster the core

[1] http://academic.research.microsoft.com/

and the interdisciplinary fields based on the similarity of the feature sets mentioned above. To understand the evolutionary landscape of a core field vis-a-vis an interdisciplinary field, we conduct a case study on one popularly accepted interdisciplinary field (WWW) and one core field (Programming Languages). The results attest to the conclusion that the interdisciplinarity occurs through cross-fertilization of ideas between the fields that otherwise have little overlap as they are studied independently. The conclusion that popularity of the interdisciplinary research now-a-days overshadows the core fields is strengthened on analyzing the core-periphery organization of the citation network at different time periods. We observe that the core region of a domain is gradually dominated by the more applied fields with interdisciplinary fields steadily accelerating towards the core.

The rich citation dataset further allows us to conduct an author-centric analysis. In particular, we analyze the diverse scientific careers of researchers in order to understand the key factors that could lead to a successful career. Essentially, we intend to answer some specific questions pertaining to a researcher's scientific career – what are the local and the global dynamics regulating a researcher's decision to select a new field of research at different points of her entire career? what are the suitable quantitative indicators to measure the diversity of a researcher's scientific career? We propose two entropy-based metrics to measure a researcher's choice of research topics. Experiments with large computer science bibliographic dataset reveal that there is a strong correlation between the diversity of the career of a researcher and her success in scientific research in terms of the number of citations. We observe that while most of the researchers are biased toward either adopting diverse research fields or concentrating on very few fields, a majority of the highly cited researchers tend to follow a typical "scatter-gather" policy – although their entire careers are immensely diverse with different types of fields selected at different time periods, they remain focused primarily in at most one or two fields at any particular time point of their career.

1.6 Community-based Applications

The group of homogeneous entities can be useful in several applications. Here we particularly focus on two major applications that are built on the citation networks and publication datasets. Prior to that, we study another important aspect of a scientific article, its growth of citation counts over time after the publication. A common consensus in the literature is that the citation profile of published articles in general follows a universal pattern – an initial growth in the number of citations within the first two to three years after publication followed by a steady peak of one to two years and then a final decline over the rest of the lifetime of the article. This observation has long been the underlying heuristic in determining major bibliometric factors such as the quality of a publication, the growth of scientific communities, impact factor of publication venues etc. We study the citation network once again and notice that the citation count of the articles over the years follows a remarkably diverse set of patterns – a profile with an initial peak (PeakInit), with distinct multiple peaks (PeakMul), that exhibits a peak late in time (PeakLate), that is monotonically decreasing (MonDec), that is monotonically increasing (MonIncr) and that cannot be categorized into any of the above (Oth)). The papers following same citation profile are assumed to form separate community. We systematically investigate the important characteristics of each of these categories.

Then we leverage this category information in order to develop a prediction model that predicts future citation count of a scientific article after a given time interval of its publication. We propose to categorize the complete set of data samples into different subparts each of which corresponds to one type of citation pattern mentioned earlier. This approach is commonly termed as *stratified learning* in the literature where the members of the stratified space are divided into homogeneous subgroups (aka strata) before sampling. We develop a *two-stage prediction model* – in the first stage, a query paper is mapped into one of the strata using a Support Vector Machine (SVM) approach that learns from a bunch of features related to the author, the venue of the publication and the content of the paper; in the second stage, only those papers corresponding to the strata of the query paper are used to train a Support Vector Regression (SVR) module to predict the future citation count of the query paper. For the same set of features available at the time of publication, the two-stage prediction model remarkably outperforms (to the extent of 50% overall improvement) the well-known baseline model. Our two-stage prediction model produces significantly better accuracy in predicting the future citation count of the highly-cited papers that might serve as an useful tool in early prediction of the seminal papers that are going to be popular in the near future. We also show that including the first few years of citations of the paper into the feature set can significantly improve the prediction accuracy especially in the long term.

Finally, we arrange citations into semantic communities based on the relation of a cited paper with the citing paper. We use this grouping to propose for the first time a framework of faceted recommendation for scientific articles, *FeRoSA* which apart from ensuring quality retrieval of scientific articles for a particular query paper, also efficiently arranges the recommended papers into different facets (categories). Our methodology is based on a principled framework of random walks where both the citation links and the content information are systematically taken into account in recommending the relevant results. First, citation links are categorized into four classes/facets, namely Background, Alternative Approaches, Methods and Comparison. Following this, for a particular query paper, we collect an initial pool of papers containing nearest citation-based neighborhoods and papers having high content-similarity with the query paper, and make an induced graph individually for each facet. Next, a random walk with restarts is performed from the query paper on each of the induced subgraphs and a ranked list of papers is obtained. We further prepare another ranked list of papers based on the content similarity. The final ranking is obtained in a principled way by combining multiple ranked lists. Our method is easy to implement and has very elegant and principled way of retrieving the relevant results irrespective of the choice of the facets. Human experts are asked to judge the recommendations of the competing systems. Experimental results show that our system outperforms the baseline systems with respect to different standard measures which are used to evaluate a recommendation system. In terms of overall precision, FeRoSA achieves an improvement of 29.5% compared to the best competing system. We also evaluate and compare the results separately for different facets (average overall precision of 0.65) and model parameters to have a thorough understanding of the performance of the system.

1.7 Contributions

In this thesis, we consider *community analysis in complex network* as a prime objective, which has been one of the active research topic for quite some time in different branches of science including computer science, physics, mathematics and biology. Despite a large volume of research in this area, few fundamental problems have remained unanswered or have not been solved satisfactorily. Here we attempt to analyze such problems. Moreover, we focus on *citation network* and study different structural and functional aspects of this network. Finally, we design two applications based on the publication dataset which leverage the community information of the underlying network. A brief report (which we shall elaborate in the forthcoming chapters) on these studies and the results obtained thereby, are presented below.

1.7.1 Constant Communities in Networks

Although enormous effort has been devoted to design efficient community detection algorithms, most of these algorithms follow a general framework – these algorithms try to optimize certain objective functions (such as modularity) by grouping vertices, which results in the partitioning of the vertices in the network. However, most of these algorithms are highly dependent on the ordering in which the vertices are processed as a result of which the algorithms produce different outputs in different iterations for a particular network. An exhaustive study of this phenomenon reveals the following interesting results:

(a) We conduct this experiment on a set of scale-free networks and observe that while the vertex orderings produce very different set of communities, some groups of vertices are always allocated to the same community for all different orderings. We define the group of vertices that remain invariant as *constant community* and the vertices that are part of the constant communities as *constant vertices*.

(b) Although constant communities are detected using the outputs obtained from certain community detection algorithms, we notice that these groups are the invariant part of a network, irrespective of the heuristic being used to detect the communities.

(c) Another issue that has not been studied earlier is whether a network at all contains community structure or not. For instance, a random network or a grid network does not have strong community structure as compared to the ring of cliques. Therefore, we propose a metric, called *sensitivity* (based on the number of constant communities within a network) which efficiently demonstrates how community-like a network is. Later in Chapter 4, we use this metric to measure the *degeneracy of solutions* of an algorithm, which although has been studied several times, is quantified here for the first time.

(d) Constant communities are quite different from the actual community structure of a network. For instance, constant communities do not always have more internal connections than external connections. Rather, the strength of the community is determined by the number of different external communities to which it is connected. Therefore, we characterize constant vertices by a metric called *relative pull*, which indicates that the constant vertices do not experience a significant "pull" from any of the external communities that will cause them not to migrate, and, therefore, their propensity to remain within their own communities is high.

(e) Further, we show that if these constant communities are identified prior to any community detection, and each constant community is combined into a super-vertex, it not only increases the efficiency of any community detection algorithm, but also reduces the variability of the final output.

(f) Finally, we conduct a case study on a specific type of labeled linguistic network constructed from the speech sound inventories of the world's languages. We discover constant communities from this network and observe that each such graph represents a natural class, i.e., a set of consonants that have a large overlap of the features. Such groups are frequently found to appear together across languages.

1.7.2 Permanence and Network Communities

Motivated by the earlier study on constant communities, we further investigate the community structure of real-world networks. Since many real-world communities are based on subjective measurements (as opposed to a formal definition), often the optimum value of the parameters are successful in identifying only a fraction of the "ground-truth" communities. Moreover, as observed in the phenomena of resolution limit [67] and degeneracy of solutions [81], the optimum parameter value sometimes produces intuitively incorrect solutions in ideal networks. As a response to these issues, new metrics are being regularly proposed [96, 231], that either produce more accurate results on a certain subclass of networks and/or can address some of these inherent problems.

Despite the on-going research in this area, an important question is whether it is always reasonable to assign every individual vertex to a community. Not all networks possess community structures of equal strength. For example, a network composed of several sparsely connected dense cliques will have strong communities whereas a grid will not have any community structure at all, and between these two extremes there exist communities of different strength as per the network structure. As of now, there is no community detection metric that can measure to what extent a vertex is a part of a community. One of the reasons for this deficiency is that the optimum value of the parameters such as modularity is not exactly related to whether the network possesses a strong community, but rather tries to identify the best community assignment, for any given network. Indeed, most algorithms output a set of communities regardless of whether the network (such as a grid) possesses a community structure or not. A corollary to this problem is that given a suboptimal answer we cannot estimate how close we are to the correct result and in the absence of ground-truth community structure, we cannot even judge whether the obtained answer is reliable or not. These are serious limitations for a field that regularly encounters new applications and datasets.

Here, we attempt to address some of the above issues by introducing *permanence* ($Perm$), which is a metric that measures the propensity of a vertex in its assigned community. The values range from 1 (the vertex is perfectly assigned to the community) to -1 (the vertex is absolutely incorrectly assigned). The values of permanence provide an estimate of how much a vertex belongs to its assigned community and the extent to which it is "pulled" by the neighboring communities. For example, if the permanence is zero, this indicates that the vertex is pulled equally by all its neighboring communities. In these cases, it might be better to assign the vertices to a *singleton* community (i.e., community containing only one vertex), rather than assigning it to one of the (larger size) neighboring communities. Similarly, we propose a generalized metric, called *overlapping permanence* (abbreviated as $OPerm$) which although is developed for overlapping community, can self-tune itself for the non-overlapping case.

The sum of the Perm (OPerm) of all vertices, normalized by the number of vertices, provides the overall Perm (overall OPerm) of the network. These values indicate to what extent, on an average, the vertices of a network are in their correct communities. This approach of combining the microscopic (vertex-level) information to obtain the mesoscopic (community-level) information provides a more fine-grained view of the modular structure of the network. Specifically, Perm (OPerm) of a graph produces high values only if the network possesses an inherent community structure across most of its vertices. As the community structure of the networks degrades, so does the value of Perm (OPerm) of the entire network. As we shall demonstrate in Chapter 4, the principal benefits of our approach are:

(a) The proposed metrics are found to be remarkably suitable for evaluating the goodness of the community structure obtained from different community identification algorithms.

(b) Perm (OPerm) is appropriately sensitive to the different perturbations of the network which should be an ideal property of a community scoring metric.

(c) OPerm is one of the rare formulations which can self-tune itself for both non-overlapping and overlapping communities depending on the network structure.

(d) OPerm provides a deep understanding of how vertices are organized within a community; specifically, OPerm values follow a Gaussian distribution and the medium-valued vertices are maximally overlapped and have the highest degree.

(e) We identify a precise rank order among the vertices within a community by arranging them into a core-periphery structure based on OPerm; this rank order can be further used as an input for various other applications (e.g., initiator selections in message spreading).

(f) Maximizing Perm (maximizing OPerm) is more successful in finding ground-truth communities as compared to state-of-the-art algorithms.

(g) Community detection using maximizing Perm (maximizing OPerm) can overcome the problem related to resolution limit, degeneracy of solutions, in many networks. Moreover, the value of Perm (OPerm) is relatively independent of the size of the network.

1.7.3 Analyzing Ground-truth Communities

Even though modeling network communities is a fundamental problem, our understanding of networks at the level of these communities has been relatively less. Moreover, the lack of reliable ground-truth makes the evaluation of such models extremely difficult. Here we study the connectivity structure of ground-truth communities of a real network, citation network of computer science domain whose nodes correspond to the scientific articles and links correspond to the citations. Our work is based on a large scale citation network where we can reliably define the notion of ground-truth communities. In this network, each paper (node) is marked by its relevant research field; thus citation interactions among papers within a same research field are relatively higher than across fields. These fields therefore act as ground-truth communities in the network. The availability of the reliable ground-truth communities has a profound effect, such as it allows us to understand the connectivity structure of the ground-truth communities and the interaction among these communities that has the potential to portray a significantly better picture of the underlying systems.

(a) To start with, we first study the temporal interaction among communities in citation network by defining a metric called "authoritativeness" which measures the impact of a community in a particular time period. These patterns of interaction, when analyzed carefully, reveal various interesting elements that are either directly or indirectly related to the overall decline in the interest in a field followed by the rise of interest in another. One of the most striking observations is that in almost all cases, the field constituting the current "hottest" area of research within the domain is overtaken in the immediate future by its strongest competitor.

(b) We further investigate the cause of such focus shifts from different and possibly orthogonal directions and observe that (a) the density of high impact publications within a field plays a pivotal role in pulling as well as sustaining the field at the forefront, (b) certain fields produce a huge number of citations (i.e., act as hubs) for a particular field and, thereby, push it to the forefront; an abrupt fall in the number of such received citations, in many cases, triggers the decline of the field currently at the forefront, (c) inception of seminal papers in a field might trigger the emergence of a field at the forefront, and (d) the extent of team work (both within and across continents) in the form of joint publications seem to significantly contribute to the shape of the evolutionary landscape.

(c) A careful analysis of the funding trends by NSF (National Science Foundation of the United States of America) shows that our results correlate very well with the number of proposals submitted in each field while they correlate moderately well with the actual funding decisions.

A common consensus among researchers is that interdisciplinarity is one of the key factors in doing research at current times. However, a pertinent question deals with identifying appropriate indicators of interdisciplinarity. Using a set of citation based indicators, here we investigate the evolution of the extent of interdisciplinary research in computer science. For this, we study the citation network from different orthogonal directions, namely citation and reference patterns of a paper, overlapping membership of the papers in different research communities, inclination of the researchers to adopt new fields, and propose several indices to quantify the degree of interdisciplinarity of a field. The new indices of interdisciplinarity corroborate with the hypothesis that the emergence of interdisciplinarity occurs through cross-fertilization of ideas between the sub-fields that otherwise have little overlap as they are studied independently. At the end, we analyze the core-periphery organization of citation networks and arrive to the conclusion that with the advancement of interdisciplinary research, the core part of the network is also changing from theoretical towards more applied fields of research. Some of our observations are as follows.

(a) The practice of interdisciplinarity in citations occurs mainly between related scientific communities, and this phenomenon has been witnessed to tremendously increase over the last few years.

(b) Few fields such as Data Mining, WWW, Natural Language Processing, Computational Biology, Computer Vision, Computer Education provide clear indications of interdisciplinarity in terms of all the metrics proposed here.

(c) Core-periphery analysis on the citation network shows that the interdisciplinary fields are accelerating steadily toward the core of computer science domain.

(d) For already very interdisciplinary fields, such as Data Mining, the indicators may have a certain "saturation" effect forcing it towards the core region of the computer science domain.

Finally, we conduct an author-level analysis where we particularly investigate the research field adaptation process of a researcher in order to understand the key factors that could lead to a successful career. To start with, we quantify the diversity of a scientific career by proposing two entropy-based measures. Then several analyses are conducted to understand the career of researchers. Major contributions here are as follows.

(a) The average behavior indicates that a researcher tends to adopt few research fields in her entire research career, and she seems to prefer to work simultaneously on all of them together.

(b) A highly-cited researcher tends to work in many fields over her entire career but remains confined to one or few fields in each time window. However, the number of such researchers is very less in our dataset.

(c) The researchers who have tried various fields in the entire career as well and in each successive time period, get low citations.

1.7.4 Community-based Applications

Once the community structure of a network is detected, a natural question would be as to how can we use this information in designing real systems. We use publication dataset, citation network and the community structure, and design two applications – future citation count prediction of a paper after publication and faceted recommendation system for scientific articles. The major contributions from this study are mentioned below.

1. We first start analyzing the citation profile of the papers and reveal six different patterns – a profile with an initial peak (PeakInit), with distinct multiple peaks (PeakMul), that exhibits a peak late in time (PeakLate), that is monotonically decreasing (MonDec), that is monotonically increasing (MonIncr) and that can not be categorized into any of the above (Oth)).

2. While analyzing the characteristic of these categories, we observe that most of the papers in PeakInit (64.35%) and MonDec (60.73%) categories are published in conferences, whereas papers belonging to PeakLate (60.11%) and MonIncr (74.74%) categories are mostly published in journals. Hence, if a publication starts receiving greater attention or citations at a later part of its lifetime, it is more likely to be published in a journal and vice versa.

3. We observe that papers in MonDec are vastly affected by the self-citation phenomenon, i.e., around 35% of papers in MonDec would have been in the 'Oth'

category had it not been due to the self-citations. The result also agrees with the observation that MonIncr category is least affected by self-citations, followed by PeakLate, PeakMul and PeakInit in that order.

4. We study the stability of each category by analyzing the migration of papers from one category to others over time. We observe that apart from the Oth category, MonDec seems to be the most stable, which is followed by PeakInit. However, papers which are assumed to fall in Oth category quite often turn out to be MonIncr papers in the later time periods.

5. We analyze the core-periphery organization of the citation network and observe that PeakMul category gradually leaves the peripheral region over time and mostly occupies the innermost shells. PeakInit and MonDec show almost similar behavior with a major proportion of papers in inner cores in the initial year but gradually shifting towards peripheral regions. On the other hand, MonIncr and PeakLate show expected behavior with their proportion increasing in the inner shells over time indicating their rising relevance as time progresses.

6. Our proposed framework for future citation count prediction incorporates a stratified learning approach in the traditional framework which in turn remarkably enhances the overall performance of the prediction model.

7. Our two-stage model produces significantly better accuracy in predicting the future citation count of the highly-cited papers that might serve as an useful tool in early prediction of the seminal papers that are going to be popular in the near future.

8. The faceted recommendation system, FeRoSA is primarily built on the semantic annotation of citations in citation network. While evaluating the system based on expert judgment, FeRoSA achieves an overall precision (OP) of 0.65, 29.5% higher than the next best system. Thus, the recommendations generated by our framework are found to be of high quality even if the method is very simple to implement.

9. FeRoSA also achieves a reasonably high precision for the query papers with low citations (OP of 0.57 with the next best system having an OP of 0.46).

1.8 Organization of the Thesis

The thesis is organized into seven chapters.

Chapter 2 presents a detailed literature survey on the state-of-the-art in community analysis for different networks and their usage in different applications.

Chapter 3 centers around our first objective of constant communities in complex networks. We detect constant communities in a brute-force manner and study their structural properties. We show that identifying constant communities prior to any community detection enhances the performance of any community detection algorithms.

Chapter 4 investigates in detail our second objective, i.e., formulation of permanence and overlapping permanence for community analysis. We further develop two community detection algorithms using these metrics.

Chapter 5 explains our third objective of analyzing the ground-truth community structure of citation network. We study three subproblems pertaining to citation network. First, we unfold the rise and fall of scientific research in computer science domain over last fifty years. Second, we propose four metrics to quantify the degree of interdisciplinarity of a research field. Third, we study the field adoption process of a researcher over her entire research career.

Chapter 6 presents our final objective of designing different community-based applications. In particular, we design two systems: (i) future citation count prediction of a scientific article after publication, and (ii) a faceted paper recommendation system for scientific articles.

Chapter 7 concludes the thesis by summarizing the contributions and pointing to a few topics of future research that have opened up from this work.

Chapter 2

Related Work

In this chapter, we discuss relevant studies related to the objectives of this thesis. Particularly, the literature review is conducted in two broad directions: first, we shall describe the metrics and methods used in community detection, and second, we shall elaborate the analysis of community structure and its usage in various applications.

2.1 Survey on Community Detection and Evaluation

In this section, we survey the current literature on the community identification problem and other closely related problems. First, we review the work on identifying non-overlapping and overlapping communities in different networks. Following this, we present various metrics used to evaluate the community structures.

2.1.1 Non-overlapping Community Detection

A wide spectrum of community detection methods have been proposed to detect disjoint communities from static networks. Interested readers are encouraged to read the following survey papers: Fortunato [66], Lancichinetti, Fortunato [121], Harenberg et al. [90]. All these algorithms can be roughly divided into the following categories.

Traditional Methods

(i) Graph partitioning: The problem of graph partitioning consists of dividing the vertices in different groups of predefined size, such that the number of edges lying between the groups is minimal. The number of edges running between clusters is called *cut size*. There are several algorithms that can do a good job, even if their solutions are not necessarily optimal [109, 181]. Another popular technique is the spectral bisection method [15], which is based on the properties of the spectrum of the Laplacian matrix. Graphs can be also partitioned by minimizing measures that are affine to the cut size, like *conductance* [29], *ratio cut* [221] and *normalized cut* [198]. Algorithms for graph partitioning are not good for community detection, because it is necessary to provide as input the number of groups and in some cases even their sizes, about which in principle has no prior information.

(ii) Hierarchical clustering: Most of the real-world graphs have a hierarchical structure, i.e., display several levels of grouping of the vertices, with small clusters included within large clusters, which are in turn included in larger clusters, and so on. In such cases, one may use hierarchical clustering algorithms [93], i.e. clustering techniques that reveal the multilevel structure of the graph. Hierarchical clustering techniques can be classified in two categories: Agglomerative (bottom-up) and Divisive (top-down) algorithms. Hierarchical clustering has the advantage that it does not require a prior knowledge of the number and size of the clusters. However, it does not provide a way to discriminate between many partitions obtained by the procedure, and to choose that or those partitions which better represent the community structure of the graph. The results of the method depend on the specific similarity measure adopted. The procedure also yields a hierarchical structure by construction, which is rather artificial in most cases, since the graph at hand may not have a hierarchical structure at all [160].

(iii) Partitional clustering: Partitional clustering assumes that the number of clusters is predefined, say k. The points are embedded in a metric space, so that each vertex is a point and a distance measure is defined between pairs of points in the space. The distance is a measure of dissimilarity between vertices. The goal is to separate the points in k clusters so as to maximize/minimize a cost function based on distances between points and/or from points to *centroids*. Few such functions include minimum k-clustering, k-clustering sum, k-center, k-median. The most popular partitional technique in the literature is k-means clustering [141]. Extensions of k-means clustering to graphs have been proposed by some authors [22, 100]. The limitation of partitional clustering is the same as that of the graph partitioning algorithms: the number of clusters must be specified at the beginning, the method is not able to derive it.

(iv) Spectral clustering: Spectral clustering includes all methods and techniques that partition the set of vertices into clusters by using the eigenvectors of matrices or other matrices derived from it. In particular, the objects could be points in some metric space, or the vertices of a graph. Spectral clustering consists of a transformation of the initial set of objects into a set of points in space, whose coordinates are elements of eigenvectors. The set of points is then clustered via standard techniques, like k-means clustering. The first contribution on spectral clustering was by Donath and Hoffmann [55]. There are three popular methods of spectral clustering: unnormalized spectral clustering and two normalized spectral clustering techniques, proposed by Shi and Malik [198] and by Ng et al. [165] respectively. However, Nadler and Galun [152] discussed the limitations of this method such as it cannot successfully cluster datasets that contain structures at different scales of size and density.

Divisive Algorithms

The philosophy of divisive algorithms is to detect the edges that connect vertices of different communities and remove them, so that the clusters get disconnected from each other. The most popular algorithm is the one proposed by Girvan and Newman [163]. The method is historically important, because it marked the beginning of a new era in the field of community detection. Here edges are selected according to the values of *edge betweenness centrality*. Tyler et al. proposed a modification of the Girvan-Newman algorithm, to improve the speed of the calculation [211]. Another fast version of the Girvan-Newman algorithm has been proposed by Rattigan et al. [185]. Here, a quick approximation of the edge betweenness values is carried out by using a network structure index, which consists of a set of vertex annotations combined with a distance measure. In this line, gradually two community detection algorithms have been proposed for overlapping community detection, namely the concept of vertex splitting [179] and CONGA (Cluster Overlap Newman-Girvan Algorithm) [83].

Modularity-based Algorithms

Modularity (introduced by Newman and Girvan [163]) is by far the most used and best known quality function. It is based on the idea that a random graph is not expected to have a cluster structure, so the actual strength of clusters is revealed by the comparison between the actual density of edges in a subgraph and the density one would expect to have in the subgraph if the vertices of the graph were attached regardless of community structure. This expected edge density depends on the chosen null model, i.e., a copy of the original graph retaining some of its structural properties but not community structure. Modularity can then be written as follows:

$$Q = \frac{1}{2m}\sum_{ij}(A_{ij} - \frac{k_i k_j}{2m})\delta(C_i, C_j) \tag{2.1}$$

where the sum runs over all pairs of vertices, A is the adjacency matrix, m the total number of edges of the graph, k_i the degree of vertex i, the δ-function yields one if vertices i and j are in the same community ($C_i = C_j$), zero otherwise. By assumption, high values of modularity indicate good partitions. All clustering techniques that require modularity, directly and/or indirectly can be classified as follows.

(i) Greedy techniques: The first algorithm devised to maximize modularity was a greedy method proposed by Newman [161]. It is an agglomerative hierarchical clustering method, where groups of vertices are successively joined to form larger communities such that modularity increases after the merging. Later on, Clauset et al. [48] proposed more efficient data structure like *max-heaps* to make Newman's algorithm faster. Danon et al. [52] suggested to normalize the modularity variation ΔQ produced by the merger of two communities by the fraction of edges incident to one of the two communities, in order to favor small clusters. Wakita and Tsurumi [217] noticed that, due to the bias towards large communities, the fast algorithm by Clauset et al. is inefficient, because it yields very unbalanced dendrograms. Another trick to avoid the formation of large communities was proposed by Schuetz and Caflisch [193]. A different greedy approach has been introduced by Blondel et al. [26] (mostly known as *Louvain algorithm*), for the general case of weighted graphs. The method consists of two phases. First, it looks for "small" communities by optimizing modularity locally. Second, it aggregates nodes of the same community and builds a new network whose nodes are the communities obtained in the first stage. These steps are repeated iteratively until a maximum of modularity is attained. The modularity maxima found by the method are better than those found with the greedy techniques by Clauset et al. [48] and Wakita and Tsurumi [217].

(ii) Simulated annealing: Simulated annealing [113] is a probabilistic procedure for global optimization used in different fields and problems. It was first employed for modularity optimization by Guimera et al. [87]. Its standard implementation combines two types of moves: *local moves*, where a single vertex is shifted from one cluster to another, taken at random; *global moves*, consisting of mergers and splits of communities. Splits can be carried out in several distinct ways. The best performance is achieved if one optimizes the modularity of a bipartition of the cluster, taken as an isolated graph. Global moves reduce the risk of getting trapped in local minima and they have proven to lead to much better optima than using simply local moves [144].

(iii) Extremal optimization: Extremal optimization is a heuristic search procedure proposed by Boettcher and Percus [28], in order to achieve an accuracy comparable with simulated annealing, but with a substantial gain in computer time. It is based on the optimization of local variables, expressing the contribution of each unit of the system to the global function being studied. This technique was used for modularity optimization by Duch and Arenas [56]. Generally, this technique maintains a good trade-off between accuracy and speed, although it sometimes leads to poor results on large networks with many communities [66].

(iv) Other optimization strategies: Agarwal and Kempe [2] suggested maximization of modularity within the framework of mathematical programming. Chen et al. [44] used integer linear programming to transform the initial graph into an optimal target graph consisting of disjoint cliques, which effectively yields a partition. Berry et al. [19] formulated the problem of graph clustering as a *facility location problem*, that attempts to minimize a cost function based on a local variation of modularity. Lehmann and Hansen [131] optimized modularity via *mean field annealing* [176]. Genetic algorithms [102] have also been used to optimize modularity.

Modifications of Modularity

In the most recent literature on graph clustering, several modifications and extensions of modularity can be found. Modularity can be easily extended to graphs with weighted edges [159], directed graphs [132]. Kim et al. [111] proposed a different definition based on diffusion on directed graphs, inspired by Google's PageRank algorithm. Rosvall and Bergstrom raised similar objections [191]. Gaertler et al. [73] introduced quality measures based on modularity's principle of the comparison between a variable relative to the original graph and the corresponding variable of a null model. Another generalization of modularity was recently suggested by Arenas et al. [7]. Expressions of modularity for bipartite graphs were suggested by Guimera et al. [88] and Barber [14]. However, community detection using modularity has certain issues including resolution limit, degeneracy of solutions and asymptotic growth [81]. To address these issues, multi-resolution versions of modularity [8] were proposed to allow researchers to specify a tunable target resolution limit parameter. He et al. [96] considered different community densities as good quality measures for community identification, which do not suffer from resolution limits. Furthermore, Lancichinetti and Fortunato [122] stated that even those multi-resolution versions of modularity are not only inclined to merge the smallest well-formed communities, but also to split the largest well-formed communities; some of these problems have been addressed and partially resolved by Chan et al. [41] recently.

Dynamic Algorithms

Here we describe methods employing processes running on the graph, focusing on spin-spin interactions, random walks and synchronization.

(i) Spin models: The Potts model is among the most popular models in statistical mechanics [224]. It describes a system of spins that can be in different states. Based on this idea, Reichardt and Bornholdt [186] proposed a method to detect communities that maps the graph onto a zero-temperature q-Potts model with nearest-neighbor interactions. In another work, Son et al. [201] presented a clustering technique based on the *Ferromagnetic Random Field Ising Model* (FRFIM).

(ii) Random walk: Random walks [106] can also be useful to find communities. If a graph has a strong community structure, a random walker spends a long time inside a community due to the high density of internal edges and consequent number of paths that could be followed. Zhou [239] used random walks to define a distance between pairs of vertices: the distance d_{ij} between i and j is the average number of edges that a random walker has to cross to reach j starting from i. A different distance measure between vertices based on random walks was introduced by Latapy and Pons [180] where the distance is calculated from the probabilities that the random walker moves from a vertex to another in a fixed number of steps. Hu et al. [103] designed a graph clustering technique based on a signaling process between vertices, somewhat resembling diffusion. Dongen, in his PhD thesis, described the *Markov Cluster Algorithm* (MCL) [214].

Statistical Inference based Methods

Statistical inference aims at deducing properties of data sets, starting from a set of observation and model hypotheses. If the data set is a graph, the model, based on hypotheses on how vertices are connected to each other, has to fit the actual graph.

(i) Generative models: Most of the methods adopted Bayesian inference [223], in which the best fit is obtained through the maximization of a likelihood (generative models). Hastings [94] chose a *planted partition model* of network with communities. Newman and Leicht [164] proposed a similar method based on a mixture model and the expectation-maximization technique. Another technique similar to that by Newman and Leicht was designed by Ren et al. [188] based on the *group fractions*. Maximum likelihood estimation was used by Čopič et al. [50] to define an axiomatization of the problem of graph clustering and its related concepts. Hofman and Wiggins [101] proposed a general Bayesian approach to the problem of graph clustering. The main limitation of these methods comes from high memory requirements.

(ii) Information theoretic approach: The modular structure of a graph can be considered as a compressed description of the graph to approximate the whole information contained in its adjacency matrix. Rosvall and Bergstrom [189] envisioned a communication process in which a partition of a graph in communities represents a synthesis of the full structure that a signaler sends to a receiver, who tries to infer the original graph topology from it. The same idea is the basis of an earlier method by Sun et al. [206], which was originally designed for bipartite graphs evolving in time. In a recent paper, Rosvall and Bergstrom [191] pursued the same idea of describing a graph by using less information than that encoded in the full adjacency matrix. The goal is to optimally compress the information needed to describe the process of information diffusion across the graph. Chakrabarti [38] has applied the minimum description length principle to put the adjacency matrix of a graph into the (approximately) block diagonal form representing the best trade-off between having a limited number of blocks, for a good compression of the graph topology, and having very homogeneous blocks, for a compact description of their structure.

Other Methods

Here we describe some algorithms that do not fit in the previous categories. Raghavan et al. [183] designed a simple and fast method based on *label propagation*. The main advantage of the method is the fact that it does not need any information on the number and the size of the clusters. It does not need any parameter, either. In a recent paper, Tibély and Kertész [210] showed that the method is equivalent to finding the local energy minima of a simple zero-temperature kinetic Potts model. A recent methodology introduced by Papadopoulos et al. [172], called *Bridge Bounding*, is similar to the L-shell algorithm, but here the cluster around a vertex grows until one “hits” the boundary edges. Another method, where communities are defined based on a local criterion, was presented by Eckmann and Moses [57]. Long et al. [138] devised an interesting technique that is able to detect various types of vertex groups, not necessarily communities. Zarei and Samani [235] remarked that there is a symmetry between community structure and anti-community (multipartite) structure, when one considers a graph and its complement, whose edges are the missing edges of the original graph.

2.1.2 Overlapping Community Detection

There has been a class of algorithms for network clustering, which allow nodes belonging to more than one community. As discussed in [226], we shall discuss the proposed algorithms by categorizing them into five classes.

Clique Percolation Algorithms

The clique percolation method (CPM) is based on the assumption that a community consists of overlapping sets of fully connected subgraphs and detects communities by searching for adjacent cliques. *CFinder* is the implementation of CPM, whose time complexity is polynomial in many applications [169]. However, it also fails to terminate in many large social networks. Following this, CPMw [64] introduces a subgraph intensity threshold for weighted networks. Only k-cliques with intensity larger than a fixed threshold are included into a community. Instead of processing all values of k, SCP [117] finds clique communities of a given size. Despite their conceptual simplicity, an usual criticism is that CPM-like algorithms are more like pattern matching rather than finding communities since they aim to find specific, localized structure in a network.

Link Partitioning Algorithms

On the other hand, few algorithms trying to partition links instead of nodes to discover community structure have also been explored. A node in the original graph is called overlapping if links connected to it are put in more than one cluster. Ahn et al. [3] proposed a method where links are partitioned via hierarchical clustering of edge similarity. Evans [61] projected the network into a weighted *line graph*, whose nodes are the links of the original graph, then applied the node partitioning algorithm. *CDAEO* [225] provides a post-processing procedure to determine the extent of overlapping. Kim and Jeong [110] extended the map equation method [191] to the line graph, which encodes the path of the random walk on the line network under the Minimum Description Length principle.

Local Expansion and Optimization Algorithms

Algorithms utilizing local expansion and optimization rely on growing a natural community or a partial community [124]. Baumes et al. [16] proposed a two-step process: first, nodes are ranked according to some criterion, then the process iteratively removes highly ranked nodes until small, disjoint cluster cores are formed. Lancichinetti et al. [121] proposed an algorithm called *LFM* which expands a community from a random seed node to form a natural community until a fitness function becomes locally maxima. Havemann et al. proposed MONC [95] which uses the modified fitness function of LFM that allows a single node to be considered a community by itself. Lancichinetti et al. further proposed *OSLOM* [125] that tests the statistical significance of a cluster [23] with respect to a global null model (i.e., the random graph generated by the configuration model [148] during community expansion).

Chen et al. [39] proposed selecting a node with maximal node strength based on two quantities: belonging degree and the modified modularity. Cazabet et al. [37] proposed *iLCD* which is capable of detecting both static and temporal communities. Given a set of edges created at some time step, iLCD updates the existing communities by adding a new node if its number of second neighbors and number of robust second neighbors are greater than expected values.

Seeds are very important for many local optimization algorithm. A clique has been shown to be a better alternative over an individual node as a seed. Shen et al. [197] in their algorithm *EAGLE* used the agglomerative framework to produce a dendrogram. Similar to EAGLE, *GCE* [128] identifies maximum cliques as seed communities.

Fuzzy Detection

Fuzzy community detection algorithms quantify the strength of association between all pairs of nodes and communities. Nepusz [155] modeled the overlapping community detection as a nonlinear constrained optimization problem which can be solved by simulated annealing methods. Zhang et al. [236] proposed an algorithm based on the spectral clustering framework [162]. There is another algorithm called *FOG* [54] which tries to infer groups based on link evidence. Similar mixture models can also be constructed as a generative model for nodes [72]. In *SSDE* [142], the network is first mapped into a d-dimensional space using the spectral clustering method. A Gaussian Mixture Model (GMM) is then trained via Expectation-Maximization algorithm. The number of communities is determined when the increase in log-likelihood of adding a cluster is not significantly higher than that of adding a cluster to random data which is uniform over the same space.

Non-negative Matrix Factorization (NMF) is a feature extraction and dimensionality reduction technique in machine learning that has been adapted to community detection. Zhang et al. [237] replaced the feature vector used in NMF with the diffusion kernel, which is a function of the Laplacian of the network. Later Zarei et al. [234] showed that the result would be better if the matrix is defined by the correlation matrix of the columns of the Laplacian. Recently, Yang and Leskovec [232] proposed BIGCLAM which is also based on NMF approach.

Ding et al. [54] extended the *affinity propagation clustering algorithm* [71] for overlapping community detection, in which clusters are identified by representative exemplars. First, nodes are mapped as data points in the Euclidean space via the commute time kernel (a function of the inverse Laplacian). The similarity between nodes is then measured by the cosine distance.

Agent-based and Dynamical Algorithms

The label propagation algorithm [183] in which nodes with same label form a community, has been extended to overlapping community detection by allowing a node to have multiple labels. Gregory proposed *COPRA* [85] in which each node updates its belonging coefficient by averaging the coefficients from all its neighbors at each time step in a synchronous fashion. Xie et al. [227] developed *SLPA* which is a general speaker-listener based information propagation process. A game-theoretic framework was proposed by Chen et al. [43], in which a community is associated with a Nash local equilibrium. A process in which particles walk and compete with each other to occupy nodes is presented by Breve et al. [34]. Different from SLPA and COPRA, this algorithm takes a semi-supervised approach. It requires at least one labeled node per class.

Other Methods

CONGO [83] extends Girvan and Newman's divisive clustering algorithm [78] by allowing a node to split into multiple copies. Gregory [84] also proposed to perform disjoint detection algorithms on the network produced by splitting the node into multiple copies using the split betweenness. Zhang et al. [238] proposed an iterative process that reinforces the network topology and propinquity that is interpreted as the probability of a pair of nodes belonging to the same community. The propinquity between two vertices is defined as the sum of the number of direct links, number of common neighbors and the number of links within the common neighborhood. Kovács et al. [114] proposed an approach focusing on centrality-based influence functions.

2.1.3 Community Scoring Metrics

Another important aspect of community detection is to evaluate the detected community structure. If we know the actual community structure of a network, it would be easier to evaluate the detected communities just by comparing them with the actual community structure. However, most of the time, collecting the actual ground-truth community structure is difficult, and therefore we rely on the structural property of the community structure. In this section, we first describe such topology-based community evaluation metrics and then briefly mention few popular validation metrics that are used to compare the detected community with the ground-truth structure.

Topology-based Community Evaluation Metric

Several metrics for evaluating the quality of community structure have been introduced. The most popular and widely accepted is Modularity [163] (see Equation 2.1). Recently, Fortunato and Barthelemy [68] presented a *resolution limit* problem of modularity, essence of which is that optimizing modularity will not find communities smaller than a threshold size, or weight [20]. The threshold depends on the total number (or total weight) of edges in the network and on the degree of interconnectedness between communities. Moreover, Good et al. [81] showed another problem of modularity called *degeneracy of solutions* that this measure admits an exponential number of high-modularity but structurally distinct solutions from a single graph. They also studied the limiting behavior of maximizing modularity for one model of infinitely modular networks (*asymptotic growth*), showing that it depends strongly both on the size of the network and on the number of modules it contains, i.e., as we add more modules to the network, the height of the modularity function converges to 1. To address the resolution limit problem, multi-resolution versions of modularity [8] were proposed to allow researchers to specify a tunable target resolution limit parameter. Lambiotte [119] proposed different types of multi-resolution quality functions to tackle resolution limit problem. Dongxiao et al. [96] considered different community densities as good quality measures for community identification, which do not suffer from resolution limits.

In the context of overlapping community evaluation, people attempted to redefine modularity for overlapping community structure. Shen et al. [197] introduced EQ, an adaptation of Newman's modularity function designed to support overlapping communities. The equation for EQ strongly resembles the original modularity function as follows:

$$EQ = \frac{1}{2m} \sum_{c \in C} \sum_{i \in c, j \in c} \frac{1}{O_i O_j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \tag{2.2}$$

where m is the number of edges in the graph, C is the set of communities, and O_v is the number of communities to which the node v belongs. The presence of an edge between two nodes v and w is represented as the value in the corresponding position of the adjacency matrix A_{vw}.

On the other hand, recently Lázár et al. [127] provided a more complex and potentially more accurate evaluation of the goodness of an overlapping community structure as follows:

$$Q_{ov} = \frac{1}{K} \sum_{r=1}^{K} \left[\frac{\sum_{i \in c_r} \frac{\sum_{j \in c_r, i \neq j} A_{ij} - \sum_{j \notin c_r} A_{ij}}{di \cdot s_i}}{n_{c_r}} \cdot \frac{n_{c_r}^e}{\binom{n_{c_r}}{2}} \right] \tag{2.3}$$

where K is the number of communities, n_{c_r} is the number of nodes and $n_{c_r}^e$ is the number of edges that the rth cluster c_r contains respectively, d_i is the degree of node i, s_i denotes the number of clusters where i belongs to and A is the adjacency matrix. Note that since the density of clusters containing one single node (when n_{c_r} = 1) is not defined (because $\binom{1}{2}$ is undefined), the modularity value is set to be zero.

Ahn et al. [3] described two simple measures to quantify the quality of a community structure. The first one is *Community Coverage* which simply counts the fraction of nodes that belong to at least one community of three or more nodes. A size of three is chosen since it is the smallest nontrivial community. This measure provides an estimate of how much of the network is analyzed. The second measure is *Overlap Coverage* which counts the average number of memberships in nontrivial communities (size at leas three) that nodes are given.

Ground-truth Based Community Validation Metrics

Evaluating the quality of a detected partitioning or cover is nontrivial, and extending evaluation measures from disjoint to overlapping communities is rarely straightforward. In this section, we discuss some of the popular evaluation metrics which are often used to compare the detected partition with the ground-truth communities.

(i) Purity (PU): The Purity measure [143] is historically the first one used in the context of community detection. Let us assume that $X = \{x_1, x_2, ..., x_I\}$ and $Y = \{y_1, y_2, ..., y_J\}$ be the two partitions of the same set. To denote the cardinalities, we use n for the total number of elements in the partitioned set, and $n_{ij} = |x_i \cap y_j|$ for the intersection of two parts. We also note $n_{i+} = |x_i|$ and $n_{+j} = |y_j|$ the part size. The purity of a part x_i relative to the other partition Y is expressed as $PU(x_i, Y) = \max\limits_{j} \frac{n_{ij}}{n_{i+}}$. The total purity of partition X relative to partition Y is obtained as follows: $PU(X, Y) = \sum_i \frac{n_{i+}}{n} PU(x_i, Y)$.

It is important to notice that purity is not a symmetric measure. Therefore, the usual approach is to take the harmonic mean of $PU(X, Y)$ and $PU(Y, X)$. The upper bound is 1, it corresponds to a perfect match between the partitions. The lower bound is 0 and indicates the opposite.

(ii) Rand Index (RI): The Rand Index [184] is a way of comparing disjoint clustering solutions that is based on pairs of the objects being clustered. Two solutions are said to agree on a pair of objects if they each put both objects into the same cluster or each into different clusters. The Rand Index can then be formalized as follows:

$$RI = \frac{(a+d)}{N} \tag{2.4}$$

where N is the number of pairs of objects, a is the number of times the solutions agree on putting a pair in the same cluster and d is the number of times the solutions agree on putting a pair in different clusters. That is, the Rand Index is the number of pairs that are agreed on by the two solutions divided by the total number of pairs.

An improvement to the Rand Index is the *Adjusted Rand Index* (ARI) [105] which adjusts the level of agreement according to the expected amount of agreement based on chance.

(iii) Omega Index: The Omega Index [49] builds on both the Rand Index and Adjusted Rand Index by accounting for disjoint solutions and correcting for chance agreement. The Omega Index considers the number of clusters in which a pair of objects is together. The observed agreement between two partitions $S1$ and $S2$ is calculated by: $Obs(S1, S2) = \sum_{j=0}^{min(J,K)} A_j/N$, where J and K represent the maximum number of clusters in which any pair of objects appears together in partitions 1 and 2 respectively, A_j is the number of the pairs agreed by both partitions to be assigned to number of clusters j, and N is again the number of pairs of objects. The expected agreement is given by: $Exp(S1, S2) = \sum_{j=0}^{min(J,K)} N_{j1}N_{j2}/N^2$, where N_{j1} is the total number of pairs assigned to number of clusters j in partition 1, and N_{j2} is the total number of pairs assigned to number of clusters j in partition 2. The Omega Index is then calculated as

$$Omega(S1, S2) = \frac{Obs(S1, S2) - Exp(S1, S2)}{1 - Exp(S1, S2)} \tag{2.5}$$

The highest possible score of 1 indicates that two solutions perfectly agree on how each pair of objects is clustered.

(iv) Normalized Mutual Information (NMI): The problem of comparing different community structures can be overcome by computing the Normalized Mutual Information (NMI) [215]. Let C be the confusion matrix. Also let N_{ij} (elements of the confusion matrix C) be the number of nodes in the intersection of the original community i and the generated community j. If C_A denotes the number of the communities in the ground truth, C_B the number of the generated communities by an arbitrary approach, N_i the sum of row i, N_j the sum of column j, and N the sum of all elements in C, then the NMI score between the ground truth partition A, and the generated partition B can be computed as shown in the following equation.

$$NMI(A,B) = \frac{-2\sum_{i=1}^{C_A}\sum_{j=1}^{C_B} N_{ij} log\frac{N_{ij}N}{N_i N_j}}{\sum_{i=1}^{C_A} N_i log\frac{N_i}{N} + \sum_{j=1}^{C_B} N_j log\frac{N_j}{N}} \tag{2.6}$$

The values of NMI range between 0 and 1 where 0 refers to no match with the ground truth and 1 refers to a perfect match. Recently, McDaid et al. [147] also provided a modified version of NMI, called *ONMI* for evaluating overlapping community structures.

However, Labatut [118] argued that these measures are not completely relevant in the context of network analysis, because they ignore the network connectivity. He proposed the modified versions of these measures where misplacing high degree vertices would incur higher penalty compared to low degree vertices. The modified formulations of NMI, ARI and Purity are the weighted versions, namely Weighted-NMI (W-NMI), Weighted-ARI (W-ARI) and Weighted-Purity (W-PU).

2.2 Survey on Post-hoc Analysis of Communities

In this section, we survey the current literature pertaining to the analysis of detected communities and how this community information can be used in the development of various systems.

2.2.1 Tracking Communities over Time

In real world, the membership of communities tend to change gradually. Backstrom et al. observed this on the communities of LiveJournal users and communities of conference publications on DBLP [10]. So it is important not only to detect communities but also to track the changes in membership over time. The questions are: which community in one snapshot metamorphoses into anther in the next snapshot? How and what fraction of membership changes in between? The problem of tracking communities is motivated by a problem in behavioral ecology in studying animals that live in fission-fusion societies such as zebra and the Asiatic wild ass [207]. A natural question is which group that we observe today is the same as that was previously observed. Groups in this setting are manifestations of perpetual communities. Inversely, a community is a consistent string of groups seen on different days. This is the problem of tracking communities over time. Loosely speaking, it is about how to string different groups from the same day into communities which span over multiple days.

There is a handful of work specifically on the problem of tracking communities over time. Berger-Wolf and Saia [18] proposed a framework which defines communities as independent local patterns. There, a community (or, metagroup) is a sequence of groups which have sufficiently high similarity. The similarity between two groups is the number of common members normalized by the sizes of the two groups. Characteristics of communities are studied via community-based statistical measures such as number of all possible communities, their sizes and life spans. Also, they proposed an approach to study the survival of the communities via finding a critical set of groups whose removal leaves only short-lived communities.

Spiliopoulou et al. [202] proposed a framework, called *MONIC*, for tracking communities over time. The framework utilizes a similarity function of groups at different time steps. The function takes into account the number of common members, the sizes of the groups, and the time decay between the groups. Then, two groups are strung together as being in the same community if their similarity is above a certain threshold. The framework not only strings groups into communities but also detects splitting and merging of communities by a separate set of threshold parameters.

Tantipathananandh et al. [209] proposed the first framework which rigorously formulates the problem of tracking communities as an optimization problem. Although the appealing aspect of this framework is the social costs model which has its roots in the social sciences view of group dynamics [174], the framework has a strong assumption that all time steps must have the same length. Tantipathananandh et al. [208] further introduced an improved framework which can handle data with time steps of variable length.

2.2.2 Analyzing Community Evolution in Networks

Slightly different from the task of community tracking is the study of the evolution of communities over time. This problem attracts a lot of research interest due to its enormous applications in real-world scenario. For example, in a blog network we might wish to detect which communities of blogs are relatively stable in size over a period of time [136]. In a mobile phone network, changes in community size over a timeframe can reveal calling patterns and customer churns [82, 168]. In other contexts such as scientific collaboration networks, communities of researchers that span many years suggest long-term research collaboration [168]. Such communities can be further investigated to identify researchers in particular fields who are consistently productive over a period of time [136]. Previous work along this line analyzed community changes using a life-cycle model comprising events such as birth, death, expand, contract, merge, and split [82,168]. Asur et al. [9] emphasized the life-cycle of nodes, an emphasis that is impractical in networks with millions of nodes and irrelevant when an overview of how communities evolve is required. Palla et al. [168] used the above events to quantify the evolution of a phone call network and a coauthorship network, whereas Greene et al. [82] used the events to investigate community evolution in a phone call network. Except for [168], little attention has been paid to modeling an event as a function of time. Recently, Lužar et al. [139] studied interdisciplinarity of research communities detected in the coauthorship network of Slovenian scientists over time.

2.2.3 Community Structure in Link Prediction

The information of community of nodes can also be leveraged in the task of link prediction. Clauset et al. [47] proposed a method to determine the hierarchical structure of a network by using MCMC sampling to create a binary dendrogram that joins nodes into groups. Since this method got introduced, a variety of similar methods and models have been proposed. Valverde-Rebaza and de Andrade Lopes [212] described experiments to analyze the viability of applying the *within and inter cluster* (WIC) measure for predicting the existence of a future link on a large-scale online social network. They further proposed three measures for the link prediction task which take into account all different communities that users belong to [213]. Sachan and Ichise [192] proposed to build a link predictor in a co-authorship network, and showed that the knowledge of a pair of researchers lying in the same dense community can be used to improve the accuracy of our predictor further. Recently, Fenhua et al. [134] proposed a link prediction method based on clustering and global information.

2.2.4 Community Structure in Information Diffusion

Communities are vehicles for efficiently disseminating news, rumors, and opinions in human social networks. Several approaches studied this phenomenon using the community structure of the network. Belak et al. [17] studied information diffusion across communities and showed that one can achieve high community-based spreading using an efficient targeting strategy. Nematzadeh et al. [154] used the linear threshold model to systematically study how community structure affects global information diffusion. Kimura et al. [112] used community analysis to find influential nodes for information diffusion on a social network under the independent cascade model. Weng et al. [222] focused on understanding interactions between community structures and information diffusion, and developed predictive models of information diffusion based on community structure. Chen et al. [40] employed the network to investigate the impact of overlapping community structure on susceptible-infected-susceptible (SIS) epidemic spreading process. Similarly, Xiangwei et al. [46] studied epidemic spreading in weighted scale-free networks with community structure. Recently, Shang et al. [195] classified vertices into overlapping and non-overlapping ones, and investigated in detail how they affect epidemic spreading.

2.2.5 Community Structure in Recommendation Systems

Community detection algorithms and clustering functions constitute a powerful tool in the development of network based recommendation system. Zhuhadar et al. [240] used the community detection method to design a visual recommender system to recommend learning resources to cyberlearners within the same community. Lisboa et al. [137] proposed a method to improve recommendation systems by taking into consideration changes in the behavior of users over time. For that, communities are first detected using a network analysis method and recommendations are made for each community using Naïve Bayes modeling. Kamahara et al. [108] proposed a recommendation method in which a user can find new interests that are partially similar to the user's taste, where partial similarity is an aspect of the user's preference which is projected by the community in which the user belongs. Musto et al. [151] particularly studied user community behavior in OSN and developed *STaR* to suggest a set of relevant keywords for the resources to be annotated. Fatemi and Tokarchuk [65] proposed novel community based social recommender system, *CBSRS* which utilizes the social data to provide personalized recommendations based on communities constructed from the users' social interaction history with the items in the target domain.

Chapter 3

Constant Communities in Networks

In this chapter we address our first objective – studying the dependence of community detection algorithms on the vertex ordering that leads to the variability in the final output obtained.

3.1 Introduction

A fundamental problem in understanding the behavior of complex networks is the ability to correctly detect communities. Mathematically, this question can be translated to a combinatorial optimization problem with the goal of optimizing a given metric of interrelation, such as modularity or conductance. The goodness of community detection algorithms (see [70] for a review) is often objectively measured according to how well they achieve the optimization.

However, these algorithms can be applied to any network, regardless of whether it possesses a community structure or not. Furthermore when the optimization problem is NP-hard, as in the case of modularity [163], the order in which the vertices are processed as well as the heuristics can change the results. These inherent fluctuations of the results associated with modularity have long been a source of concern among researchers. Indeed the goodness of modularity as an indicator of community structure has also been questioned, and there exist examples [81] which demonstrate that high modularity does not always indicate the correct community structure.

Research in addressing the fluctuations in the results due to modularity maximization heuristics include identifying stability among communities from the consensus networks built from the successive iterations of a non-deterministic community detection algorithm (such as by Seifi et al. [194]). Lancichinetti et al. [123] proposed *consensus clustering* by reweighting the edges based on how many times the pair of vertices were allocated to the same community, for different identification methods. Ovelgonne et al. [167] pointed out an ensemble learning strategy for graph clustering. Gfeller et al. [76] investigated the instabilities in the community structure of complex networks. Finally, several pre-processing techniques [56, 187] have been developed to improve the quality of the solution. These methods form an initial estimate of the community allocation over a small percentage of the vertices and then refine this estimate over successive steps.

3.2 Defining Constant Communities

All combinatorial optimization algorithms focus on compiling the differences in the results to arrive at an acceptable solution, and despite these advances a crucial question about the variance of results remains unanswered – what does the *invariance* of the result tell us about the network structure? In this chapter, we focus on the invariance in community detection as obtained by modularity maximization. Our results, on a set of scale-free networks, show that while the vertex orderings produce very different set of communities, some groups of vertices are always allocated to the same community for all different orderings. We define the group of vertices that remain invariant as *constant community* and the vertices that are part of the constant communities as *constant vertices*. Figure 3.1 shows a schematic diagram of constant communities. In this figure, two colors (red and green) indicate two communities of the network formed in each iteration. Combined results of two algorithms produce two constant communities (rectangular and circular vertices). Remaining one vertex (hexagonal shaped) is not included since it switches its community between the two algorithms. Note that not all vertices in the network belong to constant communities. This is a key difference of constant communities with the consensus methods [123] described

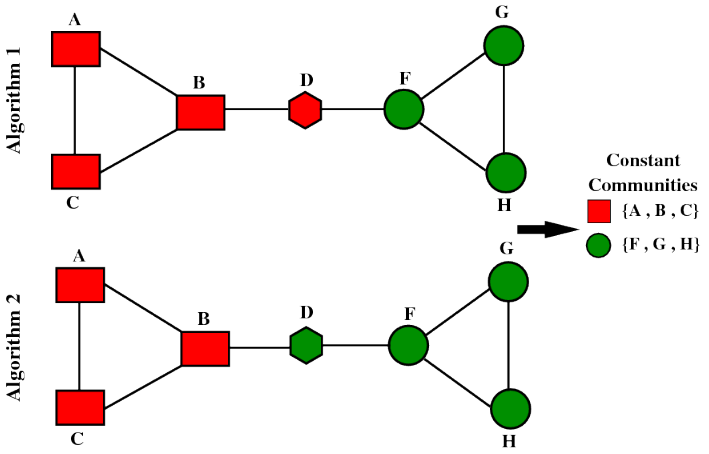


Figure 3.1: Schematic illustration of the formation of constant communities.

earlier. Consensus methods attempt to find the best (most stable or most similar) community among all available results and thus include all the vertices. Constant communities, on the other hand, focus on finding subgraphs where the cohesive groups can be unambiguously identified. As discussed earlier, communities obtained by modularity maximization may include vertices that can move from one group to another depending on the heuristic or the vertex ordering. The vertex groups obtained using constant communities are invariant under these algorithmic parameters and, thereby, provide a lower bound on the number of uniquely identifiable communities in the network. Although trivially each vertex can be considered to be a constant community by itself, our goal is to identify the largest number of vertices (i.e., at least three or more) that can be included in an invariant group.

The presence of such invariant structures can be used to evaluate the accuracy of the communities obtained when other independent methods of verifications are unavailable. However in many networks, constant communities constitute only a small percentage of the total number of vertices. To understand how other non-constant vertices are allocated to communities, we show that by using constant communities we can significantly reduce the variations in results (see Section 3.6). Thus, building from the more accurate results reduces the variance over the larger network.

3.3 Experimental Setup

3.3.1 Datasets

We conduct our experiments on networks obtained from real-world data as well as on a set of synthetically generated networks using the LFR model [120]. The set of real-world networks is obtained from the instances available at the 10th DIMACS challenge website [1]. The networks, which are undirected and unweighted, include – Jazz (network of jazz musicians; $|V| = 198, |E| = 2742$) [80], Polbooks (network of books on USA politics; $|V| = 105, |E| = 441$) [115], Chesapeake (Chesapeake bay mesohaline network; $|V| = 39, |E| = 340$) [12], Dolphin (Dolphin social network; $|V| = 62, |E| = 159$) [140], Football (American college football; $|V| = 115, |E| = 1226$) [79], Celegans (Metabolic network of C. elegans; $|V| = 453, |E| = 2025$) [56], Power (topology of the Western States Power Grid of the USA; $|V| = 4941, |E| = 6594$) [220] and Email (e-mail interchanges between members of the Univeristy Rovira i Virgili; $|V| = 1133, |E| = 5451$) [86] (note that $|V|$ refers to the number of vertices and $|E|$ refers to the number of edges).

Networks generated using the LFR model are associated with a mixing parameter μ that represents the ratio of the external connections of a node to its total degree. We create LFR networks based on the following parameters [123]: number of nodes = 500, average degree = 20, maximum degree = 50, minimum community size = 10, maximum community size = 50, degree exponent for power law = 2, community size exponent = 3. We vary the value of μ from 0.05 - 0.90. Low values of μ correspond to well-separated communities that are easy to detect and consequently these networks contain larger percentage of constant communities. As μ increases, communities get more ambiguous and community detection algorithms provide more varied results leading to fewer vertices being in significantly sized constant communities.

3.3.2 Community Detection Algorithms

We select two popular agglomerative modularity maximization techniques – the method proposed by Clauset et al. [48] (henceforth referred to as the *CNM* method) and the method proposed by Blondel et al. [26] (henceforth referred to as *Louvain* method). Both these methods initially start by assigning one vertex per community. Then at each iterative step, two communities whose combination most increases the value of modularity are joined. This process of joining community pairs is continued until the value of modularity no longer increases. The Louvain method generally produces a higher value of modularity than CNM, because it allows vertices to migrate across communities if that leads to a more optimum value.

3.3.3 Degree Preserving Order

Ideally, the total number of different orderings to be tested should be equal to the factorial of the number of vertices in the network. However, even for the smallest network in our set (Chesapeake with 39 vertices) this value is astronomical. We therefore restrict our permutations to maintain a *degree-preserving* order. The vertices are ordered such that if degree of v_i is greater than the degree of v_j, then v_i is processed prior to v_j.

In addition, to reducing the number of vertex permutation, degree-preserving permutation also has another important advantage. Recall that the networks in the test suite have few vertices with high degrees and a lot with low degrees. Therefore, arranging the high degree vertices earlier pushes most of the fluctuations towards the later part of the agglomeration process. This ensures that the sub-communities formed initially are relatively constant and only later do the divergence in community memberships take place. Clearly, such orderings based on decreasing degrees are geared towards facilitating low variance in communities. If *this ordering* does not produce constant structures, it makes a very strong case about the inherent fluctuations that underlie modularity maximization methods.

3.4 Identifying Constant Communities

In order to identify constant communities from a network, we permute the order of the vertices, and then apply a community detection algorithm to each of the permuted networks. The results vary across permutations. We select the groups of vertices that

Table 3.1: Comparison of the constant communities obtained from Louvain (LVN) with those obtained from CNM and Infomap (INFO) algorithms using NMI.

Networks		Jazz	Chesapeake	Dolphin	Football	Polbooks	Celegans	Email	Power
NMI	LVN vs. CNM	0.8856	0.8429	0.8663	0.8765	0.8950	0.9232	0.8103	0.8097
	LVN vs. INFO	0.8235	0.7928	0.9722	0.8824	0.8239	0.9144	0.8072	0.7856

are always allocated together across all the permutations and mark them as constant communities. The rationale behind this process is that these vertices must have some intrinsic connectivity properties that force them to stay together under all orderings.

To implement the vertex permutation, we adopt a stochastic degree-preserving scheme as discussed in Section 3.3.3 that can arrange the vertices based on the descending order of their degrees. The ordering of the set of vertices with the same degree is permuted. By applying this method we preserve the relative ordering of the degrees of the vertices since it is well-known that node-degrees constitute a fundamental network property. Thus, our permutations prevent us from the possibility of getting confined in a local maximum of the modularity.

In order to identify these communities, for each network in the test suite, we apply CNM (and Louvain) method over different permutations of the vertices and then isolate the common groups that are preserved across the different orderings. These common groups of vertices are marked as the constant communities for the respective network.

We further observe based on the high (> 0.80) Normalized Mutual Information (NMI) [143] (see Section 2.1.3) values that the overlap between the constant communities obtained from the two methods is considerable [205] (see Table 3.1). One might argue that the constant communities are highly dependent on the underlying optimization functions (such as modularity) or the methods (such as agglomerative method) used in the community detection algorithms. To cross-check this, we further detect the constant communities using another very popular non-deterministic community finding algorithm called *Infomap* [190] which is not an agglomerative method but tries to minimize the minimum description length of the bit sequence generated by a random walk. We observe a similar high overlap between the constant communities obtained from Louvain and Infomap (see Table 3.1). Therefore, in the interest of space and clarity we confine our discussion about the properties of constant communities to those obtained from the Louvain method.

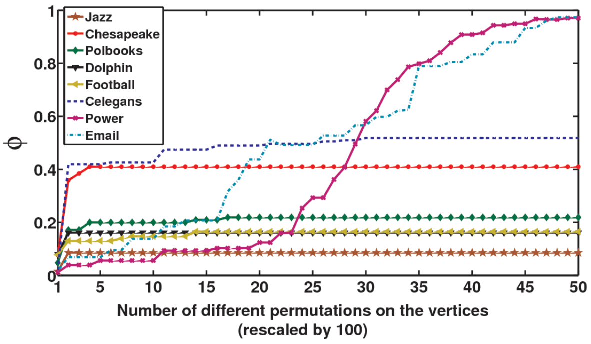


Figure 3.2: Sensitivity of each network across 5000 permutations. X-axis is rescaled by a constant factor of 100 for better visualization.

3.5 Characteristics of Constant Communities

In this section, we identify some interesting characteristic properties of constant communities observed in the real-world networks.

3.5.1 Sensitivity of Community Structure to Vertex Perturbations

In our first experiment, we study how the community structures of the networks change under vertex perturbations. Since constant communities are the groups of vertices that remain invariant, we measure the change in community structure based on the number of constant communities. We define *sensitivity* (ϕ) as the ratio of the number of constant communities to the total number of vertices. If ϕ is 1 then each vertex by itself is a constant community (the trivial case), thus there is no consensus over the set of communities obtained over different permutations. The higher the sensitivity metric, the fewer the vertices in each constant community and, therefore, this metric is useful for identifying networks that do not have a good community structure under modularity maximization. Note that this metric will be used further in Section 4.13.2 to quantify degeneracy of solutions of a community detection algorithm.

The sensitivity of each network is given in Figure 3.2. The x-axis indicates the number of different permutations of the vertices and the y-axis plots the value of the sensitivity. We observe that for most of the networks the number of constant communities become stable within the first 100 permutations, and the sensitivity values are low. This indicates that there can potentially exist very strong groups in these networks that have to be together to achieve high modularity. However, for networks such as Power and Email, the number of constant communities keeps increasing until the values of ϕ are close to 1. Thus, the community detection results for these two networks are extremely sensitive to the vertex perturbations. This implies that the communities (if any) in these two networks are not tightly knit, i.e., very "amorphous".

3.5.2 Percentage of Constant Communities

We further define the *relative size* (ξ) of a constant community as the ratio of the number of vertices in that constant community to the total number of vertices in the network and the *strength* (Θ) as the ratio of the edges internal to the constant community to the edges external (i.e., one end point of the edge is inside the constant community while the other is outside) to the constant community. Figure 3.3 plots the relative size (in percentage) of the constant communities with respect to their strength. If the strength of a constant community is above 1 (above 0 in log scale) then the number of internal edges in the community is larger than the number of external edges. The higher the value, the more tightly connected is the community. We notice that the value of relative size ranges from 0-34, with a larger cluster of values around 0-5. This shows that most of the constant communities contain very few vertices with respect to the network size. If the relative size of the constant communities is low then the remaining vertices have more freedom in migrating across communities, making the community structure weaker. We observe that, despite there being more constant communities of low relative size, there are some networks that have multiple constant communities with relative size over 15% of the total number of nodes indicating that they have a much stronger community structure. These include Jazz, followed by Dolphin and then Polbooks and Chesapeake.

Relative size and strength together provide an estimate of which networks have good community structure. If we divide the x-axis at roughly the mid-point of the range and the y-axis at 1, then we obtain four quadrants each representing different types of community structures. The first quadrant (upper right) contains communities that have high relative

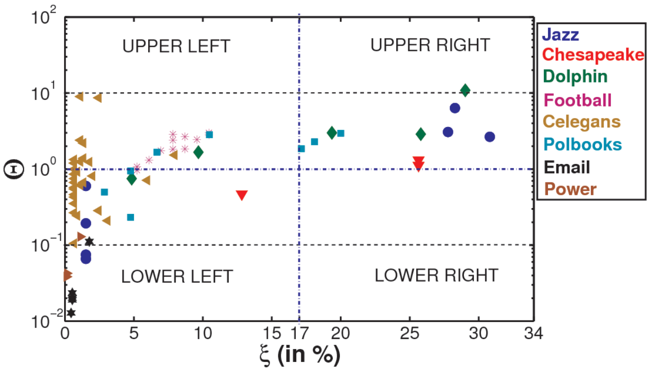


Figure 3.3: Comparison between the relative size and strength of the constant communities. X-axis plots the relative size in percentage, and Y-axis (in logarithmic scale) plots the strength. The plot is vertically divided at x = 17 that could help systematically analyze the distribution of the points.

size as well as high strength. Networks containing a large number of such constant communities are less likely to be affected by perturbations. Diagonally opposite is the third quadrant (lower left), which contains communities of low relative size and low strength. As discussed earlier, networks having communities predominantly from this quadrant will produce significantly different results under perturbations and are likely to not have a strong community structure under modularity maximization. The second quadrant (upper left) contains the groups of vertices that are strongly connected but have small relative size. This indicates that there are some pockets of the network with strong community structure. The fourth quadrant (lower right) represents communities with high relative size but low strength. In this set of experiments it is empty, and we believe that this area will be sparsely populated, if at all. This is because networks having such communities will have a very special structure: strongly connected groups of very few vertices with many spokes radiating out to account for the high number of external communities.

3.5.3 Pull from External Connections

We note in Figure 3.3 that there are several constant communities whose strength is below one, i.e., they have more external than internal connections. This is counterintuitive to the idea that a strong community should have more internal connections. Indeed, modularity maximization methods always tend to create communities whose strengths are greater than

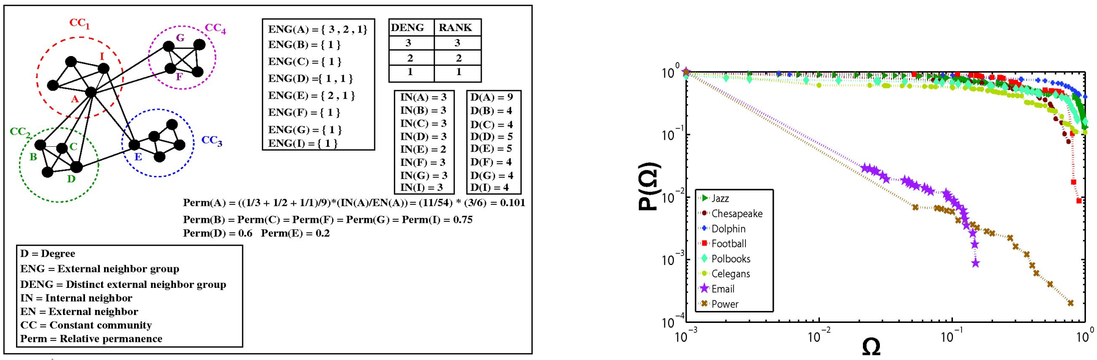


Figure 3.4: (Left) Schematic diagram illustrating the computation of the relative permanence of the vertices; (Right) distribution of relative permanence values.

one. However, the structure of some of the constant communities belies this convention.

We observe that in these cases, the external connections are distributed across different communities. Furthermore, the number of connections to any one external community is always lower than the internal connections. Based on this observation, we hypothesize that a group of vertices are likely to be placed together so long as the internal connection is greater than the connections to any one single external community. In such a scenario, the vertices within the community do not experience a significant "pull" from any of the external communities that can cause them to migrate, and therefore, their propensity to remain within their own communities is high. We quantify this observation as follows:

Let v be a vertex in a constant community; further, let $D(v)$ denote the degree of v, and $EN(v)$ and $IN(v)$ denote the number of external and internal neighbors of v respectively (i.e., $D(v) = IN(v) + EN(v)$). We also assume that the $EN(v)$ external neighbors are divided into k external groups, and $ENG(v)$ denote a set of k elements where the ith element in the set represents the number of neighbors of v belonging to the i^{th} external group. For instance, consider the vertex A in CC_1 in Figure 3.4 (left), $D(A) = 9, IN(A) = 3, EN(A) = 6$ and $ENG(A) = \{3, 2, 1\}$ (i.e., three external neighbors in CC_2, one external neighbor in CC_3, and two external neighbors in CC_4). Similarly, we calculate $ENG(v)$ for each vertex in the network and form a list $DENG(G)$ by taking union over all $ENG(v)$, that is, only unique entries across $ENG(v)$ get listed in $DENG(G)$ (see Figure 3.4 (left)). The list is then ranked in ascending order, i.e., the group with lowest number of external neighbors is ranked 1, the group with second lowest external neighbors is ranked 2 and so on. The intuition behind this ranking is that we are more interested in how distinct the external neighbor groups are, rather than the absolute size of the external neighbor groups. Moreover, by ranking, we can reduce the skewness of the range of external group size. This rank would therefore signify the intensity of the pull of the particular external community and its inverse signifies the degree of stability of the vertex v.

For a particular vertex, if the inverse rank of each of the external group is equal to one, it would point to the fact that all its external neighbors are diversely distributed (i.e., well-spread), and therefore the pull experienced should be minimum; in contrast, if the value is much lower than one, it would imply that the vertex experiences a strong pull from its external neighbors. We define the *strength* of a vertex *v*, $\theta(v)$, as the ratio of the internal neighbor ($IN(v)$) to the external neighbor ($EN(v)$) of vertex v similar to the strength (Θ) of a constant community defined earlier. Mathematically, the suitably normalized value of $relative\ permanence$, $\Omega(v)$, of a vertex v in a constant community can be expressed as:

$$\Omega(v) = \theta(v) \times \frac{\sum_{i=1}^{k} \frac{1}{Rank(ENG_i(v))}}{D(v)} \tag{3.1}$$

where $Rank(ENG_i(v))$ denote the rank (retrieved from the ***DENG(v)*** list) of the i^{th} element in $ENG(v)$. This metric indicates the propensity of a vertex to remain in the same community regardless of any algorithmic parameters.

Figure 3.4 (left) presents a schematic diagram for computing relative permanence of vertices within the communities. Figure 3.4 (right) plots the cumulative distribution of the relative permanence over the vertices in all networks. The x-axis indicates the value of the relative permanence and the y-axis, the cumulative fraction of vertices having the corresponding relative permanence value. The nature of the cumulative permanence distribution of the vertices is roughly same for all networks except Email and Power. The distinguishing nature of the curves for Email and Power graphs compared to the other graphs indicates that very few number of vertices in these two networks have higher relative permanence values and therefore experience more "pull" from the external communities. Another observation is that a high fraction of vertices in Jazz, Polbooks, Dolphin and Celegans have relative permanence close to one. These vertices are more "stable" compared to the other vertices in the respective networks.

Algorithm 1 Modularity maximization using constant communities

```
Input: A network (graph) $G = (V, E)$; Community detection algorithm $A$.
Output: Set of constant communities $CC_1, \dots CC_k$; Modularity $Q$
  procedure FINDING CONSTANT COMMUNITIES
    Sort vertices in $V$ in degree descending order
    Apply degree preserving permutation $P$ to vertices such that degree($v_i$) $\geq$ degree($v_{i+1}$) in $P$.
    $|P|$ is number of degree preserving permutations applied.
    Initialize array $Vertex[|V|][|P|]$ to -1        /* $Vertex[|V|][|P|]$ will store the community membership of the vertices in each permutation */
    Set $i = 0$        /* This variable indicates the permutation index */
    for all $P_i \in P$ do        /* Detect community memberships of the vertices in each permutation using $A$ and store them in $Vertex$ */
      Apply algorithm $A$ to find the communities of the permuted network $G_{P_i}$
      if Vertex $v$ is in community $c$ then
        $Vertex[v][i] = c$        /* Vertex $v$ in permutation $P_i$ belongs to community c after applying $A$ to $P_i$ */
      $i = i + 1$
    Set $j = 0$ /* This variable indicates the index of the constant community */
    for all $v \in V$ do /* Detecting constant communities using the community information stored in $Vertex$ */
      if vertex $v$ is not in a constant community then
        Create constant community $CC_j$
        Insert $v$ to $CC_j$        /* All $CC_j s'$ are the constant communities */
        for all $u \in V \setminus CC_j$ do
          if $Vertex[v][i] = Vertex[u][i], \forall\ i = 1$ to $|P|$ then        /* Check for the exact matching of community memberships of u and v */
            Insert $u$ to $CC_j$
      $j = j + 1$
  procedure COMPUTING MODULARITY
    Set of constant communities in $CC$
    for all $CC_j \in CC$ do /* Create intermediate small, weighted network */
      Combine vertices in $CC_j$ into a super-vertex $X_j$
      Replace edges from $X_j$ to another vertex $X_i$ by their aggregate weight        /* For the self-loop, $i = j$ */
    Sort vertices of collapsed network, $G'$, in degree descending order
    Apply community detection method $A$
    Unfold all $X_j$ in $G'$ and compute the modularity $Q$
```

3.6 Constant Communities for Improving Modularity

We note that in many networks (such as Football and Celegans) constant communities form only a small percentage of the vertices. Thus, finding only the constant communities may not provide adequate information about the relationship amongst the rest of the vertices. We therefore leverage on the invariant results in the first and second quadrants of Figure 3.3 as building blocks to identify larger communities.

We first permute the vertices 5000 times in degree-descending order i.e., each of the permutations preserves the constraint that if vertex v_i is placed before v_j in the sequence then $degree(v_i) \geq degree(v_j)$. Then for each of these permutations, we run Louvain algorithm and obtain the community structure (and the modularity value). Table 3.2 (left) shows the mean modularity (and its variance) obtained by averaging the modularity values of all iterations. Next, from these community structures obtained across the different permutations, we detect the constant communities and combine them into super-vertices. This process

Table 3.2: Modularity before and after pre-processing for real networks (left) and for different values of mixing parameter (μ) over LFR graphs (right)

Networks	Louvain			
	Before pre-processing		After pre-processing	
	Mean (m_q)	Var (σ_q)	Mean (m_q)	Var (σ_q)
Jazz	0.448	3.13e-6	0.452	0
Chesapeake	0.301	1.17e-5	0.303	3.36e-33
Polbooks	0.539	1.74e-5	0.557	1.24e-32
Dolphin	0.543	1.76e-5	0.550	0
Football	0.610	2.01e-5	0.623	0
Celegans	0.438	2.89e-5	0.442	1.33e-26
Email	0.542	6.89e-5	0.568	0.95e-12
Power	0.936	1.09e-5	0.937	2.25e-10

μ	Planted Modularity	Louvain			
		Before pre-processing		After pre-processing	
		Mean(m_q)	Var(σ_q)	Mean(m_q)	Var(σ_q)
0.05	0.878	0.834	1.98e-24	0.877	0
0.10	0.817	0.802	2.28e-28	0.817	0
0.20	0.716	0.690	5.74e-7	0.686	0
0.50	0.440	0.385	2.05e-6	0.389	1.58e-28
0.70	0.223	0.298	9.70e-10	0.219	1.04e-28
0.90	0.029	0.225	4.25e-10	0.205	5.64e-28

creates a smaller network as well as ensures that the vertices in the constant communities always stay together. Then we execute a modularity maximization algorithm over the entire network. We compute the variance in results by executing the underlying modularity maximization algorithm individually over 5000 permutations, in each case maintaining the degree-preserving order (see Algorithm 1). As shown in Table 3.2 (left), combining constant communities as a pre-processing step both increases the mean modularity value as well as reduces the variability across permutations for real-world networks.

We also observe that the variance becomes 0 or very low for the networks which have significant number of constant communities in the first and second quadrants of Figure 3.3. The results obtained from the other networks with high sensitivity, such as Email and Power, still indicate some variance although the value is less pronounced.

These observations on real-world networks lead us to believe that pre-processing using constant communities is more effective if a network has strong community structure. To test this hypothesis, we create LFR graphs with mixing parameters from 0.05 to 0.90. Low mixing parameters indicate strong community structure. For the LFR graphs, we repeat the same set of experiments as discussed above and obtain the mean modularity and its variance. As shown in Table 3.2 (right), pre-processing using constant communities also helps increase the modularity value and reduces variability of the results in the LFR graphs.

Another advantage of LFR networks is that we know the "ground-truth", i.e., the correct distribution of communities (exact number of vertices in each community and the number of in-community connections between them). We use NMI to compare the obtained com-

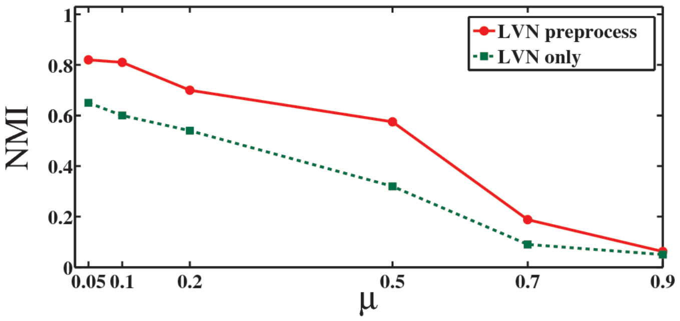


Figure 3.5: Variation of NMI for different values of mixing parameters. The broken line corresponds to the experiment without the pre-processing step and the solid line to the experiment after using the pre-processing step.

munities, with and without using the pre-processing step with the ground-truth community structures of LFR graphs for different mixing parameters. As shown in Figure 3.5, when the community structure is strong (low mixing parameter), using constant communities pushes the result towards the ground-truth. In contrast, when the community structure is not well-defined (high mixing parameter), use of constant communities does not mimic the community distribution of the ground-truth, because there can be many variations of community distribution in such networks that lead to high modularity. These results once again highlight the significance of constant communities.

3.7 Relative Ranking of Constant Communities

A constant community is meaningful if it is large in size (high ξ) or it has high relative permanence (Ω). We calculate the relative permanence of a constant community by averaging the relative permanence of its constituent vertices. We experiment to see which one of these two properties is more important in determining high modularity. To do so, we order the constant communities according to (a) decreasing order of ξ and (b) decreasing order of Ω. We combine the constant communities into super-vertices one by one following the order obtained from (a) and (b) separately. After each combination, we compute the modularity and compare the value with the average modularity (over 5000 permutations) obtained by using the Louvain method without any pre-processing.

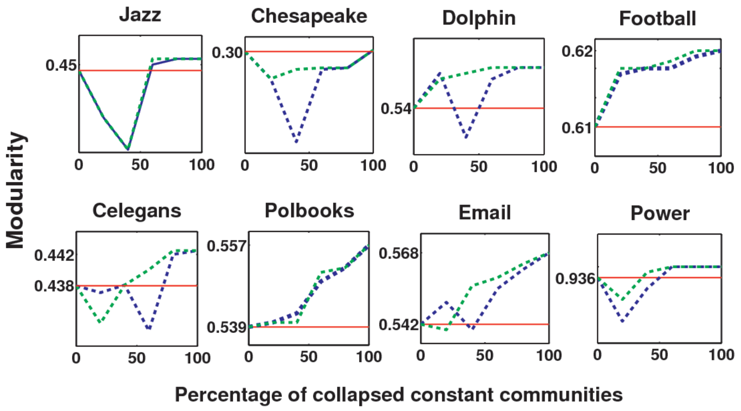


Figure 3.6: Modularity after partially collapsing the constant communities. The broken blue lines are in decreasing order of size and the broken green lines are in decreasing order of relative permanence. The red lines depict the mean modularities without using constant communities.

Figure 3.6 compares the modularity obtained by collapsing constant communities according to the order obtained from (a) (dotted blue line) and (b) (dotted green lines). For almost all the networks, there is a transition where the modularity values cross over the mean modularity (solid red line). Once this transition takes place, the modularity values generally remain above (or at least equal to) the mean modularity. This critical point indicates the smallest fraction of constant communities required to outperform the original algorithms. We observe further that the green lines (ordered according to $\Omega(v)$) generally reach the critical point earlier than the blue lines (ordered according to ξ), indicating that $\Omega(v)$ is a better indicator of constant communities.

Table 3.3: Few constant communities of PhoNet and the features they have in common.

Constant communities	**Features in common**
/p^h/, /t^h/, /k^h/	voiceless, aspirated, plosive
/^{m}b/, /^{n}d/, /^{n}g/	prenasalized, voiced, plosive
/p̰/, /t̰/, /k̰/	laryngealized, voiceless, plosive
/t/, /d/, /n/	dental
/ɭ/, /ɳ/, /ʈ/, /ɖ/	retroflex

3.8 Case Study

The significance of constant community in a network can be further understood if we consider networks where nodes have specific functionalities associated with them. We hypothesize that in such a network a constant community would represent indispensable functional blocks that reflect the defining characteristics of the network. In order to corroborate this hypothesis we conduct a case study on a specific type of linguistic network constructed from the speech sound inventories of the world's languages [150]. The sound inventory of a language comprises a set of consonants and vowels also sometimes together known as *phonemes*. In order to unfurl the co-occurrence principles of consonant inventories, Mukherjee et al. [150] constructed a network (phoneme-phoneme network or PhoNet) where each node is a consonant and an edge between two nodes denotes if the corresponding consonants have co-occurred in a language. The number of languages in which the two nodes (read consonants) co-occur defines the weight of the edge between these nodes. Note that each node here has a functional representation since it can be represented by means of a set of phonetic features (e.g., bilabial, dental, nasal, plosive etc.) that indicate how it is articulated. Since this is a weighted graph, we suitably define a threshold to construct the unweighted version. We detect constant communities of PhoNet and observe that each such graph (see Table 3.3) represents a *natural class*, i.e., a set of consonants that have a large overlap of the features [150]. Such groups are frequently found to appear together across languages, and linguists describe this observation through the principle of *feature economy* [150]. According to this principle, the speakers of a language tend to be economic in choosing the features in order to reduce their learning effort. For instance, if they have learnt to use a set of features by virtue of learning a set of sounds, they would tend to admit those other sounds in their language that are combinatorial variations of the features already learnt – if a language has the phonemes /p/ (voiceless, bilabial, plosive), /b/ (voiced, bilabial, plosive) and /t/ (voiceless, dental, plosive) in its inventory then the chances that it will have /d/ (voiced, dental, plosive) is disproportionately higher compared to any other arbitrary phoneme since by virtue of learning to articulate /p/, /b/ and /t/ the speakers need to learn no new feature to articulate /d/. Identification of constant communities therefore systematically unfolds the natural classes and provides a formal definition for the same (otherwise absent in the literature). Further, we observe that collapsing the constant communities results either in more dilute groups (still with a certain degree of feature overlap) or reproduces the same constant communities indicating that no valid dilution is possible for these functional blocks.

3.9 Summary of this Chapter

The idea of constant community has been derived by observing the variability of the community detection algorithms in terms of the results produced. We observe that constant communities are the most invariant part of the network. Therefore, the extent of presence of constant community within a network indicates how community-like a network is. Although we currently detect constant communities by comparing across different permutations, our results have uncovered some interesting facts about the community structure of networks, which can lead to improved algorithms for community detection.

- Constant communities of a network indicate the core of a community structure, in which the nodes have high probability of staying together. Moreover, we notice that constant communities are significantly different from the mere communities of a network. We characterize these constant communities using a new metric called relative permanence.
- The proposed metric, "sensitivity" indicates how well the community structure is within a network. The more the value of sensitivity, the less the extent of presence of constant communities.
- We show that prior detection of such constant communities eventually improves any community detection algorithm in discovering meaningful communities from a network.
- We also demonstrate through a labeled graph that these constant communities indeed represent the functional blocks of a network, i.e., each constant community corresponds to a functional unit of a network. Therefore, efficient detection of such blocks can be useful in several applications such as in the study of biological networks.

Chapter 4

Permanence and Network Communities

In this chapter we address our second objective of analyzing the limitations of state-of-the-art community evaluation metrics (such as modularity) and community detection algorithms (such as modularity maximization algorithms).

4.1 Introduction

Community detection algorithms primarily deal with identifying densely-connected units from within large networks. Often this is done blindly without much attention being paid toward inferring whether the network *at all possesses a community structure*. Similarly, a community detection algorithm targets for full coverage; in contrast, there might be situations when it should rather not include some of the nodes in any community. Modularity is a widely accepted metric for measuring the quality of community structure identified by various community detection algorithms. However, a growing body of research have begun to explore the limitations of maximizing modularity for community identification and evaluation; three such limitations include – resolution limit, degeneracy of solutions and asymptotic growth of the modularity value. To address these issues, we propose a new vertex-based metric called *permanence*, that can quantitatively give an estimate of the community-like structure of the network. The central idea of permanence is based on the observation that the strength of membership of a vertex to a community depends upon the following two factors: (i) the distribution of external connectivity of the vertex to individual communities and *not* the total external connectivity, and (ii) the strength of its internal connectivity and *not* just the total internal edges.

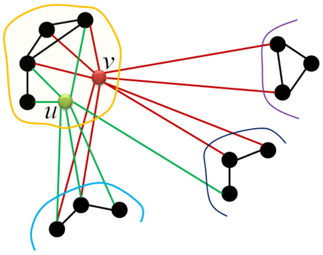
The contributions of this chapter are as follows:

- We define permanence for both disjoint and overlapping community structures. In contrast to the earlier literature where it is assumed that the constituent nodes in a community have the same level of belongingness, permanence unfolds the heterogeneity in the community membership of vertices.
- We show that permanence as compared to other standard measures, namely modularity, conductance and cut-ratio provides a more accurate estimate of a derived community structure to the ground-truth community.
- Permanence qualifies to be appropriately sensitive to perturbations in the network.
- We demonstrate that the process of maximizing permanence produces meaningful communities that concur with the ground-truth structure of the networks more accurately than the modularity based and other approaches.
- We also provide theoretical results to show that maximizing permanence can effectively reduce the limitations associated with modularity maximization as well as can indirectly help in inferring the community quality of a network.

In this chapter, we discuss the permanence measure for both disjoint and overlapping communities separately.

4.2 Permanence and Disjoint Community Structure

In this section, we explain two heuristics behind the formulation of permanence as follows. The basic idea builds on the relative permanence measure described in Section 3.5.3.



Vertex	D(.)	I(.)	E_{max}(.)	C_{in}(.)	Perm(.)
u	7	3	3	2/3	-0.191
v	9	3	2	1	0.167

Figure 4.1: Toy example depicting *permanence* of two vertices u and v.

Heuristic I: *A vertex should have more number of internal connections than the number of connections to any one of the neighboring communities.*

Most optimization metrics consider the *total number of external neighbors* of the vertex. However, earlier in Section 3.5.3 we demonstrated that a group of vertices are likely to be placed together so long as the number of internal connections is *larger* than the number of connections to *any one single external community*. In other words, the vertex which has connections to some external communities, experiences "pull" from each of these external communities, which is proportional to the number of connections to the external community. For example, in Figure 4.1, the pull that v experiences from each of its external communities is proportional to 2, whereas u experiences pull proportional to 3 from one community and 1 from the other community. However, we hypothesize that instead of considering total external connections of a vertex, one should look into how these external connections are distributed across different communities, which is mostly determined by the *maximum number of external connections to anyone of the neighboring communities* (which in this case is 2 and 3 for v and u respectively).

Heuristic II: *Within the substructure of a community, the internal neighbors of the vertex should be highly connected among each other.*

Most optimization metrics consider the internal connections of a vertex within its own community together as a whole. However, how strongly a vertex is connected to its internal neighbors may differ. The toy example of Figure 4.2 shows two networks each having two communities. Both the networks have the same number of edges; and the modularity, conductance and cut-ratio for the two divisions are exactly the same. However, the vertices on the left-hand graph are more tightly connected to each other than the

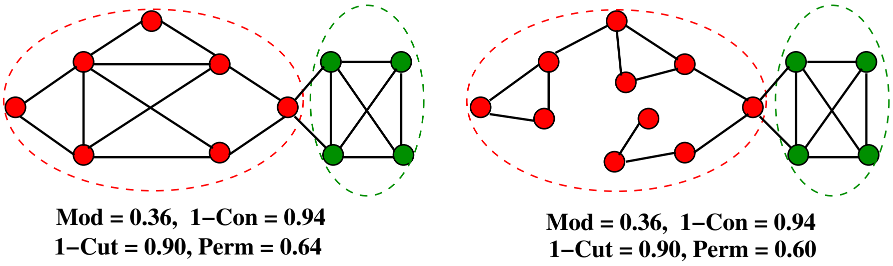


Figure 4.2: Two networks with same modularity, conductance and cut-ratio, but the left one has more prominent community structure.

vertices on the right-hand graph. To measure this internal connectedness of a vertex, one can compute the clustering coefficient of the vertex with respect to its internal neighbors. The higher this internal clustering coefficient, the more tightly the vertex is connected to its community. For instance, in Figure 4.1, the internal connectedness of v and u is 1 and $\frac{2}{3}$ respectively. We hypothesize that the *internal clustering coefficient of a vertex* can capture the connectedness of the vertex within its own community.

Based on these two heuristics together, we formulate *permanence* of a vertex in its own community. Permanence is composed of the following two ingredients.

1. The internal connections, $I(.)$, of the vertex v should be more than the maximum connections to a single external community, $E_{max}(.)$, which results in more internal pull than the maximum external pull (indicated by F1 in Equation 4.1). If the vertex has no external connections, F1 is just the value of the internal connections. We normalize this value by the total degree of the vertex, $D(v)$ (indicated by F2 in Equation 4.1), which ensures that the product of F1 and F2 will be between 0 (no internal connections) and 1 (no external connections).

2. Within a specific community, the internal neighbors of the vertex v should be highly connected among each other (i.e., its internal clustering coefficient[1], $c_{in}(v)$, should be high). This criteria emphasizes that a vertex is likely to be within a community if it is part of a near-clique substructure. For computing $c_{in}(v)$, we assume that each community should have at least three vertices and three internal connections; otherwise,

[1] Note that, internal clustering coefficient of v is obtained by considering the ratio of the existing connections and the total number of possible connections among the *internal neighbors* of v.

$c_{in}(v)$ is set to 0. When computing permanence, we impose a penalty based on low internal clustering coefficient (indicated by F3 in Equation 4.1). The less the internal clustering coefficient, the more the penalty imposed to the final outcome of the community score. This value also ranges from 0 (no penalty) to 1 (maximum penalty).

We aggregate these two criteria to formulate permanence of a vertex v as follows:

$$Perm(v) = \Big[\underbrace{\frac{I(v)}{E_{max}(v)}}_{\text{F1}} \times \underbrace{\frac{1}{D(v)}}_{\text{F2}}\Big] - \Big[\underbrace{1 - c_{in}(v)}_{\text{F3}}\Big] \tag{4.1}$$

Figure 4.1 depicts a toy example for measuring permanence of two vertices u and v. Note that this formula actually differentiates between the two cases in Figure 4.2 with higher permanence value for the case (left) where the external pull is uniform.

Boundary conditions of permanence: For vertices that do not have any external connections, $Perm(v)$ is considered to be equal to the internal clustering coefficient (i.e., $Perm(v) = c_{in}(v)$). The maximum value of $Perm(v)$ is 1 and is obtained when vertex v is an internal node and part of a clique. The lower bound of $Perm(v)$ is close to -1. This is obtained when $I(v) \ll D(v)$, such that $\frac{I(v)}{D(v)E_{max}(v)} \approx 0$ and $c_{in}(v) = 0$. Therefore for every vertex v, $-1 < Perm(v) \leq 1$. The permanence of a graph $G(V, E)$, where V is the set of vertices and E is the set of edges, is given by $Perm(G) = \frac{1}{|V|} \sum_{v \in V} Perm(v)$. For a graph $G(V, E)$, the range is $-1 < Perm(G) \leq 1$. $Perm(G)$ will be closer to 1 as more vertices have high permanence, that is more vertices are in well-defined communities. This can happen only if the network has a strong community structure. The maximum value obtained is when G consists of a series of disconnected cliques. If there is a vertex bridging between two cliques, then the highest overall permanence will be obtained if each clique acts as a separate community and bridging vertex forms a singleton community. For a grid, the best value of $Perm(G)$ will be zero, i.e., each vertex is assigned to a singleton community.

4.3 Experimental Setup

In this section, we provide a brief overview of the datasets, metrics and comparative methods that we use for our experiments.

4.3.1 Test Suite of Networks

We use the LFR benchmark model [121] to generate synthetic networks along with their ground-truth communities. Users can specify the following properties: number of nodes (n), average ($< k >$) and maximum (k_{max}) degree, the degree distribution, the community size distribution, and the mixing-coefficient (μ). The mixing coefficient represents the ratio (in average) between the external connections of a node to its degree. Thus the lower the value of μ, the stronger the community in the network. For our experiments, we set the number of nodes as 1000, and μ as 0.1, 0.3 and 0.6 (unless mentioned otherwise). For the rest of the parameters, we use the default values mentioned in the authors' implementation[2] [121].

We also use three real-world networks whose true community structures are known a priori and whose properties are summarized in Table 4.1. The last column of Table 4.1 shows that the average internal density of ground-truth communities is high, orders of magnitude higher than equivalent sized random graphs, and therefore can be considered as valid and significant communities.

Football network was proposed by Girvan and Newman [78] which contains the network of American football games between Division IA colleges during regular season Fall of 2000. The vertices in the graph represent teams (identified by their college names), and edges represent regular-season games between the two teams they connect.

Indian Railway network proposed by Ghosh et al. [77] consists of nodes representing stations, where two stations s_i and s_j are connected by an edge if there exists at least one train-route such that both s_i and s_j are scheduled halts on that route. The weight of the edge between s_i and s_j is the number of train-routes on which both these stations are scheduled halts. We filter out the low-weight edges and then make the resultant network

[2] https://sites.google.com/site/santofortunato/inthepress2

unweighted. We tag each station based on the state in India[3] to which that station belongs. The states act as communities since the number of trains within each state is much higher than the number of trains between two states.

Co-authorship network is derived from the citation dataset mentioned in Section 5.2. This dataset contains the metadata (title, author(s), related field(s)[4] of the paper, publication venue, year of publication, references and abstract) of all the papers of computer science published between 1960 to 2009 and archived in Microsoft Academic Search. We build an aggregated undirected coauthorship network where each node represents an author, and an undirected edge between a pair of authors is drawn if they were co-authors at least once. Since each paper is marked by its related field, we assume this field as the research area of the author(s) writing that paper. Therefore, an author may possess more than one area of research interests. We resolve this by tagging each author by the major field on which she has written most of the papers. These fields act as the ground-truth communities. Besides the aggregated network, we also create some intermediate networks mentioned in Table 4.6 by cumulatively aggregating all the vertices and edges over different time windows, e.g., 1960-1970, 1960-1971, 1960-1972 and so on.

Table 4.1: Properties of real-world networks; n and e are the number of nodes and edges, c is the number of communities, <k> and k_{max} its average and maximum degree, n_c^{min} and n_c^{max} the sizes of its smallest and largest communities, <ψ> its average internal connection density. The <ψ> values for the corresponding random graphs are shown within parenthesis in the last column.

Networks	n	e	<k>	k_{max}	c	n_c^{min}	n_c^{max}	<ψ>
Football	115	613	10.57	12	12	5	13	0.72 (8.1×10^{-3})
Railway	301	12,24	6.36	48	21	1	46	0.65 (5.3×10^{-3})
Coauthorship	103,677	352,183	5.53	1,230	24	34	14,404	0.31 (6.5×10^{-4})

[3] http://irfca.org/apps/station_codes

[4] Note that, the different sub-branches like Algorithms, AI, Operating Systems etc. constitute the different "fields" of computer science domain.

4.3.2 Scoring Functions for Evaluating Community Structure

The goodness of a community is often measured by how well certain scoring functions are optimized. Here we compare the optimal value of permanence for the obtained communities against three popular scoring functions, namely modularity (Mod) [156], conductance (Con) [133] and cut-ratio (Cut) [143]. In order to make the higher the better, we measure (1-Con) and (1-Cut) for conductance and cut-ratio respectively.

4.3.3 Metrics to Compare with Ground-truth

A stronger test of the correctness of the community detection algorithm, however, is by comparing the obtained community with a given ground-truth structure. We use three standard validation metrics, namely Normalized Mutual Information (NMI) [53], Adjusted Rand Index (ARI) [105] and Purity (PU) [143] to measure the accuracy of the detected communities with respect to the ground-truth community structure. Labatut [118] argues that these measures have certain drawbacks in that they ignore the connectivity of the network. We therefore also use the weighted versions of these measures, namely Weighted-NMI (W-NMI), Weighted-ARI (W-ARI) and Weighted-Purity (W-PU) as proposed in [118]. Note that all the metrics are bounded between 0 (no matching) and 1 (perfect matching).

4.3.4 Community Detection Algorithms

We use the following community detection algorithms for comparison with our proposed algorithm discussed in Section 4.6:
(i) Modularity-based: FastGreedy [161], Louvain [27] and CNM [48].
(ii) Random walk-based: WalkTrap [180].
(iii) Compression-based: InfoMod [189] and InfoMap [191].

Table 4.2: For football network, the values of the scoring functions on the output obtained from different algorithms and the scores of the validation metrics with respect to the ground-truth communities. The ranks of the algorithms (using dense ranking) are shown within parenthesis. The average ranks of all the normal (weighted) validation measures are shown in column 9 (column 13).

Algorithms	Mod	Perm	1-Con	1-Cut	NMI	ARI	PU	Avg (N)	W-NMI	W-ARI	W-PU	Avg (W)
Louvain	0.60(1)	0.36(1)	0.77(5)	0.44(5)	0.93(1)	0.99(1)	0.89(2)	1.33	0.99(2)	0.93(2)	0.99(1)	1.67
FastGreedy	0.58(2)	0.25(3)	0.81(3)	0.59(3)	0.93(1)	0.99(1)	0.91(1)	1.00	1.00(1)	0.94(1)	0.99(1)	1.00
CNM	0.55(3)	0.20(4)	0.85(1)	0.86(1)	0.67(4)	0.75(4)	0.42(5)	4.33	0.55(5)	0.63(5)	0.71(3)	4.33
WalkTrap	0.60(1)	0.36(1)	0.82(2)	0.69(2)	0.90(2)	0.98(2)	0.84(3)	2.33	0.98(3)	0.91(3)	0.99(1)	2.33
Infomod	0.60(1)	0.35(2)	0.82(2)	0.69(2)	0.89(3)	0.97(3)	0.82(4)	3.33	0.97(4)	0.89(4)	0.98(2)	3.33
Infomap	0.60(1)	0.35(2)	0.79(4)	0.51(4)	0.89(3)	0.97(3)	0.82(4)	3.33	0.97(4)	0.89(4)	0.98(2)	3.33

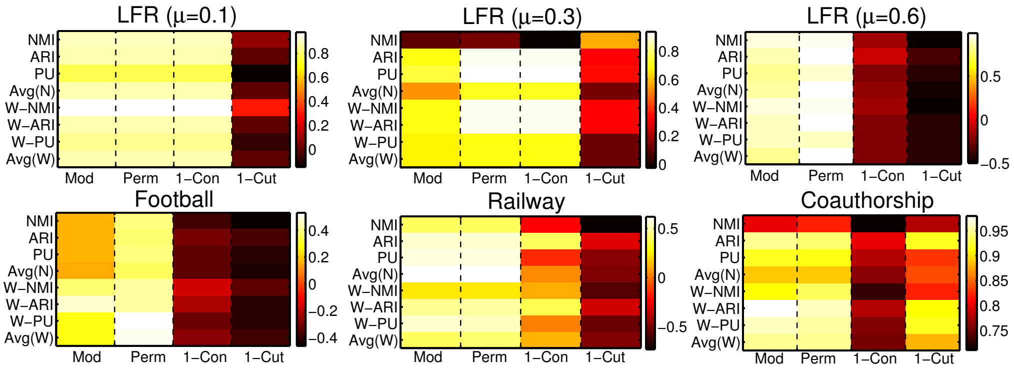


Figure 4.3: Heat maps depicting the pairwise Spearman's rank correlation between four scoring functions with six validation measures for six different networks. Avg(N) and Avg(W) are the averages of the ranks of three normal and three weighted validation measures respectively as shown in Table 4.2.

4.4 Permanence as a Community Scoring Function

In this section, we demonstrate the effectiveness of permanence as a scoring function for evaluating the goodness of detected communities, and compare it with Mod, 1-Con and 1-Cut. To do this, we perform the following experiment, on the same lines as that of [204].

The steps in our experiment are as follows: (i) we apply several community detection algorithms on a specified network and obtain the vertex-to-community assignment as given by each algorithm; (ii) we compute the values of all the community scoring functions for these communities; (iii) for each scoring function we rank the algorithms based on which one of these produces the most optimal (highest) value; (iv) we then compare the obtained community structure with the known ground-truth community structure and compute the respective validation measures, namely NMI, ARI, Purity and their weighted versions; (v) for each validation metric, we rank the algorithms based on the one that produces the highest value, i.e., best match with ground-truth.

Table 4.2 shows the results of the experiment performed on football network. Scoring functions (columns 2-5) are measures of goodness of the community set obtained. The validation metrics (columns 6-8, 10-12) measure the concurrence of the communities with the ground-truth communities. We posit that since these two types of measures are orthogonal, and because the validation metrics generally provide a stronger measure of correctness, the values of a good scoring function should “match" those of the validation metrics. That is, if a scoring function indeed identifies the correct communities, then when its value is high (low), the values of the corresponding validation metrics should also be high (low).

To compute this correlation, we compare the relative ranks, because the range of the values is not commensurate across the quantities and we are more interested in observing the “up" or “down" direction, rather than the absolute values. For each network, we measure the Spearman’s rank correlation between all pairs of scoring functions and validation measures. Note that it is not always possible to assign ranks uniquely. We used different ranking schemes to break ties. Here, we present the results using dense ranking; we have also used standard competition ranking and fractional ranking and our results are consistent across all the different methods.

Results. Table 4.2 shows the values and ranks for the different metrics for football network. For all the networks, the rank correlations of the scoring functions and the validation metrics are shown as heat maps in Figure 4.3. Lighter color indicates higher correlation and hence more similarity between the scoring function and the validation metric. For the networks having distinct community structure such as LFR ($\mu = 0.1$), football and railway networks, permanence shows comparable performance as that of other scoring functions. However for LFR network, with the increase of μ, the inter-community connection density starts increasing, and it is difficult for any community detection algorithm and/or scoring function to capture the ground-truth communities. Interestingly, we observe that the rank correlation obtained through the permanence scores and those through validation metrics is exceptionally high for LFR ($\mu = 0.6$) and coauthorship networks which seem to have poor community structure compared to the other networks (see Table 4.1). Since the ground-truth communities are not well formed, there is a wide variance in the type of community structures identified by different algorithms. Permanence score can capture this variability much better than other scoring functions.

4.5 Sensitivity of Permanence

We now evaluate the sensitivity of permanence under different perturbations of the ground-truth community structure. We posit that a good metric for evaluating communities should be stable under small perturbations of the ground-truth communities (i.e., groups of nodes that differ very slightly from the ground-truth communities). This indicates that the scoring function is robust to noise. However, if the perturbation is beyond a threshold, i.e., when the ground-truth community structure is perturbed to such an extent that it resembles a random set of nodes, then a good scoring function should assign it a low score.

Given a graph $G =< V, E >$ and *perturbation intensity* p, we start with the ground-truth community S and then modify it (i.e., change its members) by executing the perturbation strategy $p \cdot m$ times. The value of m is based on different strategies, as described below. For our experiments, we adopt three perturbation strategies motivated by the methods proposed in [231]:

(i) Edge-based perturbation picks a random inter-community edge (u, v) where $u \in S$ and $v \in S'$ (where $S \neq S'$) and then swaps the memberships (i.e., assign u to S' and v to S). It continues until $p \cdot |E|$ iterations are completed (here, $m = |E|$). This strategy preserves the size of S. However, if v is not connected to any other nodes in S except u, then it makes S disconnected.

(ii) Random perturbation takes community members and replaces them with random non-members. We pick two random nodes $u \in S$ and $v \in S'$ (where $S \neq S'$) and then swap their memberships. It continues until $p \cdot |V|$ iterations are completed (here, $m = |V|$). Random perturbation maintains the size of S but may disconnect S. Generally, it degrades

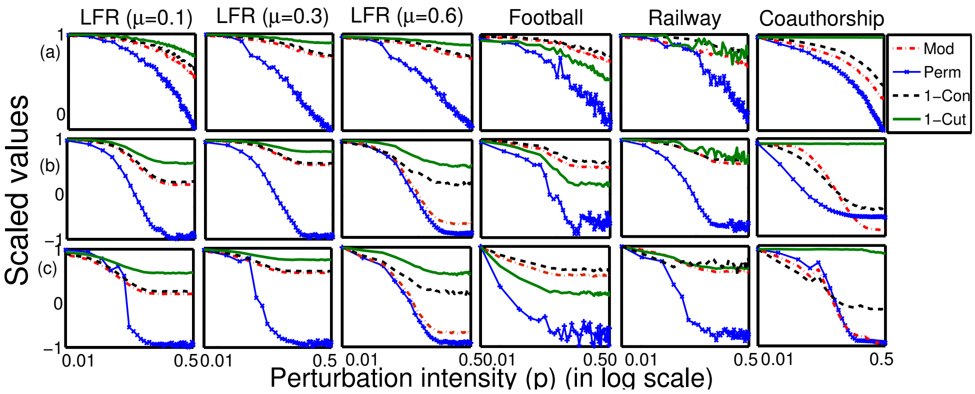


Figure 4.4: Change in the value of the scoring functions with the increase of perturbation intensity (p) in, (a) edge-based, (b) random and (c) community-based perturbation strategies. The values are normalized by the maximum value obtained from each function.

the quality of S faster than edge-based strategy, since edge-based strategy only affects the "fringe" of the community.

(iii) Community-based perturbation adopts a similar mechanism as in the edge-based strategy. However, it considers each community S from the ground-truth community structure one by one and continues the perturbation until $p \cdot |S|$ constituent nodes of the community are swapped with the other non-constituent nodes (here, $m = |S|$). This process is repeated for all the communities separately. This perturbation decreases the quality of the ground-truth communities the fastest among the three since the number of swaps is much higher than the others.

We perturb different networks using these three perturbation strategies for values of p ranging between 0.01 to 0.5. We compute the values of four community scoring functions, i.e., modularity (Mod), permanence (Perm), 1-Con and 1-Cut. For small values of p, small change of the original value of the scoring function is desirable since it indicates that the scoring function is robust to noise. For high perturbation intensities (i.e., for larger values of p), the value should drop significantly since the communities become more random.

Results. Figure 4.4 shows the results of our experiments. For a commensurate comparison, we rescale the values of each parameter by normalizing with the maximum value obtained from that function. For all three strategies, the values of the scoring functions tend to decrease with the increase of p, and the effect is most prominent in community-based strategy followed by random and edge-based strategies. For each network, once p has reached a certain threshold, the decrease is much faster in permanence. On more careful inspection, we find that this happens because the internal structure of a community completely breaks down if perturbation is taken beyond a point and thus has an avalanche effect on the value of the clustering coefficient ($c_{in}(v)$ in Equation (4.1)). This in turn quickly pulls the value of permanence down. Summarizing, the results indicate that permanence is a better measure for distinguishing true communities from randomized sets of nodes than the other parameters.

4.6 Permanence Maximization

Inspired by the effectiveness of permanence as a scoring function and its sensitivity to perturbations, we develop a community detection algorithm called **MaxPerm** (pseudocode in Algorithm 2) that identifies communities by maximizing permanence.

Our algorithm is a heuristic, that strives to obtain a high value of permanence. In this algorithm, we begin by initializing every vertex to a singleton community. A vertex is moved to a community only if this movement results in a net increase in the number of internal connections and/or a net decrease in the number of external connections to any of the neighboring communities. If such a move is not possible, then either the vertex remains as a singleton (such as in the case where moving to any one of the candidate communities will give equal permanence) or becomes a part of the community where it is more tightly connected with its neighbors (this causes the vertex to have positive permanence). This process is repeated for each vertex and the entire relocation of all vertices is repeated over several iterations until the permanence value converges. Although convergence is not theoretically guaranteed, we observe that in most cases the algorithm converges with high probability.

4.6.1 Performance Evaluation

Table 4.3 shows results of the improvement of our method (as differences) compared to the average and best performances of six competing algorithms (given in Section 4.3.4)

Algorithm 2 MaxPerm: Maximizing permanence for detecting non-overlapping communities

```
Input: A graph $G$.
Output: Permanence of $G$; Detected communities
  procedure MAX PERMANENCE($G(V, E)$)
      Each vertex is assigned to its seed community
      Set value of maximum iteration as $maxIt$
      $vertices \leftarrow |V|$
      $Sum \leftarrow 0$
      $Old\_Sum \leftarrow -1$
      $Itern \leftarrow 0$
      while $Sum \neq Old\_Sum$ and $Itern < maxIt$ do
          $Itern \leftarrow Itern + 1$
          $Old\_Sum \leftarrow Sum$
          $Sum \leftarrow 0$
          for all $v \in V$ do
              Compute current permanence of $v$
              $cur\_p \leftarrow Perm(v)$
              if $cur\_p == 1$ then
                  $Sum \leftarrow Sum + cur\_p$
                  continue;
              $cur\_p\_neig \leftarrow 0$
              for all $u \in Neig(v)$ do    /* $Neig(v)$=set of neighbors of $v$ */
                  Compute current permanence of $u$
                  $cur\_p\_neig \leftarrow cur\_p\_neig + Perm(u)$
              for all $C \in Comm(v)$ do    /* $Comm(v)$ is the set of neighboring communities of $v$ */
                  Move $v$ to community $C$
                  Compute permanence of $v$ in community $C$
                  $n\_p \leftarrow Perm(v)$
                  /* Neighbors of $v$ are affected for this movement */
                  $n\_p\_neig \leftarrow 0$
                  for all $u \in Neig(v)$ do
                      Compute new permanence of $u$
                      $n\_p\_neig \leftarrow n\_p\_neig + Perm(u)$
                  if $(cur\_p < n\_p)$ and $(cur\_p\_neig < n\_p\_neig)$ then
                      $cur\_p \leftarrow n\_p$
                  else
                      Replace $v$ to its original community
              $Sum \leftarrow Sum + cur\_p$
      $Netw\_perm = Sum/vertices$    /* Permanence of $G$ */
      return $Netw\_perm$
```

based on six ground-truth based validation metrics.

Comparable results – In LFR (μ = 0.1) and football networks, since the communities are well-separated, most algorithms efficiently capture these partitions and our method is

Table 4.3: Improvement of MaxPerm with respect to the average (left-hand value) and the best (right-hand value) performances of the six competing algorithms (separated by semicolon). Positive (negative) values indicate that MaxPerm outperforms (underperforms) the corresponding performances of the competing algorithms.

Validation metrics	LFR (μ=0.1)	LFR (μ=0.3)	LFR (μ=0.6)	Football	Railway	Coauthorship
NMI	0.04; 0.00	0.15; 0.05	-0.31; -0.78	0.04; 0.00	0.15; 0.11	0.04; -0.06
ARI	0.06; 0.00	0.21; 0.02	-0.39; -0.76	0.07; 0.00	0.03; 0.04	0.03; -0.08
PU	0.04; 0.00	0.17; 0.00	-0.38; -0.72	0.01; 0.00	0.13; 0.00	0.03; -0.06
W-NMI	0.02; 0.00	0.14; 0.00	-0.41; -0.78	0.09; 0.00	0.26; 0.00	0.05; -0.01
W-ARI	0.05; 0.02	0.19; 0.05	-0.35; -0.72	0.05; 0.00	0.02; -0.15	0.04; -0.06
W-PU	0.03; 0.01	0.17; 0.00	-0.45; -0.79	0.00; 0.00	0.05; -0.04	0.02; -0.15

comparable to the other algorithms as well.

Improved results – In LFR ($\mu = 0.3$) and railway networks, our method significantly outperforms other algorithms. Moreover in railway network, we observe that our algorithm detects three singleton communities (i.e., communities each containing only one vertex), one of which is also present in the ground-truth structure. None of the community detection algorithms is able to capture this singleton community.

Moderate results – Our method does not work well for the LFR ($\mu = 0.6$) network. For coauthorship network, we observe that though our algorithm outperforms the average performance of the competing algorithms, it performs less well than that of the two information-theoretic approaches (Infomod and Infomap).

4.6.2 Reasons behind Moderate Performance

LFR (μ=0.6) – To understand why our algorithm is not as competitive for LFR ($\mu = 0.6$), we further observe the quality of the ground-truth communities in three LFR networks. We observe that while the average internal clustering coefficient of vertices in LFR ($\mu = 0.1$) is 0.78, it deteriorates to 0.36 for LFR ($\mu = 0.6$). Moreover, 97% of vertices in ground-truth communities of LFR (μ=0.6) have less internal connections than the external connections (while LFR (μ=0.1) and LFR (μ=0.3) hardly have any such nodes). This indicates that LFR ($\mu = 0.6$) does not have modular structure in the ground-truth communities.

To further strengthen this claim, we also measure the similarity of the communities obtained by different community detection algorithms (as listed in Section 4.3.4) across different validation measures. The results in Table 4.4 clearly show the degradation of the similarity values with the increase in μ. The reason is that with the increase in μ, the communities in LFR network tend to be less well-knit, and thus the agreement of the outputs of different algorithms is also less. Therefore, the output of a good community detection algorithm should reflect such absence of modular structure in the network (hence shows poor performance).

Coauthorship network – To explain the permanence-based results obtained from coauthorship network, we further analyze the communities obtained from our algorithm. We check the title and the abstract of the papers written by the authors in each community of coauthorship network, and notice that our method splits large ground-truth communities into denser submodules. This splitting is mostly noticed in older research areas such as Algorithms and Theory, Databases etc. These submodules are actually the subfields (sub-communities) of a field (community) in computer science domain.

Table 4.4: Average values among pairwise similarities between outputs of the community detection algorithms on different LFR networks.

Validation measures	LFR (μ=0.1)	LFR (μ=0.3)	LFR (μ=0.6)
NMI	0.95	0.82	0.53
ARI	0.98	0.79	0.48
PU	0.99	0.85	0.56
W-NMI	0.94	0.85	0.54
W-ARI	0.97	0.78	0.50
W-PU	0.98	0.83	0.57

Comparison of largest community size. Many optimization algorithms have the tendency to underestimate smaller size communities (known as the resolution limit problem [81]) and sometimes tend to produce very large size communities. In our test suite, we observe a similar tendency in all the competing algorithms whereas the communities obtained by permanence are smaller in size. In Table 4.5, we show for two representative networks that the size of the largest communities detected by the other algorithms is much larger than the size of the largest community present in the ground-truth structure. We also measure the maximum similarity (using Jaccard coefficient) between the largest-size community detected by each algorithm with the communities in ground-truth structure and notice that

Table 4.5: Size of the largest communities obtained from different community detection algorithms and their similarities with the ground-truth structure for two networks (LFR (μ=0.3, N=1000) and football).

	Largest community size		Similarity	
	LFR ($\mu = 0.3$)	Football	LFR ($\mu = 0.3$)	Football
Ground-truth	49	12	–	–
Louvain	62	24	0.70	0.41
FastGreedy	95	18	0.32	0.65
CNM	91	32	0.52	0.31
Walktrap	83	15	0.51	0.57
Infomod	61	16	0.79	0.86
Infomap	59	16	0.74	0.86
MaxPerm	49	13	1	0.92

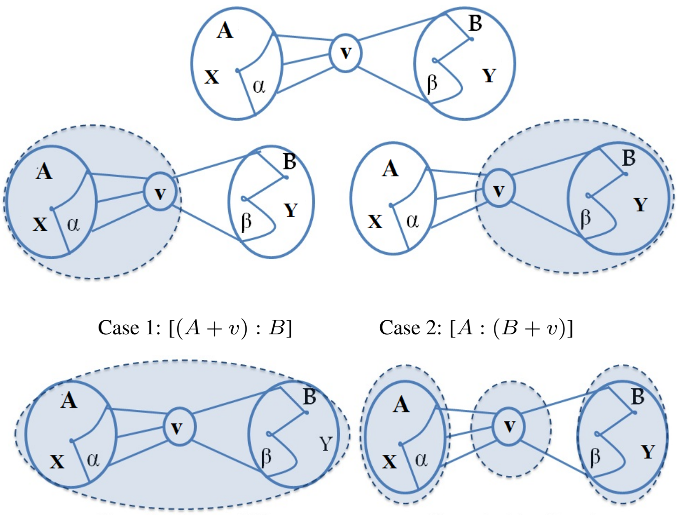
MaxPerm is able to detect largest size community which is most similar to the ground-truth structure. Therefore, we hypothesize that our algorithm has the potential to reduce the effect of resolution limit (see Section 4.7 for theoretical proofs).

4.7 Handling Modularity Maximization Issues

As discussed earlier, modularity maximization algorithms suffer from the issues including (a) resolution limit, (b) degeneracy of solution and (c) dependence on the size of the graph [81]. We now discuss how each of these problems are ameliorated by maximizing permanence.

We demonstrate that community assignments are different in a modularity-based algorithm vis-a-vis **MaxPerm** algorithm using the example in Figure 4.5. In this figure, we assume that apart from the edges through v, there is no connection between the communities A and B.

As we see in Figure 4.5, two communities A and B are connected via a vertex v. The vertex v has connections to α nodes in community A and to β nodes in community B, and these nodes form the set N_α and N_β respectively. The average internal degree of a vertex, $a \in N_\alpha$ ($b \in N_\beta$), before v is added is I_α (I_β). Similarly, the average internal clustering coefficient of a vertex, $a \in N_\alpha$ ($b \in N_\beta$), before v is added is C_A (C_B). We assume the



Case 1: $[(A+v):B]$ Case 2: $[A:(B+v)]$

Case 3: $[(A+v+B)]$ Case 4: $[A:B:v]$

Figure 4.5: An illustrative example to show the community assignment of vertex v. These are used to demonstrate four Lemmas.

values of C_A and C_B to be at least 0.5. Communities A and B have no other connections except those through v. We also assume that $\alpha \geq \beta$.

When v is added to communities A (B) then the average internal clustering coefficient of v becomes C^v_A (C^v_B), and the average clustering coefficient of the nodes in $N_\alpha(N_\beta)$ become C^α (C^β). We consider two extreme values of $C^\alpha(C^\beta)$. One case is when the nodes in the community are tightly connected and adding v does not significantly change the internal clustering coefficient. In this case, we assume $C^\alpha = C_A$ and $C^\beta = C_B$. The other case is when the nodes in the community are not as tightly connected. In this case, adding v decreases the average internal clustering coefficient.

Let the number of internal connections of nodes in N_α, before v is added, be f_x. Therefore $C_A = \frac{fx}{I_\alpha(I_\alpha-1)}$. In the second case when the communities are sparse, once v is added we assume that no new distinct connections among the pair of neighbors of v are formed, but the internal degree increases by one. Therefore $C^\alpha = \frac{fx}{(I_\alpha+1)I_\alpha} = \frac{C_A I_\alpha(I_\alpha-1)}{(I_\alpha+1)I_\alpha} = \frac{C_A(I_\alpha-1)}{(I_\alpha+1)}$. Similarly $C^\beta = \frac{C_B(I_\beta-1)}{(I_\beta+1)}$.

The combination of communities A, B and the vertex v can have four cases as follows:

- **Case 1.** v joins with community A only. We denote this configuration as $[(A+v):B]$, and its total permanence as $P_{(A+v):B}$. We assume that the combined permanence of all nodes $x \notin (N_\alpha \cup N_\beta \cup v)$ as P_x. This value will not be affected due to the re-assignments. Therefore, the total permanence is the sum of the following factors: P_x, $[\alpha C^\alpha]$ (for the nodes in N_α connected to v), $[\frac{\alpha}{(\alpha+\beta)\beta} - (1 - C_A^v)]$ (for vertex v) and $[\beta(\frac{I_\beta}{I_\beta+1} - (1 - C^\beta))]$ (for the nodes in N_β).

  $P_{(A+v):B} = P_x + \alpha C^\alpha + \frac{\alpha}{(\alpha+\beta)\beta} - (1 - C_A^v) + \beta(\frac{I_\beta}{I_\beta+1} - (1 - C_B))$

- **Case 2.** v joins with community B only. We denote this configuration as $[(A:(v+B)]$, and its total permanence as $P_{A:(v+B)}$. The values of this total permanence is the sum of the following factors: P_x, $[\alpha(\frac{I_\alpha}{I_\alpha+1} - (1 - C^\alpha))]$ (for the nodes in N_α), $[\frac{\beta}{(\alpha+\beta)\alpha} - (1 - C_B^v)]$ (for vertex v) and $[\beta C^\beta]$ (for the nodes in N_β connected to v).

  $P_{A:(v+B)} = P_x + \alpha(\frac{I_\alpha}{I_\alpha+1} - (1 - C_A)) + \frac{\beta}{(\alpha+\beta)\alpha} - (1 - C_B^v) + \beta C^\beta$

- **Case 3.** A, B and v merge together. We denote this configuration as $[(A+v+B)]$, and its total permanence as $P_{(A+v+B)}$. The values of this total permanence is the sum of the following factors: P_x, $[\alpha C^\alpha]$ (for the nodes in N_α), $[\frac{\alpha(\alpha-1)C_A^v + \beta(\beta-1)C_B^v}{(\alpha+\beta)(\alpha+\beta-1)}]$ (for vertex v) and $[\beta C^\beta]$ (for the nodes in N_β connected to v).

  $P_{(A+v+B)} = P_x + \alpha C^\alpha + \frac{\alpha(\alpha-1)C_A^v + \beta(\beta-1)C_B^v}{(\alpha+\beta)(\alpha+\beta-1)} + \beta C^\beta$

- **Case 4.** A, B and v remain as separate communities. We denote this configuration as $[(A:v:B)]$, and its total permanence as $P_{(A:v:B)}$. The values of this total permanence is the sum of the following factors: P_x, $[\alpha(\frac{I_\alpha}{I_\alpha+1} - (1 - C^\alpha))]$ (for the nodes in N_α), 0 (for vertex v) and $[\beta(\frac{I_\beta}{I_\beta+1} - (1 - C^\beta))]$ (for the nodes in N_β).

  $P_{(A:v:B)} = P_x + \alpha(\frac{I_\alpha}{I_\alpha+1} - (1 - C_A)) + \beta(\frac{I_\beta}{I_\beta+1} - (1 - C_B))$

Lemma 4.1 *Given $C^\alpha = C_A$ and $C^\beta = C_B$, let $Z = \frac{\alpha-\beta}{\alpha\beta} + (C_A^v - C_B^v) + \left(\frac{\alpha}{I_\alpha+1} - \frac{\beta}{I_\beta+1}\right)$. The assignment $[(A+v):B]$ will have a higher permanence than $[A:(v+B)]$, if $Z > 0$ and a lower permanence if $Z < 0$.*

Proof. Here we are comparing between Case 1 and Case 2. The difference in total permanence between these two assignments by assuming $C^{\alpha} = C_A$ and $C^{\beta} = C_B$ is:

$$\begin{aligned} P_{(A+v):B} - P_{A:(v+B)} &= \frac{\alpha}{(\alpha+\beta)\beta} + C_A^v + \beta(\frac{I_\beta}{I_\beta+1} - 1) \\ &- (\alpha(\frac{I_\alpha}{I_\alpha+1} - 1) + \frac{\beta}{(\alpha+\beta)\alpha} + C_B^v) \\ &= \frac{\alpha}{(\alpha+\beta)\beta} - \frac{\beta}{(\alpha+\beta)\alpha} + (C_A^v - C_B^v) + (-\beta\frac{1}{I_\beta+1} - \alpha\frac{-1}{I_\alpha+1}) \\ &= \frac{\alpha-\beta}{\alpha\beta} + (C_A^v - C_B^v) + (\frac{\alpha}{I_\alpha+1} - \frac{\beta}{I_\beta+1}) \end{aligned} \tag{4.2}$$

If this difference is greater than zero then $[(A+v):B]$ will have a higher permanence. If the difference is less than zero then $[A:(v+B)]$ will have higher permanence.

Lemma 4.2 Merging the communities A, B and v, gives higher permanence than joining v to community A if $C^{\beta} = C_B$, and $\frac{\gamma}{(\gamma+1)\beta} + \frac{C_A^v(2\gamma+1) - C_B^v}{(\gamma+1)^2} - \frac{\beta}{I_\beta+1} < 1$; where $\gamma = \alpha/\beta$, and also if $C^{\beta} = C_B\frac{I_\beta-1}{I_\beta+1}$, and $\frac{\gamma}{(\gamma+1)\beta} + \frac{C_A^v(2\gamma+1) - C_B^v}{(\gamma+1)^2} + \frac{\beta(2C_B-1)}{I_\beta+1} < 1$.

Proof. We are comparing Case 1 and Case 3 and in this case $C^{\beta} = C_B\frac{I_\beta-1}{I_\beta+1}$. The difference in total permanence is:

$$\begin{aligned} P_{(A+v):B} - P_{(A+v+B)} &= \frac{\alpha}{(\alpha+\beta)\beta} - 1 + C_A^v + \beta(\frac{I_\beta}{I_\beta+1} - 1 + C_B) - (\frac{\alpha(\alpha-1)C_A^v + \beta(\beta-1)C_B^v}{(\alpha+\beta)(\alpha+\beta-1)} + \beta C^{\beta}) \\ &= \frac{\alpha}{(\alpha+\beta)\beta} - 1 + C_A^v - \frac{\beta}{I_\beta+1} + \beta(C_B - C^{\beta}) - \frac{\alpha(\alpha-1)C_A^v + \beta(\beta-1)C_B^v}{(\alpha+\beta)(\alpha+\beta-1)} \\ &\text{Substituting } \gamma = \alpha/\beta \text{ and } C^{\beta} = C_B\tfrac{I_\beta-1}{I_\beta+1} \\ &= \frac{\gamma}{(\gamma+1)\beta} - 1 + C_A^v - \frac{\beta}{I_\beta+1} + \beta C_B\frac{2}{I_\beta+1} - \frac{\gamma(\gamma-1/\beta)C_A^v + (1-1/\beta)C_B^v}{(\gamma+1)(\gamma+1-1/\beta)} \end{aligned} \tag{4.3}$$

The value of $1/\beta$ will become lower as β increases. We therefore ignore its effect. The equation then becomes:

$$\begin{aligned} P_{(A+v):B} - P_{(A+v+B)} &= \frac{\gamma}{(\gamma+1)\beta} - 1 + C_A^v - \frac{\beta}{I_\beta+1} + \beta C_B\frac{2}{I_\beta+1} \\ &- \frac{\gamma^2 C_A^v}{(\gamma+1)(\gamma+1)} - \frac{C_B^v}{(\gamma+1)(\gamma+1)} \\ &= \frac{\gamma}{(\gamma+1)\beta} - 1 + \frac{C_A^v(2\gamma+1) - C_B^v}{(\gamma+1)^2} + \beta\frac{2C_B-1}{I_\beta+1} \end{aligned} \tag{4.4}$$

If this difference is less than 0 then higher permanence is obtained by merging. Therefore, the condition to merge A, B and v altogether rather than v joining with A is: $\frac{\gamma}{(\gamma+1)\beta} + \frac{C_A^v(2\gamma+1) - C_B^v}{(\gamma+1)^2} + \beta\frac{2C_B-1}{I_\beta+1} < 1$.

If we consider $C^{\beta} = C_B$, then

$$P_{(A+v):B} - P_{(A+v+B)} = \frac{\gamma}{(\gamma+1)\beta} - 1 + \frac{C_A^v(2\gamma+1) - C_B^v}{(\gamma+1)^2} - \frac{\beta}{I_\beta + 1} \tag{4.5}$$

In this case, the condition to merge is:

$$\frac{\gamma}{(\gamma+1)\beta} + \frac{C_A^v(2\gamma+1) - C_B^v}{(\gamma+1)^2} - \frac{\beta}{I_\beta + 1} < 1 \tag{4.6}$$

`Corollary 4.5` *If $\beta = 1$, $C^{\beta} = C_B \frac{I_\beta - 1}{I_\beta + 1}$, $C_A^v > 1/2$ then v will join community A rather than the three pieces merging.*

`Corollary 4.6` *If $C_B \approx 1$, $C^{\beta} = C_B \frac{I_\beta - 1}{I_\beta + 1}$, $\beta \geq I_\beta + 1$ and $C_A^v \geq C_B^v/3$ then v will join community A rather than the three pieces merging.*

Proof of Corollary 4.5: If $\beta = 1$, then

$$\begin{aligned}
P_{(A+v):B} - P_{(A+v+B)} &= \frac{\gamma}{(\gamma+1)\beta} - 1 + C_A^v - \frac{\beta}{I_\beta+1} + \beta C_B \frac{2}{I_\beta+1} - \frac{(\gamma(\gamma-1/\beta)C_A^v + (1-1/\beta)C_B^v}{(\gamma+1)(\gamma+1-1/\beta)} \\
&= \frac{\gamma}{(\gamma+1)} - 1 + C_A^v + \frac{(2C_B-1)}{I_\beta+1} - \frac{\gamma(\gamma-1)C_A^v}{(\gamma+1)(\gamma)} \\
&= \frac{-1}{(\gamma+1)} + C_A^v \frac{2\gamma}{\gamma+1)\gamma} + \frac{(2C_B-1)}{I_\beta+1} \\
&= \frac{2C_A^v - 1}{(\gamma+1)} + \frac{(2C_B-1)}{I_\beta+1}
\end{aligned} \tag{4.7}$$

This value will be positive so long as $C_A^v > 1/2$. In this case, joining v to community A is favored.

Proof of Corollary 4.6: If $C_B \approx 1$, $\beta \geq I_\beta + 1$ and $C_A^v \geq C_B^v$, then

$$\begin{aligned}
P_{(A+v):B} - P_{(A+v+B)} &= \frac{\gamma}{(\gamma+1)\beta} - 1 + C_A^v - \frac{\beta}{I_\beta+1} + \beta C_B \frac{2}{I_\beta+1} - \frac{\gamma(\gamma-1/\beta)C_A^v + (1-1/\beta)C_B^v}{(\gamma+1)(\gamma+1-1/\beta)} \\
&\text{Ignore } 1/\beta\text{, because } \beta \text{ is high} \\
&= \frac{\gamma}{(\gamma+1)\beta} - 1 + C_A^v + \beta\frac{2C_B-1}{I_\beta+1} - \frac{\gamma(\gamma-1/\beta)C_A^v}{(\gamma+1)(\gamma+1)} - \frac{C_B^v}{(\gamma+1)(\gamma+1)} \\
&= \frac{\gamma}{(\gamma+1)\beta} + \frac{C_A^v(2\gamma+1) - C_B^v}{(\gamma+1)(\gamma+1)} + (\frac{\beta}{I_\beta+1} - 1)
\end{aligned} \tag{4.8}$$

The first term is positive. Since the smallest value of $\gamma = 1$, and $C_A^v \geq C_B^v$, the second term is positive, and since $\beta \geq I_\beta + 1$, the third term is also positive. Therefore v will join community A rather than merging all the components.

Lemma 4.3 *If $C^{\alpha} = C_A$ and $C^{\beta} = C_B$ then the communities will merge (i.e., $[(A{+}v{+}B)]$), rather than remain separate (i.e., $[A : B : C]$). If $C^{\alpha} = C_A\frac{(I_\alpha-1)}{(I_\alpha+1)}$ $C^{\beta} = C_B\frac{(I_\beta-1)}{(I_\beta+1)}$ and then the communities will merge if: $\frac{\gamma^2 C_A^v + C_B^v}{(\gamma+1)^2} > \alpha\frac{(2C_A-1)}{I_\alpha+1} + \beta\frac{(2C_B-1)}{I_\beta+1}$.*

Proof: We are comparing Case 3 and Case 4, and the case $C^{\beta} = C_B\frac{I_\beta-1}{I_\beta+1}$. The difference in total permanence is:

$$\begin{aligned}
P_{(A+v+B)} - P_{(A:v:B)} &= \alpha C^{\alpha} + \frac{\alpha(\alpha-1)C_A^v + \beta(\beta-1)C_B^v}{(\alpha+\beta)(\alpha+\beta-1)} + \beta C^{\beta} - (\alpha(\frac{I_\alpha}{I_\alpha+1} - (1-C_A)) + \beta(\frac{I_\beta}{I_\beta+1} - (1-C_B))) \\
&= -\alpha(C_A - C^{\alpha}) - \beta(C_B - C^{\beta}) + \frac{\alpha(\alpha-1)C_A^v + \beta(\beta-1)C_B^v}{(\alpha+\beta)(\alpha+\beta-1)} + \frac{\alpha}{I_\alpha+1} + \frac{\beta}{I_\beta+1} \\
&= \frac{\alpha(\alpha-1)C_A^v + \beta(\beta-1)C_B^v}{(\alpha+\beta)(\alpha+\beta-1)} + \frac{\alpha}{I_\alpha+1} + \frac{\beta}{I_\beta+1} - (\alpha\frac{2C_A}{I_\alpha+1} + \beta\frac{2C_B}{I_\beta+1}) \\
&\text{Substituting } \gamma = \alpha/\beta \\
&= \frac{\gamma(\gamma-1/\beta)C_A^v + (1-1/\beta)C_B^v}{(\gamma+1)(\gamma+1-1/\beta)} - (\alpha\frac{(2C_A-1)}{I_\alpha+1} + \beta\frac{(2C_B-1)}{I_\beta+1})
\end{aligned} \tag{4.9}$$

The value of $1/\beta$ will become lower as β increases. We therefore ignore its effect. The equation then becomes

$$P_{(A+v+B)} - P_{(A:v:B)} = \frac{\gamma^2 C_A^v + C_B^v}{(\gamma+1)^2} - (\alpha\frac{(2C_A-1)}{I_\alpha+1} + \beta\frac{(2C_B-1)}{I_\beta+1}) \tag{4.10}$$

If $C^{\alpha} = C_A$ and $C^{\beta} = C_B$, then

$$\begin{aligned}
P_{(A+v+B)} - P_{(A:v:B)} &= \alpha C_A + \frac{\alpha(\alpha-1)C_A^v + \beta(\beta-1)C_B^v}{(\alpha+\beta)(\alpha+\beta-1)} + \beta C_B - (\alpha(\frac{I_\alpha}{I_\alpha+1} - (1-C_A)) + \beta(\frac{I_\beta}{I_\beta+1} - (1-C_B))) \\
&= \frac{\alpha(\alpha-1)C_A^v + \beta(\beta-1)C_B^v}{(\alpha+\beta)(\alpha+\beta-1)} + \frac{\alpha}{I_\alpha+1} + \frac{\beta}{I_\beta+1}
\end{aligned} \tag{4.11}$$

This value is always positive so the communities will merge.

Lemma 4.4 *If $C^{\alpha} = C_A$ and $C^{\beta} = C_B$ then the communities will remain separate (i.e., $[A : v : B]$) rather than v joining with community A (i.e., $[(A+v) : B]$), if $\alpha(\frac{1}{I_\alpha+1} + \frac{1}{(\alpha+\beta)\beta}) < (1 - C_A^v)$. Otherwise, if $C^{\alpha} = C_A\frac{(I_\alpha-1)}{(I_\alpha+1)}$; $C^{\beta} = C_B\frac{(I_\beta-1)}{(I_\beta+1)}$ and then the communities will remain separate if $\alpha(\frac{2C_A-1}{I_\alpha+1}) + (1 - C_A^v) \geq \frac{\alpha}{(\alpha+\beta)\beta}$*

Proof: We are comparing Case 1 and Case 4 for the case $C^{\alpha} = C_A\frac{(I_\alpha-1)}{(I_\alpha+1)}$; $C^{\beta} = C_B\frac{(I_\beta-1)}{(I_\beta+1)}$.

The difference in total permanence is:

$$\begin{aligned} P_{(A+v):B} - P_{(A:v:B)} &= \alpha C^{\alpha} + \frac{\alpha}{(\alpha+\beta)\beta} - (1 - C^v_A) - (\alpha(\frac{I_\alpha}{I_\alpha+1} - (1 - C_A)) \\ &= \alpha(C^{\alpha} - C_A + \frac{1}{I_\alpha+1}) + \frac{\alpha}{(\alpha+\beta)\beta} + (C^v_A - 1) \\ &= \alpha(\frac{1-2C_A}{I_\alpha+1}) + \frac{\alpha}{(\alpha+\beta)\beta} + (C^v_A - 1) \end{aligned} \tag{4.12}$$

This value will be negative (favor merge) if: $\alpha(\frac{2C_A-1}{I_\alpha+1} + (1 - C^v_A)) > \frac{\alpha}{(\alpha+\beta)\beta}$.

If we consider the case $C^{\alpha} = C_A$ and $C^{\beta} = C_B$, then

$$\begin{aligned} P_{(A+v):B} - P_{(A:v:B)} &= \alpha C^{\alpha} + \frac{\alpha}{(\alpha+\beta)\beta} - (1 - C^v_A) - (\alpha(\frac{I_\alpha}{I_\alpha+1} - (1 - C_A)) \\ &= \frac{\alpha}{I_\alpha+1} + \frac{\alpha}{(\alpha+\beta)\beta} + (C^v_A - 1) \\ &= \alpha(\frac{1}{I_\alpha+1} + \frac{1}{(\alpha+\beta)\beta}) \\ &+ (C^v_A - 1) \end{aligned} \tag{4.13}$$

This value will be negative (favor merge) if: $\alpha(\frac{1}{I_\alpha+1}) + \frac{1}{(\alpha+\beta)\beta}) < (1 - C^v_A)$.

Corollary 4.7 *If $\alpha = \beta$, $C^{\beta} = C_B\frac{I_\beta-1}{I_\beta+1}$, $C^v_A = C^v_B$ then communities A, B and v will merge, rather than v joining community A, if $\frac{1}{2\beta} + \frac{C^v_A}{2} + \beta\frac{2C_B-1}{I_\beta+1} < 1$.*

Corollary 4.8 *If $\alpha = \beta$, $C^{\beta} = C_B\frac{I_\beta-1}{I_\beta+1}$, then communities A, B and v will remain separate rather than v joining community A, if $\alpha(\frac{2C_A-1}{I_\alpha+1}) + (1 - C^v_A) \geq \frac{1}{2\alpha}$. If $\alpha = \beta = 1$, then $C^v_A = 0$ the communities will remain always separated.*

Proof of Corollary 4.7: If $\alpha = \beta$, $C^v_A = C^v_B$, $C^{\beta} = C_B\frac{I_\beta-1}{I_\beta+1}$, then comparing Case 1 and Case 3 we get

$$\begin{aligned} P_{(A+v+B)} - P_{(A:v:B)} &= \frac{\gamma}{(\gamma+1)\beta} - 1 + \frac{C^v_A(2\gamma+1) - C^v_B}{(\gamma+1)^2} + \beta\frac{2C_B-1}{I_\beta+1} \\ &= \frac{1}{2\beta} - 1 + \frac{3C^v_A - C^v_A}{4} + \beta\frac{2C_B-1}{I_\beta+1} \\ &= \frac{1}{2\beta} - 1 + \frac{2C^v_A}{4} + \beta\frac{2C_B-1}{I_\beta+1} \end{aligned} \tag{4.14}$$

This values is negative(favors merging) if $\frac{1}{2\beta} + \frac{C^v_A}{2} + \beta\frac{2C_B-1}{I_\beta+1} < 1$.

Proof of Corollary 4.8: If $\alpha = \beta$, $C^{\beta} = C_B\frac{I_\beta-1}{I_\beta+1}$, $C^v_A = C^v_B$ then

$$\begin{aligned} P_{(A+v):B} - P_{(A:v:B)} &= \alpha(\frac{1-2C_A}{I_\alpha+1}) + \frac{\alpha}{(\alpha+\beta)\beta} + (C^v_A - 1) \\ &= \alpha(\frac{1-2C_A}{I_\alpha+1}) + \frac{1}{2\alpha} + (C^v_A - 1) \end{aligned} \tag{4.15}$$

This value will be negative (favor merge) if: $\alpha(\frac{2C_A-1}{I_\alpha+1}) + (1 - C^v_A)) > \frac{1}{2\alpha}$.

If $\alpha = 1$, then $C^v_A = 0$. Then the condition is: $\frac{2C_A-1}{I_\alpha+1} + 1 > \frac{1}{2}$, Which is always true.

Degeneracy of solution is a problem where a community scoring function (e.g., modularity) admits multiple distinct high-scoring solutions and typically lacks a clear global maximum, thereby, resorting to tie-breaking [81]. For our example, when α = β, modularity maximization algorithm will assign v arbitrarily to A or B. However, in the case of permanence as we see in the earlier proofs, v will remain as a separate community so long as the following condition is maintained:

Condition. If $\alpha = \beta$, $C^\beta = C_B\frac{I_\beta-1}{I_\beta+1}$, then communities A, B and v will remain separate rather than v joining community A, if $\alpha(\frac{2C_A-1}{I_\alpha+1}) + (1 - C^v_A) \geq \frac{1}{2\alpha}$.

We observe that when $\alpha = \beta = 1$, then $C^v_A = 0$ and the communities will always remain separate. Furthermore, as α increases, the left-hand side of the above condition will become larger than the right, thus increasing the chance of separate communities.

Resolution limit is a problem where communities of certain small size are merged into larger ones [81]. A classic example where modularity cannot identify communities of small size is a cycle of m cliques. Here maximum modularity is obtained if two neighboring cliques are merged.

In the case of permanence as we see in the earlier proofs, we can determine that whether two communities A and B would merge (as in modularity) or whether v would join community A (we select A, but similar analysis can also be done for the case when v joins B), by the following condition:

Condition. Joining v to community A gives higher permanence than merging the communities A, B and v if $C^\beta = C_B$, and ($\frac{\gamma}{(\gamma+1)\beta} + \frac{C^v_A(2\gamma+1)-C^v_B}{(\gamma+1)^2} - \frac{\beta}{I_\beta+1}$)>1; where $\gamma = \alpha/\beta$ and

Table 4.6: Change in scoring functions with the (near-)symmetric growth of coauthorship network. N: number of nodes, c: number of communities, $< I >$: average internal degree, $< k >$: average degree, $< c_{in} >$: average internal clustering coefficient, $< E_{max} >$: average maximum external connectivity. The value of permanence is less affected by the growth.

Coauthorship	Network properties	n	964	1515	1991	2681	3386	4836	6284	7814	9001	10386
		c	24	24	24	24	24	24	24	24	24	24
		$\frac{<I>}{<k>}$	0.082	0.095	0.093	0.091	0.089	0.104	0.111	0.112	0.115	0.113
		$\frac{1}{<E_{max}>}$ $(\times 10^{-4})$	3.8	3.2	2.9	3.9	2.8	2.11	2.39	2.92	2.69	3.22
		$< (1 - c_{in}) >$	0.239	0.248	0.246	0.251	0.251	0.260	0.265	0.269	0.270	0.274
	Modularity		0.369	0.374	0.395	0.392	0.421	0.422	0.465	0.471	0.493	0.501
	Permanence		0.094	0.092	0.092	0.096	0.095	0.095	0.097	0.097	0.097	0.098

also if $C^{\beta} = C_B \frac{I_{\beta}-1}{I_{\beta}+1}$, *and* $(\frac{\gamma}{(\gamma+1)\beta} + \frac{C_A^v(2\gamma+1)-C_B^v}{(\gamma+1)^2} + \frac{\beta(2C_B-1)}{I_{\beta}+1}) > 1$.

This result is independent of the size of the communities. Moreover, so long as A and B are almost cliques (internal clustering coefficients > 0.5), C_A^v is sufficiently high and C_B^v is sufficiently small (e.g., C_A^v >2/3 and C_B^v=0), v will join community A rather than merging. Thus, in general, *the highest permanence is obtained if v joins the community to which it is very tightly connected rather than the one to which it is loosely connected.*

Asymptotic growth of value of a metric implies a strong dependence on the size of the network and the number of modules the network contains [81]. Rewriting Equation 4.1, we get the permanence of the entire network G as follows: $Perm(G) = \frac{1}{|V|}\sum_{v \in V}\left[\frac{I(v)}{D(v)E_{max}(v)}\right] - \frac{1}{|V|}\sum_{v \in V}[(1 - c_{in}(v))]$. We can notice that most of the parameters in the above formula are independent of the network size and the number of communities. Table 4.6 illustrates that with the symmetric growth of the network size in coauthorship network, the modularity increases consistently, while permanence remains almost constant.

4.8 Permanence and Overlapping Community Structure

In Section 4.2, we showed that the extent of membership of a vertex to a community depends on the following two factors: (i) the distribution of external connections of the vertex to individual communities, and (ii) the density of its internal connections. Based on these observations, we proposed a vertex-based scoring function, called *permanence* to measure the extent to which a vertex belongs to a *non-overlapping community*. Here, we formulate a *generalized version* of this metric called *overlapping permanence* (abbreviated as *OPerm*) that, although is developed for overlapping community, translates to the non-overlapping case under special boundary conditions. Note that this is one of the rarest formulations which can be useful for both non-overlapping and overlapping community analysis.

We formulate OPerm for each node as follows: consider a vertex v with degree $D(v)$ that belongs to a set of communities, C. OPerm is computed by taking into account two factors that determine the membership of the vertex in the communities.

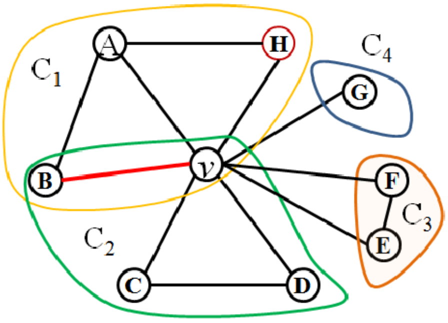
The first factor measures the extent to which other vertices "pull" v towards their communities. There are two types of pull: internal and external. The internal pull is computed as $I^c(v) = \sum_{e \in \Gamma_v^c} \frac{1}{x_e}$, where Γ_v^c denotes the set of internal edges of v in community c, and x_e for an edge $e = (u, v)$ denotes the number of communities that both the vertices u and v (i.e., the edge e) share. The total internal pull of v, over all the communities in C is given by $I(v) = \sum_{c \in C} I^c(v)$. The external pull measures how well the vertex is connected to vertices in communities not in C. Let $E_{max}(v)$ be the maximum pull from an external community. This represents the largest force that can pull v away from its current community set. The opposing internal and external pulls experienced by the vertex v appropriately weighted by its degree are represented by $\frac{I^c(v)}{E_{max}(v)} \times \frac{1}{D(v)}$. If $E_{max}(v)$ is zero, we set this factor to 1.

The second factor in deciding community membership is how well the vertex is integrated within each of its constituent communities. This is given by the internal clustering coefficient $c_{in}^c(v)$ of v in community $c \in C$. If the number of internal vertices is less than three, we set the internal clustering coefficient to 1. This value is weighted by the fraction indicating the extent of internal pull and we measure this integration as $(1 - c_{in}^c(v)) \cdot \frac{I^c(v)}{I(v)}$.

Taking these two factors together, we compute *overlapping permanence*, $P_{ov}^c(v)$ of v in community c as follows:

$$P_{ov}^c(v) = \frac{I^c(v)}{E_{max}(v)} \times \frac{1}{D(v)} - (1 - c_{in}^c(v)) \cdot \frac{I^c(v)}{I(v)} \tag{4.16}$$

Equation 4.16 ensures that the value of $P_{ov}^c(v)$ is between 1 (vertex is completely integrated



$$D(v) = 8,\ I(v) = 5$$

$$P_{ov}^{C_1}(v) = \frac{1+1+\frac{1}{2}}{2\times 8} - (1-\tfrac{2}{3}) \times \frac{(1+1+\frac{1}{2})}{5} = -0.01$$

$$P_{ov}^{C_2}(v) = \frac{1+1+\frac{1}{2}}{2\times 8} - (1-\tfrac{1}{3}) \times \frac{(1+1+\frac{1}{2})}{5} = -0.18$$

$$P_{ov}(v) = P_{ov}^{C_1}(v) + P_{ov}^{C_2}(v) = -0.19$$

Figure 4.6: Toy example depicting OPerm of a vertex v which belongs to both C_1 and C_2 and has two external neighboring communities, C_3 and C_4. The red-colored edge shares membership in both C_1 and C_2.

within a clique) to -1 (vertex is wrongly assigned). A vertex in a singleton community (i.e., community with only one vertex) will have $P_{ov}^c(v)$ as zero. The total OPerm of v over all its communities $c \in C$ is computed as $P_{ov}(v) = \sum_{c\in C} P_{ov}^c(v)$. The total OPerm of the network, with the vertex set V, is the average of the OPerm values of all the vertices, i.e., $P_{ov} = \frac{1}{|V|}\sum_{v\in V} P_{ov}(v)$. Equation 4.16 indeed reduces to the non-overlapping case (see Equation 4.1) when the number of internal communities is 1. An example of how overlapping permanence is computed is presented in Figure 5.16.

4.9 Experimental Setup

In this section, we briefly discuss the state-of-the-art metrics and algorithms that are used in the experiments presented in the subsequent sections.

4.9.1 Datasets

To study the properties of OPerm, we observe its behavior on LFR[5] benchmark networks [121] that take into account heterogeneity in degree and community size distributions of a network (as mentioned in Section 4.3.1). Unless otherwise stated, LFR

[5] http://sites.google.com/site/andrealancichinetti/files.

Table 4.7: Properties of the real-world networks used in the experiments. N: number of nodes, E: number of edges, C: number of communities, ρ: average edge-density per community, S: average size of a community, $\bar{O}_m$: average number of community memberships per node.

Networks	Node type	Edge type	Community type	N	E	C	ρ	S	$\bar{O}_m$	Reference
LiveJournal	User	Friendship	User-defined group	3,997,962	34,681,189	310,092	0.536	40.02	3.09	[232]
Amazon	Product	Co-purchased products	Product category	334,863	925,872	151,037	0.769	99.86	14.83	[232]
Youtube	User	Friendship	User-defined group	1,134,890	2,987,624	8,385	0.732	43.88	2.27	[232]
Orkut	User	Friendship	User-defined group	3,072,441	117,185,083	6,288,363	0.245	34.86	95.93	[232]
Flickr	User	Friendship	Joined group	80,513	5,899,882	171	0.046	470.83	18.96	[219]
Coauthorship	Researcher	Collaborations	Research area	391,526	873,775	8,493	0.231	393.18	10.45	[170]

graph is generated with the following configuration: μ = 0.2, N=1000, O_m=4, O_n=5%; other parameters being set to their default values.

We also use six real networks whose underlying ground-truth community structures are known a priori. The properties of these networks are summarized in Table 4.7.

Sampling real-world networks. As noted in [232], most of the baseline community detection algorithms (mentioned in Section 4.9.3) do not scale for networks of large size. Therefore, we use the following technique proposed by Yan and Leskovec [232] to obtain several small subnetworks with overlapping community structure from the large real networks. We pick a random node u in the given graph G that belongs to at least two communities. We then take the subnetwork to be the induced subgraph of G consisting of all the nodes that share at least one ground-truth community membership with u. In our experiments, we create 500 different subnetworks for each of the six real-world datasets and the results are averaged over these 500 samples.

4.9.2 Overlapping Community Scoring Metrics

The following metrics are often used to quantify the quality of the detected overlapping community structure.

- **Modularity:** Shen et al. [89] introduce EQ, an adaptation of Newman's modularity function [156] designed to evaluate overlapping communities as follows:

$$EQ = \frac{1}{2m} \sum_{c \in C} \sum_{i \in c, j \in c} \frac{1}{O_i O_j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \tag{4.17}$$

where, A_{ij} indicates (i, j) entry in the adjacency matrix A, m is the number of edges in the graph, C is the set of communities, and O_i is the number of communities to which the node i belongs.

On the other hand, recently Lázár et al. [127] provide a more complex evaluation metric of the goodness of an overlapping community structure as follows:

$$Q_{ov} = \frac{1}{|C|} \sum_{c \in C} \left[\frac{\sum_{i \in c} \frac{\sum_{j \in c, i \neq j} A_{ij} - \sum_{j \notin c} A_{ij}}{d_i \cdot s_i}}{n_c} \cdot \frac{n_c^e}{\binom{n_c}{2}} \right] \tag{4.18}$$

where C is the set of communities, n_c and n_c^e are the number of nodes and edges that community c contains respectively, d_i is the degree of node i, and s_i is the number of communities to which i belongs.

- **Community Coverage (CC):** As described in [3], this metric counts the fraction of nodes that belong to at least one community of three or more nodes. A size of three is chosen since it constitutes the smallest non-trivial community structure.

- **Overlap Coverage (OC):** As described in [3], this metric counts the average number of node memberships in non-trivial communities (size at least three).

4.9.3 Baseline Algorithms

We choose the following state-of-the-art overlapping community detection algorithms for our experiments. The algorithms are chosen such that they are relatively new and can cover all types of overlapping community detection heuristics mentioned in [226]:

- **Local expansion and optimization:** OSLOM [6] [126], EAGLE [7] [89].
- **Agent-based dynamical algorithms:** COPRA[8] [85] and SLPA[9] [228].
- **Fuzzy detection using mixture model:** MOSES[10] [146], BIGCLAM[11] [232].

4.9.4 Community Validation Metrics

Given the ground-truth community structure of a network, a stronger test for evaluating the quality of the detected community would be to compare it with the ground-truth structure. For this purpose, we use the following validation metrics that quantify the level of correspondence between these two types of communities: (a) Overlapping Normalized Mutual Information (ONMI) [147], (b) Omega (Ω) Index [85], (c) F-Score [232]. Note that higher the value of these metrics, the closer is the match with the ground-truth structure.

4.10 Inferences from OPerm Values

Although a community is generally conceived as a homogeneous entity, in reality it is not so. Within a community, the extent of involvement and activity may not be same for all members - OPerm precisely captures this heterogeneity. The value of OPerm of a node v belonging to a community c indicates the extent to which the node belongs to c. Using this value several inferences can be drawn about the communities present in the network. For instance, it inherently creates a gradation/ranking of the constituent vertices in a community. This ranking may be important in many cases; one such perspective is to explore the core-periphery structure of a community. We also describe the utility of such ranking through a real-world application – initiator selection during message spreading in

[6] http://www.oslom.org.

[7] http://code.google.com/p/eaglepp/

[8] http://www.cs.bris.ac.uk/~steve/networks/software/copra.html.

[9] https://sites.google.com/site/communitydetectionslpa.

[10] http://sites.google.com/site/aaronmcdaid/moses.

[11] http://snap.stanford.edu

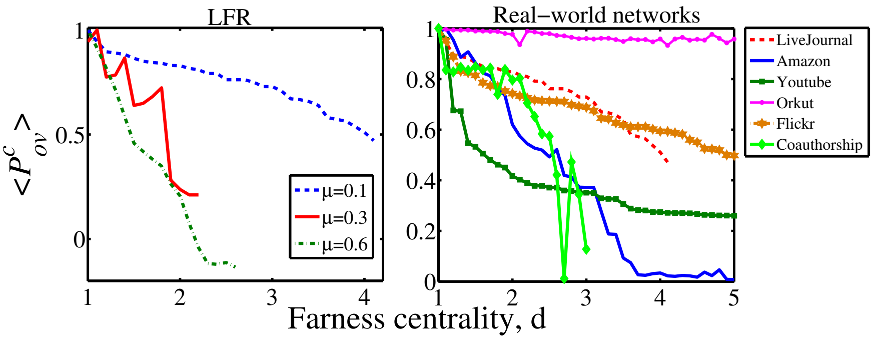


Figure 4.7: Community-wise average overlapping permanence, $< P^c_{ov} >$ of vertices as a function of farness centrality d for LFR and real-world networks.

networks. Finally, we provide a general overview discussing the distribution of nodes in the constituent communities.

4.10.1 Core-periphery Structure of Community

We intend to explore the relation of OPerm of a vertex with its position vis-a-vis core of a community. To do so, we use *farness centrality* (d) proposed in [233] as a measure to locate the position of a vertex within a community. In order to measure farness centrality for each community, we construct the induced subgraph constituting all the nodes in the community and measure average shortest path for each vertex within this subgraph[12]. Thus, the lower the value of d for a vertex, the closer the vertex is to the core part of the community. We plot average P^c_{ov} of vertices as a function of d in Figure 4.7. We observe that for both LFR and real-world networks, average P^c_{ov} decreases with the distance from the center of the community. Therefore, the value of OPerm can act as a strong indicator of the position of the vertex in the community. Moreover, we shall see in Figure 4.9(i) - 4.9(j) that nodes exhibiting high P^c_{ov} tend to have low degree. This leads to an interesting realization that the core of a community is mostly shaped without high-degree nodes.

The next investigation reveals the manner in which the OPerm value of vertices decreases from the core. A smooth decrease in value would indicate that the nodes in a community are arranged in layers with each layer of vertices roughly having similar OPerm. In order

[12] Farness centrality is just the opposite of closeness centrality in a connected component.

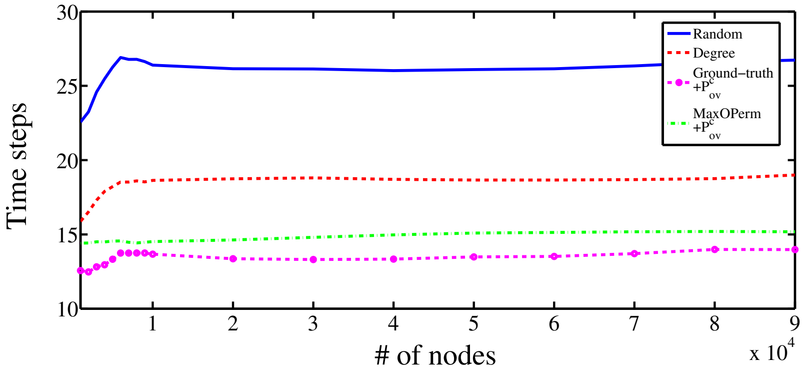


Figure 4.8: Number of time steps required to spread a message in LFR network by varying the number of nodes.

to understand the mixing pattern of vertices, we measure assortativity (r)[13] [158] based on OPerm to observe the preference for a network's nodes in a community c to attach to other nodes that have nearly similar OPerm. We divide the values of P^c_{ov} into 20 bins so that nodes within a bin are treated to exhibit exactly same P^c_{ov}, and then measure r for the corresponding community. For comparison, we also measure degree-based assortativity of vertices in each community. The average of assortativity scores of all the communities per network is reported in Table 4.8. We observe that both synthetic and real-world networks are highly assortative in terms of OPerm. This result indeed indicates that in general, a community is organized into several layers, where each layer is composed of vertices exhibiting similar OPerm, and vertices tend to be highly connected *within each layer* than across different layers. The positive impact of such layer is next illustrated by considering the task of message spreading.

Table 4.8: Average of the assortativity scores, $< r >$ (degree-based and P^c_{ov}-based) of the communities per network.

$< r >$	LFR ($\mu = 0.1$)	LFR ($\mu = 0.3$)	LFR ($\mu = 0.6$)
Degree-based	-0.045	-0.018	0.139
P^c_{ov}-based	0.645	0.483	0.421

$< r >$	LiveJournal	Amazon	Youtube	Orkut	Flickr	Coauthorship
Degree-based	0.037	-0.275	-0.182	0.221	-0.098	0.281
P^c_{ov}-based	0.465	0.497	0.438	0.528	0.402	0.469

[13] Assortativity (r) lies between -1 and 1. When r = 1, the network is said to have perfect assortative patterns, when r = 0 the network is non-assortative, while at r = -1 the network is completely disassortative.

4.10.2 Initiator Selection for Message Spreading

Message spreading is one of the challenging problems in complex networks [45]. Starting with a set of source nodes/ initiators having a message, the protocol proceeds in a sequence of synchronous rounds. At every time step, each node in the system having the message communicates with one node (not having the message) in its neighborhood and transfers the message. The algorithm terminates when all the nodes in the system have received the message.

A fundamental issue in message spreading is the selection of initiators. Since OPerm produces a ranked list of vertices within a community and vertices with higher P_{ov}^{c} form the core of the community, we posit that initiator selection based on P_{ov}^{c} would help in disseminating the message more quickly. For this, we consider LFR network and vary the number of nodes from 1,000 to 90,000, keeping the other parameters constant (see Section 4.9.1). We select multiple initiators by picking one node per community present in the ground-truth structure based on the following criteria separately: (i) random, (ii) highest degree, (iii) highest P_{ov}^{c}. Side by side, we detect communities using MaxOPerm (see Section 4.12) and choose initiators from the communities based on highest P_{ov}^{c}. For each network configuration, we run 500 simulations and report in Figure 4.8 the average number of time steps required for the message to reach all the nodes in the network. We observe that P_{ov}^{c}-based initiator selection from ground-truth communities requires minimum time steps to spread the message compared to the degree-based selection. P_{ov}^{c}-based initiator selection for the communities obtained from MaxOPerm performs almost as good as that selected from ground-truth community (we shall discuss this issue more in Section 4.13). These results thus highlight the importance of P_{ov}^{c}-based ranking within a community.

4.10.3 Explaining the Community Structure

In this section, we investigate the distribution of OPerm corresponding to each node-community pair, which in turn might be effective to explain general characteristics of the community structure. To do so, we compute P_{ov}^{c} of each vertex on the ground-truth communities of the benchmark networks. As shown in Figure 4.9(a) - 4.9(b), we divide the

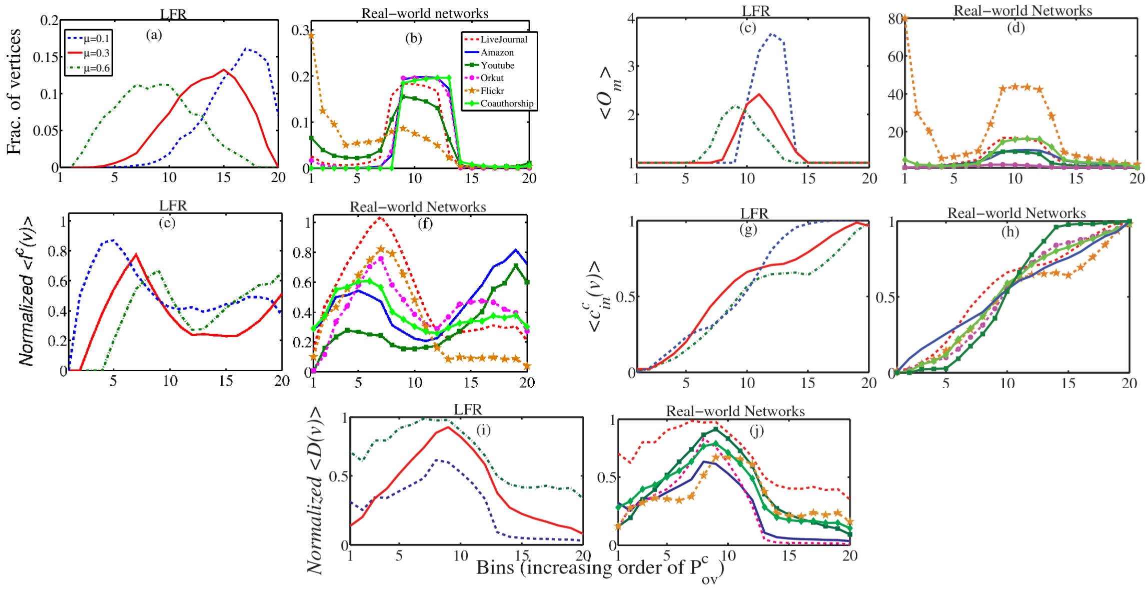


Figure 4.9: The relation of average P^c_{ov} with (a)-(b) fraction of vertices, (c)-(d) $< O_m >$, average community memberships per node, (e)-(f) $< I^c(v) >$, average internal degree (normalized by the maximum value), (g)-(h) $< c^c_{in}(v) >$, average internal clustering coefficient, (i)-(j) $< D(v) >$, average degree of nodes for LFR and real-world networks. The value of P^c_{ov} of vertices in each community is equally divided into 20 buckets indicated in x-axis (bin 1: $-1 \leq P^c_{ov} < -0.9$, ..., bin 20: $0.9 \leq P^c_{ov} \leq 1$).

values of P^c_{ov} ranging from -1 to 1 into 20 bins on x-axis where the low (high) numbered bins contain nodes with lower (higher) P^c_{ov}, and for each bin, we plot in y-axis the fraction of vertices present in the network. We observe that this curve follows a Gaussian-like distribution, i.e., there are few vertices with very high or very low P^c_{ov} values, and majority have intermediate values. In Figure 4.9(a), the peak shifts from left to right with the decrease of μ value in LFR network (keeping the other parameters of LFR constant). The shift in the peak shows that as the structure of the communities gets more well-defined with the decrease of μ, most vertices move towards higher OPerm zone. The real-world networks, except Flickr show a similar Gaussian distribution in Figure 4.9(b), where most of the vertices fall in medium P^c_{ov} range. For Flickr network, we notice in Table 4.7 that the communities are large in size (high S) and sparse in terms of edge density (low ρ) compared to those for the other networks. We also observe that most of the vertices in Flickr network have low internal clustering coefficient (0.12, where the average c^c_{in} of vertices for the other networks is 0.31), thus producing very low P^c_{ov}.

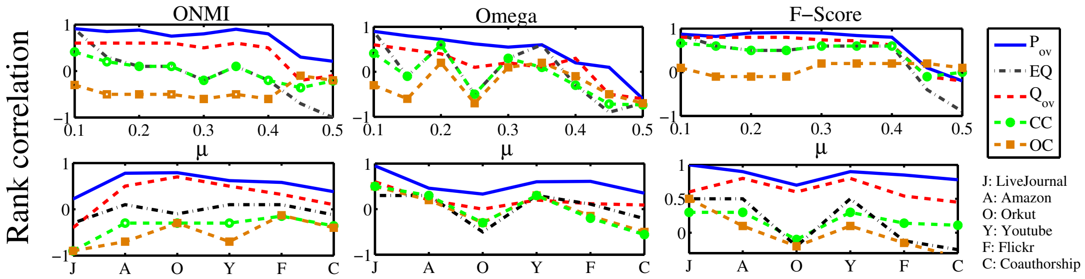


Figure 4.10: Spearman's rank correlation between five scoring functions with three validation measures for (top panel) LFR by varying μ and (bottom panel) six real-world networks.

To investigate further, for a community we enumerate the number of (other) community memberships of each constituent node vis-a-vis its P^c_{ov}. As shown in Figure 4.9(c) - 4.9(d), this pattern also follows a Gaussian distribution for LFR networks. This indicates that typically in a community, vertices exhibiting average P^c_{ov} tend to belong to multiple communities. This is non-intuitive because generally P^c_{ov} of a vertex tends to decrease with the increase in the magnitude of its belongingness to multiple communities ($I^c(v)$ becomes lower in Equation 4.16). However surprisingly, here nodes sharing multiple communities still exhibit medium to high P^c_{ov} value. We speculate that some other factors in Equation 4.16 might push the value of P^c_{ov} up to a certain extent.

To check this, we further plot in Figure 4.9(e) - 4.9(f) the average value of $I^c(v)$ of vertices within each P^c_{ov} bin. For each network, the values of $I^c(v)$ are normalized by the maximum value. We observe the minimum value of $I^c(v)$ in the middle bin; this is because the nodes participating in many communities contribute a small fraction of internal edges to each community. Next, we plot another important ingredient of P^c_{ov}, average $c^c_{in}(v)$ of the vertices in each bin. Here we notice that for all the networks, the constituent nodes of a community with medium P^c_{ov} exhibit significantly high $c^c_{in}(v)$. This can be the possible reason for having P^c_{ov} in a medium range for highly overlapped nodes. We also observe in Figure 4.9(i) - 4.9(j) that nodes with higher degree exhibit medium P^c_{ov}. Therefore, comparing all the results in Figure 4.9 together we conclude that the high-degree nodes are part of more communities, thus most of their connected edges are shared by multiples groups. In other words, they maintain medium P^c_{ov} in every community they participate.

4.11 OPerm as Community Scoring Function

In this section, we perform exhaustive experiments to show that OPerm serves as a better scoring metric compared to those outlined in Section 4.9.2.

4.11.1 Correspondence to Ground-truth Structure

We first adopt the same rank correlation based approach [204] described in Section 4.4 and compare OPerm with the other overlapping community scoring metrics mentioned in Section 4.9.2.

Figure 4.10 shows the correlation values for different LFR networks (where μ is varied[14]) and six real-world networks (the values are reported by averaging over 500 subnetworks in each case). Each vertical panel in the figure corresponds to a validation measure. Each line in a panel corresponds to a scoring function. We observe that for all the cases, the lines corresponding to P_{ov} dominate other scoring metrics, which is followed by Q_{ov} and EQ. The performance of CC and OC are same and worst among the others. Therefore, we conclude that OPerm can capture the variability much better than other scoring metrics.

4.11.2 Robustness to Perturbations

So far we have examined the ability of different scoring metrics to rank algorithms according to their goodness. In this section, we further evaluate community scoring metrics using a set of perturbation strategies for communities. We posit that a metric is robust to any perturbation if its value under small perturbations to the ground-truth, changes slightly. However, if the ground-truth labels are highly perturbed such that the underlying community structure gets highly deformed, then a good community scoring metric should diminish to a low score.

We adopt three perturbation strategies mentioned in Section 4.5. We perturb different

[14] Results are same for the other LFR networks obtained by varying O_m and O_n.

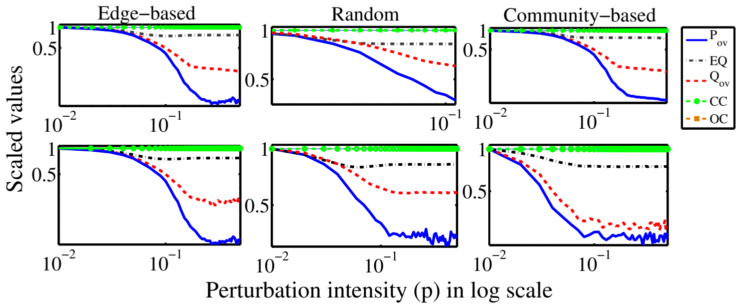


Figure 4.11: Change in the value of five overlapping community scoring functions with the increase of perturbation intensity p in (a) edge-based, (b) random and (c) community-based strategies for one LFR (top panel) and one real network (Flickr, bottom panel). The values of each metric are normalized by the maximum value obtained from that metric. Most cases, the lines for community coverage (CC) and overlap coverage (OC) are juxtapose because of their high similarity.

networks using these strategies for values of p ranging between 0.01 to 0.5, and compute five community scoring metrics, i.e., P_{ov}, EQ, Q_{ov}, CC, CC. Figure 4.11 shows the representative results of our experiments for one LFR and one real-world network (Flickr) (the results are same for the other cases). For all three strategies, the value of the scoring metrics tends to decrease with the increase in p; the effect is most pronounced in community-based strategy. For each network, once p has reached a certain threshold, the decrease in value is much faster in OPerm. This happens because the internal structure of a community completely breaks down if the perturbation is taken beyond a point and thus has an avalanche effect on the value of the clustering coefficient ($c_{in}^{c}(v)$ in Equation 4.16). This in turn quickly reduces the value of OPerm, thus making it appropriately robust.

4.12 Overlapping Community Detection by Maximizing OPerm

We develop MaxOPerm (pseudocode in Algorithm 3), a greedy agglomerative algorithm that iteratively maximizes OPerm and, thereby, detects the overlapping community structure of a network. It starts with a random community assignment where each edge is assumed to be a community. Then in every iteration, OPerm of a vertex v is computed

Algorithm 3 MaxOPerm: Maximizing OPerm for detecting overlapping communities

```
Input: A connected graph $G = (V, E)$
Output: Detected overlapping communities and OPerm of $G$
  Assign each edge (two end vertices) to a separate community
  $Ver \leftarrow |V|$
  $GOPerm \leftarrow 0.0$, $SumOPerm \leftarrow 0.0$
  $OldOPerm \leftarrow -1.0$
  Set the value of maximum iteration as $MaxItern$
  $Itern \leftarrow 0$
  while $Itern < MaxItern$ and $SumOPerm \neq OldOPerm$ do
      $Itern \leftarrow Itern + 1$
      $OldOPerm \leftarrow SumOPerm$
      for each vertex $v$ do
          $CurComm$ is the set of communities to which $v$ belongs
          Find $CurOPerm$, the OPerm of $v$ in $CurComm$
          if $CurOPerm == 1$ then
              $SumOPerm \leftarrow SumOPerm + CurOPerm$      /* Communities of $v$ are set to $CurComm$ */
              continue;
          Endif
          Determine $CNeigh$, the set of neighboring communities of $v$
          Find overlapping permanence of $v$ in $CNeigh$
          $TempOPerm \leftarrow 0.0$
          $TempComm \leftarrow \emptyset$
          for each community $c$ in $CNeigh$ do
              Temporarily assign $v$ to community $c$
              Calculate $v\_P_{ov}^{c}$, OPerm of $v$ in $c$
              if $v\_P_{ov}^{c} > 0$ then
                  $TempOPerm \leftarrow TempOPerm + v\_P_{ov}^{c}$
                  $TempComm \leftarrow TempComm \cup c$
              Endif
          Endfor
          if $TempOPerm > CurOPerm$ then
              $CurComm \leftarrow TempComm$
              $CurOPerm \leftarrow TempOPerm$
          Endif
          $SumOPerm \leftarrow SumOPerm + CurOPerm$
      Endfor
  Endwhile
  $GOPerm \leftarrow SumOPerm/Ver$      /* OPerm of the graph */
  return $GOperm$
```

by temporarily assigning it into each of its neighboring communities. Then, the overall change in OPerm is computed. A vertex v is assigned to newer communities if there is a positive gain in OPerm due to this assignment. The algorithm converges either when there is no improvement in OPerm for all the vertices or the maximum number of allowable iterations is reached.

4.13 Evaluation of MaxOPerm

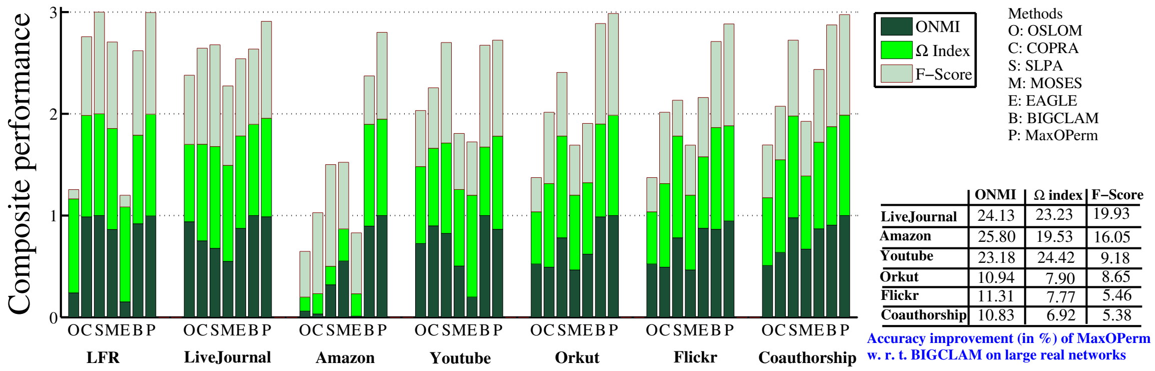
In order to evaluate MaxOPerm, we (a) compare the detected community structure with the ground-truth community structure and measure the similarity (Section 4.13.1), and (b) check how sensitive its output is due to the change in the initial vertex ordering (Section 4.13.2).

4.13.1 Comparison with Baseline Algorithms

We run MaxOPerm along with six other algorithms mentioned in Section 4.9.3 and compare their performance for networks whose ground-truth communities are known. Since the baseline methods do not scale for large-size real networks, we use the sampled subnetworks as mentioned in Section 4.9.1. For the LFR benchmark however, the results are reported on the entire network. The results are shown for the following setting of the LFR network: n=1000, μ=0.2, O_n=5% and O_m=4. For each real network, we measure the average value of each validation metric for 500 different samples.

For each validation metric (ONMI, Ω Index, F-Score), we separately scale the scores of the methods so that the best performing community detection method has the score of 1. Finally, we compute the composite performance by summing up the 3 normalized scores. If a method outperforms all the other methods in all the scores, then its composite performance is 3.

Figure 4.12 displays the composite performance of the methods for different networks. On an average, the composite performance of MaxOPerm (2.88) significantly outperforms other competing algorithms: 6.27% higher than that of BIGCLAM (2.71), 18.03% higher than that of SLPA (2.44), 101.3% higher than that of OSLOM (1.43), 36.4% higher than that of COPRA (2.11), 48.4% higher than that of MOSES (1.94), and 77.8% higher than that of EAGLE (1.62). The absolute average ONMI of MaxOPerm for one LFR and six real networks taken together is 0.85, which is 4.93% and 26.8% higher than the two most competing algorithms, i.e., BIGCLAM (0.81), and SLPA (0.67) respectively. In terms of absolute values of scores, MaxOPerm achieves the average F-Score of 0.84 and average Ω



	ONMI	Ω index	F–Score
LiveJournal	24.13	23.23	19.93
Amazon	25.80	19.53	16.05
Youtube	23.18	24.42	9.18
Orkut	10.94	7.90	8.65
Flickr	11.31	7.77	5.46
Coauthorship	10.83	6.92	5.38

Accuracy improvement (in %) of MaxOPerm w. r. t. BIGCLAM on large real networks

Figure 4.12: Performance of various competing algorithms (ranging from 0 to 3) to detect the ground-truth communities. The table shows the performance improvement of MaxOPerm over BIGCLAM in detecting communities in large real networks.

Index of 0.83. Overall, MaxOPerm gives the best results, followed by BIGCLAM, SLPA, COPRA, MOSES, EAGLE and OSLOM.

Comparison with BIGCLAM for large networks: As most of the baseline algorithms except BIGCLAM do not scale for large real networks [232], we separately compare MaxOPerm with BIGCLAM (which is also the most competing algorithm) on actual large real datasets. The table in Figure 4.12 shows the percentage improvement of MaxOPerm over BIGCLAM for different real networks. On average, MaxOPerm achieves 17.67% higher ONMI, 14.96% higher Ω Index, and 10.78% higher F-Score. Overall, MaxOPerm outperforms BIGCLAM in every measure and for every network. The absolute values of the scores of MaxOPerm averaged over all the networks are 0.81 (ONMI), 0.82 (Ω Index), and 0.81 (F-Score). Therefore, the improvement of MaxOPerm over BIGCLAM is higher considering the entire network in comparison to that in the sampled networks.

4.13.2 Degeneracy of Solutions

Degeneracy of solutions is the phenomenon where the same optimal value can output different community assignments. Most of the community detection algorithms are based on optimizing certain functions (such as modularity), and the values are heavily dependent on the order in which vertices are processed [123].

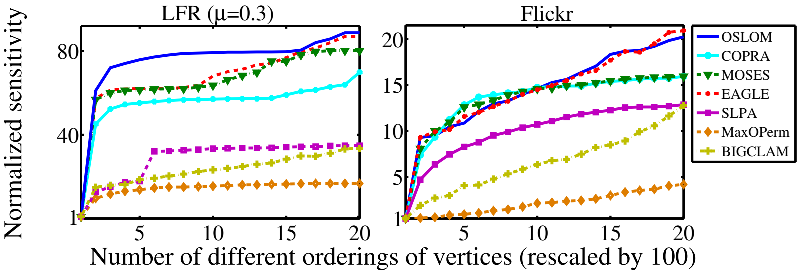


Figure 4.13: Sensitivity (rescaled by the minimum value) of each algorithm across 2000 different vertex orderings. The x-axis is rescaled by a constant factor of 100.

An intrinsic goodness of an algorithm can be measured by the number of invariant groups of vertices (termed as "constant communities" as discussed in Chapter 3) that remain in the same community across different vertex orderings. To quantify the effect of vertex ordering, we introduced a metric, called *sensitivity* (ϕ), which measures the ratio of the number of constant communities to the total number of nodes (see Section 3.5.1). In the worst case, the number of constant communities would be equal to the number of nodes with each node being a community. Here, we use this metric to measure the degeneracy of an algorithm. If the value of ϕ for an algorithm remains constant and small over different vertex orderings, then the algorithm is less susceptible to degeneracy of solutions.

We plot the value of sensitivity for different vertex orderings for each algorithm in Figure 4.13 (one LFR and one real-world network are considered to illustrate the results). The x-axis indicates the number of different vertex orderings and the y-axis plots the normalized value of ϕ. We observe that MaxOPerm is the most consistent algorithm. This result demonstrates that by producing lesser number of competing solutions our algorithm is able to significantly reduce the problem of degeneracy of solutions. We emphasize that this is the *first time a metric has been proposed to quantify the degeneracy of solutions*, and this metric can be used for evaluating any newly proposed community detection algorithm.

Another example of the advantage of MaxOPerm has been shown in initiator selection for message spreading (see Section 4.10.2). As shown in Figure 4.8, if one uses MaxOPerm to detect the overlapping communities, then the performance is almost as good, as that of using the ground-truth community.

4.14 Summary of this Chapter

In this chapter we introduced two vertex-based metrics, called permanence (Perm) and overlapping permanence (OPerm) respectively for disjoint and overlapping community analysis. Subsequently, we proposed two greedy algorithms, MaxPerm and MaxOPerm to detect disjoint and overlapping communities respectively from the networks. The contributions of this chapter are manifold:

- The values of Perm and OPerm act as a strong indicator of the existence of community structure in a network.
- We presented the first generalized formula, OPerm that can identify both overlapping and non-overlapping communities depending on the underlying structure of the network.
- We demonstrated for the first time how vertices are organized within a community through proper statistical quantities.
- We identified a precise rank order among the vertices within a community by arranging them into a core-periphery structure based on OPerm.
- Being vertex-based metrics, both Perm and OPerm are significantly localized; they therefore allow partial estimation of communities in a network whose entire structure is not known.

Chapter 5

Analyzing Ground-truth Communities

In this chapter we address our third objective of analyzing ground-truth community structure of a real-world network. In particular, we consider a scientific network, called *citation network*, and analyze its ground-truth community structure.

5.1 Introduction

Several works on detecting and tracking communities in a temporal environment have been conducted [24, 203]. However, the interactive patterns of the detected communities over a temporal scale still remain unexplored mainly due to the lack of standard real-world ground-truth communities. For those networks, whose ground-truth community structures are known to us, the lack of appropriate metadata information has remained a barrier in exploring the dynamics of the community interactions. This chapter stresses on developing ground-truth overlapping communities in terms of the research fields of a large-scale directed citation network of computer science domain and explores the inter-cluster interactive patterns on a longitudinal scale (i.e., with the progress of time) that in turn explains the rise and fall of the impact of scientific research over the last fifty years.

The contributions of this chapter are threefold:

- **Ground-truth communities and their temporal interactions:** In first part of this chapter, we describe a large-scale paper-paper directed citation network of the computer science domain with the research areas/fields annotated thus representing the natural partitioning of the network into ground-truth communities. Next, we propose a simple edge-centric measurement called "inwardness" of a community to capture the dynamics of inter-cluster interactions across time points. Subsequently, to understand this phenomena at a more granular level, we postulate several explanations for such a dynamical behavior of research communities. Finally, we validate our proposed framework with the evidence of additional statistics obtained in the form of the project funding decisions made by NSF (National Science Foundation of the USA).

- **Interdisciplinary nature of research fields:** In the second part, we systematically unfold the dynamics and emergence of connections across the fields, which in turn quantify the interdisciplinary research activities. The degree of interdisciplinarity of a particular field is measured using four indicators that neatly separate out the core from the interdisciplinary fields in a fully unsupervised fashion. After that, we perform a two-fold analysis of the results. As a first objective, we compare the evolutionary landscape of a core and an interdisciplinary field, while as a second objective we perform core-periphery analysis of the citation network at different time points and observe that the popularity of the interdisciplinary research now-a-days overshadows the core fields.

- **Understanding scientific career of researchers:** In the last part of this chapter, we analyze the diverse scientific careers of researchers in computer science domain in order to understand the key factors that could lead to a successful scientific career. In particular, we investigate by proposing two entropy-based metrics how the researchers make choices to select their fields of research at different points in their career. We observe that most of the prominent researchers tend to follow typical "scatter-gather" policy – although their entire careers are immensely diverse with different types of fields selected at different time periods, they remain very focused in one or at most two fields at different shorter time spans of their careers.

Table 5.1: General information of raw and filtered datasets.

	Raw dataset	Filtered dataset
Number of valid entries	2,473,171	1,549,317
Number of entries with no venue	343,090	–
Number of entries with no author	45,551	–
Number of entries with no publication year	191,864	–
Partial data of the years before 1970 and 2011-2012	343,349	–
Number of authors	1,186,412	821,633
Avg. number of papers per author	5.18	5.04
Avg. number of authors per paper	2.49	2.67
Number of unique venues	6,143	5,938
Percentage of entries with multiple fields	9.08%	8.68%

5.2 A Large Publication Dataset

The traditional information pertaining to citation networks like papers and citation distributions are not adequate in this study to meet all the experimental needs. The analysis needs several other related information about each paper, e.g., publication year, publication venue (journal/conference), research field, authors and their continents.

5.2.1 Curation of a Large Publication Dataset

We crawled one of the largest publicly available datasets from Microsoft Academic Search (MAS)[1]. We collected all the papers specifically published in the computer science domain and indexed by MAS. The crawled dataset[2] contains more than 2 million distinct papers altogether which are further distributed over 24 fields of computer science domain (see Table 5.2). Moreover, each paper comes along with various bibliographic information – the title of the paper, a unique index for the paper, its author(s), the affiliation of the author(s), the continent of the author(s), the year of publication, the publication venue, the related field(s) of the paper, the abstract and the keywords of the paper, and the references of the paper.

In general, scientific focus shifts are affected manifold by contributory papers than by

[1] academic.research.microsoft.com

[2] The dataset is available at http://cnerg.org for the research community.

Table 5.2: Percentage of papers in various fields and their average inwardness (see Section 5.3) in each decade (for each decade, top and second ranked inwardness measures are in bold font).

No.	Subject	Abbreviation	% of papers	Average Inwardness				
				60-69	70-79	80-89	90-99	00-10
1.	Artificial Intelligence	AI	15.30	0.02	0.67	**4.94**	**5.14**	**3.29**
2.	Algorithms and Theory	ALGO	14.09	**4.13**	**4.49**	3.39	2.12	0.55
3.	Networking	NW	8.63	0.19	0.53	1.06	**3.42**	1.76
4.	Databases	DB	8.12	**3.75**	**3.67**	1.80	1.14	0.17
5.	Distributed and Parallel Computing	DIST	7.63	0.02	2.02	2.86	1.55	0.56
6.	Hardware & Architecture	ARC	7.29	0.41	2.49	2.29	1.12	1.04
7.	Software Engineering	SE	6.40	1.98	3.21	1.89	1.67	0.52
8.	Machine Learning and Pattern Recognition	ML	6.09	0	0.43	2.51	2.97	**2.62**
9.	Scientific Computing	SC	4.02	0	1.14	2.38	2.91	0.19
10.	Bioinformatics & Computational Biology	BIO	3.88	0	0	0.71	1.27	0.56
11.	Human-Computer Interaction	HCI	3.42	0	0.03	1.65	2.05	1.39
12.	Multimedia	MUL	3.34	0	0.53	2.51	2.22	1.33
13.	Graphics	GRP	3.32	0	0.56	2.58	2.63	1.07
14.	Computer Vision	CV	3.03	0	0.86	1.29	2.73	1.27
15.	Data Mining	DM	3.02	0	0.27	1.80	1.83	1.02
16.	Programming Languages	PL	3.00	0.41	2.49	**3.86**	2.46	1.29
17.	Security and Privacy	SEC	2.94	0	0.86	3.80	2.56	1.59
18.	Information Retrieval	IR	2.26	0	0.42	1.32	2.62	1.79
19.	Natural Language and Speech	NLP	2.11	0	0.13	1.16	2.82	1.92
20.	World Wide Web	WWW	1.76	0	0	1.86	2.10	1.83
21.	Computer Education	EDU	1.67	0	0	0.80	0.83	0.39
22.	Operating Systems	OS	1.07	0.31	1.73	1.39	1.98	1.20
23.	Real Time Embedded Systems	RT	0.90	0	0.67	1.56	2.52	0.54
24.	Simulation	SIM	0.14	0	0.30	1.20	2.70	0.87

reviews, surveys and text books, and therefore we exclude these items from our data. Further, in order to make our data bounded we consider only those papers that cite or are cited by at least one paper. Moreover, we consider only those papers published in between 1970 and 2010 because this set of papers contains the most reliable and significant entries. In the filtered dataset, 8.68% papers belong to multiple fields (act as interdisciplinary papers). Some of the general information pertaining to the filtered dataset are presented in Table 5.1.

5.2.2 Constructing Citation Networks

Since our method is primarily based on suitable statistical analysis of various properties of paper-paper citation network, the next task is to construct the citation network from the tagged dataset. Formally, a citation network is defined as a graph $G =< V, E >$ where each node $v_i \in V$ represents a paper and a directed edge e_{ji} pointing from v_j to v_i indicates that the paper corresponding to v_j cites the paper corresponding to v_i in its references. From our tagged dataset, a citation network is constructed by the papers representing nodes and the citations representing directed edges from the citing paper to the cited paper. At a higher tier, each field (i.e., a collection of papers) can be thought of as a single community, and two communities can again be linked by a directed edge with edge-weight calculated using Equation 5.1 mentioned in Section 5.3. Following this strategy, we essentially obtain a field-field directed and weighted network on top of the paper-paper citation network which attempts to capture the interaction patterns of the scientific communities. Note that in each year, there are at most 24 communities (if there exists at least one paper from each of the fields) and the size of each community changes over the years depending upon the number of publications in that field. A community at time t can interact with any other communities at or before t.

5.3 Time Transition of Scientific Communities

In this section, we analyze the time transition of the scientific focus showing how one field has taken over another during the time evolution of the computer sciences. In particular, we measure the impact of a field so as to construct the time transition diagram reflecting the trend shifts. Some of the previous experimental results [59, 60] show that the trend of citations received by a paper after its publication period is not linear in general; rather there is a fast growth of in-citations within the initial few years after the publication, followed by an exponential decay. We notice the same property in our dataset and observe that the average number of inward citations per paper peaks within three years from its publication and then slowly declines over time (see Figure 5.1). Note that this property is also prevalent across the different fields of the domain (see inset of Figure 5.1). Therefore,

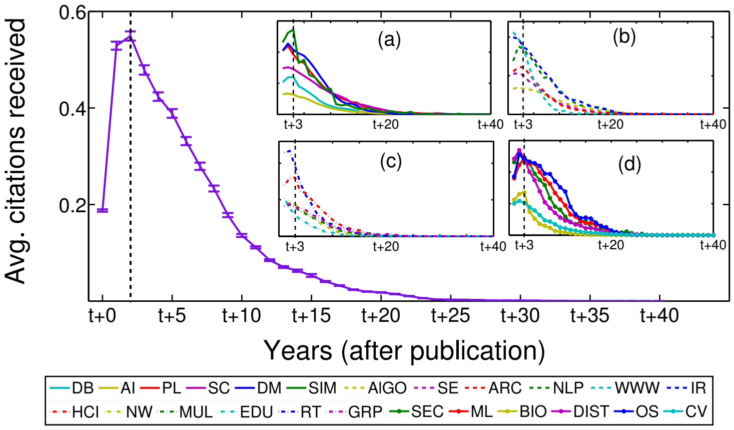


Figure 5.1: Average distribution pattern of inward citations (with variances) for a paper after publication (inset: same measure for every field).

in order to measure the importance of a paper (or a field) around its time of publication, all our analysis throughout the rest of the chapter assumes only the citations received by the paper within three years from its publication. We quantify the importance of a paper (aka *inwardness*) in terms of the total number of inward citations to the paper. Consequently, the temporal inwardness of a field f_i at time t denoted by $In(f_i^t)$ that captures the local citation count (within three-year window) suitably normalized by the number of papers in that field can be defined as follows:

$$In(f_i^t) = \sum_{j \neq i} w_{j \to i}^t \tag{5.1}$$

where $w_{j\to i}^t = \frac{c_{j\to i}^t}{p_i^t}$ with $c_{j\to i}^t$ corresponding to the number of citations received by the papers of field f_i from the papers of field f_j, p_i^t corresponding to the total number of papers in field f_i and $1 \leq t \leq 3$. Note that for all our estimates, in addition to this three-year window we also include the year of publication of the paper.

In order to investigate the global time transition pattern (i.e., the worldwide behavior) we compute the inwardness of each field (Equation 5.1), rank them and plot the top two values (see the solid and broken lines respectively in Figure 5.2(a)) as a function of time. Each field is uniquely color coded and the relative height of the y-axis shows the inwardness of the field for a particular year. In each focus-window, we also mention the name of the top hub (backup) field that on an average brings in the largest number of citations for the

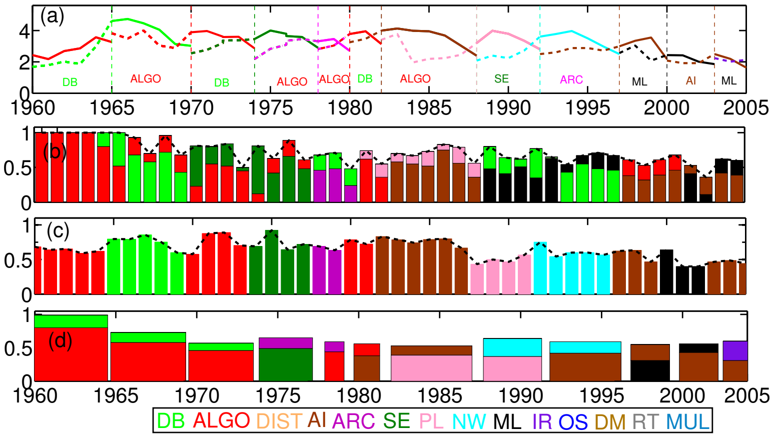


Figure 5.2: Time transition of scientific communities and probable reasons behind such transition. (a) Top two scientific community (based on inwardness) at the forefront of scientific research trend (names of topmost backup community for the community in the forefront of every trend-window are mentioned). Cause analysis: (b) fraction of papers for the top and the second ranked communities among the 10% high impact papers in each year; (c) change of citations from the topmost backup communities; (d) fraction of papers for the top and second ranked communities among the 10% highly influential papers in each trend-window. To smoothen the curves, the best sliding window size of five years has been used.

top ranking field. This information, as we shall see in Section 5.3.1, forms one of the major reasons for focus shifts. The total number of transitions of research focus during 1960 to 2005 is 11 (i.e., there are 12 trend-windows in the global time transition diagram). A careful inspection of the behavior of the curves shows that in every focus-window, a similar pattern is followed with the inwardness first rising and then gradually declining near the transition. Simultaneously, the second rank field which comes to the top position in the next focus-window in every case starts reflecting a relative growth of inwardness at the middle of the current focus-window. Another important issue is that the differences of inwardness between the top and the second top ranked fields in the long-ranged and short-ranged focus-windows are largely different. We investigate this property in further detail in the rest of the section.

Table 5.3: Ranking of top fields in each trend-window in terms of collaborative papers, multi-continent papers and diversity (average ranks of top fields in two segments of 6 trend-windows are shown in third, fifth, seventh, tenth, twelfth and fourteenth rows).

		1960-1964	1965-1969	1970-1973	1974-1977	1978-1979	1980-1981
Collaborative	Rank	13	8	13	11	3	13
	Avg.	10.16					
Multi-continent	Rank	12	8	12	10	1	12
	Avg.	9.87					
Diversity	Rank	11	8	11	13	12	11
	Avg.	11					

		1982-1987	1988-1991	1992-1996	1997-1999	2000-2002	2003-2005
Collaborative	Rank	6	12	2	6	1	6
	Avg.	5.5					
Multi-continent	Rank	7	11	3	7	2	7
	Avg.	6.17					
Diversity	Rank	3	9	10	3	4	3
	Avg.	5.33					

5.3.1 Reasons for Transitions

In this section, we conduct a diverse set of experiments to investigate the reasons behind the typical dynamics of scientific communities in the longitudinal scale observed earlier. We focus on different orthogonal characteristics all of which converge to reasons for the transitions observed. While the first cause that we propose is from an overall estimate of the data, the following three are time-varying estimates of the data.

Cause I: Impact of collaborations: Here we show that, in the current years, the expansion of collaborative work within and across continents as well as the diversity in research interest can have direct influence on the emergence of a scientific community at the forefront. To this purpose, we measure the impact of collaborative research by ranking all fields globally based on (i) the number of papers in that field having multiple authors (collaborative papers), (ii) the number of papers involving authors from multiple continents (multi-continent papers) and (iii) the diversity of a field (say, f) measured by the average number of fields that the authors of f have worked. These three ranks act as three different indicators of collaboration. Note that in case (iii), the more the diversity the higher is the rank of the field. Moreover, we suitably normalize each of the above three factors for any particular field by the total number of papers in that field. Thus, each factor indicates the average collaborative nature of a field. We then rank the fields based on each of the three normalized scores. Table 5.3 notes the ranks in cases (i), (ii) and (iii) for those fields that are at the forefront in terms of inwardness in each trend-window and the average rank of these fields in two segments each composed of six trend-windows. We observe that in all the three cases the average rank in the second segment is much higher[3] than that in the first segment. This indicates that in the current years, those fields that enjoy a higher number of collaborations and a higher overall diversity in the research interests of its constituent authors have an increased chance of emerging at the forefront. The collaborative ranks of the top fields in the earlier time periods are lower mainly because of the less proportion of the collaborative/multi-continent/diverse papers in those fields. We also observe that during the earlier time periods the high ranked collaborative fields are mostly the newly emerging fields such as AI, ML, NLP. Earlier, these emerging fields contained very few papers compared to the papers in the core fields. It seems that in the early years, the top ranking fields like Algorithms and Databases (the so-called core-fields of computer science) acted as the only and therefore indispensable sources of citation for any other field. Therefore, they were able to maintain their high ranks at least in the initial years even without having much collaborations. This is precisely the reason for their low collaboration score in spite of a high inwardness score.

Cause II: High impact papers: We extract the top 10% of the papers that have the highest number of in-citations (considering the last three years and the current year) from among all the papers published in a year. We call them as *high-impact papers*. Next we measure the fraction of papers out of this 10% that belong to a particular field. The fields are then ranked by this fraction and the fractional values are plotted in Figure 5.2(b) for the top and the second ranked fields. We observe that in 9 out of 11 cases a decline in the fraction of high-impact papers of the top ranked field and the simultaneous increase of high-impact papers in the second ranked field trigger a transition in Figure 5.2(a). Another important point to note is that in the later years, out of the 10% high impact papers, the fractions from the top and the second ranked fields diminish rapidly. While in the initial years this fraction is found to be close to 1, in the later years it drops to around 0.5. This partially indicates the maturity of the computer science domain as a whole, whereby several fields become effective and now have a place in the list of 10% high-impact papers unlike in the earlier years.

[3] Note that in this case, the rank x is higher than rank y if $x < y$ conforming to the usual notion of any ranking system.

Cause III: Citation patterns of backup communities: The impact of a paper in our experiment is determined by the citations received from other papers. Therefore, one of the important factors that helps a particular scientific community to rise up to the top is the contribution of its backup communities that direct most of their outward citations to push this community to the top. In Figure 5.2(c), we plot bars for each year indicating the fraction of citations that the top ranked community (according to Figure 5.2(a)) received from its primary backup community (i.e., the backup community that brings in the largest number of citations). Note that in 75% of the cases, the citation received from the primary backup community falls abruptly close to the transition indicating that they play a pivotal role in keeping the dominant field "dominant". This abrupt fall could be possibly caused because the citations coming from the backup communities start getting shared by other competing communities and the current community at the forefront start losing its charm owing to its member topics slowly becoming dated, thereby, losing the "timeliness" advantage. We observe that the backup fields for a particular top field are not same in all time windows. Moreover, the citations from the backup fields which are mostly focused towards the top field in the initial time periods, split among multiple fields at the time of transition. We observe that the increase in the diversity of citations from the backup field is one of the main triggering factors behind the time transition, and this might be possibly tied to the overall maturity of the backup field itself to emerge as an altogether new scientific paradigm.

Cause IV: Effect of seminal papers: The two causes discussed above have a direct bearing with the time transition of the research trend. However, there can be indirect factors affecting the rank of a community – one such factor could be the inception of seminal papers that have potential to completely change the direction of research in the immediate future. In this section, we attempt to quantify the impact of such papers by introducing a metric called *Influence*. In particular, we consider only those citations that a paper receives from the papers belonging to its own field published within the three-year window, however, ensuring that the paper being cited does not have any author in common with the paper citing it. This expresses how important a particular paper is within its own scientific community. The influence ($Influence(p_i^t)$) of paper p_i at time t is defined as follows:

$$Influence(p_i^t) = \sum_{p_j \in P^t} \frac{1}{d_{p_j}} \tag{5.2}$$

where P^t is the set of all papers that cite p_i within the three year window ($1 \leq t \leq 3$) and belong to the same field as of p_i, and d_{p_j} corresponds to the total number of outward citations from the paper p_j - the fraction is used to suitably normalize the impact of citation.

We extract the top 10% influential papers in each trend-window and find out from among them the fraction of influential papers for each field. We then rank the fields based on this fraction and plot once again the top and second ranked influential fields in each trend-window in Figure 5.2(d). The results corroborate our hypothesis that the top rank field (inwardness based) in a certain trend-window has the highest number of influential papers in the previous window (almost in 65% cases). In the earlier years (1960 to 1975), the two fields, namely Algorithms and Databases completely shadow all other fields in terms of papers and citations. The competitive pressure starts to appear mainly after 1975. If we measure this fraction from after 1975, we observe that in six out of seven cases (excluding the last window) the field that sees the birth of the largest number of influential papers in a trend-window emerges in the forefront in the immediate next trend-window. This observation points to the fact that the influential papers can play a very crucial role in shaping future research.

5.3.2 Correlation with Research Funding

It could be interesting as well as important to validate our measurements with other extraneous real-world statistics directly or indirectly reflecting the evolution of scientific research in the computer science domain. To this purpose, we collect the fund disbursal data of one of the major funding agencies of the United States – the National Science Foundation (NSF)[4]. Although this agency has a long funding history, the publicly available data that we could gather is from 2003 to 2009. In Table 5.4, we compare the top three fields ranked by our inwardness metric with the top three fields ranked by (i) the number of NSF proposals submitted and (ii) the number of proposals accepted in that field. The high-impact fields predicted by our method match accurately with the trend of proposal submission. To

[4] http://www.nsf.gov/

compare the two statistics, we propose a similarity metric τ that is defined as

$$\tau = \frac{s}{n} \tag{5.3}$$

where s is the number of similar pairs and n is the number of data points. In Table 5.5, we report the pairwise similarity (τ) between the fields ranked by our method and fields ranked by (a) the number of proposals submitted and (b) the number of proposals granted in those fields. As the number of data points are not many, exact similarity might again be a very strict assumption in this case. Therefore, while measuring the similarity using Equation 5.3, we increment the value of s when (i) at least one field is matching, and (ii) at least two fields are matching with 50% weight for each matching. We report the similarity values in the first row (REC vs. SUBMIT) and fourth row (REC vs. AWARD) of Table 5.5 for the same year where REC refers to what is recommended by our method based on inwardness. The results clearly show that our predictions are very well aligned with proposal submission while it is moderately aligned with the fund disbursal patterns.

Table 5.4: Funding statistics compared with the inwardness results (top three ranked fields are tabulated from left to right).

		NSF	
Years	Inwardness results	Proposal submitted	Proposal awarded
2003	AI/IR/NW	NW/AI/HCI	NW/ALGO/SE
2004	AI/IR/NW	AI/HCI/RT	RT/ARC/DIST
2005	AI/IR/NW	AI/ML/HCI	GRP/SE/ALGO
2006	IR/ML/AI	ML/ALGO/SEC	ALGO/SEC/ML
2007	ML/AI/ALGO	ALGO/ML/HCL	ALGO/HCI/SEC
2008	ML/AI/ALGO	ML/ALGO/SE	ALGO/ML/SE

It is often observed that the current funding patterns significantly affect the research directions of the future. Further, at times, the current research trend seems to strongly influence the funding decisions of the immediate future. The above observations can be illustrated quantitatively here. In order to do so, we introduce lagging and leading similarities between fields ranked by the inwardness metric (REC) and those ranked by the number of proposals submitted/awarded. We measure two different similarity values – $lead(\text{fund}, \text{REC}, t = 1)$ and $lag(\text{fund}, \text{REC}, t = 1)$. From the results depicted in Table 5.5, we observe that the influence of funding decisions on the future research trend is much more (lead) than the influence of the current research trend on the future funding decisions (lag). This shows that our results are remarkably in line with the decisions made by the expert researchers involved in such important proposal selection committees. However, we remark that all our results are based on only a small number of data points and should therefore be considered indicative.

Table 5.5: Correlations between our recommendations (REC) with the submit (SUBMIT) and award (AWARD) patterns of grants.

		τ	
Pairs		At least 1 matching	At least 2 matching
REC	Same year	1	0.78
vs.	$lead(\text{SUBMIT}, \text{REC}, t = 1)$	1	0.83
SUBMIT	$lag(\text{SUBMIT}, \text{REC}, t = 1)$	0.83	0.50
REC	Same year	0.71	0.50
vs.	$lead(\text{AWARD}, \text{REC}, t = 1)$	0.75	0.42
AWARD	$lag(\text{AWARD}, \text{REC}, t = 1)$	0.33	0.25

5.4 Measuring Interdisciplinarity of Scientific Research

"Interdisciplinary research is the only way to do research in current times."

– Fritjof Capra, The Turning Point

A field is any comparatively self-contained and isolated domain of human experience which possesses its own community of experts, with distinctive components such as shared goals, concepts, facts, tacit skills and methodologies. Interdisciplinary field, on the other hand, brings in together distinctive components of two or more fields in research or education, leading to new knowledge which would not be possible without this integration. Despite a reasonable number of works promoting the increasing trend of cross-field research, researchers [149, 216] still believe that there is a lack of proper quantitative indicator that could efficiently identify interdisciplinary fields (interdisciplinary papers) in a certain domain. Here we propose four indicative metrics for measuring interdisciplinarity; three of these are directly related to the topological structure of the citation network, while the fourth is an external indicator based on the attractiveness of a field for the in-coming researchers. We measure the significance of each of these features in characterizing interdisciplinarity independently and then systematically accumulate them to build an unsupervised classification model.

5.4.1 Features for Identifying Interdisciplinarity

In this subsection, we propose some possible features for each of the fields that can serve as indicators for interdisciplinarity. The rest of the subsection elaborately describes the proposed features one by one and their significance in unfolding the interdisciplinary nature of a field.

(i) Reference Diversity Index (RDI): The references of a paper reflect the diversity of knowledge sources, i.e., the related subject areas from where the paper has been motivated. Moreover, it is quite intuitive that the more is the breadth of the references of a paper, the more interdisciplinary it should be. Therefore, to formulate the diversity of references, we propose a simple quantitative measure described below.

Definition 1: ***Reference Diversity Index (RDI):*** *The RDI of a paper is the entropy of its reference set in terms of different fields the paper cites. The RDI of a field is the average of the RDIs of all the papers belonging to that field.*

Let X_i be a paper of field f_i, and it refers to papers of k different fields namely $f_1, f_2, ..., f_k$ (f_i may be one of the fields in f_1 to f_k). The *Reference Diversity Index (RDI)* of paper X_i denoted by $RDI(X_i)$ describes the heterogeneity in the distributions of references as follows:

$$RDI(X_i) = -\sum_j p_j log(p_j) \tag{5.4}$$

where p_j is the proportion of references of X_i that are given to the papers of field f_j. In other words, it is the ratio of the number of references made to the field f_j by the paper X_i to the total number of references that the paper X_i makes. The average is taken over all the papers in field f_i to get the RDI score of f_i.

Figure 5.3 illustrates the results of the RDI measured for the fields of computer science domain in four different time windows. All the results are sorted in descending order of RDI to get an idea of the rank of the fields in each time window. The more the RDI value of a field the more it should be interdisciplinary in nature. After 1975-1979, the interdisciplinary work mainly started emerging and the fields like Data Mining, World Wide Web, Human Computer Interaction, Information Retrieval consistently remain at the

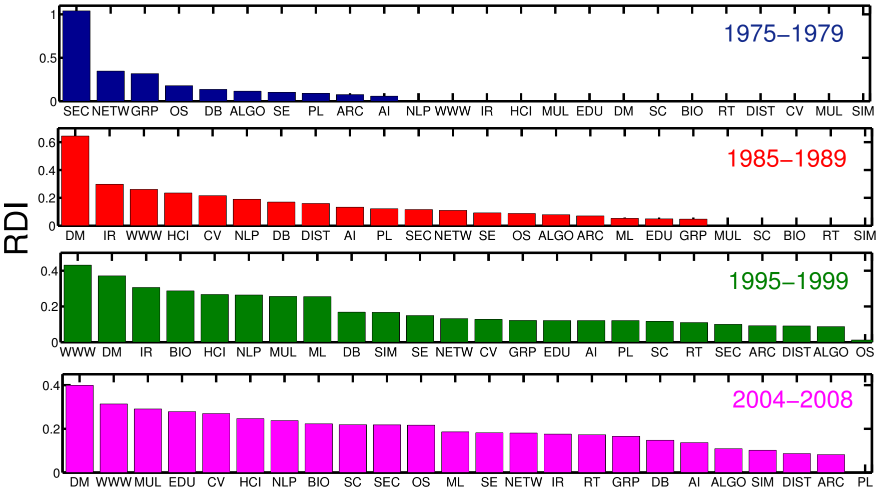


Figure 5.3: Reference Diversity Index (RDI) of all the fields in computer science domain in four time-windows. The x-axis is sorted (descending order) by the RDI value.

top positions in terms of their RDI values (Figure 5.3). At the same time, the fields like Algorithms, Operating Systems, Hardware and Architecture, Databases, Programming Languages steadily accelerate to the bottom of the rank list. Another important observation is that the degree of interdisciplinarity in terms of RDI for all the fields gradually seems to get uniform over the years (the bars in Figure 5.3 for all the fields gradually acquire equal height over the years). This is a clear indication of an increasing rate of interdisciplinary activities manifesting across the entire domain over the last few decades.

(ii) Citation Diversity Index (CDI): When analyzing the inward citation distribution patterns of the fields in our dataset, we notice that though the skewness of the inward citation pattern (i.e., breadth of the incoming citations of a paper coming from different fields) is reasonably similar for all the fields, there exist few fields exhibiting a sudden sharp rise of citation diversity at certain time points. We quantitatively measure the diversity of the inward citations of a field in the following paragraph.

Definition 2: ***Citation Diversity Index (CDI):*** *The CDI of a paper in a particular time window is the entropy of its incoming citations coming from papers of different fields published in that time window. The CDI of a field is the average of the CDIs of all the papers belonging to that field.*

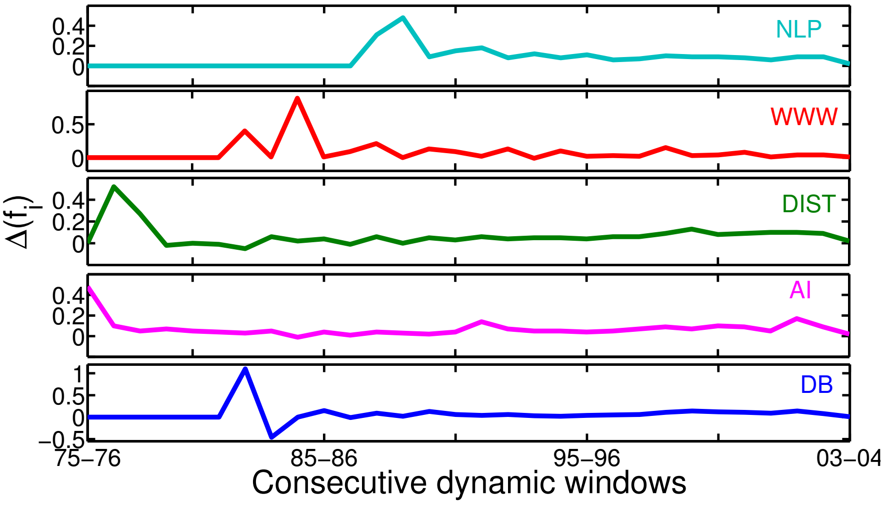


Figure 5.4: Drift of CDIs in two consecutive time windows for those fields showing sudden fluctuations in their temporal spectrum. The value in y-axis corresponding to the label $(t_i - t_{i+1})$ in x-axis indicates the difference of the CDI values obtained from the time windows t_i and t_{i+1}.

Let X_i be a paper of field f_i published in the time window t_i[5], and is cited by the papers (also published in t_i) of k different fields namely $f_1, f_2, ..., f_k$ (f_i may be one of the fields in f_1 to f_k). The *Citation Diversity Index (CDI)* of paper X_i in time window t_i denoted by $CDI_{t_i}(X_i)$ is defined to capture the diversity of the inward citations of a paper using the following equation.

$$CDI_{t_i}(X_i) = -\sum_j p_j log(p_j) \tag{5.5}$$

where p_j is the proportion of citations of paper X_i received from the papers (published in the time window t_i) of field f_j. The average is taken over all the papers in field f_i to get the CDI score of f_i. Similarly, we can find out the CDI of X_i in time window t_{i+1}, i.e., $CDI_{t_{i+1}}(X_i)$ by the diversity of the citations received from the papers published in t_{i+1}. This indicates the diversity of new citations for the same paper in the next time window. Then for a field f_i, the difference in the diversity of inward citations between two successive time windows (t_i and t_{i+1}) which we call *drift* can be expressed as

$$\Delta_{t_i}(f_i) = CDI_{t_{i+1}}(f_i) - CDI_{t_i}(f_i) \tag{5.6}$$

[5] Note that by the term “time window t_i” we refer to the five year time period from t_i to $t_i + 4$.

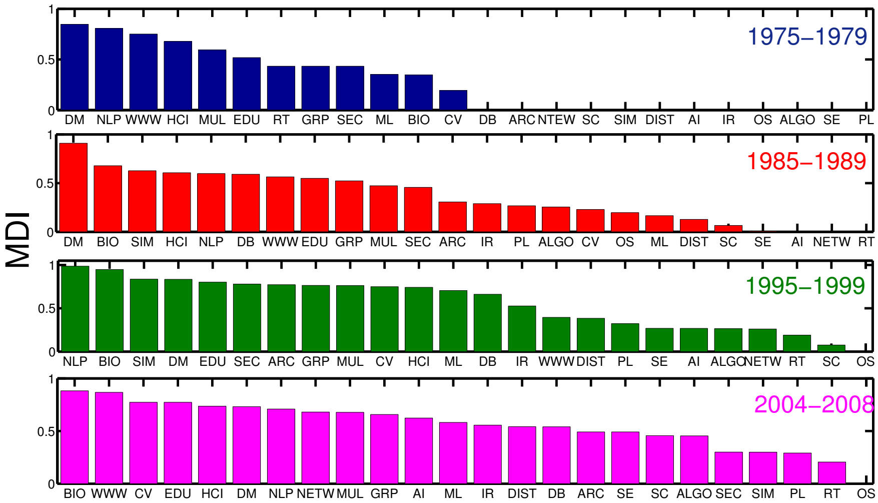


Figure 5.5: Membership Diversity Index (MDI) of all the fields in computer science domain in four time-windows. The x-axis is sorted (descending order) by the MDI value.

The interpretation of this difference Δ is as follows. If the temporal profile of Δ is roughly stable for a field then it would mean that the diversity of inward citations does not change over time. However, there are certain fields where at some point Δ rises abruptly indicating a sudden huge difference in the diversity between t_i and t_{i+1}. Following this point, the diversity remains high at all time points thus keeping the difference Δ stable once again for the rest of the time span. In Figure 5.4, we plot the Δ values of those fields for which we are able to detect such a large fluctuation at some time point in the entire profile. As shown in Figure 5.4, WWW shows a sudden peak between the time windows 1984-1988 and 1985-1989 and then gets stabilized. Similar behavior is observed for NLP between the time windows 1988-1992 and 1989-1993. Other fields mentioned in Figure 5.4 indicate similar characteristics. However, the only exception in Figure 5.4 is the Databases field which although seems to be a relatively core area of research shows a peak in Δ at around 1982-1986. Within a very short period, the Δ falls abruptly again (1983-1987) which is unlike the case of other fields discussed earlier. A closer inspection of our data shows that during the years 1982-1986, Databases received a variety of citations from fields like Computer Vision, Security and Privacy and Operating Systems. However, in the later years such citations to the Databases field are not found any more. A possible reason could be that in the later years Data Mining that had its birth from Databases (see Figure 5.8 later) started enjoying the cross-field citations rather than the Databases field itself.

(iii) Membership Diversity Index (MDI): The communities in citation network of a domain generally indicate different areas of research (see Section 5.2) where the intra-community citation density is higher than across communities [42]. We hypothesize that the diverse range of membership of a paper in different communities could be an indicator of its degree of interdisciplinarity. To verify our hypothesis, we conduct a community-centric measurement on the networks of four dynamic-windows (1975-1979, 1985-1989, 1995-1999 and 2004-2008). We use SLPA (Speaker listener Label Propagation Algorithm) [229] to detect overlapping communities in each dynamic-window. Then based on the membership of the overlapping nodes (papers) in each field, we define another metric called *Membership Diversity Index (MDI)* for each field as a measure of its interdisciplinarity.

Definition 3: ***Membership Diversity Index (MDI):*** *The MDI of a paper is the entropy expressing the extent of its membership to different communities. The MDI of a field is the average of the MDIs of all the papers belonging to that field.*

We run SLPA on the network of each dynamic-window that extracts the overlapping communities (say, $c_1, c_2, ..., c_n$). Since we know the actual field information of the papers, for each community c_j we can then find out the major field f_i such that c_j contains most of the papers from f_i. In this way, we can mark each community with a field tag that roughly signifies the research area indicated by this community. Note that it might be possible that more than one communities are marked by the same field tag since we have very few field categories (24 fields in the computer science domain) compared to the number of communities in each dynamic-window. Now for the field f_i, we extract only those papers that are part of overlapping communities in that time-window. These papers 'flagged' as overlapping papers within the field f_i form the basic constituent of the MDI measure.

We find out the membership of each such overlapping paper in the different field-tagged communities. Now, the MDI of the field f_i in a particular time-window is defined by the following equation:

$$MDI(f_i) = -\sum_{j=1}^{m} p_j log(p_j) \tag{5.7}$$

where p_j is the fraction of papers flagged as overlapping in f_i and is a member of the community tagged as f_j, while m is the number of fields (i.e., $m = 24$). The more the

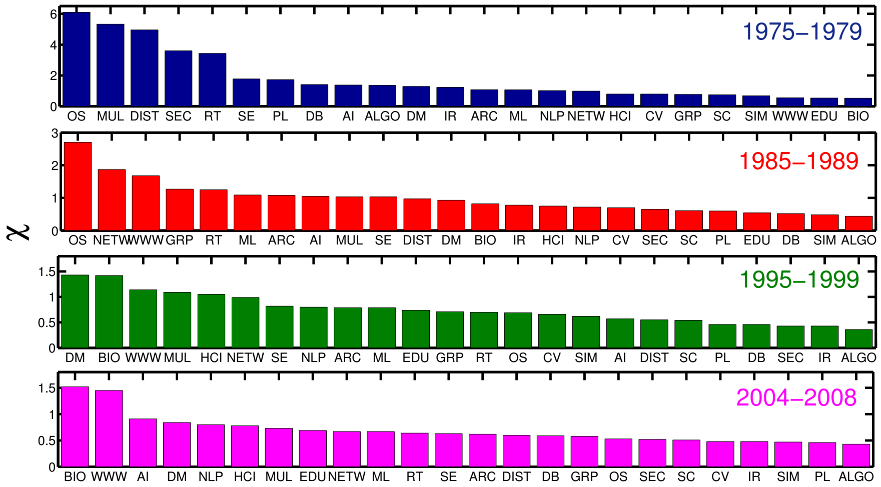


Figure 5.6: Attraction Index of all the fields in computer science in four time-windows. The x-axis is sorted (descending order) by the χ value.

MDI value of a field the more is its chance of interdisciplinarity.

Note that since the overlaps are measured in different dynamic sliding windows, a node that belongs to a specific community in one dynamic window may move to a different community (communities) in the subsequent dynamic window because its surrounding connectivity might change in the next time window. Figure 5.5 shows the fields of computer science domain in four different time windows in decreasing order of MDI. Here, while in the time windows (1975-1979) and (1985-1989), Data Mining is consistently found to be at the top, in the later years the fields like NLP and Computational Biology seem to acquire the top positions.

(iv) Attraction Index: The selection of the new research field for both the budding and experienced researchers mostly depends on the impact and popularity of the existing fields in any particular time period. Therefore, the study of inclination of the authors to adopt a new field can be one of the real and relevant evidences supporting the popularity of the fields in that time period. To quantify the attractiveness of a field, we use a simple measurement called *Attraction Index* (χ) discussed below.

Definition 4: ***Attraction Index*** *(χ): The Attraction Index of a field in a time window is defined by the number of new authors (normalized by the number of papers in that time window) who start research in that field in that time window.*

Let us assume that the number of unique authors from the beginning to the year t_i and to the the year t_{i+4} who published papers in field f are n_i and n_{i+4} respectively. The number of papers of field f published in time window $(t_i - t_{i+4})$ is c_i. Therefore, the *Attraction Index* of a field f at that time window denoted by χ_f is measured by the following equation.

$$\chi_f = \frac{n_{i+4} - n_i}{c_i} \tag{5.8}$$

In Figure 5.6, we plot the value of χ for all the fields (in decreasing order of χ) in four different time windows. We can observe that though the fields like OS, Networking hold the top few positions in terms of χ in the earlier two time windows (1975-1979 and 1985-1989), in the recent years, these positions are gradually occupied by the fields like Computational Biology, WWW, Data Mining. We posit that this observation can be a distinctive factor to categorize core and interdisciplinary fields.

5.4.2 Unsupervised Classification Model

In the previous section, we have proposed four features with the intention that they would be indicative to explore the degree of interdisciplinarity of a field as well as help classifying the core and interdisciplinary fields. In this section, we propose an unsupervised classification model that can effectively cluster the fields based on the similarity of these features. Note that we only consider the most recent time period of 1995-2008 for this classification[6]. In this model, each field f is represented by a feature vector of size four. The entries of the vector correspond to the value of four features namely $RDI(f)$, Δ_f, $MDI(f)$ and χ_f. Then we create a symmetric adjacency matrix $A_{24\times24}$ whose (i,j) cell, $A(i,j)$, denotes the cosine similarity of the feature vectors corresponding to the fields f_i and f_j. For instance, let us assume that V_i and V_j represent the feature vectors corresponding to the fields f_i and f_j respectively. Then $A(i,j)$ represents the cosine-similarity

[6] The features are most discriminative in this time-window.

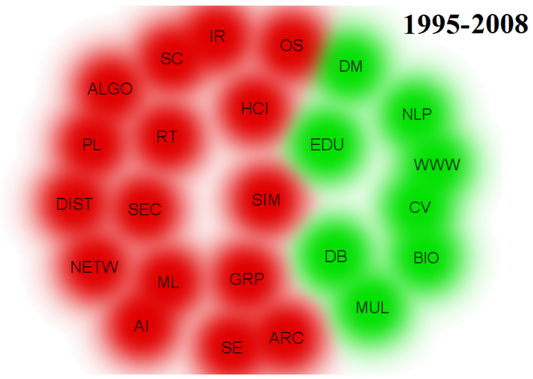


Figure 5.7: The result of the unsupervised classification model. Two clusters are represented by two different colors (red and green).

between the feature vectors V_i and V_j as indicated by the following equation:

$$A(i,j) = cos(V_i, V_j) = \frac{\sum_{k=1}^{4} V_{ik} \times V_{jk}}{\sqrt{\sum_{k=1}^{4} V_{ik}^2} \times \sqrt{\sum_{k=1}^{4} V_{jk}^2}} \tag{5.9}$$

An undirected and weighted network is created based on the adjacency matrix A, and the network is fed into the classification module. We use the algorithm proposed by Waltman et al. [218] for the unsupervised clustering.

The results of the clustering algorithm is pictorially depicted in Figure 5.7. It is apparent from the figure that the fields get divided into two distinct clusters. The cluster represented by the green color comprises eight fields; all of them seem to be interdisciplinary fields except Databases. The reason could be that the fields like WWW, NLP, Data Mining got the major motivation and ideas from Databases when emerging as separate fields (see Figure 5.8 for further details). Therefore, though individual features could not reflect this similarity properly, their combination efficiently unveils the latent similarity in the clustering results. On the other hand, the cluster represented by the red color consists mainly of the fields which show their consistent existence from the very beginning. Therefore, this cluster seems to be representing the core fields of computer science. To the best of our knowledge, this is the first attempt to present a quantitative definition of interdisciplinarity in terms of a set of distinctive features that neatly separates out the core from the interdisciplinary research areas.

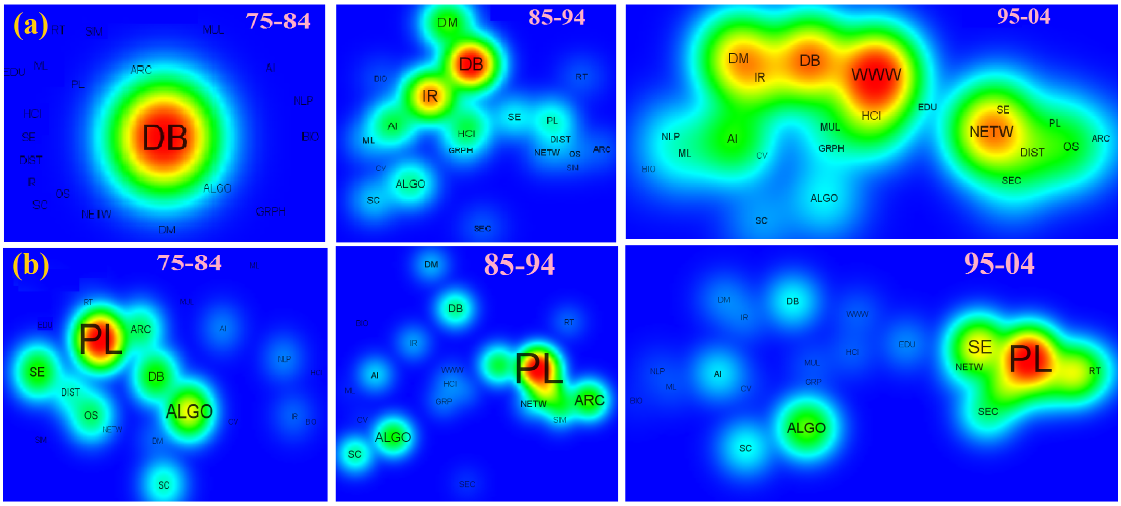


Figure 5.8: Evolutionary landscape of (a) WWW and (b) Programming Languages (PL) based on the references. Top panel shows a constant level of interaction among Databases, Data Mining, IR resulting in a new field - WWW; whereby core field like PL remains same over the years.

5.4.3 Evolutionary Landscape of Interdisciplinary Fields

Since from the previous section, we obtain two distinct clusters of core and interdisciplinary fields in the computer science domain, the immediate question we ask is that how such an interdisciplinary field could have evolved from the cross-fertilization of the various core fields. Are the citation-based evidences capable of unfolding the evolutionary landscape of an interdisciplinary field? To answer this question, we concentrate on the temporal interaction patterns among the fields through citations over the last four decades. We hypothesize that if an interdisciplinary field has evolved from two or more fields (say, $f_1, f_2, ..., f_n$), the interactions among the fields $f_1, f_2, ..., f_n$ over the years should show a steady growth due to the sharing of knowledge and principles through cross-citations.

For this purpose, we construct a field-field citation network $G_f =< V_f, E_f >$ on top of the paper-paper citation network in each time window, where each node $f_i \in V_f$ indicates a field f_i (a collection the papers related to f_i), and a directed and weighted edge $e_{ij} \in E_f$ from f_i to f_j denotes the number of citations from the papers of field f_i to the papers of field f_j. Thus, in our experiment, we have maximum 24 vertices (if there exists at least one paper in a field it qualifies as a vertex) in G_f at any time point. Then, we study the temporal interactions of the vertices in each time window. For the sake of conciseness, here we present the evolutionary landscape of one interdisciplinary field (WWW) and one core field (Programming Languages) which exhibit a consistent ranking for all the metrics discussed in section 5.4.1. In Figure 5.8, we draw the contour heat maps showing the evolution pattern of WWW (top panel) and PL (bottom panel) over the last four decades. This figure has following two utilities. First, it takes into account the distance of two vertices as the inverse of the edge weight connecting them and groups them accordingly (green regions). In addition, the size of the font and the red circle around each vertex (field) indicates the relative importance of the vertex. Here the size of each vertex in Figure 5.8(a) indicates the amount of citation received by the field (corresponding to the vertex) from the papers of WWW (similarly from the papers of PL in Figure 5.8(b)). Furthermore in each time step, an automated threshold is defined to exclude the fields which have not received sufficient amount of citations compared to the others. As shown in Figure 5.8(a), the papers of WWW have cited only the papers of Databases in early times (1975-1984); but in the later time window (1985-1994), the citations get divided among Databases, IR and Data Mining. Moreover, a distinct group comprising Databases and Information Retrieval starts evolving with small contributions from Data Mining, AI and Human Computer Interaction. Till this point, WWW is missing from the frame due to the small number of inward citations. In the latest time stamp (1995-2004), WWW is found to receive huge self citations and the previous group is enlarged with the pronounced involvement from WWW, DM, DB and IR. It clearly explains the evolution dynamics of WWW. However, another group is noticed in the last time window consisting mainly of Networking, Software Engineering, Distributed Systems and Security & Privacy. This probably indicates another line of interdisciplinary research manifesting in the form of secured distributed networking. On the other hand, if we look at Figure 5.8(b) demonstrating the evolution of Programming Languages (PL), a constant appearance of PL is noticed from the very beginning. This indicates that Programming Languages was one of the contributory fields in computer science domain earlier and remains significant afterwards. This could be the first and fundamental study to understand the basic ingredients responsible for the formulation of a new field of research and helps develop a prediction system capable of recommending the probable fields whose cross-fertilization can produce another field of research in the near future.

5.4.4 Core-periphery Analysis

We understand the impact of the research fields on the entire domain in a systematic way by studying the core-periphery organization [35] of the citation network. The idea is to decompose the fields into various shells in a particular year (or in a dynamic time window) such that a high $k^s - shell$ index of a field reflects a central position in the core of the network. As mentioned earlier, both the inward and outward citations play pivotal roles in determining the impact of a field in its domain. Therefore, we take into account both of them separately to perform the k-core decomposition in four different dynamic windows (i.e., 1975-1979, 1985-1989, 1995-1999, 2004-2008).

We start by recursively removing nodes that have single link until no such nodes remain in the network. These nodes form the 1-shell of the network ($k^s - shell$ index k^s = 1). Similarly, by recursively removing all nodes with degree 2, we get the 2-shell. We continue increasing k until all nodes in the network have been assigned to one of the shells. The union of all the shells with index greater than or equal to k^s is called the k^s-core of the network. We repeat the experiment both for in-citation and out-citation of a node separately. Since the shell index is assigned to each paper, we calculate the fraction of papers of a field in each k^s-core of the network in each dynamic window to identify the fields of a domain that sit at the core of the network.

The multi-level pie charts in Figure 5.9 (a) in four dynamic time-windows show how the different branches of computer science are positioned with respect to the core-periphery organization of the citation network (considering inward citations). Each level of the pie-chart represents one of the k^s-shell regions, i.e., the innermost layer represents Region I (largest k^s-shell index), followed by Region II, Region III, and finally the outermost layer represents the peripheral Region IV. In each layer, we show the fraction of papers belonging to a field. The pie charts for the time windows 1975-1979 and 1985-1989 show that the Region I consists mostly of core fields like Databases, Programming Languages and Software Engineering; while after that it is dominated by the more applied fields like Networking, Distributed Systems, Data Mining with a small contribution from Hardware & Architecture and Databases. In all other regions, all branches of computer science are present. From these results, we can infer that the core of the computer science is gradually

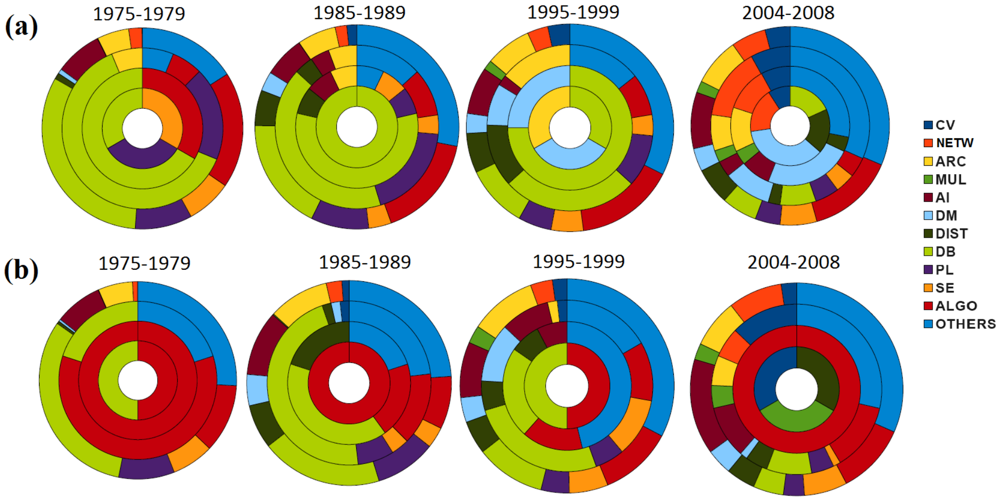


Figure 5.9: Multilevel pie-chart for the dynamic windows 1975-1979, 1985-1989, 1995-1999 and 2004-2008 showing the core-periphery organization of the citation network of computer science with respect to (a) inward citations and (b) outward citations.

being shaped by the more applied fields.

As mentioned earlier, while inward citation represents the *authoritativeness* of a field, the outward citation shows the *hubness* of a field, i.e., the propensity of a field to cite others. The degree of hubness of a field is equally important to measure its impact since the high degree hub papers (fields) usually act as the connectivity backbone of the network, sometimes creating paths between distant fields thereby, unfolding a scope for the emergence of new transdisciplinary fields. Therefore, we extract the core-periphery organization of citation network with respect to the outward citations as shown in Figure 5.9 (b). Surprisingly, while Algorithms and Theory has been consistently appearing at the periphery region in Figure 5.9 (a), the core regions are heavily dominated by Algorithms in Figure 5.9 (b) along with an additional contribution from Databases. Recently, the core region is covered by the emerging fields like Computer Vision, Multimedia and Distributed Systems. In short, Figure 5.9 presents a clear indication of the position of different fields within the domain and that the interdisciplinary fields are accelerating steadily toward the core of the domain.

5.5 Understanding Scientific Career of Researchers

"It is really important to do the right research as well as to do the research right. You need to do 'wow' research, research that is compelling, not just interesting."

– Richard M. Reis, Stanford University

Of all the decisions we make as an emerging scientist, none is more important than identifying the right research area, and in particular, the right research topic. The success of scientific career gets determined by these two choices. Change in scientific research career can be defined as any major change in work-role requirements or work context [32,33,166] and as a process that may result in a change of job, profession, or a change in one's orientation of work while continuing in the same job [5, 62]. People believe that many factors act as an active role to regulate these changes. For instance, researchers might try to align themselves with the cutting-edge research at the current time and as a result of this a change in scientific research career becomes unavoidable [173]. On the other hand, this career shift might be described as an effect of "saturation" in the field of a researcher leading to a switch to the other fields [177].

Here, we use the same bibliographic dataset of computer science domain mentioned in Section 5.2 and attempt to analyze the local and global dynamics regulating a researcher's decision to select new field of research over the entire career. Essentially, we intend to answer some specific questions pertaining to a researcher's scientific career – how are the local and the global dynamics regulating a researcher's decision to select new field of research over the entire career? what are the suitable quantitative indicators to measure the diversity of a researcher's scientific career? We further build a stochastic model that can reproduce the real-world phenomenon of field selection process. Evaluations of our model through the real-world data lead us to conclude that our model, quite accurately, mimics the field selection process for all the researchers present in the dataset. Note that, we use the terms "author" and "researcher" interchangeably in the rest of the chapter.

5.5.1 Diversity Measures

Diversity of an author's research career can be understood as the degree of variation/changes in research fields over the entire career. Since "diversity" of a sequence can be efficiently measured by Shannon's entropy [196], we propose two different versions of entropy measurement to quantify diversity of an author's research career. If F is the set of unique fields of papers written by an author a, the ***plain entropy*** of author a (denoted by $H_p^a(F)$) is calculated over the number of times author a writes papers in a particular field in her entire research career as defined by the following equation:

$$H_p^a(F) = -\sum_{i \in F} p_i log(p_i) \tag{5.10}$$

where $p_i = \frac{number\ of\ papers\ written\ by\ a\ in\ field\ i}{total\ number\ of\ papers\ written\ by\ a}$. Zero plain entropy implies that an author worked in a single field throughout her career whereas a high value indicates that she has worked in various fields in different time spans of her career. However, the plain entropy does not capture the order information of a particular value within the sequence; it just considers probability distributions, i.e., if we interchange the position of different entries in a sequence keeping the frequency contribution of each individual field same, the plain entropy of the sequence remains constant. In our case, since our primary interest is to understand the change in research fields adopted by an author in different time periods, the ordering information of fields in the sequence turns out to be important. Therefore in order to capture the local diversity, we propose another measure of field diversity for an author a called the ***window entropy*** (denoted by $H_w^a(F)$) defined as follows – a window of size k slides over the sequence of fields in F, the plain entropy for the sequence contained within that window at each position is calculated, and the mean of all these positions is computed to measure the window entropy of the entire sequence as described in the following equation:

$$H_w^a(F) = -\frac{1}{n-k+1} \sum_{i=1}^{n-k+1} H_p^a(w_i) \tag{5.11}$$

where w_i is the set of fields in the i^{th} sliding window of size k, ranging from i to $(i+k-1)$. The window entropy indicates the diversity in the selection of fields in short spans of time by considering only previous k fields in the sequence. The motivation behind these measures is to understand whether the author is working simultaneously in diverse fields throughout the career or she is following the "scatter-gather" policy, i.e., while working in diverse fields at the macro scale, at the micro scale, concentrating on only one particular field within a given time slice. Low $H_w^a(F)$ indicates that the author indeed follows a "scatter-gather" policy; whereas high values indicate that the author has a tendency to work in many different fields simultaneously within short periods of time.

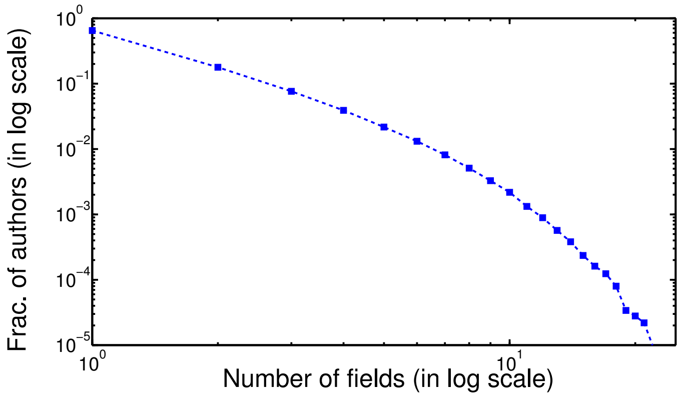


Figure 5.10: Distribution of fields adopted by the authors (plotted in log-log scale). The y-value corresponding to the x-value indicates the fraction of authors contributing to x number of fields in their careers.

5.5.2 Experimental Results

Statistical analysis of authors' careers: We first plot in Figure 5.10 the distribution of fields selected by the authors over their entire career. It follows a truncated power-low behavior and shows that around 64% of the total authors worked only in one field, 18% of the total authors worked in two fields and so on. In Figure 5.11(a), we show the average number of fields in which an author contributed in a particular year from the start of her career (i.e., after her first publication). It can be observed that as the career of an author progresses over time, the number of distinct fields she has contributed to increases till around fifteen years and then mostly stabilizes. This may be understood as an author's career profile, in the initial years a scientist is usually more actively developing skillset in a field and is keen setting up collaborations in the field as well as in the closely associated ones. Eventually, after fifteen years of her scientific career, she would generally tend to focus on a fixed set of fields where she has considerable expertise; thus a decline forward at the end of the curve is observed in Figure 5.11(a). In Figure 5.11(b), we plot the total

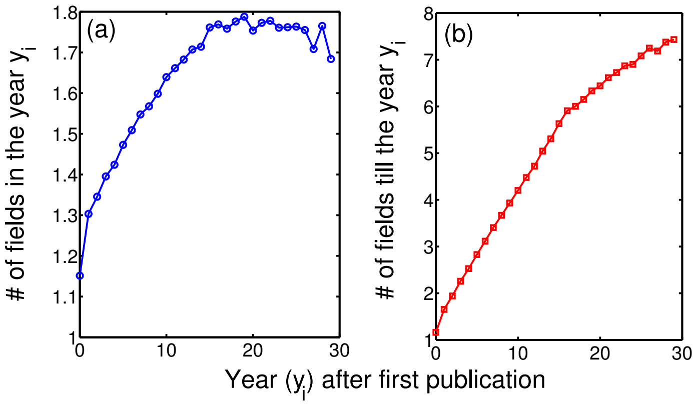


Figure 5.11: (a) Average number of fields contributed by an author in a year after the first publication and (b) total number of fields in which an author contributes till a particular year after the first publication. Both of these are calculated as an average over all authors present in the data set.

number of fields contributed by an author (on an average) up to a certain year after her first publication. It is to be noted that this plot cannot be obtained directly by cumulating the result in Figure 5.11(a) since we measure the total number of distinct fields in an author's career in Figure 5.11(b); whereas cumulating over Figure 5.11(a) might count a field more than once. It can be observed in Figure 5.11(b) that the total number of fields to which an author contributes increases till around fifteen years and at a relatively lower rate afterward. This plot also indicates the average number of years that an author usually takes to start contributing to the i^{th} field. For example, an author starts contributing to the third field in about 6-7 years from the start of her career.

Diversity of authors' scientific career: Next, we analyze the diversity of an author's scientific career at different time points in terms of two proposed entropy-based measures, namely the plain entropy ($H_p^a(F)$) and the window entropy ($H_w^a(F)$). Figure 5.12(a) shows the plain entropy probability histogram of all authors, i.e., fraction of authors with plain entropy ranges divided into several buckets. It can be observed that high fraction of authors tend to have small plain entropy since most of them have worked in very few research fields. Figure 5.12(b) shows the average plain entropy of sequence of fields of an author till a certain number of publications. We take an empirical cutoff of 80 publications while computing the entropy measures because most of the authors fall in this region. The publications of authors after this cutoff are ignored for the entropy calculation. One can observe the increase, relative to the number of fields selected by an author which is faster

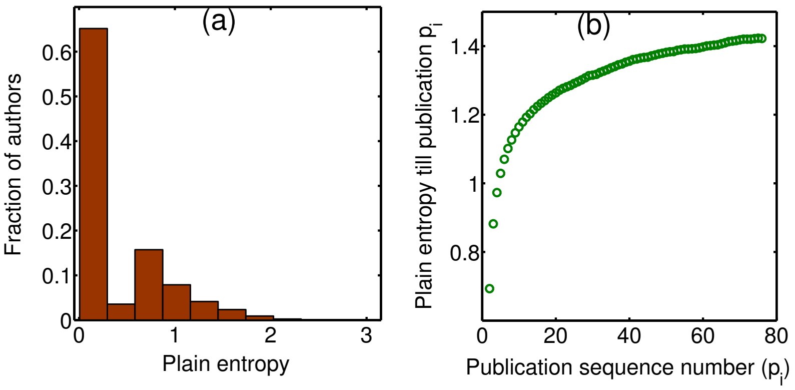


Figure 5.12: (a) Plain entropy distribution histogram of authors and (b) plain entropy till a certain number of publications (averaged over all the authors).

at the early stages following a gradual stabilization toward the end.

Similarly, we plot both these figures for the window entropy in Figure 5.13. Interestingly, the major concentration of authors lies in 0.5-1.2 window-entropy region plus a significant amount of mass in the first bucket. From both the histograms of plain and window entropies, one can conclude that though the average behavior indicates that an author tends to select only a few research fields in her entire research career, she seems to prefer working simultaneously in multiple fields at a particular time point. We observe that such fields are quite related to each other in the sense that if we can take more coarse-grained field classification scheme, these fields might come within the category of a top-level field. For instance, Graphics and Multimedia are so closely related fields in computer science that one can assume these as two subfields under the field called “Image Processing.” The scatter plot in Figure 5.13(b) indicates that researchers, at the beginning of their research, tend to select a number of fields simultaneously thus making the average entropy higher. The reason could be that they are initially not very confident which particular fields of research they should select in order to survive in the scientific community. Gradually with experience, they tend to get stabilized after publishing 20-25 papers, thus reaching a lower entropy at the middle of the curve. Following this, again a steady growth of the curve indicates that possibly they start collaborating with other researches from the other fields and hence, by doing this, they tend to advocate interdisciplinary research toward the end of their careers.

Correlation between plain and window entropies: In Table 5.6, we present a confusion matrix indicating the correlation between the two diversity measures. We classify the

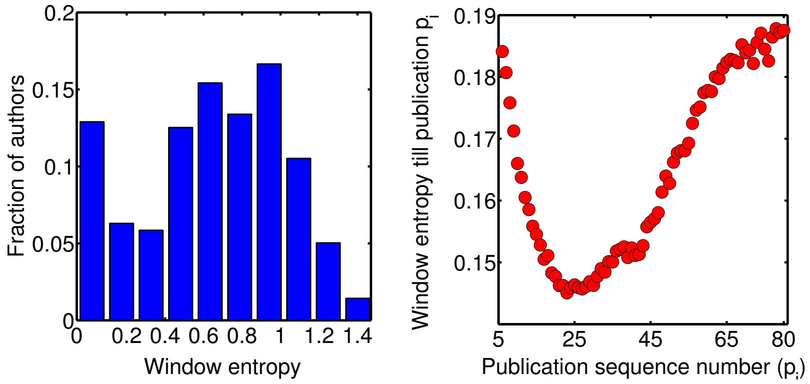


Figure 5.13: (a) Window entropy distribution histogram of authors and (b)window entropy till a certain number of publications (averaged over all the authors).

Table 5.6: Confusion matrix indicating the population density of authors (in %) and average number of citations obtained by an author (in parenthesis) in different regions.

		Window entropy	
		Low	High
Plain entropy	Low	43.37 (11.61)	6.6 (11.21)
	High	6.61 (13.36)	43.41 (10.38)

entire population of authors into four parts corresponding to the four cells of the matrix. Note that the low (high) entropy values correspond to values below (above) the median value of the respective type of entropy. Each region on the matrix corresponds to different types of career profile. For instance, the region corresponding to low window entropy and high plain entropy indicates that the authors here do not work simultaneously in multiple fields; rather they choose to work in fields one after the other. On the other hand, the region indicating high plain and window entropies corresponds to those authors who have worked in diverse areas and also contributed simultaneously to multiple fields at any particular time point. Table 5.6 shows the population density (in percentage) of authors in each region. In parallel, what would be more interesting to investigate is the importance of each such region, i.e., what would be the preferred strategy a new author should adopt in order to acquire higher importance in scientific community. In any bibliographic dataset, a raw measure to quantify the importance of an author is usually the number of citations she has received by publishing papers. Therefore, we measure the importance of each region by

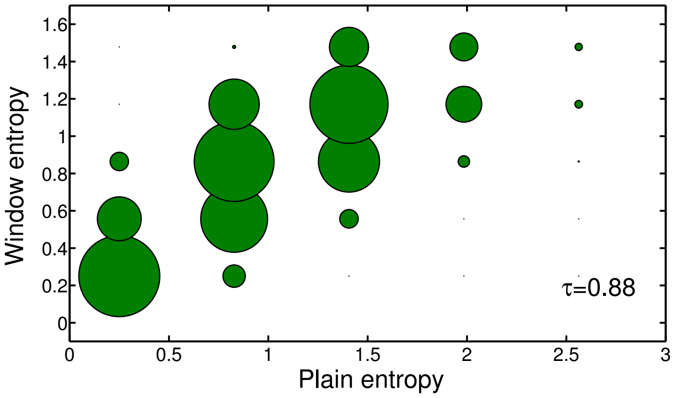


Figure 5.14: Scatter plot showing the correlation between plain entropy and window entropy with size of the circle proportional to number of points in the region surrounding the center of the circle.

calculating the average citations an author of the corresponding region has received (the value within parenthesis in each region of Table 5.6 indicates this importance). The key observation is that the region with high plain entropy and low window entropy has high average citation value compared to all other regions. This indicates that the highly cited authors follow a "scatter-gather" policy, i.e., work in diverse fields over their entire career but remain confined to a few fields in each time slice. The authors in the region with high plain entropy and high window entropy have least average citation count which indicates that the authors who have worked in a large majority of fields in the entire career as well as in each shorter time slice get low citations. It is worth noting that in this experiment, we consider only those authors who have published at least five articles because the size of the sliding window for calculating the window entropy is assumed to be five. We further vary the window size, but similar results are observed in all the regions for different window values.

Next we measure the correlation between plain and window entropies using Pearson correlation coefficient (τ) as shown in Figure 5.14. The size of the circle is proportional to number of points in the region surrounding the center of the circle. The correlation is very high (τ=0.88) between these two entropies which indicates a strong dependency between them. These results once again agree with the proportion of authors shown as two most popular regions on the principal diagram of the confusion matrix (i.e., low plain entropy and low window entropy, high plain entropy and high window entropy). This correlation increases with the increase in the size of the window since k-window entropy with large

k value gradually approaches the plain entropy. Furthermore, it would be interesting to understand to what extent an author is influenced by her other colleagues (coauthors) in selecting a new field. More particularly, we intend to measure the correlation between the entropy of an author with her strongest collaborator (with whom she has published the largest number of papers) and to what extent this correlation changes when compared to any arbitrary coauthor. Figure 5.15(a) shows the scatter plot of the plain entropy of an author with the plain entropy of her strongest collaborator. The Pearson correlation between them is quite high (0.52) compared to any arbitrary coauthor (τ=0 as shown in Figure 5.15(b)). This result can have two implications: (a) an author either tries to align herself in the direction of her strongest collaborator in choosing research fields or (b) an author chooses such a collaborator with whom the research interests have maximum alignment. Similar correlation exists for the case of window entropy (since plain and window entropies are highly correlated as discussed in Figure 5.14).

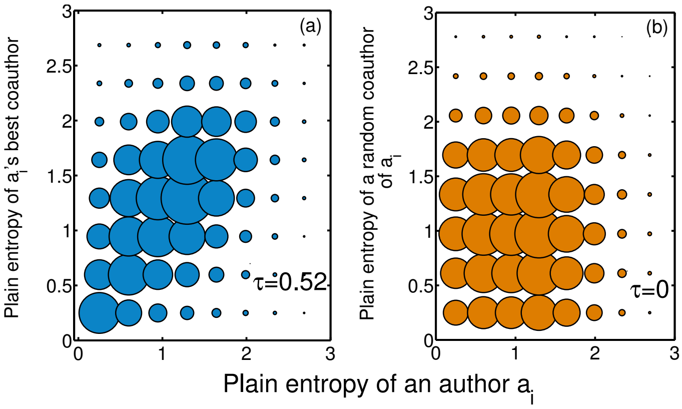


Figure 5.15: Scatter plot of the plain entropy of an author (a) with her strongest collaborator and (b) with any arbitrary coauthor.

5.6 Formation of Circles in Coauthorship Networks

The availability of an overwhelmingly large amount of bibliographic information including citation and co-authorship data makes it imperative to have a systematic approach that will enable an author to organize her own personal academic network profitably. An effective method could be to have one's co-authorship network arranged into a set of "circles",

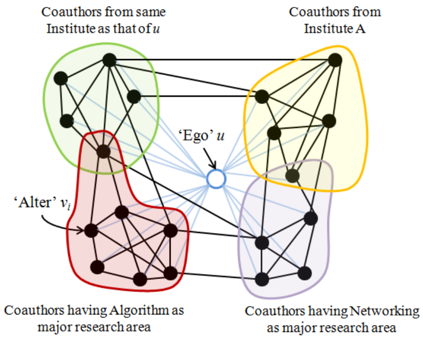

Coauthors from same Institute as that of u
Coauthors from Institute A
'Ego' u
'Alter' v_i
Coauthors having Algorithm as major research area
Coauthors having Networking as major research area

Figure 5.16: A hypothetical example showing an ego network of an author u with labeled circles.

which has been a recent practice for organizing relationships (e.g., friendship) in many online social networks.

Here, we study the problem of automatically discovering an author's academic circles. In particular, given a single author with her co-authorship network, our goal is to identify her circles, each of which is a subset of her coauthors. Some examples of real-world circles in an author's co-authorship network are shown in Figure 5.16. The "owner" of such a network (the "ego") may wish to form circles based on common bonds and attributes among her coauthors (the "alters"). An author could have several reasons behind initiating a new collaboration. Some common tendencies exhibited by authors include collaborations with the people from her own Institute or with people sharing the same research interest with her. Therefore, the problem of deciding upon a single dimension to both characterize the circles and categorize the coauthors appropriately becomes extremely challenging. Moreover, circles are author-specific, as each author organizes her personal network of coauthors independent of all other authors with whom she is not connected. This leads to a problem of designing an automatic method that organizes an author's academic network, more precisely, categorizes her surrounding neighborhoods into meaningful circles.

5.6.1 An Unsupervised Model for Discovering Ego-centric Circles

Our model for detecting ego-centric circles applies to any general ego network, where each node is considered as an ego and the set of her one-hop neighbor nodes constitute the set of alters. The ego is said to spawn the ego network, but is not considered as a part of the network. Our method intends to discover circles in this ego network in an unsupervised fashion, leveraging properties specific to nodes as well as properties of the network. Our model requires each node to have a profile, which is essentially the feature vector characterizing the node in a feature space. We now describe the algorithm for circle formation in more details. The input to our algorithm is an ego network $G =< V, E >$. Each node $v \in V$ has an N-dimensional profile vector F_v = {f_{1v}, f_{2v}, f_{3v}, ..., f_{Nv}}, where f_{iv} denotes the value of the i^{th} feature of the node v. The ego node u, often referred to as the *center* node, is responsible for spawning the ego network, but does not itself feature as a part of the network. So the ego network of u is essentially the subgraph induced by the alters of u. Let $D(x, y)$ be the Euclidean distance between the profile vectors of nodes x and y given by Equation 5.12.

$$D(x, y) = D(y, x) = \sqrt{\sum_{i=1}^{N}(f_{ix} - f_{iy})^2} \tag{5.12}$$

The aim of the method is to identify a set of circles $\hat{C}$= {C_1, C_2,.....,C_K}. Given a circle $C_j \in \hat{C}$ and a node $y \in V$, we define the distance of y from C_j, say $D'(C_j, y)$, as the average distance of y from all other nodes in C_j. Also, the profile similarity measure between a pair of nodes x and y, denoted by $Sim(x, y)$ is defined to be the reciprocal of $D(x, y)$. Analogously, the similarity between node y and circle C_j, denoted by $Sim'(C_j, y)$ is defined to be the reciprocal of $D'(C_j, y)$.

Each circle C_j in our model has a similarity threshold parameter τ_j associated with it such that if node $y \in V$ is in C_j then the following constraint is satisfied:

$$Sim'(C_j, y) \geq \tau_j \tag{5.13}$$

Based on our assumption that nodes within a common circle at any point of time have a higher probability of forming an edge in the network, our model predicts the circles estimated at each step to be cliques, and distinct circles not to share any edge at all. Given a set of K circles $\hat{C}$= {C_1, C_2,.....,C_K}, along with a set of threshold parameters $\hat{\tau}$ = {τ_1, τ_2,...,τ_K} in any iteration of the algorithm, we define a *closeness* estimator for a pair of nodes $(x, y) \in V \times V$ in terms of their circle membership, denoted by $\beta(x, y)$. Let $\beta_1(x, y)$

and $\beta_2(x,y)$ be defined as follows.

$$\beta_1(x,y) = \sum_{C_j:\{x,y\}\subseteq C_j} (Sim(x,y) - \tau_j + \lambda)^{-1} \tag{5.14}$$

$$\beta_2(x,y) = \sum_{C_j:\{x,y\}\nsubseteq C_j} (Sim(x,y) - \tau_j + \lambda)^{-1} \tag{5.15}$$

Note that $\{x,y\} \subseteq C_j$ if both x and y are members of the circle C_j, while $\{x,y\} \nsubseteq C_j$ if C_j does not contain one or both of x and y. The constant λ is kept large enough to ensure that no term in the summation is negative and may simply be taken as the maximum of all threshold values, i.e., $max\{\tau_1, \tau_2, ..., \tau_K\}$. Note that $\beta_1(x,y)$ is high if x and y share common circles with very high thresholds, while $\beta_2(x,y)$ is high if x and y do not share common circles with high thresholds.

Now, we define the *closeness* estimator $\beta(x,y)$ as follows.

$$\beta(x,y) = \exp\{[\beta_1(x,y)]^2 - [\beta_2(x,y)]^2\} \tag{5.16}$$

Note that $\beta(x,y)$ is purely a circle-membership based similarity metric for the pair (x,y), and increases with increase in the *number* and *threshold values* of the *common circles* which x and y are part of. Thus, the *closeness* estimator emphasizes not only the common circle memberships of nodes but also the thresholds of the circles they are part of.

From the *closeness* information so estimated, the probability that the pair (x,y) forms an edge in G is modeled by:

$$p((x,y) \in E) = \frac{\beta(x,y)}{1+\beta(x,y)} \tag{5.17}$$

Similarly, for the node-pair (x,y) which does not belong to E, the probability is estimated as follows:

$$p((x,y) \notin E) = 1 - p((x,y) \in E) = \frac{1}{1+\beta(x,y)} \tag{5.18}$$

Quite evidently, $p(x,y)$ increases with increase in $\beta(x,y)$ and is normalized using *add-one smoothing*. Thus we get a predicted probability of existence for each possible edge in the network given $\hat{C}$ and $\hat{\tau}$. The rationale underlying the prediction is that the *closeness* of a pair of nodes (x,y) is proportional to the similarity of their profiles as well as the number and similarity thresholds of common circles that they are a part of. Now the model must ensure that this predicted network indeed corresponds to the real network, for which we present the following analysis.

Assuming independent generation of each edge in the graph, the joint probability of G and $\hat{C}$ can be written as

$$P_{\hat{\tau}}(G;\hat{C}) = \prod_{(x,y)\in E} p((x,y)\in E) \prod_{(x,y)\notin E} p((x,y)\notin E) \tag{5.19}$$

We define the following notation 5.20 for ease of expression:

$$\phi(x,y) = \log\left(\beta(x,y)\right) = ([\beta_1(x,y)]^2 - [\beta_2(x,y)]^2) \tag{5.20}$$

Taking logarithm of Equation 5.19, and using notation 5.20 we can express the log likelihood of G given $\hat{C}$ and $\hat{\tau}$ as:

$$\begin{aligned} l_{\hat{\tau}}(G;\hat{C}) &= \log\left(P_{\hat{\tau}}(G;\hat{C})\right) \\ &= \sum_{(x,y)\in E} \log\left(p((x,y)\in E)\right) + \sum_{(x,y)\notin E} \log\left(p((x,y)\notin E)\right) \\ &= \sum_{(x,y)\in E} \log\left(\beta(x,y)\right) - \sum_{(x,y)\in V\times V} \log(1+\beta(x,y)) \\ &= \sum_{(x,y)\in E} \phi(x,y) - \sum_{(x,y)\in V\times V} \log(1+\exp\{\phi(x,y)\}) \end{aligned} \tag{5.21}$$

5.6.2 Unsupervised Learning of Model Parameters

In this section, we describe the method used to find the set of circles $\hat{C}$ by maximizing the log likelihood in Equation 5.21. Initially, each node is in a different circle with a very high threshold value. At each iteration t, for each node $y \in V$ we alter the circle membership of y by randomly adding it to some circles it previously did not belong to and deleting it from some circles it belonged to. The circle thresholds are then updated accordingly such that the constraint in Equation 5.13 is not violated.

The general idea is that larger the number of circles a node y is already part of after time step t, lesser is the extent to which the circle membership of y is disturbed in time step $t+1$.

We denote by $\hat{C_t}$ the set of circles and by $\hat{\tau_t}$ the corresponding set of thresholds after time step t, where $\hat{C_t}$ = $\{C_1(t)$, $C_2(t)$,...,$C_K(t)\}$ and $\hat{\tau_t}$ = $\{\tau_1(t)$, $\tau_2(t)$,...,$\tau_K(t)\}$. Also, let the log likelihood of G given $\hat{C_t}$ and $\hat{\tau_t}$ be $l_{\hat{\tau}}(G; \hat{C_t})$. The following are the main steps of the algorithm to update the circle in time step $t+1$:

Step 1: For each node $y \in V$, we capture the circle membership of y at time t by defining two sets $S1_{y,t}$ and $S2_{y,t}$:

$$S1_{y,t} = \{C_j(t)|C_j(t) \in \hat{C_t} \wedge y \in C_j(t)\} \tag{5.22}$$

$$S2_{y,t} = \{C_j(t)|C_j(t) \in \hat{C_t} \wedge y \notin C_j(t)\} \tag{5.23}$$

Step 2: Now we intend to compute the number of circles to add y to and to remove y from, given by the two variables - $AddCircle(y, t+1)$ and $RemoveCircle(y, t+1)$:

$$AddCircle(y, t+1) = \left\lceil \frac{K1 + |S1_{y,t}|}{|S1_{y,t}|} \right\rceil \tag{5.24}$$

$$RemoveCircle(y, t+1) = \left\lceil \frac{K2 + |S1_{y,t}|}{|S1_{y,t}|} \right\rceil \tag{5.25}$$

Here, $K1$ is a randomly chosen integer with $1 \leq K1 < |S2_{y,t}|$, such that the value of $AddCircle(y, t+1)$ is less than or equal to $|S2_{y,t}|$, i.e., the number of circles that y is currently not part of. Similarly, $K2$ is a randomly chosen integer with $1 \leq K2 < |S1_{y,t}|$ such that the value of $RemoveCircle(y, t+1)$ is less than or equal to $|S1_{y,t}|$, i.e., the number of circles that y is currently part of. Note that both $AddCircle(y, t+1)$ and $RemoveCircle(y, t+1)$ are low for high values of $|S1_{y,t}|$. This ensures that the more the number of circles y is currently part of, lesser is the disturbance to the circle membership of y (and vice versa).

Step 3: Add y to $AddCircle(y, t + 1)$ many randomly chosen circles from $S2_{y,t}$ and remove y from $RemoveCircle(y, t + 1)$ many randomly chosen circles from $S1_{y,t}$. The corresponding circles are updated accordingly.

Step 4: Once Steps 1, 2 and 3 are over for each node, we have the set $\hat{C}_{t+1}$ containing the augmented circles. Next, we update the corresponding thresholds by setting $\tau_j(t + 1)$ corresponding to the circle $C_j(t + 1)$ to the minimum value such that for each node $y \in C_j(t + 1)$ the constraint in Equation 5.13 is not violated. Thus the updated $\tau_j(t + 1)$ for $C_j(t + 1)$ is given by:

$$\tau_j(t+1) = min\{Sim'(C_j(t+1), y)|y \in C_j(t+1)\} \tag{5.26}$$

Step 5: If the threshold $\tau_j(t + 1)$ for $C_j(t + 1)$ falls below a constant lower limit τ_L, we discard $C_j(t + 1)$. The value of τ_L is empirically determined. In our experiments, we tested over a wide range of τ_L and set it to 0.2 for best results (see Figure 5.17).

Step 6: We then compute the log likelihood $l_{\hat{\tau}_{t+1}}(G; \hat{C}_{t+1})$ using Equation 5.21. If $l_{\hat{\tau}_{t+1}}(G; \hat{C}_{t+1}) > l_{\hat{\tau}_t}(G; \hat{C}_t)$, then retain newly computed sets $\hat{C}_{t+1}$ and $\hat{\tau}_{t+1}$; else set $\hat{C}_{t+1} = \hat{C}_t$ and $\hat{\tau}_{t+1} = \hat{\tau}_t$.

The process continues till we reach a maxima and the log likelihood does not increase any further for sufficiently many iterations. We then report the set of circles so obtained as the optimal set of circles.

5.6.3 Feature Extraction

Profile information of each author node in the ego network is represented as a feature vector consisting of a set of features. These features can be divided into two broad categories – *general* and *ego-centric* features. Having these two separate categories, the feature set emphasizes the fact that members of common circles should not only have high feature similarity with each other but also share similar relationships with the ego.

Given an author x with all her publications, and the set of fields of research $F = \{r_1, r_2, ..., r_{24}\}$[7], we define the *versatility vector* $\hat{V}(x)$ of an author x as $\{r_{i,x}; r_i \in F\}$ such that $r_{i,x}$ is the fraction of publications of x in field r_i. Also, given a set of decades DEC = {1960-1970, 1971-1980, 1981-1990, 1991-2000, 2001-2009}, we define the *persistence vector* $\hat{D}(x)$ for x as $\{d_{j,x}; 1 \leq j \leq 5\}$, where $d_{j,x}$ denotes the number of papers published by x in decade $DEC(j)$. We also define the major field of work $R(x)$ for x, where she has maximum number of publications.

The *general* features capture independent characteristics of each author in the ego network and are listed below:

- The normalized number of citations the author has received (size 1)
- The normalized number of citations *per paper* that the author has received (size 1)
- The normalized h-index of the author (size 1)
- The normalized number of coauthors of the author (size 1)
- The *versatility vector* of the author (size 24)
- The normalized number of papers written by the author (size 1)
- The *persistence vector* of the author (size 5)
- The major field of the author (size 1)

On the other hand, the *ego-centric features* capture the relationship of an alter with its ego. Such features include:

- The fraction of papers coauthored by the alter with the ego in each of the five decades (size 5)

[7] Note that there are 24 research fields present in our dataset.

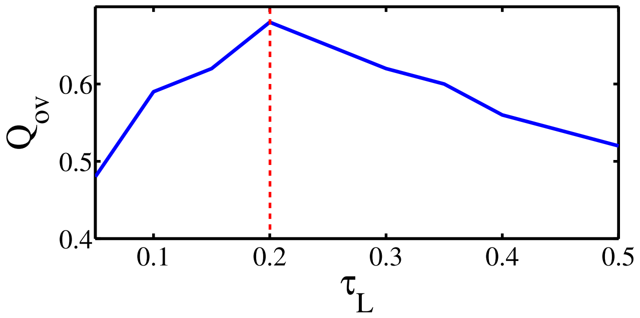
- The fraction of papers coauthored by the alter with the ego in each of the 24 fields (size 24)
- The normalized number of common coauthors that the alter has with the ego (size 1)
- The fraction of papers authored by the alter in the major field of the ego (size 1)
- The fraction of papers authored by the ego in the major field of the alter (size 1)

5.6.4 Evaluation of Detected Circles

We use the citation network discussed in Section 5.2. The co-authorship network constructed from this dataset has authors as nodes and edges between authors who have written at least one paper together. We consider the ego networks corresponding to each node (author) present in our dataset, thus obtaining 821,633 ego networks.

In this section, we intend to evaluate the quality of the circles detected by our proposed methodology. Evaluation is especially important to judge the quality of the detected circles. We compare the circles detected by our model with that obtained from four other recent overlapping community detection algorithms, namely BIGCLAM [232], SLPA [228], OSLOM [126] and COPRA [85]. We also detect the circles using the coordinate ascent method (CA) [145]. Since we intend to show that research field of the authors is not the proper information for creating the circles, we also compare our output with the circles obtained simply from research fields. For comparison, we use *overlapping modularity* Q_{ov} [89] which is probably the most widely used measure for evaluating the goodness of a community structure without a ground-truth.

First, to show the change in Q_{ov} with respect to the threshold τ_L as described in Section 5.6.2, we plot this quality function in Figure 5.17 by varying τ_L from 0.05-0.5. We observe that Q_{ov} reaches maximum at $\tau_L = 0.2$. Then for each competing algorithm, we measure the value of Q_{ov} for each ego and take an average over all the egos present in our dataset. The table adjacent to Figure 5.17 shows that our method outperforms the traditional topology based community finding algorithms in detecting meaningful circles. Our method achieves Q_{ov} of 0.68 which is 6.25% higher than coordinate ascent



Algorithms	Q_{ov}
BIGCLAM	0.60
SLPA	0.56
OSLOM	0.59
COPRA	0.58
Field based	0.45
Coordinate ascent	0.64
Our method	**0.68**

Figure 5.17: (Left) Change in overlapping modularity Q_{ov} with the increase in τ_L; (Right) comparison of the baseline algorithms with our method.

method, 13.33% higher than BIGCLAM, 15.25% higher than OSLOM, 17.24% higher than COPRA, and 21.42% higher than SLPA.

5.6.5 Task Based Evaluation

We further evaluate the quality of the circles through a task based evaluation framework – the task of collaboration prediction. We choose two supervised learning models: linear regression (LR) [11] and supervised random walks (SRW) [11]. Then we demonstrate that inclusion of the ego-centric circles detected by our model as a feature in the feature set would eventually enhance the performance of this model with respect to the one in which the circle information is missing.

Feature Set: We use a set of node- and edge-level features for the learning models. The following set of node-level features (denoted by N) are used. Each feature is normalized by the maximum value of the corresponding feature so that the values range between 0 to 1.

- Normalized number of citations received by an author
- Normalized h-index of an author
- Normalized number of coauthors of an author
- Fraction of papers by an author in each of the 24 fields
- Normalized number of papers written by an author
- Fraction of papers published by an author in each of the five decades (between 1960-2009)

Further, given an edge $e = (x, y)$ in the co-authorship network, we additionally use the following edge-level features (denoted by E). Each feature is appropriately normalized to a value between 0 and 1.

- Fraction of papers coauthored by x and y in each of the five decades
- Normalized number of common coauthors of x and y
- Fraction of papers authored by x in the major field of y
- Fraction of papers authored by y in the major field of x

We refer to the combined set of both node- and edge-level features by NE. We provide this set NE of node and edge attributes as an input to the learning model which then takes care of determining how to combine them with the network structure to make predictions [11]. Note that if we take the dataset till t for training the model, all the features mentioned above will be calculated based on the statistics of each vertex till t in order to avoid information leakage.

Table 5.7: Comparison of BIGCLAM (BIG), coordinate ascent method (CA) [145] and our model (CIRC) after including their detected circle information into the feature set of Linear Regression (LR) and Supervised Random Walks (SRW) frameworks across three time periods and different feature sets (N: node-level, E: edge-level, NE: node- and edge-level, NEB: adding the binary circle information to NE, NEBC: adding the numerical circle information to NEB).

Time period	Area Under the ROC Curve (AUC)																	
	Linear Regression (LR)									Supervised Random Walks (SRW)								
	N	E	NE	NEB			NEBC			N	E	NE	NEB			NEBC		
				BIG	CA	CIRC	BIG	CA	CIRC				BIG	CA	CIRC	BIG	CA	CIRC
1996-1999	0.5872	0.5914	0.6451	0.6569	0.6689	0.6791	0.6989	0.7195	**0.7235**	0.6332	0.6478	0.7659	0.7908	0.7895	0.8275	0.7971	0.8296	**0.8303**
2001-2004	0.5890	0.5907	0.6528	0.6529	0.6437	0.6659	0.6845	0.7011	**0.7012**	0.6419	0.6514	0.7591	0.8067	0.8035	0.8249	0.8098	0.8149	**0.8356**
2006-2009	0.5916	0.5891	0.6436	0.6439	0.6510	0.6509	0.6905	0.7001	**0.7198**	0.6360	0.6608	0.7609	0.8001	0.8101	0.8295	0.8111	0.8279	**0.8321**
Average	0.5893	0.5904	0.6472	0.6512	0.6545	0.6653	0.6913	0.7069	**0.7148**	0.6370	0.6533	0.7620	0.7992	0.8101	0.8273	0.8060	0.8279	**0.8327**

Time period	$Prec$@20																	
	Linear Regression (LR)									Supervised Random Walks (SRW)								
	N	E	NE	NEB			NEBC			N	E	NE	NEB			NEBC		
				BIG	CA	CIRC	BIG	CA	CIRC				BIG	CA	CIRC	BIG	CA	CIRC
1996-1999	0.137	0.124	0.152	0.155	0.161	0.158	0.164	0.173	**0.177**	0.165	0.172	0.201	0.205	0.209	0.210	0.207	0.215	**0.223**
2001-2004	0.141	0.143	0.156	0.162	0.159	0.169	0.175	0.175	**0.185**	0.158	0.163	0.198	0.200	0.210	0.209	0.215	0.220	**0.225**
2006-2009	0.147	0.142	0.161	0.162	0.165	0.171	0.179	0.178	**0.189**	0.161	0.169	0.199	0.208	0.209	0.212	0.211	0.217	**0.224**
Average	0.142	0.136	0.156	0.160	0.162	0.166	0.173	0.175	**0.184**	0.161	0.168	0.199	0.204	0.209	0.210	0.211	0.217	**0.224**

Evaluation Methodology: In order to demonstrate that predictions are robust irrespective of the time stamp considered for dividing the dataset into training and test sets, we run the competing models in three different time periods: (i) the dataset till 1995 is considered for training and the accuracies of the models are measured by comparing the new edges formed between 1996-1999, (ii) similarly, the dataset till 2000 for training and 2001-2004 for checking the accuracy, and (iii) the dataset till 2005 for training and 2006-2009 for checking the accuracy.

In each time stamp, we evaluate the methods on the test set, considering two performance metrics: the Area under the ROC curve (AUC) and the Precision at Top 20 ($Prec@20$), i.e., for each node s, what fraction of top 20 nodes suggested by each model actually receive links from s later. This measure is particularly appropriate in the context of link-recommendation where we present a user with a set of suggested coauthors and aim that most of them are correct.

Performance Evaluation: We compare the predictive performance of two learning models including the circle information in three different time periods. We iterate each of these collaboration prediction models using different sets of features: (i) only node-level features (*Model: N*), (ii) only edge-level features (*Model: E*), (iii) both node and edge level features (*Model: NE*), (iv) besides node and edge level features, including a binary feature B that checks whether a pair of nodes (x, y) belong to at least one common ego-centric circle or not (*Model: NEB*), and (v) besides node-level and edge-level features and the binary circle information, including a numeric feature C indicating the number of common circles a pair of nodes (x, y) is a part of (*Model: NEBC*). The circles are detected by our model, the coordinate ascent method (CA) [145] and BIGCLAM separately.

Table 5.7 shows the performance of these two prediction models with different feature sets. We notice that edge features are more effective than node features, and the performance improves incrementally after combining different features together. A general observation is that inclusion of circle information in the feature set improves the performance of both the prediction models irrespective of the time periods. For instance, it improves the performance by 9.87% and 15.25% on average in terms of AUC and $Prec@20$ respectively compared to the case, where the circle information is not present (NE).

We further observe that the inclusion of circle information detected by our model significantly outperforms the case where the circles are obtained by BIGCLAM and CA in each time stamp. Including the binary circle information (NEB) from our model achieves an average AUC improvement of 2.16% and 3.51% respectively for LR and SRW models (similarly, in terms of $Prec@20$, the improvement is 3.75% and 2.94% respectively for LR and SRW models) compared to BIGCLAM (CA).

Further, including the count of common circles for a node pair ($NEBC$) in the feature set leads both LR and SRW to achieve even better performance. We observe an average AUC improvement of 3.41% (1.11%) and 3.31% (0.57%) respectively for LR and SRW models using our circle information as compared to that obtained from BIGCLAM (CA) (similarly, in terms of $Prec@20$, the improvement is 6.35% (5.14%) and 6.16% (3.22%) respectively for LR and SRW models).

5.7 Summary of this Chapter

The lack of reliable ground-truth communities has made network community detection a very challenging task. In this chapter, we developed ground-truth overlapping communities of a directed paper-paper citation network that emerge from the natural grouping of research papers in various fields of the computer science domain. We conduced a set of experiments to understand this network and the community structure from diverse perspectives. We conclude by summarizing our main observations in this chapter as follows:

- Quite remarkably, for the last fifty years one observes a very robust behavior of the dynamics – the field that is the strongest contender of the field currently at the forefront almost surely emerges as the top ranked field after the transition.
- The key factors that keep a field at the forefront include the citations from the backup field, the inception of the seminal papers and the existence of high-impact papers.
- Funding statistics obtained from NSF is in very good agreement with the results predicted by our method.
- Four indicative features quite efficiently unfold the extent of interdisciplinarity of a field that further help in building the classification model.
- For already very interdisciplinary fields, such as Data Mining, the indicators may have a certain "saturation" effect forcing it towards the core region of the computer science domain.

- The average behavior of a researcher's career indicates that a researcher tends to adopt few research fields in her entire research career, though she seems to prefer to work simultaneously on various fields.

- Researchers who have worked in many fields in their entire careers but remained confined in few fields in each time window get high importance in terms of citations compared to the others.

- Finally, we proposed a simple yet effective method of detecting ego-centric circles in co-authorship networks.

Chapter 6

Community-based Applications

In this chapter we address our fourth objective of leveraging the community information of networks in order to design different applications.

6.1 Introduction

Community detection or clustering constitutes a fundamental framework in the development of various applied systems, e.g., search system, recommendation system etc. In this chapter we show how the citation network and various types of community markings of this network can be leveraged to develop two applications.

The contributions of this chapter are threefold:

- We analyze the scientific dataset mentioned in Section 5.2 to understand the citation growth of each paper after its publication. We discover that the pattern of citation growth can be clustered into at least six different categories which are in contrast with the earlier observation that the citation profile of published articles in general follows a universal pattern. We further conduct a deeper investigation of the papers in the different categories that leads us to deduce a series of conclusions about the characteristic properties of these categories.

- In particular, we adopt a *stratified learning approach* for the prediction task, whereby, we propose a two-stage prediction model for the task of future citation count prediction where we make use of the six categories of citation pattern that we observe.

- Finally, we propose for the first time a framework for *faceted recommendation* of scientific articles, *FeRoSA* which apart from ensuring quality retrieval of scientific articles for a particular query paper, also efficiently clusters the recommended papers into different semantic categories (facets).

6.2 Categorization of Scientific Citation Profiles

There has been a plethora of research done on the citation network and on its evolution as a collective system. There is already a well-accepted belief on the dynamics of citations that a scientific article receives after publication – an initial growth (growing phase) in the number of citations within the first two/three years after publication followed by a steady peak of one to two years (saturation phase) and then a final decline over the rest of the lifetime of the article (decline and obsolete phase) as shown in Figure 6.1 [58, 59]. In most cases, the above observation has been drawn from the analysis of a very limited set of publication data, thus, obfuscating the true characteristics. For instance, Eom and Fortunato [58] used 14,977 papers published in journals of the American Physical Society (APS) from 1893 to 2008. Here, we conduct our experiment on a massive bibliographic dataset of the computer science domain comprising more than 1.5 million papers published between 1970 and 2010 (see Section 5.2). Strikingly, unlike earlier observations about citation profile of a paper, we notice six different patterns of citation profiles prevalent in the dataset (namely, PeakInit, PeakMul, PeakLate, MonDec, MonIncr and Oth). We exhaustively analyze these profiles to exploit the micro-dynamics controlling the actual growth of the underlying citation network that has remained unexplored in the existing literature. We believe that this observation will not only help in reformulating the existing bibliographic indices such as Journal Impact Factor (JIF), but will also enhance the general bibliometric research such as citation link prediction, information retrieval and self-citation characterization.

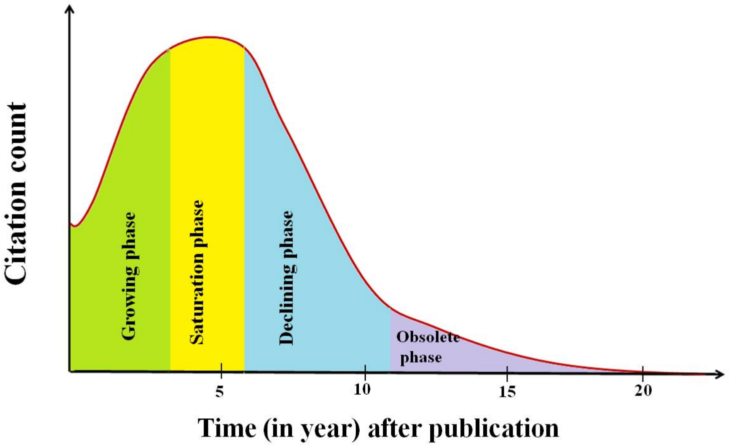


Figure 6.1: A hypothetical example showing the traditional belief in the pattern of citation profile of a scientific paper after publication.

6.2.1 Six Categories of Citation Profiles

Since the primary focus of our study is to analyze citation growth of a paper after publication, an in-depth understanding of how the number of citations after publication of a paper varies over the years is necessary. We therefore conduct an exhaustive analysis of the citation patterns of different papers present in our dataset. Some of the previous experimental results [75] show that the trend of citations received by a paper after its publication date is not linear in general; rather there is a fast growth of citations within the initial few years, followed by an exponential decay. Here we take all the papers having at least 10 years of citation history, and consider maximum 20 years of their citation history. Then we design our own heuristics based on the number and the position of the peaks to categorize the citation profile of each paper.

Algorithm to categorize citation profiles: In order to decipher the trends of citation, we perform various processing on the data set. First of all, to smoothen the time series data points in the citation profile of a paper, we use five-years moving average filtering; then, we scale the data points by normalizing them with the maximum value present in the time series (i.e, maximum citations received by the paper in a particular year); finally, we run local peak detection algorithm[1] to detect peaks in the citation profile. Over and above,

[1] The peak detection algorithm is available in Matlab Spectral Analysis package - http://www.mathworks.in/help/signal/ref/findpeaks.html; we use 'MINPEAKDISTANCE'=2 and 'MINPEAKHEIGHT'=0.75 and the default values for the other parameters.

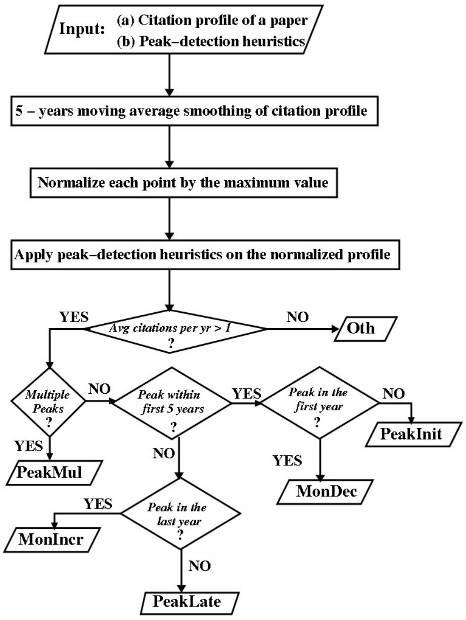

Input:
(a) Citation profile of a paper
(b) Peak–detection heuristics
5 – years moving average smoothing of citation profile
Normalize each point by the maximum value
Apply peak–detection heuristics on the normalized profile
Avg citations per yr > 1 ?
YES
NO
Oth
Multiple Peaks ?
NO
YES
PeakMul
Peak within first 5 years ?
YES
NO
Peak in the first year ?
NO
PeakInit
YES
MonDec
Peak in the last year ?
YES
MonIncr
NO
PeakLate

Figure 6.2: A systematic flowchart demonstrating the rules for classifying the citation profiles.

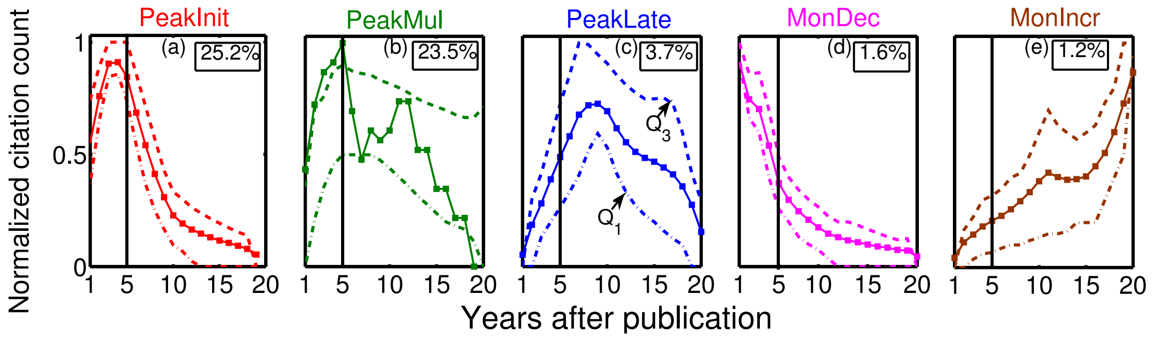


Figure 6.3: Citation itineraries for the first five categories. In each frame, the belt bounded by the lines Q_1 and Q_3 represent the first and the third quartiles of the data points respectively. For each category, one representative citation itinerary is shown at the middle of the belt. The percentage mentioned in each frame indicates the proportion of papers in each category. The major proportion of papers (44.8%) lies in category ‘Oth’ which does not have any specific pattern and is not shown in this diagram.

we apply the following two heuristics to specify peaks: (i) the height of a peak should be at least 75% of the maximum peak-height, and (ii) two consecutive peaks should be separated by more than 2 years, otherwise they are treated as a single peak. A systematic flowchart to detect each category is shown in Figure 6.2.

Remarkably, we notice that a major proportion of papers do not follow the traditional citation profile mentioned in the earlier studies, rather there exist six different types of citation profiles of research papers based on the count and the position of peaks present in the profile as shown in Figure 6.1 (in each frame, a citation belt is formed by the lines Q_1 and Q_3 which represent the first (10% points lie below this line) and third (10% points lie above this line) quartiles of the data points respectively (i.e., effectively 80% points are within the citation belt), and the solid line drawn within the citation belt represents the average behavior of all the profiles corresponding to that category.) The definitions of six types of citation profiles with the individual proportions in the entire dataset are give below.

(i) PeakInit: Papers whose citation count peaks within 5 years of publication (but not in the first year) followed by an exponential decay (proportion: 25.2%) (Figure 6.3(a)).

(ii) PeakMul: Papers having multiple peaks at different time points of the citation itinerary (proportion: 23.5%) (Figure 6.3(b)).

(iii) PeakLate: Papers having very few citations at the beginning and then a single peak after at least 5 years of the publication which is followed by an exponential decay in citation count (proportion: 3.7%) (Figure 6.3(c)).

(iv) MonDec: Papers whose citation count peaks in the immediate next year of the publication followed by a monotonic decrease in the number of citations (proportion: 1.6%) (Figure 6.3(d)).

(v) MonIncr: Papers having a monotonic increase in the number of citations from the year of publication till the date of observation (i.e., it can be after 20 years of its publication) (proportion: 1.2%) (Figure 6.3(e)).

(vi) Oth: Apart from the above types, there exist a large number of papers which on an average usually receive less than one citation each year. For these papers, the evidences are not significant enough for assigning them into one of the above categories, and, therefore, they remain as a separate group altogether (proportion: 44.8%).

6.2.2 Contribution of Categories in Different Citation Ranges

One of the fundamental aspects of analyzing scientific publications is to measure how acceptable they are to the research community. This is often measured by the raw citation count – the more an article receives citations from other publications, the more it is

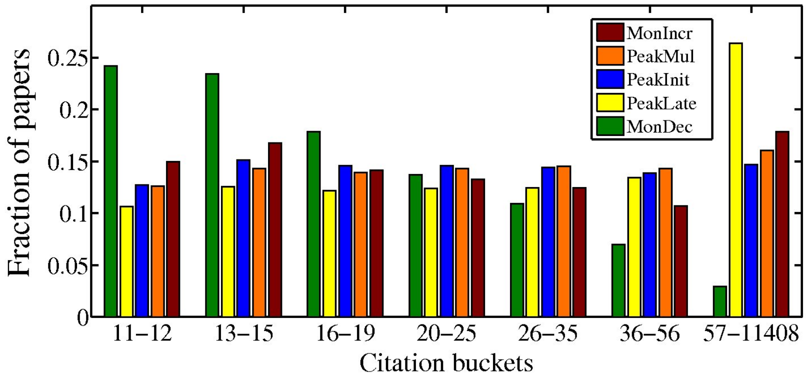


Figure 6.4: Contribution of papers of each category in different citation buckets.

assumed to be admired by the researchers and hence the more is the scientific impact [30]. In this current context, an interesting question is – which among the six categories contain papers that are admired most in terms of citations. In order to answer this question, we conduct a systematic study – the total citation range is divided into four buckets (the citation ranges are: 11-12, 13-15, 16-19, 20-11408) such that each citation bucket would contain almost equal number of papers. For a deeper analysis of the highest citation range, we further divide the last bucket (20-11408) into four more ranges, thus obtaining seven buckets altogether. Then we measure the proportion of papers contributed by a particular category to a citation bucket (see Figure 6.4). Note that in each citation bucket, the number of papers contributed by a category is normalized by the total number of papers belonging to that category. Therefore, this figure is a histogram of conditional probability distribution – probability that a randomly selected paper falls in citation bucket i given that it belongs to category j. The normalization is required in order to avoid population bias across different categories. We observe that the higher region of citation is mostly occupied by the papers in PeakLate and MonIncr categories followed by PeakMul and PeakInit. We also notice that the MonDec category which has the minimum proportion in the last citation bucket shows a monotonic downward fall in the fraction of papers as the citation range increases. These initial evidences present a general and non-intuitive interpretation of citation profiles that if a paper does not obtain high citations within the immediate few years after its publication, it does not necessarily mean that it will continue to remain low impact all through its lifetime; rather in future its citation growth rate might accelerate and it could indeed turn out to be a well accepted paper in the scientific community. We further explain this behavior in the subsequent parts of this section.

Table 6.1: Mean publication year (Y) (its standard deviation, $\sigma(Y)$) and the proportion of papers (in %) in conferences and journals for each category of citation profile.

Category	Mean publication year ($\sigma(Y)$)	% of conference papers	% of journal papers
PeakInit	1994 (5.19)	64.35	35.65
PeakMul	1991 (6.68)	39.03	60.97
PeakLate	1992 (6.54)	39.89	60.11
MonDec	1994 (5.44)	60.73	39.27
MonIncr	1993 (7.36)	25.26	74.74

6.2.3 Characterizing Different Citation Profiles

The rich metadata information of the publication dataset further allows us to understand the characteristic features of each of these six categories at finer levels of detail.

Influences of publication year and publication venues on the categorization: One might raise an immediate question that this categorization might be influenced by the time (year) when the papers are published, i.e., the papers published earlier might be following the well-known behavior whereas the papers published recently might indicate a different behavior. In order to verify that the categorization is not biased by the publication time period, we measure the average year of publication of the papers in each category. From the second column of Table 6.1, we can conclude that the citation pattern of the papers is not biased by the publication year since the average years more or less point to the same time period.

On the other hand, the mode of publication in conferences is significantly different from that of journals, and therefore the citation profiles of papers published in these two venues are also expected to be different. To analyze the venue effect on the categorization, we measure the fraction of papers published in journals vis-a-vis in conferences for each category as shown in the third and the fourth columns of Table 6.1 respectively. We observe that while most of the papers in PeakInit (64.35%) and MonDec (60.73%) categories are published in conferences, papers belonging to PeakLate (60.11%) and MonIncr (74.74%) categories are mostly published in journals. Hence, if a publication starts receiving greater attention or citations at a later part of its lifetime, it is more likely to be published in a journal and vice versa. These results put forth two immediate conclusions. First, due to the increasing popularity of conferences in an applied domain like computer science, the conference papers get quick publicity within a few years after publication, which is also the reason for the rapid decay of their popularity. In contrast, journal papers usually take time to get published and hence to get popularity, thus being mostly admired much later after publication. However, most of the journal papers remain consistent in receiving citations even after long years of their publications. Another interesting point to be noted from these results is that although the existing formulation of the journal impact factor [75] has been defined taking into consideration the citation profile as shown in Figure 6.1, most of the journal papers which fall in PeakLate or MonIncr do not follow such a profile at all; at least for papers in PeakLate category, the metric does not focus on the most-relevant time frame of the citation profile (mostly after first 5 years of publication). In the light of the current results, the appropriateness of the formulation of the bioliogaphic metrics such as journal impact factor remain doubtful.

Effect of self-citation on the categorization: Another factor that often affects citation rate is self-citation. We also conduct a similar experiment to notice the effect of self-citation on the categorization of citation profiles. Essentially, we first dispose the citation from the dataset if the citing and the cited papers have at least one author in common, and then measure what fraction of papers in each category migrate to the other category due to this disposal. We observe in Table 6.2 that papers in MonDec are vastly affected by the self-citation phenomenon. We find that around 35% of papers in MonDec would have been in the ‘Oth’ category had it not been due to the self-citations. We also observe that the self citation is usually used in initial periods of the publication by the authors in attempt to increase the visibility of their publications in the scientific community. This effect is more prominent for MonDec category which is followed by Oth and PeakInit.

6.2.4 Analyzing Stabilities of Different Categories

The number of citations for a paper changes over time depending on its long/short lasting effect on the scientific community which in turn might change the shape of the citation profile. Therefore, studying the temporal evolution of each citation profile can help us un-

Table 6.2: Confusion matrix representing the transition of categories due to the removal of self-citations. A value x in the cell (i, j) represents that x fraction of papers in category i would have fallen in category j if self-citations were absent in the entire dataset. Note that, no row has been specified for Oth category because papers from this category can never be moved to the other categories by any deletion of citations.

Category	PeakInit	PeakMul	PeakLate	MonDec	MonIncr	Other
PeakInit	0.72	0.10	0.03	0.01	0	0.15
PeakMul	0.02	0.81	0.04	0	0.1	0.11
PeakLate	0.01	0.06	0.86	0	0.01	0.06
MonDec	0.05	0.14	0	0.41	0	0.35
MonIncr	0	0.02	0.01	0.01	0.88	0.09

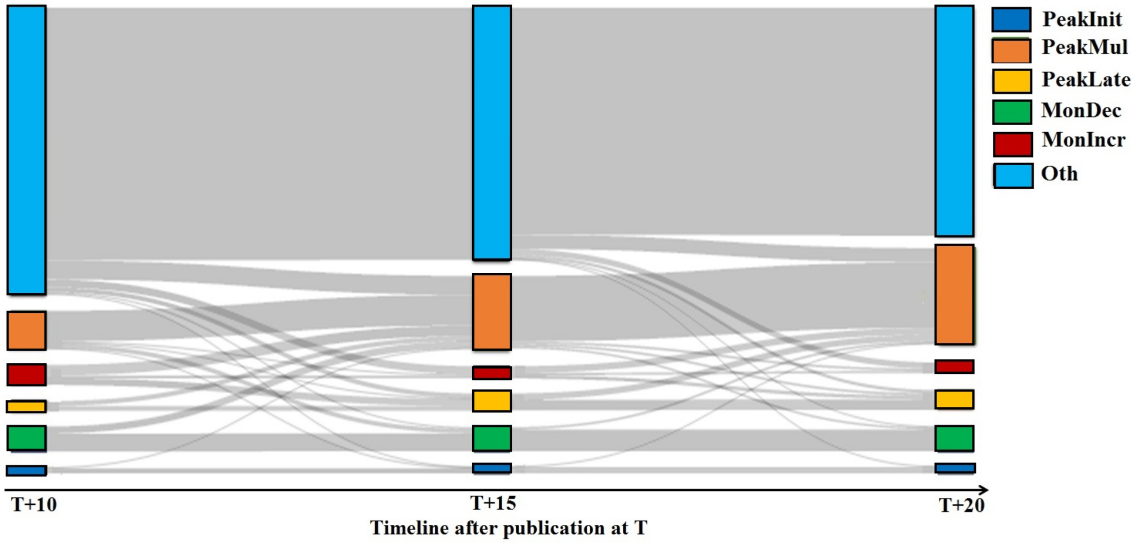


Figure 6.5: Alluvial diagram representing the evolution of papers in different categories and the flows between the categories in time $T + 10$, $T + 15$ and $T + 20$. The colored blocks correspond to different categories. The size of the block indicates the number of papers in that category, and the shaded waves joining the regions represent flow of papers between the regions, such that the width of the flow corresponds to the fraction of papers. The total width of incoming flows is equal to the width of the corresponding region.

derstand the stability of the categories individually. Since, we know the category of those papers that have at least 20 years of citation history, for each such paper we further analyze how the shape of the profile evolves through this 20 years timeline. Essentially, after publication of a paper at time T, we identify its category at time $T+10$, $T+15$ and $T+20$ based on the heuristics discussed earlier. We hypothesize that a stable citation category tends to maintain its shape throughout the entire timeline. The colored blocks of the alluvial diagram in Figure 6.5 correspond to the different categories for three different timestamps. We observe that apart from the Oth category which has a major proportion of papers, MonDec seems to be the most stable, which is followed by PeakInit. However, papers which are assumed to fall in Oth category quite often turn out to be MonIncr papers in the later time periods. This analysis indeed demonstrates a systematic approach to unfold the transition from one category to another taking place in scientific research with the increase of citations.

6.2.5 Core-periphery Analysis

Although Figure 6.4 provides the impact of different categories in terms of raw citation count, it neither indicates the significance of the papers in each category forming the core of the network nor gives us any information regarding the temporal evolution of the structure. For a better and more detailed understanding, we perform k-core analysis [92] of the evolving citation network by decomposing the network for each year into its k^s-shells, such that an inner shell index of a paper reflects a central position in the core of the network. The idea is to show how the papers in each category (identified at the year 2000) migrates from one shell to another after getting citations in the next 10 years. It also allows us to observe how persistent a category is in a particular shell. In Figure 6.6, we notice that the majority of papers in the Oth category lie in the periphery and its proportion in the periphery increases over time which indicates that the papers in this category are becoming increasingly less popular in time. PeakMul category gradually leaves the peripheral region over time and mostly occupies the two innermost shells. PeakInit and MonDec show almost similar behavior with a major proportion of papers in inner cores in the initial year but gradually shifting towards peripheral regions. On the other hand, MonIncr and PeakLate show expected behavior with their proportion increasing towards inner shells over time indicating their rising relevance as time progresses. This study helps us identify temporal evolution of the importance of different categories in terms of how each of them contributes to the central position of the citation network.

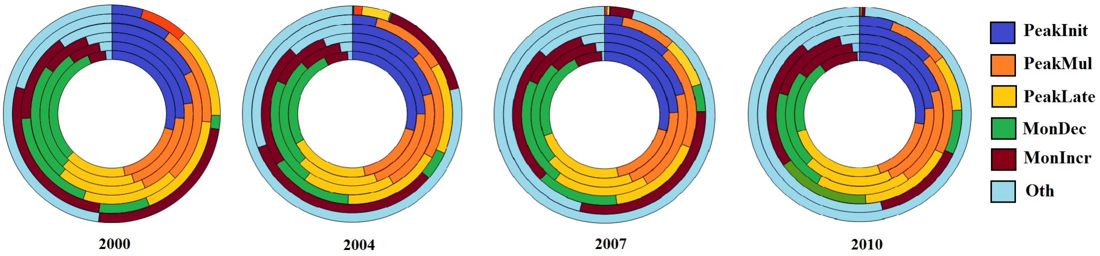


Figure 6.6: Multi-level pie chart for year 2000,2004, 2007 and 2010 showing the composition of each of the categories in different k^s-shell regions; where the colors represent different categories and the area covered by each colored region in each k^s-shell denotes the proportion of papers in the corresponding category occupied in that shell. The innermost shell is the core region and the outermost shell is the periphery region.

6.3 Predicting Future Citations of Scientific Articles

The next objective is to show that the categorization of citation profiles has significant consequences to early prediction of citation itinerary of scientific papers. Such a prediction scheme can be of significant interest not only for the scholars at universities and research institutes but also for the engineers and policy makers in business and government domains. The very limited number of studies on this topic [230] have mostly modeled the problem as a learning task – given a set of features and a particular time interval, a regression model is trained on the entire set of the training population, and accordingly, the future citation count of a query paper is estimated. A common underlying implicit assumption in these approaches is that the citation itinerary of all published papers have similar characteristics. However, we observe that such an assumption is flawed and therefore seriously affects the accuracy of the prediction. Consequently, we propose to categorize the complete set of data samples into different subparts each of which corresponds to one of the six citation itineraries observed. This approach is commonly termed as *stratified learning* [91] in the literature where the members of the stratified space are divided into homogeneous subgroups (aka strata) before sampling. This indeed reduces the extent of variability and increases the representativeness of the data samples in each individual strata thus enhancing the learning scheme.

6.3.1 Distinctive Features

Here we provide a brief description of the set of features learned by the classifiers. The features can be broadly classified into three classes, namely the author-centric features, the venue-centric features and the paper-centric features. Note that for a particular paper, all the features are calculated with respect to the year of its publication. For features which are still unobserved, e.g., new authors or new venues, we do not assign zero values; instead we set them to the minimum value observed across all the samples available at that particular time point.

(i) Author-centric features: For all the author-centric features mentioned here, we measure both the average (**Avg**) and the maximum (**Max**) values for each paper to incorporate the notion of both team-effect and individual leadership respectively in the final citation count prediction.

(a) Author productivity: Yan et al. [230] noticed that the more papers an author publishes (productivity of the author), the higher average citation counts she can expect. Therefore, for each paper, we calculate the productivity of its authors (**ProAuth**) that indeed indicates how the influence of productive authors regulates the citation profile of a paper.

(b) Author h-index: H-index is a standard metric to measure both the productivity and the impact of the published work of an author [99]. Therefore for each paper, we measure the h-index (**Hindex**) of authors.

(c) Author diversity: The diversity of an author a denoted by $AuthDiv(a)$, indicating the breadth of expertise of a. It is measured by the entropy of the research fields where she publishes and is given by

$$AuthDiv(a) = -\sum_{i=1}^{24} p(n_i|n) \times log(p(n_i|n)) \tag{6.1}$$

where n_i denotes the number of papers written by author a belonging to the field i (total 24 fields are available in the dataset), and n denotes the total number of papers written by a. For each paper, we include the diversity of authors (**AuthDiv**) as a feature. Note that these two features have not been considered in earlier works [230].

(d) Sociality of author: Since the authors tend to cite papers of their previous collaborators, it is natural to assume that the paper from a widely connected authors has a larger probability to be cited by different coauthors. A simple measurement is to count the number of coauthors (**NOCA**) of each author present in a paper [230].

(ii) Venue-centric features: We consider the three features listed below to signify the importance of venue.

(a) Long term venue prestige: To measure the prestige of a publication venue (**VenPresL**), we calculate the average citations received by the papers published so far in that venue.

(b) Short term venue prestige: It is measured as the average number of citations received per paper published in that venue during *at most* two preceding years (**VenPresS**). The basic difference between **VenPresL** and **VenPresS** is that while the former measures the overall impact of a venue by considering all the papers published so far in that venue, the latter only measures the recent impact of the venue. **VenPresS** is conceptually similar to the impact factor of a journal as defined in [74].

(c) Venue diversity: **VenDiv** can be measured by considering the different fields covered by the papers published in that venue. A formula similar to Equation 6.1 gives a quantitative measure of **VenDiv**. This is another new feature introduced in this study for the first time.

(iii) Paper-centric features: Among the paper-centric features mentioned below, third and fourth features are newly introduced in this study.

(a) Team-size: It has been observed that there exists a critical value of team-size corresponding to which the citation accumulation is maximum. Hence, we directly take into account the number of authors of a paper (**Team**).

(b) Reference count: Sometimes, only the number of references serves as a feature to distinguish regular and survey papers. Therefore, we directly use the reference count of a paper (**RefCount**) as a feature in our study.

(c) Reference diversity: Earlier in Section 5.4.1 we proposed a measure called Reference

Diversity Index (**RDI**) as an indicator of interdisciplinarity that attempts to quantify the diversity in terms of the number of fields being cited by a paper. It is also measured similarly using Equation 6.1; here n (n_i) indicates the total number of references (number of references to the papers belonging to field i).

(d) Keyword diversity: As mentioned in Section 5.2, MAS assigns keywords, from a global set of keywords, against each paper in order to characterize it properly. For each paper, we measure how diverse its keywords are (**KDI**) similarly by Equation 6.1; here n_i indicates the fraction of keywords of paper x belonging to the field i. Note that a keyword may appear in multiple fields. For them, we consider multiple instances one for each field.

(e) Topic diversity: We use the unsupervised Latent Dirichlet Allocation[2] [25] as mentioned by Yan et al. [230] to discover topics for each paper. We empirically set the number of topics as 100, i.e., for each of our 100 topics, the topic model calculates $p(topic_i|d)$, the inferred probability of topic i in document d (**Topic**). The topic distribution $\tau(d)$ over all topics in the document d is then: $\tau(d) = \{p(topic_1|d), p(topic_2|d), ..., p(topic_{100}|d)\}$.

6.3.2 Proposed Framework

The prediction is done through a two-stage learning process where the learning task is defined as follows:

Learning task: Given a set of features $F = \{f_1, f_2, ..., f_n\}$, our goal is to learn a function ψ to predict the citation count of an article d at a given time period Δt after its publication. Formally, this can be written as:

$$\psi(d|F, \Delta t) \rightarrow C_T(d|\Delta t) \tag{6.2}$$

where *citation count*, C_T is as defined below.

Citation count: As defined by Yan et al. [230], given the set of scientific articles D, the citation count ($C_T(.)$) of an article $d \in D$ is defined as:

[2] We use GibbsLDA++ (http://gibbslda.sourceforge.net/) with the default settings.

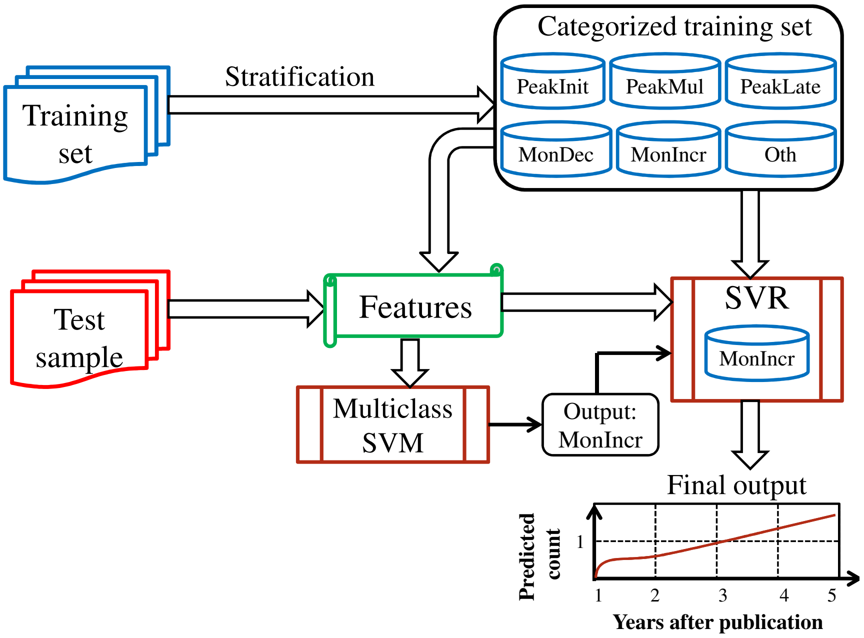


Figure 6.7: A schematic of our proposed framework (SVM: Support Vector Machine, SVR: Support Vector Regression). Here we assume that the query paper is mapped to 'MonIncr' class by the SVM module.

$$citing(d) = \{d' \in D : d' \; cites \; d\}$$
$$C_T(d) = |citing(d)|$$

Note that in this paper, we consider $\Delta t \in [1, 5]$.

We now elaborate the two-stage learning process undertaken to accomplish the above mentioned task.

Two-stage prediction model

The schematic diagram of our proposed two-stage model for predicting future citation count is shown in Figure 6.7. In the first stage, a sample (paper) is classified into one of the six identified categories which is done by using a multi-class SVM. In the second stage, the actual citation count of the paper is computed by employing a customized SVR model. In the rest of the section, we explain each of the stages separately.

Stage I: For each training sample, we identify its category among the six defined categories using the set of rules mentioned in Section 6.2.1. We also extract the features (mentioned in Section 6.3.1) for each training sample. Hence the multi-class Support Vector Machine (SVM) [130] receives the category and the feature (author-centric, paper-centric, venue-centric) information of each member in the training set. Subsequently, given a test sample (query paper) along with its set of features, the multi-class SVM outputs the category of the sample. For training and classification phases of SVM, we use Weka-LibSVM toolkit[3] applying pairwise multi-class decision approach. The best results are obtained for the polynomial kernel setting. The overall accuracy and the importance of each feature in the classification task are reported in Section 6.3.3.

Stage II: Support Vector Machine can be applied not only to classification problems but also to the case of regression often termed as *Support Vector Regression* (*SVR*) [200]. We use LibSVM (epsilon-SVR)[4] for this analysis with the default parameter settings. We train separate SVR models for each category C as well as for each time instance Δt; each SVR is identified by the notation $SVR(C, \Delta t)$. Recently, Yan et al. [230] used four prediction models, namely Linear Regression, k-Nearest Neighbor, CART and SVR and showed that SVR outperforms other models in predicting future citation counts. Therefore, in our experiment we use SVR for the final prediction.

The training set for SVR pertaining to a certain category (say, MonIncr) contains the papers whose citation patterns fall into that category. Besides taking the features of the papers as input, $SVR(C, \Delta t)$ also takes as input the number of citations the constituent training papers in that category have received at Δt time after their publication. That is, if Δt = 5, the citation count of a paper at the fifth year of its publication available from the training sample is taken into consideration. For example, if a paper has been published in 1975 (1978), the number of citations it received in the year 1980 (1983) is taken as input.

Handling information leakage: In order to make predictions for the query paper, we

[3] http://www.cs.waikato.ac.nz/ml/weka/

[4] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

Table 6.3: Performance of the baseline model (columns 2-4) and our proposed system at various time intervals for the test papers published between 1996-2000 (columns 5-7) and test papers published between 2001-2005 (columns 8-10).

	Baseline			Our model					
	1996-2000			1996-2000			2001-2005		
	R^2	θ	ρ	R^2	θ	ρ	R^2	θ	ρ
Δt=1	**0.55**	**5.45**	**0.59**	0.87	2.66	0.86	0.89	1.95	0.88
Δt=2	0.54	6.36	0.57	**0.90**	**1.46**	**0.88**	**0.91**	**1.20**	**0.90**
Δt=3	0.53	7.67	0.56	0.83	3.11	0.85	0.82	3.22	0.80
Δt=4	0.50	9.16	0.52	0.77	3.86	0.84	0.77	3.76	0.79
Δt=5	0.48	12.09	0.49	0.74	4.18	0.75	0.71	4.08	0.73

always consider the information available before the publication of the query paper (i.e., we avoid any information *at or after* the publication year of the query paper). For instance, when predicting the future citation count of an article (published in 1996) 5 years after its year of publication, all the articles published in the year 1990 or before are processed in the training samples; all the other articles published after 1990 are discarded. The reason is that for 5-years future citation prediction of the papers published in 1996, if we use the papers published in 1992 in the training phase, their citation counts in the year 1997 would become the data points in the training space of the regression model for $\Delta t = 5$. This implies that we are using the information of the citations at 1997 in order to predict the citation count of the paper published in the year 1996, which leads to information leakage.

6.3.3 Performance Evaluation

In this section, we analyze the performance of the baseline system and our proposed model in predicting future citation count of a given paper at the time of publication. For the baseline system, we design a model which is similar to that proposed by Yan et al. [230] (except that we are using a lot more features). It is identical to our proposed model except that it does not include the first stage of our model. Thus, for a query paper, it takes into account all the training samples and the set of features discussed in Section 6.3.1, and applies SVR to predict the citation count of the paper. Essentially, we intend to show the significance of the preprocessing stage (first stage of our model) in the task of future citation prediction.

Evaluation Metrics: For the evaluation purpose, we use the following metrics: coefficient of determination (R^2)[5] [230], mean squared error (θ) and Pearson correlation coefficient (ρ). Note that the more the value of R^2 and ρ, the better the accuracy of the model; but for θ, the reverse argument is true.

Dataset: The filtered dataset contains 1,549,317 scientific articles which need to be divided further for training and testing. However, for the evaluation of SVM, we need those papers whose true categorizations are known to us, i.e., those papers which have at least 10 years history (published between 1970-2000); though for measuring SVR accuracy, this might not be the criteria. Therefore for the sake of uniformity, we consider the papers published between 1970-1995 for training (505,149 papers), and the papers published between 1996-2000 (146,620 papers) for testing (for baseline as well as our algorithm) throughout the evaluation (unless explicitly mentioned). However, we also report the final prediction accuracy for the papers published between 2001-2005.

Performance of the baseline model: The predictive performances of the baseline system for each of the consecutive five years after publication are shown in Table 6.3 (columns 2-4). We observe that the baseline system achieves the highest accuracy (R^2=0.55, θ=5.45 and ρ=0.59) at the immediate next year after publication of a paper. We also observe that the accuracy of the predicted citation count is moderately overestimated for longer number of years which in turn decreases the accuracy of the baseline system in the later time periods.

Performance of our model: Table 6.3 shows the final performance of our model in each time interval after the time of publication. In this table, apart from the citation prediction for the papers published between 1996-2000, we also show the accuracy for the papers published between 2001-2005 (in that case, the training set consists of papers published between 1970-2000, and papers published between 2001-2005 constitute the test samples). Contrary to the performance of the baseline model where the highest accuracy is achieved at the immediate next year after publication, we achieve the best performance of our model 2 years after the year of publication. Remarkably, we observe that for all the cases, our

[5] R^2 is defined as: $R^2 = 1 - \frac{\sum_{d \in D}(C(d) - C^{'}(d))^2}{\sum_{d \in D}(C(d) - C(D))^2}$, where D is the set of test documents, $C(d)$ is the actual citation count for article d, $C^{'}(d)$ is the predicted citation count for article d in the test set D, $C(D) = \frac{1}{|D|}\sum_{d \in D} C(d)$ is the mean of the actual citation count for an article present in D. $R^2 \leqslant 1$, and a larger R^2 indicates a better performance.

Table 6.4: Confusion matrix depicting the performance of SVM at the first stage of our prediction model. The last column indicates the accuracy of the classification system for each individual category. The correct classification results (diagonal elements) are highlighted in bold font.

	PeakInit	PeakMul	PeakLate	MonDec	MonIncr	Oth	Accuracy
PeakInit	**9550**	70	20	20	0	2419	0.79
PeakMul	29	**15261**	2500	3	0	3000	0.73
PeakLate	7	718	**4842**	2	489	518	0.73
MonDec	398	444	157	**2247**	0	453	0.61
MonIncr	2	403	0	0	**2789**	0	0.87
Oth	55	5142	5	2	0	**154188**	0.96
Overall accuracy							0.78

model achieves nearly 50% higher accuracy compared to the baseline system (especially for θ and R^2). Note that the performance in 2001-2005 is also quite significant - even better than the previous regime as the system is getting trained with more data. We further observe that the typical situation where the system performs poorly is when a new venue gets introduced and quickly becomes popular; it takes certain number of years of learning for the system to predict accurately. Another important observation is that the predicted citation counts are almost always overestimated (not underestimated) for the later years. The reason behind this is not exactly clear but the result itself provides a future opportunity to estimate a linear offset to predict more accurately.

Performance of SVM classification: We have discussed the accuracy of the prediction model but this in turn depends on the underlying first stage of classification which is done using multi-class SVM. Table 6.4 shows the confusion matrix describing the performance of the SVM classification used in the first stage of our model. Each column of the matrix represents the instances in a predicted class, while each row represents the instances in an actual class. Therefore, all correct guesses are located in the diagonal of the table. Bethard and Jurafsky [21] mentioned that 90% of papers that have been published in academic journals are never cited. We have also observed in Figure 6.3 that our dataset is highly biased towards the population of the low-cited papers (i.e, ‘Oth’). Therefore, SVM also slightly overestimates ‘Oth’ category in the classification. The overall accuracy of the classification system is 0.78 which is quite promising considering the biased training samples and the fact that no feature after the publication of the paper is considered to classify the papers. Besides

‘Oth’ category, we also observe higher accuracy for class ‘MonIncr’ (0.87) which is followed by ‘PeakInit’ (0.79), ‘PeakMul’ (0.73) and ‘PeakLate’ (0.73). The lowest accuracy is obtained for category ‘MonDec’ (0.61). One possible reason could be that this is one of the rarest categories in the dataset. Thus, the lack of enough evidences might have accounted for the low final accuracy of the SVM model in classifying the papers into this category.

Performance assuming perfectly accurate SVM: In Table 6.4, we notice that in the first stage of our model, we achieve overall 78% accuracy and the error in this stage propagates in the second stage of our model. We believe that this performance can be improved a lot in future with more efficient feature selection and thus remains a potential area of future research. However, one might argue that if the SVM model could have achieved nearly 100% accuracy, how much improvement one would expect from the final prediction model. This might also answer how the error which propagates from the first stage of the model to the next stage affects the final output of citation prediction. Since we know the true category of each of the test papers, we use only those training samples belonging to the true category for training SVR, thus forcing 100% accuracy in the first stage. Table 6.5 shows the performance improvements (differences) of our model in comparison to the earlier results shown in Table 6.3 for different values of Δt (test set constitutes papers published within 1996-2000). One can clearly notice a significant improvement over the baseline model and our earlier results especially for the higher values of Δt. This indicates that the error propagating from the first stage SVM model to the next stage significantly affects long term citation prediction, and improvements in the first stage can highly enhance the overall performance of the system.

Robustness of categories: Earlier results show that the systematic categorization of the training samples improves the performance of the prediction system in comparison to the baseline system. A pertinent question could be that how robust are these categories for the final prediction, i.e., if the (near-)similar/dissimilar categories are merged together, how does it affect the final output of the model. Note that in Figure 6.3, the categories ‘PeakInit’ and ‘MonDec’ (‘PeakLate’ and ‘MonIncr’) are nearly similar in terms of the number of peaks and whether the peak occurs in the first/last half of the citation profile; others are reasonably different. Now the question is that if we merge the near-similar categories together to reduce the total number of categories, how does it affect the final prediction. The extreme case would be the baseline system itself where all the categories

Table 6.5: Performance improvement (differences) of our model in comparison to the earlier results shown in Table 6.3 for different values of Δt, while considering 100% accuracy in SVM model.

	Improvement over baseline model			Improvement over our earlier results		
	R^2	θ	ρ	R^2	θ	ρ
$\Delta t = 1$	0.34	-3.54	0.31	0.02	-0.75	0.04
$\Delta t = 2$	0.37	-5.09	0.34	0.01	-0.19	0.03
$\Delta t = 3$	0.37	-5.85	0.33	0.07	-1.26	0.04
$\Delta t = 4$	0.35	-7.22	0.36	0.08	-1.92	0.04
$\Delta t = 5$	**0.41**	**-10.19**	**0.37**	**0.15**	**-2.28**	**0.11**

Table 6.6: (Left) Performance of the two-stage prediction model for two different types of categorization schemes; (Right) performance of the baseline model and our proposed system at various time intervals after including the first year's citation count as another feature.

	Cat-1			Cat-2		
	R^2	θ	ρ	R^2	θ	ρ
Δt=1	0.87	2.05	0.85	0.59	5.23	0.63
Δt=2	**0.88**	**1.94**	**0.88**	**0.61**	**4.67**	**0.68**
Δt=3	0.79	3.38	0.80	0.55	6.86	0.61
Δt=4	0.75	4.01	0.76	0.51	8.89	0.54
Δt=5	0.71	4.10	0.72	0.50	9.58	0.49

	Baseline			Our model		
	R^2	θ	ρ	R^2	θ	ρ
Δt=2	**0.60**	**4.92**	**0.65**	**0.92**	**1.02**	**0.90**
Δt=3	0.59	5.06	0.64	0.85	2.56	0.82
Δt=4	0.58	5.44	0.62	0.83	3.16	0.81
Δt=5	0.54	6.56	0.56	0.81	3.88	0.79

are combined. Apart from this, we reconfigure the categorization in two different ways: **[Cat-1]** combining near-similar categories and keeping others separate ([PeakInit + MonDec], [PeakLate + MonIncr], [PeakMul], [Oth]), **[Cat-2]** combining one pair of dissimilar categories ([PeakInit + PeakMul], [PeakLate], [MonDec], [MonIncr], [Oth]). In this case also, we use the default set of training and test samples as mentioned in earlier in the dataset and run the two-stage prediction model separately for two types of categorization.

Table 6.6 (left) shows the final performance of the two-stage model for the two categorization schemes. One can easily notice two immediate consequences of these schemes – (i) combining two near-similar categories (as done in Cat-1) does not make much effect on the final prediction in comparison to combining two different categories (as followed in Cat-2), since the decrease in accuracy from the actual results (shown in Table 6.3) is significantly less for Cat-1 than that for Cat-2; (ii) while combining two major categories in Cat-2, the accuracy of the final prediction decreases drastically from the actual results of Table 6.3, and it tends to be closer to the baseline system. The results for Cat-1 are still worse (although slightly) than the original six category system. Hence, a natural question stays whether dividing the data into further categories would improve performance. We have tried different variations; all of them tend to introduce more noise in the SVM classification module thus decreasing the overall performance.

Impact of early citation information: In earlier papers [36], it has been shown that the citation count of a paper in the initial few years after publication plays an important role in predicting the future citation count of the paper. However, in our experiments, we have only considered those features of a paper that one can get at the time of its publication since our objective is to predict the future impact of a paper as early as possible. However, we also believe that the initial few years' citation counts can boost up the prediction of the final citation counts since these initial citations seem to be the early crowd-sourced feedback of the scientific community about the paper. Therefore, to see its impact in the final prediction, we conduct another set of experiments – we include the citation count of a paper in the immediate next year (Δt=1) of its publication as a feature and predict the citation count of each paper for Δt between 2 and 5 years. Table 6.6 (right) shows the accuracy for both the baseline system and the two-stage prediction model. As compared to Table 6.3, we can see a clear improvement of the system mostly in the higher values of Δt. Moreover, this also improves the SVM classification where we achieve 84% overall accuracy. With this information, the baseline system also improves a lot as mentioned in [230].

6.4 Faceted Recommendation for Scientific Articles

One of the most common ways of doing any literature survey is perhaps the following – start from a known article and then traverse along those articles which have either cited the known article or have been cited by the known article. In particular, when a researcher reads the known article, she starts ruminating and asking recurrent questions pertaining to it that can further lead her to browse the other articles. These questions are most often synthesized from the *knowledge context* of the end users. For instance, an expert in a certain area, while reading a paper might want to find papers presenting "alternative approach" of the query paper, thus expecting the recommendation engine to return only "alternative approach" related papers; while on the other hand a naïve user might be interested to understand the "background" of the query paper. A smart recommendation engine should be able to organize the recommended papers into multiple such facets/tags. This would not only reduce the tedious effort of searching related articles in accordance to the knowledge context of the end user, but also should answer a more fundamental question: what is the *role* of a recommended paper in relation to the query paper. However, the traditional paper recommendation systems primarily aim at improving the *relevance* of the recommendations and therefore tend to overlook the above fundamental aspect.

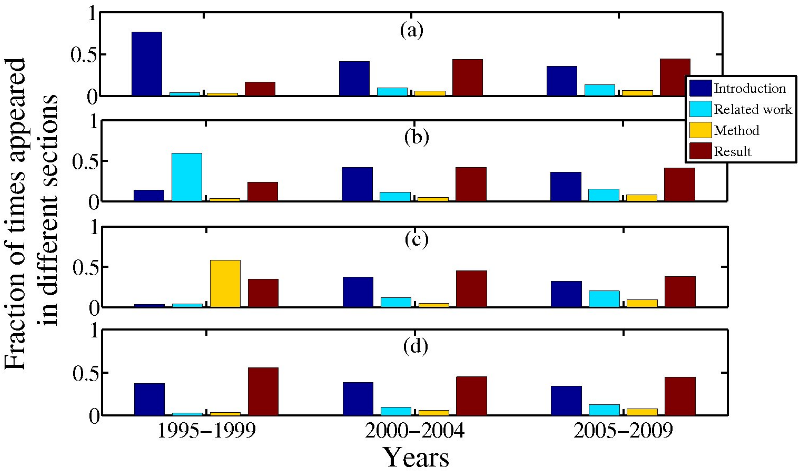
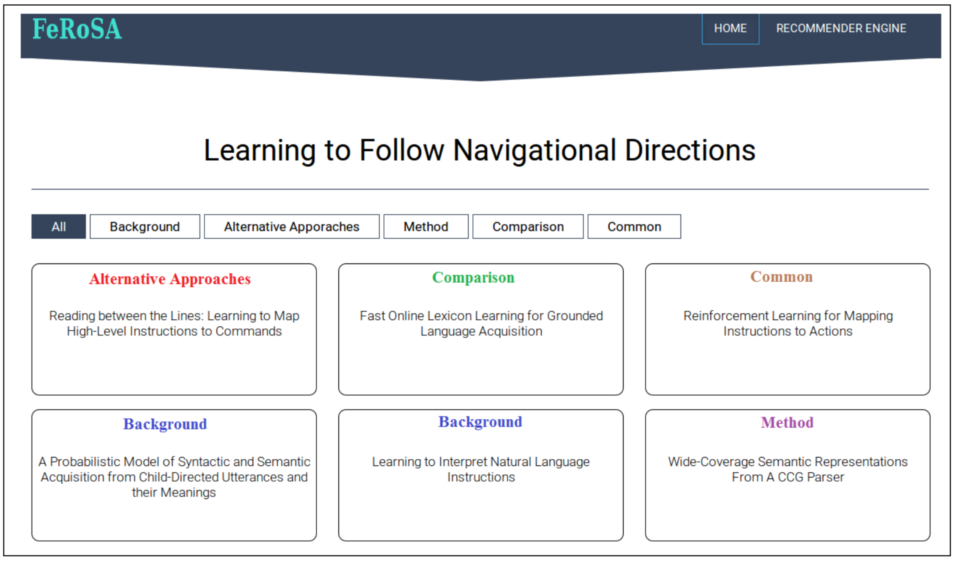


Figure 6.8: Change in role of the papers over the years. Papers which mostly appeared in (a) Introduction, (b) Related work, (c) Comparison or (d) Result section earlier have been referred to in other sections quite significantly over the years.

With respect to the content of any paper, a possible way to capture the variable knowledge context is through the section information. For instance, the "Related work" section of a paper often contains papers solving similar problems. In general, a paper might be referred to for different reasons (thus playing various roles), and the role of a referred paper can also change over the years. For instance, we observe that those papers which were mostly referred to in the "Introduction" section earlier have started appearing in "Method" section as well quite significantly (see Figure 6.8). Therefore, in order to build a simple working system one can assume that the sections serve as representatives for the different knowledge contexts. In particular, we posit that an efficient recommendation system should be able to capture the changing role of a recommended paper with respect to the query paper. This in turn calls for designing a recommendation engine that apart from ensuring the appropriateness of the recommended papers, is capable of organizing the recommendations into semantic groups or facets.

In this paper, we attempt to build a "Faceted Recommendation System for Scientific Articles" (*FeRoSA*) that given a query paper, in addition to recommending the relevant scientific papers, also organizes the recommendations into facets, thereby, suitably catering to the appropriate knowledge context of the end user. FeRoSA groups the recommendations into four naturally observed facets, namely, Background, Alternative Approaches, Methods and Comparison. This grouping has been formulated from the most intuitive forms of the knowledge context of the end users, which directly map to the different broad sections of any paper; however, the current system can very easily adapt to any other suitable form of grouping. An initial prototype version of the system is shown in Figure 6.9 and can be accessed from www.ferosa.org. Our methodology is based on a principled framework of random walk with restarts that attempts to simulate the traversal mechanism of an user initiating from the known article. The model takes into consideration both the citation links as well as the content information to systematically produce the most relevant results.



Learning to Follow Navigational Directions

Alternative Approaches

Reading between the Lines: Learning to Map High-Level Instructions to Commands

Comparison

Fast Online Lexicon Learning for Grounded Language Acquisition

Common

Reinforcement Learning for Mapping Instructions to Actions

Background

A Probabilistic Model of Syntactic and Semantic Acquisition from Child-Directed Utterances and their Meanings

Background

Learning to Interpret Natural Language Instructions

Method

Wide-Coverage Semantic Representations From A CCG Parser

Figure 6.9: The propotype of FeRoSA publicly available online. The definitions of the four facets are described in Section 6.4.1.

6.4.1 Dataset

We collect the AAN dataset[6] [182] which is a assemblage of all papers included in ACL (Association for Computational Linguistics[7]) publication venues. In the full dataset, most of the papers have raw text. The texts are pre-processed where the sentences, paragraphs and the sections are properly separated using different tags. A significant part of the corpus had word splits and word joins. These are rectified in the whole corpus using a dictionary based approach. A preliminary statistics pertaining to the used dataset is shown in Table 6.7.

Table 6.7: Statistics of the used datasets.

Number of papers	9,843
Average number of references (within ACL only)	6.21
Number of unique authors	7,892
Number of unique venues	280

Extraction of Section Heading: As discussed earlier, we categorize the citation links based on their occurrence in various sections of the paper. To extract the section heading, a list of 25,483 unique headings is collected and manually annotated into five different categories: Introduction, Related Work, Method, Results and Conclusion. The categories are further mapped into four facets, namely Background (Introduction), Alternative Approaches (Related Work), Method (Method) and Comparison (Results and Conclusion), as also suggested by Zhigang et al. [104]. A brief description of the facets/tags is as follows:

- **Background (BG)**: These are the citations which are prerequisite for understanding the basic notions of the citing paper. These citations generally point either to some seminal papers in that particular area, or to some papers which describe certain concepts that are relevant in understanding the framework of the citing paper.
- **Alternative Approaches (AA)**: If there are citations to the approaches, which can be seen as alternative to the method proposed in the citing paper, then such citations are categorized as AA. These references are often found in system oriented research papers where new methods/frameworks are proposed.

[6] http://clair.eecs.umich.edu/aan/xml/

[7] https://www.aclweb.org/

- **Methods (MD)**: If the citing paper borrows any such tools, techniques, datasets, measures or other concepts from the paper, or if both the papers have some overlap in usage of any of the entities mentioned above then such a citation is treated as MD.
- **Comparison (CM)**: As mentioned in [135], a relation is said to be comparable if the citing paper has been compared to a cited paper in terms of differences or resemblances. Most of the times, these types of references tend to occur in the evaluation section of the citing paper. Essentially, one can argue that all the AA-tagged papers can be treated as CM papers. However, the AA-tagged papers may be irreproducible or difficult to be reimplemented, and thus may not be used for comparison. We only consider the cited paper as CM if it is used by the citing paper for comparison.

In this context, Liang et al. [135] and Nanba and Okumura [153] classified the citation relations into three major categories, namely *Based-on* (same as the combination of BG and AA), *Comparable* (same as CM) and *General* (those which are not classified). However, we argue that these categories are too coarse to unfold the specific relations between citing and cited papers. While manually mapping the sections, we observe that there are few references falling in Based-on category, which are mentioned mostly in the Related work section with the explicit description and often appear alone in a sentence (average 1.05 references per line); whereas other references in this category appear as one among many others in a single sentence (average 3.35 references per line). While the latter references are mostly used to understand the background of the paper, the former are more important and often point to the papers describing alternative approaches related to the same problem. Therefore, we further divide the Based-on relation into two facets, BG and AA. In addition, we see that most of the system oriented papers often borrow techniques from the other papers to design a new framework. On the other hand, to establish the performance of a proposed system, researchers often compare their systems with the state-of-the-art. We argue that these two types of references are different and should be explicitly distinguished. Therefore, we further propose two new facets, MD and CM. In Table 6.8, we show that the experts[8] also concur with us most of the times on these four categories (facets). Note that the set of facets to be used may vary for different domains of research; however the underlying faceted recommendation framework used in this study is independent of the choice of facets.

[8] The expert opinion was taken from the annotators, who were later involved in evaluating the systems as discussed in Section 6.4.3. For a direct reference of a paper, we ask experts whether the reference indicates BG, AA, MD or CM and then compare their opinion with our section annotation (in four categories).

Table 6.8: Confusion matrix showing the agreement of the experts' judgment (indicated by column) with our section mapping (indicated by row).

	BG	AA	MD	CM
BG (Introduction)	**26**	7	6	3
AA (Related Work)	19	**63**	8	7
MD (Method)	4	2	**18**	1
CM (Comparison)	5	6	2	**28**

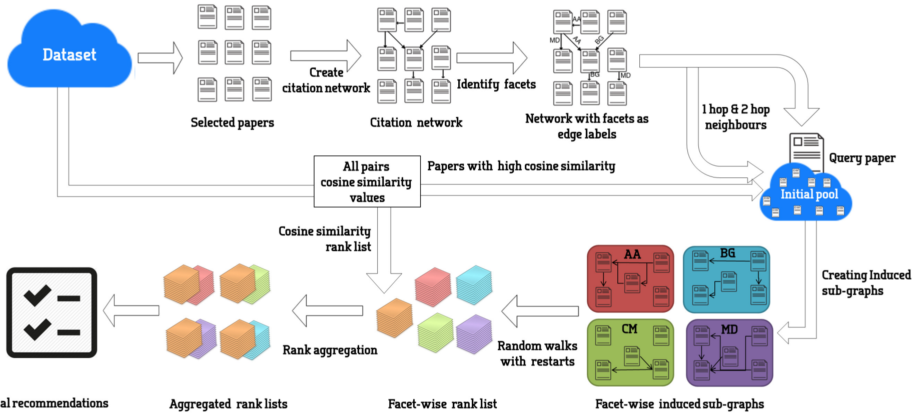


Figure 6.10: The work-flow diagram of FeRoSA.

Extraction of Citation Contexts: In addition to the citing-cited paper pair for each citation, we also need to know the context and the section heading of where the citation has occurred, in order to assign the facet. We use *Parscit* [51] to identify the citation contexts from the dataset and then extract the section headings for the pair of papers within the network. A facet is assigned to each pair of citing-cited paper, depending on the section information. The statistics regarding extracted citation contexts are shown in Table 6.9. Note that if a cited paper occurs multiple times in different sections of a cited paper, multiple facets would get assigned to this paper pair.

Table 6.9: Statistics of various facets in the annotated dataset.

Number of citation contexts extracted	61,051
Number of BG edges	23,022
Number of AA edges	10,797
Number of MD edges	8,828
Number of CM edges	18,404

6.4.2 Recommendation Method

In this section, we describe in detail the working principle of our proposed recommendation system, *FeRoSA*. Figure 6.10 shows a schematic diagram of the proposed work-flow. From the AAN dataset, we first construct a citation network, where each edge is labeled with one or more of the four facets, as described in Section 6.4.1. Given a query paper, an induced subgraph is constructed by taking its 1-hop and 2-hop neighbors in the citation network and the papers with high content similarity to the query paper based on cosine similarity measure. This graph is further used to construct an induced subgraph for each of the facets. Next, a random walk with restarts is performed to find papers in each induced subgraph, that are important for the query paper. A facet-wise rank list of papers is constructed for each query paper, along with a rank list based on papers with high cosine similarity. Finally, a rank aggregation framework is used to combine these multiple ranked lists. Below, we describe each of these modules in further details.

The citation network: We build the citation network which is a directed graph $G = (V, E)$ with edge labels. The labeling is a mapping from the edge set E to set of facets based on the data obtained from citation contexts. An edge may be tagged with multiple facets, if a paper cites another paper in multiple sections.

The induced subgraphs: We construct an induced subgraph of the network for each query paper. An initial pool of vertices is obtained by following two criteria: (i) we consider all the papers which are at 1-hop or 2-hop distance from the query paper in the citation network irrespective of the label and directionality of edges; (ii) we also consider those papers as nodes that have a cosine similarity of at least 0.49 with the query paper (top 100 papers if the number of papers exceeds 100). Then we construct an induced subgraph of nodes present in the initial pool for each facet individually. For instance, for AA we only consider those citation edges in the induced subgraph which are labeled as AA. Therefore, the graph structure for a particular query node is different for different facets. Note that in this process, few nodes might get disconnected or remain isolated. In the rest of this section, we discuss how this graph can be utilized in a systematic way to generate recommendations.

Random walk on the induced subgraphs: In order to obtain the importance of the nodes with respect to the query paper, we perform random walk with restarts (RWR) [171] on the induced subgraph with query paper being the starting node. Random walk with restarts is defined in Equation 6.3: consider a random walker that starts the walk from node i. The walker iteratively moves to its neighborhood with a probability proportional to the edge weights. At each step of the random walk, it has some probability c to return to the starting node i. The relevance score of node j with respect to node i is defined by the steady-state probability $r_{i,j}$ that the walker will finally stay at node j and is given by

$$\overrightarrow{r}_i = (1-c)\hat{A}\overrightarrow{r_i} + c\overrightarrow{e}_i \tag{6.3}$$

where $\overrightarrow{r}_i = [r_{i,j}]$ is an $n \times 1$ ranking vector; $r_{i,j}$ is the relevance score of node j with respect to node i; c is the restart probability, $0 \leq c \leq 1$; $\hat{A}$ is the normalized weighted matrix associated with the weighted adjacency matrix $A = [a_{ij}]$; $\overrightarrow{e}_i$ is the restart vector, with all its elements 0 except the i^{th} element. We consider the restart probability c as 0.4.

Apart from the citation links in the induced subgraph, we also consider the isolated nodes by assigning a *teleportation probability* (i.e., a probability of randomly jumping to any one of the isolated nodes) as 0.3, thus eliminating the chance of the isolated nodes remaining unreachable by the random walker.

Rank aggregation: We use the above framework to obtain a rank list of nodes present in the induced subgraph for each facet separately. Additionally, we consider content similarity by measuring the cosine-similarity between the query paper and each of the papers present in the induced subgraph. Next, we utilize a rank aggregation method to combine these two types of rankings. In our work, we use *RankAggreg*, an R package developed by Pihur et al. [178], where they consider rank aggregation as an optimization problem

$$\delta^* = argmin \sum_{i=1}^{m} w_i d(\delta, L_i)$$

The optimization problem finds an ordered list that minimizes the total distance between each of the provided lists and the list δ, where δ is the ideal super-list, L_i is the i^{th} ordered list and w_i is its corresponding weight, which is equally distributed in our case. $d(\delta, L_i)$ is the distance measure [178]. We have employed Spearman footrule [178] as the distance measure which is nothing but the summation over absolute difference between ranks of all the nodes while considering any two lists.

Note that for each facet T, we aggregate the ranking obtained for T and the cosine-similarity based ranking to obtain the final rank list. In addition, we also perform a total rank aggregation in order to design a flat version of FeRoSA (*f-FeRoSA*) by combining all the facet-wise rankings and the cosine similarity based ranking together (see Section 6.4.3).

6.4.3 Experimental Results

In this section, we present the performance of FeRoSA. Since evaluating the performance of such kind of systems requires domain knowledge, we design a new evaluation scheme, consisting of three independent steps. First, we ask experts with sufficient domain knowledge to generate a limited set of ground-truth dataset, based on which we evaluate our system for faceted recommendation. Second, we ask a set of researchers having partial knowledge of the domain ("semi-experts") for mass-scale evaluation. Third, we shortlist a few papers and request one of the authors of each paper to judge the quality of the recommendations returned by FeRoSA for their own papers. Further, we show that FeRoSA can appropriately be used for flat recommendation as well that significantly outperforms existing state-of-the-art systems. Towards the end, we conduct a detailed analysis of the systems for different model parameters.

Evaluation metrics: We use the following metrics to evaluate FeRoSA and the baseline systems.

1. **Comparative evaluation:** The following metrics are used to compare the competing systems:

    - **Overall Precision (OP):** It measures the ratio between the number of relevant recommendations (according to the experts' judgments) and the total number of recommendations provided for a query paper by each competing system. The overall precision of each system is then measured by averaging OPs for all the query papers.
    - **Overall Impression (OI):** It measures that among all the query papers, in how many cases a particular system is rated to have an overall better performance.

We measure this value for a system on the basis of precision majority, i.e., for what fraction of query papers, the system has a higher OP than others.

2. **Faceted evaluation:** We use facet-based precision (TP_T) for the evaluation of the facets of FeRoSA. It measures the overall precision of the recommended papers under each individual facet T, denoted by TP_T. Initially, for each paper TP_T is calculated and then the average of all TP_Ts is reported for each tag T.

As mentioned earlier, while the quality of the recommendation system is partially judged by OI, OP and TP, due to the lack of information about the actual number of relevant papers for each query paper, it is difficult to measure the recall of the system.

Evaluation of faceted recommendation:

In this section, we first describe the process of ground-truth generation, followed by a brief description of the baseline algorithms, and then elaborate the comparative evaluation of all the systems for faceted recommendation.

Ground-truth generation: Because of the unavailability of a benchmark dataset for the evaluation of scientific article recommendation especially for the faceted recommendation, we conducted an expert judgment for generating a set of faceted and flat recommendations (used later in this section) as our ground-truth. For this purpose, we shortlisted a set of 30 query papers that cover the fields of expertise of the experts. For each query paper, we presented 30 recommendations that we pulled from four separate systems: FeRoSA (12 recommendations), Google Scholar (GS)[9] (6 recommendations), Microsoft Academic Search (MAS)[10] (6 recommendations) and a graph based paper recommendation system (we call it LLQ) proposed by Liang et al. [135] (6 recommendations). Note that the three latter systems, namely GS, MAS and LLQ, which are quite popular for paper recommendation are further used in this section as competing systems to FeRoSA for flat recommendation. The reason behind taking more recommendations from FeRoSA in the ground-truth dataset

[9] http://scholar.google.co.in

[10] http://academic.research.microsoft.com/

is that we intended to evaluate other two proposed faceted baseline systems using this annotation, each of which might not cover six recommendations if we would have taken less. The experts were provided with web based interfaces[11], in which they were shown a query paper and the 30 recommendations for each query paper (the name of the systems remained anonymous). Each expert had to mark whether each recommended paper was relevant to the query paper, and if so, the possible facet(s).

Baseline systems: To the best of the authors' knowledge, FeRoSA is the first and unique faceted recommendation engine for scientific articles and hence no other systems are available, which can be considered as a baseline. We, therefore, design two competitive baseline systems to compare with FeRoSA.

(i) VanillaPR: For each query paper we form a single induced subgraph $G'(V', E')$ which is exactly same as that of FeRoSA ignoring the facet labeling. Once the graph is formed, we perform RWR with query paper as the restart node. We then retrieve the nodes having the highest values from RWR. Finally, to label the retrieved documents with facets, we train a supervised model using the ground-truth data we collected in the previous section. The main motivation was that since the section heading is quite correlated with the facet of the paper, this information can be used for labeling a recommended paper with the possible facet. Based on this idea, we use the following three types of features to train the supervised model. For each pair of the query and the recommended paper, we use a total of 12 (4 Boolean + 8 real valued) features as follows: (i) section of the recommended paper, if it appears in one hop distance of the query paper (4 Boolean features, one for each section), (ii) within the vertex set V' for the query paper, fractional number of times a particular recommended paper appears in a given section (4 real-valued features, one for each section), and (iii) for a given recommended paper, fraction of times it is cited in a given section by any paper in the whole dataset (4 real-valued features, one for each section). We then learn the weights for features with a rankSVM model [129]. We report the average precision of the system after performing a three fold cross-validation (20 instances for training and 10 for testing for each iteration) over the ground truth data.

(ii) FeRoSA-CS: Our second baseline recommends papers by relying only on RWR, performed on subgraphs of papers within 2-hop distance from the query paper, without con-

[11] http://www.ferosa.org/evaluation/

sidering the content-similarity (CS) based papers. This in turn answers the necessity of considering cosine-similarity based papers while constructing the initial pool.

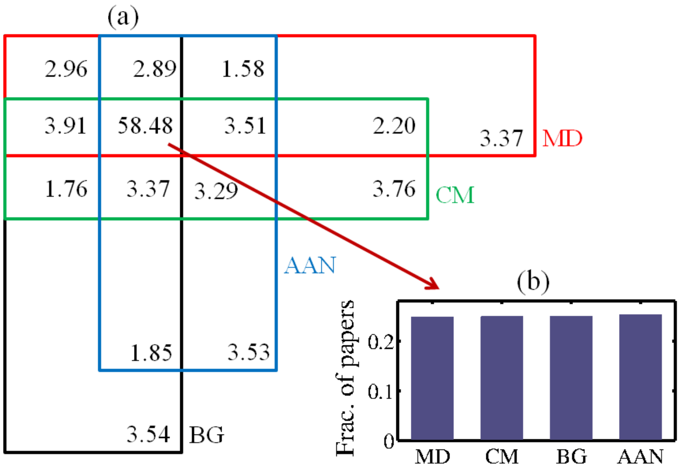


Figure 6.11: (a) Venn diagram of the recommended papers in four facets; (b) distribution of papers which are tagged by all four facets.

Comparative analysis: We conduct an empirical study on the results obtained from FeRoSA. Figure 6.11(a) represents a Venn diagram of all the recommended papers under different facets, i.e., what all facets have a particular paper been recommended under various different queries. For example, 3.54% of the recommended papers appear only as BG to any of the query paper. We observe that 58.48% of the recommended papers appear under all the 4 facets for various queries. For these papers, we further show in Figure 6.11(b) their distribution in different facets, which seems to be fairly uniform.

Table 6.10: Faceted evaluation of all the competing faceted recommendation systems. For VanillaPR, the performance is reported by performing 3-fold cross validation.

Facets	**VanillaPR**	**FeRoSA-CS**	**FeRoSA**
BG	0.65	0.51	**0.79**
AA	0.48	0.34	**0.56**
MD	**0.62**	0.39	**0.62**
CM	0.44	0.38	**0.62**
Average	0.55	0.40	**0.65**

We report in Table 6.10 the average precision of all the systems for different facets. We observe that FeRoSA attains the highest average precision (0.65) amongst all other systems, which is 18% higher than VanillaPR and 62.5% higher than FeRoSA-CS. The

Table 6.11: Confusion matrix for the faceted evaluation (FPR: false positive rate). For VanillaPR, we report the confusion matrix for that run where maximum average precision for all the facets is achieved.

(a) VanillaPR

Facets	AA	BG	CM	MD	FPR
AA	41	6	12	4	0.08
BG	13	87	11	17	0.09
CM	7	9	17	13	0.06
MD	11	6	7	32	0.05

(b) FeRoSA-CS

Facets	AA	BG	CM	MD	FPR
AA	23	14	9	17	0.09
BG	18	69	13	28	0.14
CM	13	8	18	7	0.06
MD	11	14	5	26	0.05

(c) FeRoSA

Facets	AA	BG	CM	MD	FPR
AA	33	11	6	13	0.07
BG	14	99	5	10	0.07
CM	2	6	28	10	0.04
MD	3	7	12	34	0.04

maximum precision of FeRoSA is obtained for the BG tag (0.79), which is followed by MD (0.62), CM (0.62), and AA (0.56). The pattern is also similar for FeRoSA-CS. For the case of MD, however, we observe similar performance for VanillaPR and FeRoSA.

For further analysis, we present the confusion matrix for the facets in Table 6.11 along with the false positive rate for each system. The false positive rates are quite low for the facets (i.e., the specificity values of the facets are quite high). To understand the reason behind the misclassification, we unfold the tagged citation network once again and observe that it arises due to the frequent occurrences of a single edge being tagged by multiple facets. For instance, Table 6.11 shows that AA is mostly misclassified to MD for FeRoSA. In the tagged citation network, we observe the same phenomenon that among the multi-faceted edges with AA tag, around 50% of edges are tagged by both AA and MD (similarly for CM and MD with 36.6% of occurrences together).

Mass-scale evaluation: To broaden the evaluation of FeRoSA, we perform a mass-scale evaluation, aiming for more coverage on the system output and targeting a wider set of evaluators. All the selected evaluators had a good knowledge of the NLP domain. This time we reverse engineer the process by selecting few papers from the ground-truth data, each of which appear in the recommendation of multiple query papers. To start with, we shortlisted a collection of 31 such recommended papers. For each recommended paper, we enlisted the set of query papers (and the facets) in which the recommended paper has appeared[12]. The evaluators then evaluated the relevance of the recommended paper, as well as the relevance of the facet with respect to each query paper in which the given recommended paper has appeared. A total of 26 experts participated in this evaluation task. Table 6.12 provides details about the experiment conducted. For each recommended

[12] This indeed reduced the evaluators' effort of reading multiple papers.

paper, we calculate the faceted precision for all its corresponding query papers and show the results in Table 6.13(a). Similarly, we calculate the average faceted precision for all query papers and report it in Table 6.13(b). We observe that for both the cases, FeRoSA significantly outperforms other baselines.

To validate our hypothesis, that the facet of a recommended paper can vary from one query paper to other query paper, we categorize our set of 31 recommended papers into two sets. We call the first set as 'steady facets', where each recommended paper appears in a particular facet for more than 75% of the time. Other papers fall in the set 'changing facets'. We observe that the average facet-wise precision is 0.75 and 0.64 for steady facets and changing facets respectively.

Table 6.12: Statistics of the mass-scale evaluation.

Number of experts	26
Number of recommended papers evaluated	31
Average number of query papers for each recommended paper	23.32
Total number of query-recommendation pairs evaluated	723

Table 6.13: Facet-wise precision for (a) recommended paper to query paper and (b) query paper to recommended paper.

(a)

Facets	**Vanilla**	**FeRoSA**	**FeRoSA**
	PR	**-CS**	
BG	0.57	0.70	**0.73**
AA	0.43	0.41	**0.53**
CM	0.37	0.59	**0.64**
MD	0.58	0.55	**0.69**
Avg.	0.49	0.56	**0.64**

(b)

Facets	**Vanilla**	**FeRoSA**	**FeRoSA**
	PR	**-CS**	
BG	0.66	0.82	**0.85**
AA	0.45	0.48	**0.54**
CM	0.42	0.54	**0.73**
MD	0.51	0.68	**0.77**
Avg.	0.51	0.63	**0.72**

We are further interested to observe whether our system performs better for the highly-cited query papers, or whether the same accuracy is achieved for all citation ranges of the query papers. Generally, a standard recommendation system should perform equally well for all ranges of query papers. Here we divide the entire range of incoming citations of the query paper into three buckets and measure the faceted precision of all the competing systems for each bucket separately. In Table 6.14, we see that FeRoSA performs significantly better than the other baseline systems even for low-cited query papers.

Evaluation by the authors: There is no better alternative than the authors themselves

Table 6.14: Performance of three competing faceted systems for different query papers divided into three citation ranges (Low: 0 to 6, Medium: 7 to 28, High: 29 to 343).

Facets	VanillaPR			FeRoSA-CS			FeROSA		
	Low	Medium	High	Low	Medium	High	Low	Medium	High
BG	0.53	0.73	0.71	0.44	0.54	0.55	0.65	0.84	0.87
AA	0.41	0.52	0.49	0.28	0.35	0.41	0.53	0.56	0.61
MD	0.57	0.59	0.71	0.40	0.34	0.44	0.65	0.55	0.67
CM	0.29	0.55	0.48	0.33	0.39	0.41	0.56	0.62	0.69
Avg.	0.45	0.59	0.59	0.36	0.40	0.45	**0.59**	**0.64**	**0.71**

when it comes to evaluating the recommendation for a particular paper. We were curious to know whether FeRoSA could impress the authors with its recommendations. We designed the judgment experiment by selecting a set of 30 authors and we sent each of them, a judgment form, where we specified one of his/her papers as query paper, and one (top) recommendation from FeRoSA for each facet. The author had to make a binary judgment about the relevance of recommendation to the query as well as the relevance of the facet for the recommendation separately. We obtain an average faceted precision of 0.50 (BG: 0.49, AA: 0.42, MD: 0.52, CM: 0.59). In 75% cases the recommended papers are marked as relevant. Four authors marked three out of four faceted recommendations as relevant. Overall, the authors appreciated the attempt of designing a faceted recommendation system for scientific articles.

Evaluation of flat recommendation:

We further posit that FeRoSA can also be used as a flat recommendation system if the rank lists obtained from the different facets and the cosine-similarity based ranking can be appropriately combined. Therefore, we use the rank-aggregation method in order to obtain a flat recommended list. In this section, we discuss the performance of the flat version of FeRoSA (*f-FeRoSA*) and compare it with three state-of-the-art flat baseline systems.

Baseline systems: we consider three flat baseline systems: Google Scholar (GS), Microsoft Academic Search (MAS) and a graph based paper recommendation system proposed by Liang et al. [135] (we call it LLQ from the initials of the three authors of the paper). We consider LLQ as a baseline system because similar to our approach, it also classifies citation relations into three categories, namely Based-on, Comparable and General using the approach proposed in [153], and these categories are further used to compute a final combined score. Note that while GS and MAS are mostly known for searching scientific papers, an inherent nature of ranking of the retrieved results has lead us in using them as potential baseline systems.

Table 6.15: (a) Flat evaluation of the competing systems; (b) overall precision of f-FeRoSA at different number of recommendations.

(a)

Systems	OI@3	OP@3
GS	0.27	0.61
MAS	0.17	0.45
LLQ	0.13	0.41
f-FeRoSA	**0.43**	**0.79**

(b)

OP	f-FeRoSA
OP@3	0.79
OP@5	0.78
OP@10	0.71

Comparative analysis: We perform a broad analysis of the performance of all the competing methods. Table 6.15(a) reports the values of individual metrics mentioned earlier, averaged over all the judgments conducted by the experts. For top 3 recommendations per system, f-FeRoSA achieves OP of 0.79 which is 29% higher than GS, 75% higher than MAS, and 62% higher than LLQ. One can also notice that for 43% of the cases, f-FeRoSA fares better than all other systems in terms of overall impression. Clearly, f-FeRoSA is preferred nearly twice more than Google Scholar, which is the second best performing system. This indeed shows that f-FeRoSA outperforms the state-of-the-art recommendation systems by a reasonable margin. We also see in Table 6.15(b) that f-FeRoSA is quite consistent in recommending highly relevant papers within top rank list.

Detailed analysis of system performance: Here, we conduct a detailed analysis of the results obtained from the baseline systems and f-FeRoSA. In particular, we intend to measure the performance of the systems for parameters such as hop distance, incoming citations of the query paper etc. Note that for better comparison, we divide the entire range of parameter values obtained from our system into three buckets such that all the buckets contain nearly equal number of elements, except few cases such as hop distance and age difference of query and recommended papers where we instead use the actual number of papers falling in a certain range. Note that the analysis is performed for those query papers

Table 6.16: Comparative evaluation based on Overall Precision (OP) of the baseline systems and f-FeRoSA for different parameters (P_q: query paper, P_r: recommended paper, Y_x: year of publication of paper x). The range of each bucket is shown in the column heading. The fraction of papers for each bucket is also mentioned within parenthesis in each cell of the table.

Systems	Hop distance			Age difference ($Y_{P_q} - Y_{P_r}$)			Incoming citations of P_q		
	1-hop	2-hop	>2-hop	Newer (-5 to -15)	Same Time (-4 to 3)	Older (4 to 20)	Low (0 to 6)	Medium (7 to 28)	High (29 to 343)
f-FeRoSA	**0.81** (0.35)	**0.79**(0.34)	**0.76** (0.31)	**0.86** (0.33)	**0.77** (0.35)	**0.71** (0.32)	**0.79** (0.40)	**0.71** (0.30)	**0.88** (0.30)
GS	0.57 (0.32)	0.52 (0.40)	0.71 (0.28)	0.52 (0.30)	0.72 (0.41)	0.61 (0.29)	0.51 (0.40)	0.58 (0.30)	0.68 (0.30)
MAS	0.43 (0.37)	0.38 (0.39)	0.62 (0.24)	0.57 (0.44)	0.62 (0.28)	0.65 (0.28)	0.41 (0.40)	0.46 (0.30)	0.52 (0.30)
LLQ	0.47 (0.34)	0.28 (0.35)	0.44 (0.32)	0.43 (0.25)	0.48 (0.28)	0.26 (0.47)	0.44 (0.40)	0.42 (0.30)	0.36 (0.30)

along with their recommendations, which were annotated by the experts. Various ranges might differ if this analysis is repeated for the whole dataset. We keep the ranges similar for the baseline systems for fair comparison. We also report the fraction of papers, falling under a particular range for a competing system.

Hop distance: Here, we intend to measure the effectiveness of the recommended papers for each hop distance. For this, we calculate the OP of recommended papers at each hop (i.e., among the recommended papers that fall in i-hop distance, what fraction is marked as relevant by the experts, where $i \in \{1, 2, > 2\}$). Table 6.16 shows that all the systems except LLQ attain the maximum accuracy for recommendations at a distance larger than 2-hops. Therefore, recommending only the 1-hop papers may not be ideal in terms of precision of the system for f-FeRoSA, GS and MAS. Note that the fraction of recommendations from different hops seems to be almost similar for all the competing systems with the papers in 2-hop being recommended the most. f-FeRoSA, however, maintains a reasonable balance and outperforms the competing systems for all the hop distances.

Age difference between query (P_q) and recommended (P_r) papers (i.e., $Y_{P_q} - Y_{P_r}$): A crucial factor for the competing systems is the age difference of the query and the recommended papers. We measure whether the older recommended papers relative to the query paper (denoted by positive value of $Y_{P_q} - Y_{P_r}$) achieve higher accuracy as compared to the recent papers published after the query paper (denoted by negative value of $Y_{P_q} - Y_{P_r}$) in the relevance judgment. We again divide the entire data into three buckets, and Table 6.16 shows that f-FeRoSA outperforms other baseline systems for all the buckets attaining a maximum accuracy when the recommended papers are newer than the query papers. GS achieves the highest accuracy in the middle bucket, which is the same as f-FeRoSA. While GS tends to recommend papers more from the same time as the query paper, MAS recommends mostly the newer papers. f-FeRoSA maintains a nice balance by recommending papers from all the three ranges.

Incoming citations of query paper P_q: Here we divide the entire range of incoming citations of the query paper into three buckets and measure the accuracy in each bucket separately. Note that since the query papers used in evaluation are same for the all the systems, the fraction of papers in each bucket is also the same for all the systems. We notice in Table 6.16 that f-FeRoSA achieves maximum accuracy for the papers with high citation. Interestingly, f-FeRoSA outperforms all the baseline systems by a huge margin, for the query papers with low citations and high citations. This feature is very critical because while it is easier to obtain a good evidence for the highly cited paper, it is much more difficult to find relevant papers for the not-so-highly cited query papers.

6.5 Diversified Citation Recommendation System

Finding relevant scholarly articles from the literature for a given topic is an important task for several scientific activities. This may be required, for example, to understand the current state-of-art in the topic, or to provide the citations while writing a research article. With more than one hundred thousand new papers published each year, performing a complete literature survey to find relevant articles has become a difficult task for research community. Researchers typically rely on manual methods such as keyword-based search via web search engines, reading proceedings of conferences, and browsing publication lists of known experts in the respective fields. These techniques are laborious as well as time-consuming, and they allow to reach only a limited set of articles in a reasonable time. For these reasons, there has been a significant effort from research community to develop automatic recommendation systems that help researchers to find relevant articles [97, 98].

While there is a significant body of work on the design of citation recommendation systems, the state-of-the-art on this problem suffers with the following three limitations.

First, some of the existing citation recommendation systems require the entire article [98] Further, some other existing methods [97] need the citation context as input to suggest the appropriate references for the given article. This implies the assumption that the person conducting the search is confident of the novelty of contributions in the article (which is why he/she chooses to invest sufficient time to create the article). This reduces the usefulness of such citation recommendation systems to only as a refinement tool, and not a tool which can potentially be used at the beginning of the acedamic research. Third, one of the problems in the keyword based citation recommendation is that the search results typically would not include the semantically correlated articles if these articles do not use exactly the same keywords.

6.5.1 Proposed Ranking Method: DiSCern

Our proposed model primarily builds on a time-variant random walk process, known as the *vertex-reinforced random walk* (*VRRW*) [175], which contrary to the general *PageRank* algorithm, takes into account both *prestige* and *diversity* in order to rank the vertices in a network. Here we first describe the key ingredients of our proposed citation recommendation system and then we present the algorithms.

Citation Network Construction: Since our method is primarily based on a network, we construct a paper-paper citation network as follows. A citation network is defined as a graph $G =< V, E >$ where each node $v_i \in V$ represents a paper and a directed edge e_{ji} pointing from v_j to v_i indicates that the paper representing v_j cites the paper representing v_i in its references. We also add a self-link to each node.

Keyword Network Construction: Since the preliminary idea of our system is to recommend diversified citations for a particular search query, we intend to expand the input query to obtain a set of similar keywords that can cover different semantics of the query. For this, we use the keyword meta data available with the articles. We then construct the keyword-keyword graph as follows: We construct the keyword-keyword graph as an undirected and weighted graph $G_k(V_k, E_k)$ where each node in V_k represents a keyword and two nodes v_i^k and v_j^k ($v_i^k, v_j^k \in V_k$) are connected by an edge e_{ij}^k ($\in E_k$) if there is at least one article that contains both the keywords corresponding to these two vertices. The weight $w^k_{i,j}$ associated with an edge e^k_{ij} is determined by the number of articles where both the keywords corresponding to v^k_i and v^k_j appear.

Query Expansion by Clustering Keywords: Our next task is to cluster similar keywords from the topological structure of the keyword-keyword network. We utilize *Louvain* [27], a well known state-of-the-art algorithm to find communities in the keyword-keyword graph. Now given an input query, the system first identifies the community membership of this query and then all the constituent keywords present in that community are fetched for the next step of the framework. We refer to the query expansion step as **QExpn**.

Retrieving Diverse and Relevant Citations: After expanding the given input query, we obtain a *expanded query* containing a set of similar keywords using QExpn described as above. Then the articles corresponding to the expanded query are collected to further determine an induced subgraph from the original citation network. We now run DiSCern on this induced subgraph to come up with the citation recommendations. We refer to this as *LocDiSCern* since we use DiSCern along with QExpn. Note that, LocDiSCern is convenient enough to accept multiple query keywords as well. In that case, we map each of the input query keyword to its corresponding community in the keyword-keyword graph and collect the articles of all the mapped communities, which in turn form the expanded query of the keywords.

Note that one might argue on the use of the keyword expansion step in LocDiSCern. However, we can also run DiSCern directly on the entire citation network by omitting the QExpn step. We refer to this as *GloDiSCern*. Other alternatives can also be possible wherein DiSCern is applied on the entire graph first, followed by QExpn and then the top K results can be returned. We have also attempted this approach, but did not get significantly good results.

6.5.2 Experimental Results

We use two datasets: computer science publication dataset (see Section 5.2) and theoretical high-energy physics dataset[13]. As mentioned earlier in Section 6.5.1, a natural competitor

[13] http://www.cs.cornell.edu/projects/kddcup/datasets.html

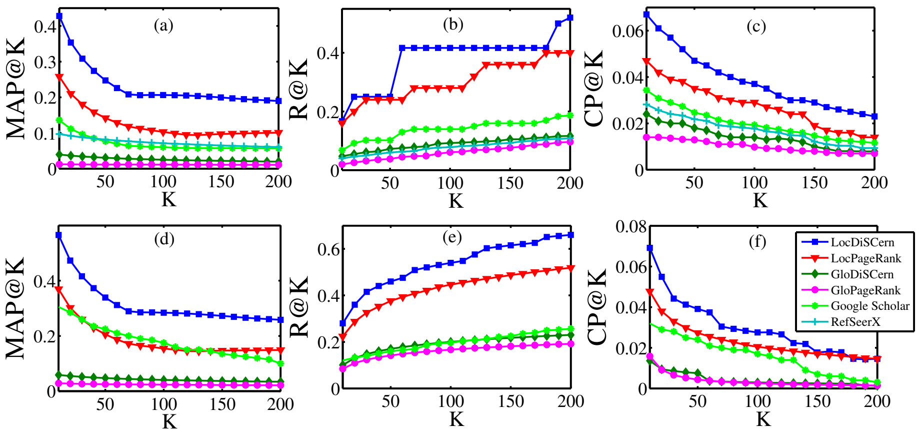


Figure 6.12: Comparison of network-based ranking algorithms in recommending citations for relevance measures: MAP, recall and co-cited probability for computer science (top panel: (a) – (c)) and high-energy physics (bottom panel: (d) – (f)) datasets.

for DiSCern is PageRank. Other baseline algorithms include LocPageRank (local PageRank), GloPageRank (global PageRank) and GloDiSCern. For evaluating our system, we manually collected a set of survey papers from both the datasets. We searched for the keywords such as “literature”, “survey”, “review” in the title of the papers. Now for each paper in the gold-standard, we know the references that the current paper has cited. We assume that these references are diverse and serve as the gold-standard in our experiment.

We evaluate the quality of the citation recommendations suggested by both the proposed and the baseline algorithms with a number of measures [116]: **Relevancy measures:** Recall ($R@K$), Mean Average Precision ($MAP@K$) and co-cited probability ($CP@K$); **Diversity measures:** l-hop graph density ($den_l@K$) and l-expansion ratio ($\sigma_l(S)$); **Other measures:** average publication year ($T@K$) and difference ratio ($DR@K$).

In Figure 6.12(a) and Figure 6.12(d), we observe that both LocDiSCern and LocPageRank outperform GloDiSCern and GloPageRank with respect to the $MAP@K$ measure. Further, the performance of LocDiSCern is more about 40% than that of LocPageRank for both the datasets. We also notice that the performance GloPageRank seems to be the lowest among the four algorithms. In Figure 6.12(b) and Figure 6.12(e), we plot the value of recall for the four algorithms on computer science and physics datasets respectively. The observation is quite similar to that of Figure 6.12(a) and Figure 6.12(d). Surprisingly, we notice that the plots of recall for LocDiSCern and LocPageRank almost behave like a step-function for

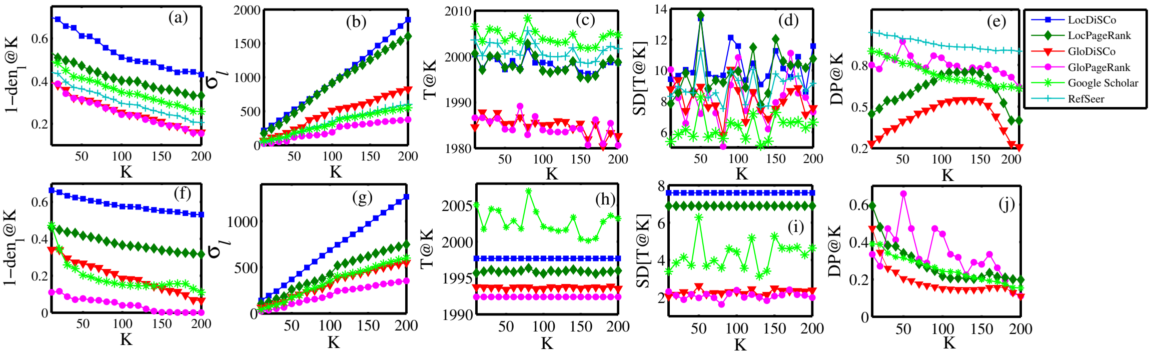


Figure 6.13: Comparison of network-based ranking algorithms in recommending citations for different diversity (l-hop graph density, l-expansion ratio) and other (average publication year, standard deviation (SD) of the publication year and the differences ratio) measures for computer science (top panel: (a) – (e)) and high-energy physics (bottom panel: (f) – (j)) datasets. In frames (e) and (j), the differences of all the other systems are measured with respect to the LocDiSCern; therefore, no line for LocDiSCern appears in that panel.

the computer science dataset. It essentially indicates that for a set of values within a certain range of K, each of them performs nearly similar. After a certain point of time, the recall value tends to increase suddenly. However, this behavior is not observed for GloDiSCern and GloPageRank. In Figure 6.12(c) and Figure 6.12(f), the value of co-cited probability is plotted for the four algorithms on both the datasets. Here again, the overall observation is similar to the earlier two scenarios. We observe that the pattern of CP tends to decrease with the increase of K. It essentially indicates that even if the recommended candidates do not appear in the gold-standard dataset, they seem to be quite relevant to the input query since there exist a significant amount of papers that cite both these (recommended and gold-standard papers) simultaneously. This indeed corroborates our earlier hypothesis that our system can also recommend better citations that might not appear in the gold-standard set.

In Figures 6.13 (a)-(b) and (f)-(g), we present the results using the two diversity measures. In Figure 6.13(a) and Figure 6.13(f), we observe that both LocDiSCern and GloDiSCern clearly outperform LocPageRank and GloPageRank. On similar lines, both LocDiSCern and GloDiSCern significantly outperform their counterparts using the l-step expansion ratio (see Figure 6.13(b) and Figure 6.13(g)), which is related to the coverage of the network with the recommendations. Both LocPageRank and GloPageRank perform convincingly worse with respect to these diversity metrics. In particular, the results obtained from the LocPageRank and GloPageRank are more clustered in the network compared to that of LocDiSCern as well as GloDiSCern. In Figure 6.13(c) and Figure 6.13(h), we plot the average publication year of all the recommended candidates using each of the four algorithms on two datasets respectively. Interestingly, we notice that our proposed two DiSCern-based algorithms outperform the two baseline systems. It indicates that the vertex-reinforced random walk based methods tend to recommend mostly the recent papers as compared to the PageRank based methods. However, one might argue that the higher value of publication year might not be a good indicator to judge the time span covered by the recommended candidates. A superior citation recommendation system should recommend citations that covers a large time span, i.e., the standard deviation of the publication years of the recommended papers should be higher. In Figure 6.13(d) and Figure 6.13(i), we plot the standard deviation of the publication years of recommended papers for the four algorithms on two datasets. Here also, we observe that our approach outperforms the baselines. Therefore, we conclude that vertex-reinforced random walk based algorithms not only recommend high quality citations based on relevancy, but also they tend to recommend both older and recent citations. In Figure 6.13(e) and Figure 6.13(j), we measure the differences of the outputs obtained using the remaining three algorithms with respect to LocDiSCern. As expected, we observe that the difference is most prominent for the results obtained from GloPageRank. However, we also notice that the patterns for GloDiSCern and LocPageRank in computer science dataset are similar – an initial increase is followed by a decrease (hyperbolic shape). The reason could be that there might exist a critical value of K after which these systems tend to return almost same set of results. A good recommendation system should adopt this critical K value while recommending results for different queries. In physics dataset, the pattern is similar for GloDiSCern and LocPageRank; however it decreases exponentially with the increase of K.

6.6 Summary of this Chapter

In this chapter, we utilize the citation network and the community structure together to build real applications. The contributions of this chapter are as follows.

- The categorization of citation profiles offers a necessary first step towards reformulating the existing quantifiers available in Scientometrics (e.g., impact factor) that should leverage on the different categories of citation patterns in order to enhance their meaningfulness.

- We further use the category information in a prediction system where the training samples are stratified to enhance the accuracy of the predictions.
- We also perform a semantic stratification of the data that further helps us designing FeRoSA which outperforms the baselines in both faceted and flat recommendations.
- Finally, we develop DiSCern, a novel framework that balances prestige and diversity in the task of citation recommendation. The model is tested on a large publication dataset of computer science domain and a dataset of physics domain. The experimental results show that our proposed approach is quite efficient and it outperforms the state-of-the-art algorithms in terms of both relevance and diversity.

Chapter 7

Conclusion and Future Work

In this chapter we elaborate important contributions from this thesis and finally wrap up this thesis by pointing to some future research directions that have been opened by this thesis.

7.1 Summary of Contributions

Community analysis of a network has remained in constant focus among the researchers since last one and half decades. Most of the work tried to design algorithms for community detection. In this thesis, the major focus has been to interpret the notion of belongingness of a node within a community, which has often been ignored due to the assumption that nodes have an equal extent of belongingness within a community. To explore this point, we have started our investigation to observe the variability of a community detection algorithm in producing output for a certain network. Then we have proposed different metrics to measure the extent to which a vertex belongs to a (non-overlapping or overlapping) community. Next, we have developed algorithms to detect communities from the networks. Following this, we have exhaustively studied the real-world community structure of a large citation network. Finally, we use the community information further to design two applications. In the following, we summarize the contributions for each problem separately.

7.1.1 Constant Communities in Networks

Constant communities are regions of the network whose community structure is invariant under different perturbations and for community detection algorithms. They, thereby, represent the similar relationships in the network. The existence of multiple results for community detection is well known; however, this is one of the first studies of the invariant subgraphs that occur in a network. The contributions of this work are summarized as follows.

- First, we observe that constant communities do not always have more internal connections than external connections. Rather, the strength of the community is determined by the number of different external communities to which it is connected. We propose a metric to quantify the pull that a vertex experiences from the external communities, and the relative permanence of the said vertex indicates its inertia to stay in its own community.
- Second, in most networks, constant communities cover only a subset of the vertices. Depending on the size of the constant communities it may not be correct or necessary to assign every vertex to a community, as is the focus of most community detection algorithms. Furthermore, when we insist on assigning a vertex to a community, the constant communities can be leveraged to produce results with higher modularity and lower variance.
- Third, the high functional cohesion among the vertices of the constant community can render meaning to the community structure of the networks. This conclusion is much more apparent for labeled graphs where the vertices are associated with certain functional properties. If we stop at detecting only the constant communities and treat them as the actual community structure of the graph, we observe that sometimes they act as a hard bound since no further community detection might be possible. Therefore, we suggest that the prior detection of these building blocks is always significant in order to decide how to merge them into more coarse-grained communities pertaining to a diluted functional cohesion.
- The fourth and most important observation is that not all networks have significant constant community structure. Two such examples in our test suites are Power and

Email graphs. The absence of constant communities in the networks indicates that either communities in general do not exist (such as Power network) or they are highly overlapped and therefore do not have a significant constant region. A set of professional emails within correspondents in the same university is likely to have more overlaps than clear cut communities.

- Finally, we demonstrate evidence that the modularity measure is not enough to judge the inherent compartmental structure of a network. For instance, Email and Power networks have reasonably higher modularity values compared to the others. Still, no consensus is observed in their community structures. Rather their sensitivity measures indicate that each node might separate out as individual constant community in the further iterations. Therefore, the goodness metric of the community detection algorithm should be redefined in a way that can effectively capture the modular structure of the network.

7.1.2 Permanence and Network Communities

In this chapter, we introduce two vertex-based metrics, permanence (Perm) and overlapping permanence (OPerm) for evaluating the goodness of communities in networks. From our experiments we observe that the scores of these metrics have a good correlation with the quality of the ground-truth communities. In addition, these two metrics also provide some significant advantages compared to other popular community scoring functions. We summarize the contributions of this chapter as follows.

- The values of Perm and OPerm strongly correlate to the community like structure of the network. Therefore, these metrics can also be used to identify whether the network is at all suitable for community detection.
- We believe that the advantages of the proposed metrics arise because these are local vertex-based metrics as opposed to the more common global/mesoscopic metrics. At the same time, these metrics also derive the benefits of a global metric to a certain extent by looking into the exact community assignments of the external neighbors of the vertex considered. Perfectly global metrics tend to aggregate the

effect of the connections of all the vertices in a community, which can lead to a loss of information, particularly if the distribution of the connections is skewed. A vertex-based metric is more fine-grained, and therefore allows partial estimation of communities in a network whose entire structure is not known.

- The algorithms, named MaxPerm and MaxOperm are able to detect meaningful communities from both synthetic and real-world networks. Moreover, these are highly resilient to the problems, such as resolution limit, degeneracy of solutions that are often observed in most of the state-of-the-art algorithms.
- Finally, for the first time the community assignment of a vertex has been studied in such finer details by checking the community assignment of each individual vertex in a network. This in turn establishes more strongly the correctness of the algorithm in finding the modular structure of a network.

7.1.3 Analyzing Ground-truth Communities

In this chapter, we analyze the communities (research areas) of a large scale citation network. The ground-truth labeling has allowed us to study rise and fall of scientific research in computer science domain over the last 50 years. Next, we study the interdisciplinary activities in computer science domain and unfold the evolution dynamics of core and interdisciplinary fields. Finally, we study the research field adaptation process of a researcher in her research career and develop a stochastic model to mimic this real-world phenomenon. In summary, this chapter shows that the usual consensus on the fact that suggesting an efficient community detection technique usually marks the “endpoint” in research in this area might not be true; in contrast, it possibly triggers the beginning of a new dimension of research, whereby, the temporal interaction, influence, shape and size of the communities so obtained can be suitably analyzed thus allowing for newer insights into the complex system under investigation. The contributions of this chapter are as follows:

- We provide a large scale real-world network with the labeled ground-truth community structure. We believe this dataset would help in the evaluation of various future community detection algorithms.

- The longitudinal analysis of the community interactions has revealed a complete picture of the paradigm shift in computer science domain. We also draw a correlation of this shift with the NSF funding statistics.

- We propose a bunch of metrics to measure the interdisciplinarity of the research fields. Few fields such as Data Mining, WWW, Natural Language Processing, Computational Biology, Computer Vision, Computer Education provide clear indications of interdisciplinarity in terms of all the metrics proposed here. These metrics further allow us to develop a classification model to identify core and interdisciplinary fields of a particular domain.

- The core-periphery organization of citation network reveals that the interdisciplinary fields are accelerating steadily toward the core of computer science domain.

- We explain the field adaptation process of a researcher through a dynamic model. We notice that the highly-cited researchers typically follow "scatter-gather" process by working on diverse fields throughout the entire career, while remaining focused on a single field in each time period.

7.1.4 Community-based Applications

In this chapter, we design two applications pertaining to the citation networks by leveraging the community information of the network. First, we analyze various citation profiles of scientific articles after publications and categorize them into six classes. We exhaustively study these categories separately and design a growth model to substantiate these categories in the real citation network. Then we leverage this information to develop a stratified learning framework that can predict the number of citations that an article would receive after certain years from its publication. Finally, we design a faceted recommendation system for scientific articles (FeRoSA) that in addition to recommending the relevant scientific papers for a given query paper, would provide the information as to how these recommended papers are related to the query paper. The contributions of this chapter are as follows:

- The categorization of scientific citation profiles provides a set of new approaches to characterize each individual category as well as to study the dynamics of their evolution over time.

- The category information is proved to be remarkably useful in predicting future citation counts within a stratified learning model where we first divide the training samples into different strata and systematically use these strata for predicting future citation count of an article.

- We introduce a bunch of features in the task of future citation prediction. We observe that author-centric features are the most distinguishing ones; among these, average productivity of authors seems to make a paper attractive.

- We further show that adding the citation counts accumulated within the first year after publication as a feature can improve the prediction accuracy.

- The idea of stratification is also used in the task of designing faceted recommendation system where we divide the dataset into four facets and conduct the random walks with restarts separately for the different facets. To the best of our knowledge this is the first recommendation system for scientific papers where the recommendations are further divided into different facets depending of the semantic relation to the query paper.

- FeRoSA achieves a reasonably high precision for the query papers with low citations and low cosine similarity, thus indicating the robustness of the proposed framework.

- FeRoSA is designed to be lightweight, so that it can easily be deployed as an online system.

7.2 Future Direction

In this section, we discuss several new avenues of research that have been opened up by this thesis.

7.2.1 Constant Communities in Networks

Future directions of this works are mentioned as follows:

- Most of the experiments conducted in this chapter focused solely on agglomerative modularity maximization methods. We plan to continue our studies on the effect of vertex perturbations on other types of community detection algorithms such as divisive and spectral methods as well as different optimization objectives.
- It is important to understand how the randomness of a network in the community assignment could be quantified in order to develop algorithms that take into account the variation in randomness for determining the quality of the communities.
- Most importantly, we would like to develop an automated algorithm that can detect such constant communities from a network.

7.2.2 Permanence and Network Communities

From this chapter, several interesting extensions are possible.

- Since Perm and OPerm are vertex-centric metric, we plan to use these metrics for large networks whose complete information is missing. In this direction, we would also like to detect meaningful communities from noisy incomplete networks.
- We plan to extend these metrics for dynamic and weighted networks. We believe that this metric will help in formulating a strong theoretical foundation for identifying community structures where the ground-truth is not known.
- We showed that the layered structure of a community is nicely revealed through the value of OPerm. Moreover, these values provide a ranking of vertices within a community, which can be leveraged in different applications, such as initiator selection during message spreading. Therefore, another direction of research could be to have a deeper understanding of this layered structure and to apply the proposed metrics in several other applications.

7.2.3 Analyzing Ground-truth Communities

The interesting future research agenda that can be enumerated from this chapter are as follows.

- The present empirical study marks the foundation for the design and implementation of a specialized recommendation engine that would be capable of answering search queries pertaining to the (a) impact of papers/authors, (b) field at the forefront (currently and in the near future), (c) seminal papers within a field and many such other factors. These results can be useful for (i) the funding agencies to make appropriate decisions as to how to distribute project funds, (ii) the universities in their faculty recruitment procedure.
- To prove the robustness of the proposed metrics for measuring the interdisciplinarity of a research field, we would like to apply the set of metrics to other domains such as physics and biology.
- Finally, we would like to explain how the global dynamics of scientific paradigm shift influences a researcher's career and vice versa.

7.2.4 Community-based Applications

The possible future agenda that can be formulated from this chapter are as follows.

- The categorization of citation profiles offers a necessary first step towards reformulating the existing quantifiers available in Scientometrics that should leverage the signature of different citation patterns in order to formulate robust measures.
- We plan to extend our studies on the datasets of other domains such as physics and biology to verify the universality of such categorizations.
- We are keen to understand the micro-level dynamics controlling the behavior of PeakMul category which is significantly different from the others. In future, we

would like to conduct a detailed analysis to understand different characteristic features particularly for the PeakMul category.

- Regarding the task of future citation count prediction, we plan to extend this work by looking into different research fields separately.

- We plan to further explore new features that can provide additional signals not captured by the features used in this study. We suspect that the content features seem to provide weak signals because of the coarse representation of the content in terms of topic modeling. A more sophisticated and systematic mining of meaningful features from the content is an immediate future task.

- We also intend to investigate whether similar techniques could be used to predict the scholarly impact of higher level entities (e.g., researchers and universities).

- Regarding FeRoSA, we are interested in the design aspects related to the ergonomics of the user interface so that it can significantly reduce user's cognitive overload, while providing high user satisfaction at the same time.

- In general, the framework used in FeRoSA can be used to provide faceted recommendations for items such as movies, books, videos etc.

Bibliography

[1] 10th DIMACS Implementation Challenge - Graph Partitioning and Graph Clustering. http://www.cc.gatech.edu/dimacs10/archive/clustering.shtml (12.01.2012).

[2] G. Agarwal and D. Kempe. Modularity-maximizing graph communities via mathematical programming. *The European Physical Journal B*, 66(3):409–418.

[3] Y.-Y. Ahn, J. P. Bagrow, and S. Lehmann. Link communities reveal multiscale complexity in networks. *Nature*, 466:761–764, 2010.

[4] R. Albert, H. Jeong, and A. Barabási. Internet: Diameter of the world-wide web. *Nature*, 401(6749):130–131, 1999.

[5] S. Albert, B. Ashforth, and J. Dutton. Organizational identity and identification: Charting new waters and building new bridges. *Academy of Management Review*, 25:13–17, 2000.

[6] L. A. Amaral, A. Scala, M. Barthelemy, and H. E. Stanley. Classes of small-world networks. *Proceedings of the National Academy of Sciences of the United States of America*, 97(21):11149–11152, October 2000.

[7] A. Arenas, A. Fernandez, S. Fortunato, and S. Gomez. Motif-based communities in complex networks. *Journal of Physics A*, 41(22):224001, Sept. 2008.

[8] A. Arenas, A. Fernández, and S. Gómez. Analysis of the structure of complex networks at different resolution levels. *New Journal of Physics*, 10(5):053039, 2008.

[9] S. Asur, S. Parthasarathy, and D. Ucar. An event-based framework for characterizing the evolutionary behavior of interaction graphs. *ACM Trans. Knowl. Discov. Data*, 3(4):16:1–16:36, Dec. 2009.

[10] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan. Group formation in large social networks: Membership, growth, and evolution. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pages 44–54, New York, NY, USA, 2006. ACM.

[11] L. Backstrom and J. Leskovec. Supervised random walks: predicting and recommending links in social networks. In *WSDM*, pages 635–644, 2011.

[12] D. Baird and R. E. Ulanowicz. The Seasonal Dynamics of The Chesapeake Bay Ecosystem. *Ecol. Monogr.*, 59(4):329–364, 1989.

[13] A.-L. Barabasi and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[14] M. J. Barber. Modularity and community detection in bipartite networks. *Physical Review E*, 76(6), 2007.

[15] E. R. Barnes. An algorithm for partitioning the nodes of a graph. Technical Report RC 08690, IBM US Research Centers (Yorktown,San Jose,Almaden, US), 1981.

[16] J. Baumes, M. K. Goldberg, M. S. Krishnamoorthy, M. Magdon-Ismail, and N. Preston. Finding communities by clustering a graph into overlapping subgraphs. In *IADIS AC*, pages 97–104. IADIS, 2005.

[17] V. Belak, S. Lam, and C. Hayes. Towards maximising cross-community information diffusion. In *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 171–178, Aug 2012.

[18] T. Y. Berger-Wolf and J. Saia. A framework for analysis of dynamic social networks. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pages 523–528, New York, NY, USA, 2006. ACM.

[19] J. W. Berry, B. Hendrickson, R. A. LaViolette, V. J. Leung, and C. A. Phillips. *eprint arXiv:0710.3800*, 2007.

[20] J. W. Berry, B. Hendrickson, R. A. LaViolette, and C. A. Phillips. Tolerating the community detection resolution limit with edge weighting. *Physical Review E*, 83(5):056119, May 2011.

[21] S. Bethard and D. Jurafsky. Who should i cite: learning literature search models from citation behavior. In *CIKM*, pages 609–618, New York, NY, USA, 2010. ACM.

[22] J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA, 1981.

[23] G. Bianconi, P. Pin, and M. Marsili. Assessing the relevance of node features for network structure. *Proceedings of the National Academy of Sciences*, 106(28):11433–11438, 2009.

[24] D. M. Blei and J. D. Lafferty. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, ICML '06, pages 113–120, New York, USA, 2006. ACM.

[25] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, Mar. 2003.

[26] V. D. Blondel, J. L. Guillaume, and E. Lambiotte, R.and Lefebvre. Fast unfolding of community hierarchies in large networks. *J. Stat. Mech.*, abs/0803.0476:P10008, 2008.

[27] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *J. Stat. Mech.*, 2008(10):P10008, Oct. 2008.

[28] S. Boettcher and A. G. Percus. Optimization with extremal dynamics. *Complex.*, 8(2):57–62, Nov. 2002.

[29] B. Bollobás. *Modern Graph Theory*. Graduate texts in mathematics. Springer, Heidelberg, corrected edition, 1998.

[30] L. Bornmann and H.-D. Daniel. What do citation counts measure? A review of studies on citing behavior. *Journal of Documentation*, 64(1):45–80, 2008.

[31] U. Brandes, M. Gaertler, and D. Wagner. Experiments on graph clustering algorithms. In G. D. Battista and U. Zwick, editors, *ESA*, volume 2832 of *Lecture Notes in Computer Science*, pages 568–579. Springer, 2003.

[32] J. Brett. Job transfer and well-being. *Journal of Applied Psychology*, 67:450–463, 1982.

[33] J. Brett. Job transitions and personal and role development. *Research in personnel and human resources management*, 2(2):155–185, 1984.

[34] F. Breve, L. Zhao, and M. Quiles. Uncovering overlap community structure in complex networks using particle competition. In *Proceedings of the International Conference on Artificial Intelligence and Computational Intelligence*, AICI '09, pages 619–628, Berlin, Heidelberg, 2009. Springer-Verlag.

[35] S. Carmi, S. Havlin, S. Kirkpatrick, Y. Shavitt, and E. Shir. A model of Internet topology using k-shell decomposition. *PNAS*, 104(27):11150–11154, July 2007.

[36] C. Castillo, D. Donato, and A. Gionis. Estimating the number of citations using author reputation. In *SPIRE*, volume 4726 of *LNCS*, pages 107–117, 2007.

[37] R. Cazabet, F. Amblard, and C. Hanachi. Detection of overlapping communities in dynamical social networks. In *2010 IEEE Second International Conference on Social Computing (SocialCom)*, pages 309–314, Aug 2010.

[38] D. Chakrabarti. Autopart: Parameter-free graph partitioning and outlier detection. In J.-F. Boulicaut, F. Esposito, F. Giannotti, and D. Pedreschi, editors, *PKDD*, volume 3202 of *Lecture Notes in Computer Science*, pages 112–124. Springer, 2004.

[39] D. Chen, M. Shang, Z. Lv, and Y. Fu. Detecting overlapping communities of weighted networks via a local algorithm. *Physica A: Statistical Mechanics and its Applications*, 389(19):4177–4187, 2010.

[40] J. Chen, H. Zhang, Z.-H. Guan, and T. Li. Epidemic spreading on networks with overlapping community structure. *Physica A: Statistical Mechanics and its Applications*, 391(4):1848–1854, 2012.

[41] M. Chen, T. Nguyen, and B. Szymanski. A new metric for quality of network community structure. *ASE Human Journal*, 1(4):226–240, 2013.

[42] P. Chen and S. Redner. Community structure of the physical review citation network. *J. Informetrics*, 4(3):278–290, 2010.

[43] W. Chen, Z. Liu, X. Sun, and Y. Wang. A game-theoretic framework to identify overlapping communities in social networks. *Data Min. Knowl. Discov.*, 21(2):224–240, Sept. 2010.

[44] W. Y. C. Chen, A. W. M. Dress, and W. Q. Yu. Community structures of networks. *Mathematics in Computer Science*, 1(3):441–457, 2008.

[45] F. Chierichetti, S. Lattanzi, and A. Panconesi. Rumour spreading and graph conductance. In *SODA*, pages 1657–1663. SIAM, 2010.

[46] X. Chu, J. Guan, Z. Zhang, and S. Zhou. Epidemic spreading in weighted scale-free networks with community structure. *Journal of Statistical Mechanics: Theory and Experiment*, 2009(07):P07043, 2009.

[47] A. Clauset, C. Moore, and M. E. J. Newman. Hierarchical structure and the prediction of missing links in networks. *Nature*, 453:98–101, 2008.

[48] A. Clauset, M. E. J. Newman, , and C. Moore. Finding community structure in very large networks. *Physical Review E*, pages 1– 6, 2004.

[49] L. M. Collins and C. W. Dent. Omega: A general formulation of the rand index of cluster recovery suitable for non-disjoint solutions. *Multivariate Behavioral Research*, 23(2):231–242, 1988.

[50] J. Copic, M. O. Jackson, and A. Kirman. Identifying community structures from network data, 2005.

[51] I. G. Councill, C. L. Giles, and M.-Y. Kan. Parscit: an open-source crf reference string parsing package. In *LREC*, Marrakech, Morocco, 2008.

[52] L. Danon, A. Díaz-Guilera, and A. Arenas. The effect of size heterogeneity on community identification in complex networks. *Journal of Statistical Mechanics*, 2006(11):P11010, 2006.

[53] L. Danon, A. Díaz-Guilera, J. Duch, and A. Arenas. Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(09):P09008, 2005.

[54] F. Ding, Z. Luo, J. Shi, and X. Fang. Overlapping community detection by kernel-based fuzzy affinity propagation. In *Intelligent Systems and Applications (ISA), 2010 2nd International Workshop on*, pages 1–4, May 2010.

[55] W. E. Donath and A. J. Hoffman. Lower bounds for the partitioning of graphs. *IBM J. Res. Dev.*, 17(5):420–425, Sept. 1973.

[56] J. Duch and A. Arenas. Community detection in complex networks using extremal optimization. *Phys. Rev. E*, 72:027104, Aug 2005.

[57] J.-P. Eckmann and E. Moses. Curvature of co-links uncovers hidden thematic layers in the world wide web. *Proceedings of the National Academy of Sciences*, 99(9):5825–5829, 2002.

[58] Y.-H. Eom and S. Fortunato. Characterizing and modeling citation dynamics. *PLoS ONE*, 6(9):e24926, 09 2011.

[59] G. Eugene. Citation indexing for studying science. *Nature*, 227(5259):669–671, 1970.

[60] G. Eugene. Citation analysis of sports medicine research, 1981-1996: Productivity, impact and influence of nations, institutions and researchers. `http://www.garfield.library.upenn.edu/papers/sportsmed.html`, 1997.

[61] T. S. Evans. Clique graphs and overlapping communities. *CoRR*, abs/1009.0638, 2010.

[62] R. Evered and M. R. Louis. Alternative perspectives in the organizational sciences: "Inquiry from the inside" and "inquiry from the outside.". *Academy of Management Review*, 6:385–395, 1981.

[63] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the internet topology. *SIGCOMM Comput. Commun. Rev.*, 29(4):251–262, Aug. 1999.

[64] I. Farkas, D. Ábel, G. Palla, and T. Vicsek. Weighted network modules. *New Journal of Physics*, 9(6):180, 2007.

[65] M. Fatemi and L. Tokarchuk. A community based social recommender system for individuals amp; groups. In *2013 International Conference on Social Computing (SocialCom)*, pages 351–356, Sept 2013.

[66] S. Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75 – 174, 2010.

[67] S. Fortunato and M. Barthelemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, Jan. 2007.

[68] S. Fortunato and M. Barthelemy. Resolution limit in community detection. *PNAS*, Jan. 2007.

[69] S. Fortunato and A. Lancichinetti. Community detection algorithms: A comparative analysis. In *Proceedings of the Fourth International ICST Conference on Performance Evaluation Methodologies and Tools*, VALUETOOLS '09, pages 27:1–27:2, ICST, Brussels, Belgium, Belgium, 2009. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[70] S. Fortunato and A. Lancichinetti. Community detection algorithms: A comparative analysis: Invited presentation, extended abstract. In *Proceedings of the Fourth International ICST Conference on Performance Evaluation Methodologies and Tools*, VALUETOOLS '09, pages 27:1–27:2, ICST, Brussels, Belgium, Belgium, 2009. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[71] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315:972–976, 2007.

[72] Q. Fu and A. Banerjee. Multiplicative mixture models for overlapping clustering. In *Eighth IEEE International Conference on Data Mining (ICDM)*, pages 791–796, Dec 2008.

[73] M. Gaertler, R. Görke, and D. Wagner. Significance-driven graph clustering. In *Proceedings of the 3rd International Conference on Algorithmic Aspects in Information and Management*, AAIM '07, pages 11–26, Berlin, Heidelberg, 2007. Springer-Verlag.

[74] E. Garfield. Impact factors, and why they won't go away. *Nature*, 411(6837), 2001.

[75] E. Garfield, I. H. Sher, and R. J. Torpie. *The Use of Citation Data in Writing the History of Science*. Institute for Scientific Information Inc., 1984.

[76] D. Gfeller, J.-C. C. Chappelier, and P. De Los Rios. Finding instabilities in the community structure of complex networks. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 72(5 Pt 2), Nov. 2005.

[77] S. Ghosh, A. Banerjee, N. Sharma, S. Agarwal, and N. Ganguly. Statistical analysis of the indian railway network: a complex network approach. *Acta Physica Polonica B Proceedings Supplement*, 4:123–137, 2011.

[78] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *PNAS*, 99(12):7821–7826, June 2002.

[79] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

[80] P. Gleiser and L. Danon. Jazz musicians network: List of edges of the network of jazz musicians. *Adv. Complex Syst*, 6(565):016118, July 2003.

[81] B. Good, Y. D. Montjoye, and A. Clauset. Performance of modularity maximization in practical contexts. *Physical Review E*, 81(4):046106, 2010.

[82] D. Greene, D. Doyle, and P. Cunningham. Tracking the evolution of communities in dynamic social networks. In *2010 International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 176–183, Aug 2010.

[83] S. Gregory. An algorithm to find overlapping community structure in networks. In *Proceedings of the 11th European Conference on Principles and Practice of Knowledge Discovery in Databases*, PKDD 2007, pages 91–102, Berlin, Heidelberg, 2007. Springer-Verlag.

[84] S. Gregory. Finding overlapping communities using disjoint community detection algorithms. In S. Fortunato, G. Mangioni, R. Menezes, and V. Nicosia, editors, *Complex Networks*, volume 207 of *Studies in Computational Intelligence*, pages 47–61. Springer, Berlin / Heidelberg, 2009.

[85] S. Gregory. Finding overlapping communities in networks by label propagation. *New J. Phys.*, 12(10):103018, 2010.

[86] R. Guimerà, L. Danon, A. Daz-Guilera, F. Giralt, and A. Arenas. Self-similar community structure in a network of human interactions. *Physical Review*, E 68(065103), 2003.

[87] R. Guimera, M. Sales-Pardo, and L. Amaral. Modularity from fluctuations in random graphs and complex networks. *Physical Review E*, 70(2):025101, 2004.

[88] R. Guimerà, M. Sales-Pardo, and L. Amaral. A network-based method for target selection in metabolic networks. *Bioinformatics*, 23(13):1616–1622, July 2007.

[89] K. C. H. Shen, X. Cheng and M. B. Hu. Detect overlapping and hierarchical community structure in networks. *Physica A*, 388(8):1706–1712, 2009.

[90] S. Harenberg, G. Bello, L. Gjeltema, S. Ranshous, J. Harlalka, R. Seay, K. Padmanabhan, and N. Samatova. Community detection in large-scale networks: a survey and empirical evaluation. *Wiley Interdisciplinary Reviews: Computational Statistics*, 6(6):426–439, 2014.

[91] G. Haro, G. Randall, and G. Sapiro. Stratification learning: Detecting mixed density and dimensionality in high dimensional point clouds. In *NIPS*, pages 553–560. 2006.

[92] J. Harris, J. Hirst, and M. Mossinghoff. *Combinatorics and Graph Theory*. Springer Undergraduate Texts in Mathematics and Technology. Springer, 2008.

[93] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

[94] M. B. Hastings. Community detection as an inference problem. *Phys. Rev. E*, 74:035102, Sep 2006.

[95] F. Havemann, M. H. 0003, A. Struck, and J. Gläser. Identification of overlapping communities and their hierarchy by locally calculating community-changing resolution levels. *CoRR*, abs/1012.1269, 2010.

[96] D. He, D. Liu, W. Zhang, D. Jin, and B. Yang. Discovering link communities in complex networks by exploiting link dynamics. *CoRR*, abs/1303.4699, 2013.

[97] Q. He, D. Kifer, J. Pei, P. Mitra, and C. L. Giles. Citation recommendation without author supervision. In *WSDM*, pages 755–764, 2011.

[98] Q. He, J. Pei, D. Kifer, P. Mitra, and L. Giles. Context-aware citation recommendation. In *WWW*, pages 421–430, USA, 2010. ACM.

[99] J. E. Hirsch. An index to quantify an individual's scientific research output. *PNAS*, 102(46):16569–16572, 2005.

[100] A. Hlaoui and S. Wang. Median graph computation for graph clustering. *Soft Comput.*, 10(1):47–53, 2006.

[101] J. M. Hofman and C. H. Wiggins. A bayesian approach to network modularity, Sept. 2007.

[102] J. H. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. MIT Press, Cambridge, MA, USA, 1992.

[103] Y. Hu, H. Chen, P. Zhang, M. Li, Z. Di, and Y. Fan. Comparative definition of community and corresponding identifying algorithm. *Phys. Rev. E*, 78:026121, Aug 2008.

[104] Z. Hu, C. Chen, and Z. Liu. Where are citations located in the body of scientific articles? a study of the distributions of citation locations. *Journal of Informetrics*, 7(4):887–896, 2013.

[105] L. Hubert and P. Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

[106] B. Hughes. *Random Walks and Random Environments: Random walks*. Number v. 1 in Oxford science publications. Clarendon Press, 1995.

[107] H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, and A. L. BarabÃąsi. The large-scale organization of metabolic networks. *Nature*, 407(6804):651–654, October 2000.

[108] J. Kamahara, T. Asakawa, S. Shimojo, and H. Miyahara. A community-based recommendation system to reveal unexpected interests. In *Multimedia Modelling Conference, 2005. MMM 2005. Proceedings of the 11th International*, pages 433–438, Jan. 2005.

[109] B. W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *Bell Sys. Tech. J.*, 49(2):291–308, 1970.

[110] Y. Kim and H. Jeong. The map equation for link community. *CoRR*, abs/1105.0257, 2011.

[111] Y. Kim, S. W. Son, and H. Jeong. Link Rank: Finding communities in directed networks. Technical Report arXiv:0902.3728.

[112] M. Kimura, K. Yamakawa, K. Saito, and H. Motoda. Community analysis of influential nodes for information diffusion on a social network. In *IJCNN*, pages 1358–1363. IEEE, 2008.

[113] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[114] I. A. Kovács, R. Palotai, M. S. Szalay, and P. Csermely. Community landscapes: an integrative approach to determine overlapping network module hierarchy, identify key nodes and predict network dynamics. *PLoS One*, page 12528, 2010.

[115] V. Kreb. Books on us politics`http://www.orgnet.com/`.

[116] O. Küçüktunç, E. Saule, K. Kaya, and Ü. V. Çatalyürek. Diversifying citation recommendations. *CoRR*, abs/1209.5809, 2012.

[117] J. M. Kumpula, M. Kivelä, K. Kaski, and J. Saramäki. Sequential algorithm for fast clique percolation. *Phys. Rev. E*, 78:026109, Aug 2008.

[118] V. Labatut. Generalized measures for the evaluation of community detection methods. *CoRR*, abs/1303.5441, 2013.

[119] R. Lambiotte. Multi-scale modularity in complex networks. In *WiOpt*, pages 546–553. IEEE, 2010.

[120] A. Lancichinetti and S. Fortunato. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Phys. Rev. E*, 80(1):016118, July 2009.

[121] A. Lancichinetti and S. Fortunato. Community detection algorithms: A comparative analysis. *Phys. Rev. E*, 80:056117, Nov 2009.

[122] A. Lancichinetti and S. Fortunato. Limits of modularity maximization in community detection. *Phys. Rev. E*, 84:066122, 2011.

[123] A. Lancichinetti and S. Fortunato. Consensus clustering in complex networks. *Nature Scientific Reports*, 2, 2012.

[124] A. Lancichinetti, S. Fortunato, and J. Kertész. Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*, 11(3):033015, 2009.

[125] A. Lancichinetti, F. Radicchi, J. J. Ramasco, and S. Fortunato. Finding statistically significant communities in networks. *PLoS ONE*, 6(4):e18961, 04 2011.

[126] A. Lancichinetti, F. Radicchi, J. J. Ramasco, and S. Fortunato. Finding statistically significant communities in networks. *PLoS ONE*, 6(4):e18961, 2011.

[127] A. Lázár, D. Ábel, and T. Vicsek. Modularity measure of networks with overlapping communities. *Europhys. Lett.*, 90(1):18001, 2010.

[128] C. Lee, F. Reid, A. Mcdaid, and N. Hurley. Detecting highly overlapping community structure by greedy clique expansion. In *In Proceedings of the 4th Workshop on Social Network Mining and Analysis, (SNA/KDD10)*, pages 33–42, Aug 2010.

[129] C.-P. Lee and C.-J. Lin. Large-scale linear ranksvm. *Neural computation*, 26(4):781–817, 2014.

[130] Y. Lee, Y. Lin, and G. Wahba. Multicategory support vector machines: Theory and application to the classification of microarray data and satellite radiance data. *Journal of the American Statistical Association*, 99(465):67–82, 2004.

[131] S. Lehmann and L. K. Hansen. Deterministic modularity optimization. *The European Physical Journal B*, 60(1):83–88.

[132] E. A. Leicht and M. E. J. Newman. Community structure in directed networks. *Phys. Rev. Lett.*, 100:118703, Mar 2008.

[133] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics*, 6(1):29–123, 2009.

[134] F. Li, J. He, G. Huang, Y. Zhang, and Y. Shi. A clustering-based link prediction method in social networks. *Procedia Computer Science*, 29(0):432–442, 2014. 2014 International Conference on Computational Science.

[135] Y. Liang, Q. Li, and T. Qian. Finding relevant papers based on citation relations. In *Web-Age Information Management*, volume 6897, pages 403–414. Springer, 2011.

[136] Y.-R. Lin, Y. Chi, S. Zhu, H. Sundaram, and B. L. Tseng. Facetnet: A framework for analyzing communities and their evolutions in dynamic networks. In *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, pages 685–694, New York, NY, USA, 2008. ACM.

[137] P. Lisboa, H. AL-Mamory, and B. W. Timproving recommendation systems by modeling the stability of implicit behaviour. In *The Post Graduate Network Symposium (PGNet2013)*, pages 354–361, Dec 2013.

[138] B. Long, Z. M. Zhang, X. Wu, and P. S. Yu. Relational clustering by symmetric convex coding. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 569–576, New York, NY, USA, 2007. ACM.

[139] B. Luëar, Z. Levnajić, J. Povh, and M. Perc. Community structure and the evolution of interdisciplinarity in slovenia's scientific collaboration network. *PLoS ONE*, 9(4):e94429, Dec. 2014.

[140] D. Lusseau, K. Schneider, O. Boisseau, P. Haase, E. Slooten, and S. Dawson. The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations. *Behavioral Ecology and Sociobiology*, 54(4):396–405, 2003.

[141] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In L. M. L. Cam and J. Neyman, editors, *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.

[142] M. Magdon-Ismail and J. T. Purnell. Ssde-cluster: Fast overlapping clustering of networks using sampled spectral distance embedding and gmms. In *SocialCom/PASSAT*, pages 756–759. IEEE, 2011.

[143] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[144] C. P. Massen and J. P. K. Doye. Identifying communities within energy landscapes. *Physical Review E*, 71(046101), 2005.

[145] J. J. McAuley and J. Leskovec. Learning to discover social circles in ego networks. In *NIPS*, pages 548–556, 2012.

[146] A. McDaid and N. Hurley. Detecting highly overlapping communities with model-based overlapping seed expansion. In *ASONAM*, pages 112–119, Washington, DC, USA, 2010.

[147] A. F. McDaid, D. Greene, and N. J. Hurley. Normalized mutual information to evaluate overlapping community finding algorithms. *CoRR*, abs/1110.2515, 2011.

[148] M. Molloy and B. Reed. A critical point for random graphs with a given degree sequence. *Random Struct. Algorithms*, 6(2/3):161–179, Mar. 1995.

[149] F. Morillo, M. Bordons, and I. Gómez. An approach to interdisciplinarity through bibliometric indicators. *Scientometrics*, 51(1):203–222, Apr. 2001.

[150] A. Mukherjee, M. Choudhury, A. Basu, and N. Ganguly. Modeling the co-occurrence principles of the consonant inventories: A complex network approach. *International Journal of Modern Physics C*, 18:281–295, 2008.

[151] C. Musto, P. Lops, F. Narducci, G. Semeraro, and M. D. Gemmis. A tag recommender system exploiting user and community behavior.

[152] B. Nadler and M. Galun. Fundamental limitations of spectral clustering methods. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*. MIT Press, Cambridge, MA, 2007.

[153] H. Nanba and M. Okumura. Towards multi-paper summarization using reference information. In *IJCAI*, pages 926–931. Morgan Kaufmann, 1999.

[154] A. Nematzadeh, E. Ferrara, A. Flammini, and Y.-Y. Ahn. Optimal network modularity for information diffusion. *Phys. Rev. Lett.*, 113:088701, Aug 2014.

[155] T. Nepusz, A. Petróczi, L. Négyessy, and F. Bazsó. Fuzzy communities and the concept of bridgeness in complex networks. E-print, 2007.

[156] M. E. Newman. Modularity and community structure in networks. *PNAS*, 103(23):8577–8582, June 2006.

[157] M. E. J. Newman. The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences of the United States of America*, 98(2):404–409, January 2001.

[158] M. E. J. Newman. Mixing patterns in networks. *Phys. Rev. E*, 67(2):026126, Feb. 2003.

[159] M. E. J. Newman. Analysis of weighted networks. *Phys. Rev. E*, 70(5):056131, Nov. 2004.

[160] M. E. J. Newman. Detecting community structure in networks. *The European Physical Journal B - Condensed Matter and Complex Systems*, 38(2):321–330, Mar. 2004.

[161] M. E. J. Newman. Fast algorithm for detecting community structure in networks. *Phys. Rev. E*, 69:066133, Jun 2004.

[162] M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Phys. Rev. E*, 74:036104, Sep 2006.

[163] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review*, E 69(026113), 2004.

[164] M. E. J. Newman and E. A. Leicht. Mixture models and exploratory analysis in networks. *Proceedings of the National Academy of Sciences*, 104(23):9564–9569, 2007.

[165] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*, pages 849–856. MIT Press, 2001.

[166] N. Nicholson. A Theory of Work Role Transitions. *Administrative Science Quarterly*, 29(2):172–191, 1984.

[167] M. Ovelgónne and A. Geyer-schulz. An ensemble learning strategy for graph clustering. In *10th DIMACS Implementation Challenge Graph Partitioning and Graph Clustering*, 2012.

[168] G. Palla, A.-L. Barabasi, and T. Vicsek. Quantifying social group evolution. *Nature*, 446(7136):664–667, April 2007.

[169] G. Palla, I. Derényi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, June 2005.

[170] G. Palla, I. J. Farkas, P. Pollner, I. Derényi, and T. Vicsek. Fundamental statistical features and self-similar properties of tagged networks. *New J. Phys.*, 10(12):123026, 2008.

[171] J.-Y. Pan, H.-J. Yang, C. Faloutsos, and P. Duygulu. Automatic multimedia cross-modal correlation discovery. In *SIGKDD*, pages 653–658, Seattle, WA, 2004.

[172] S. Papadopoulos, A. Skusa, A. Vakali, Y. Kompatsiaris, and N. Wagner. *eprint arXiv:0902.08*, 2009.

[173] B. Passi and S. Mishra. Selecting Research Areas and Research Design Approaches in Distance Education: Process issues. *The International Review of Research in Open and Distance Learning*, 5(3):329–340, 2004.

[174] M. Pearson. Drifting smoke rings: Social network analysis and markov processes in a longitudinal study of friendship groups and risk-taking, connections, 2003.

[175] R. Pemantle. Vertex reinforced random walk. *Prob. Th. and Rel. Fields*, pages 117–136, 1992.

[176] C. Peterson and J. R. Anderson. A mean field theory learning algorithm for neural networks. *Complex Systems*, 1:995–1019, 1987.

[177] A. M. Pettigrew. Longitudinal field research on change: Theory and practice. *Organization Science Special Issue: Longitudinal Field Research Methods for Studying Processes of Organizational Change*, 1(3):267–292, 1990.

[178] V. Pihur, S. Datta, and S. Datta. Rankaggreg, an r package for weighted rank aggregation. *BMC bioinformatics*, 10(1):62, 2009.

[179] J. W. Pinney and D. R. Westhead. Betweenness-based decomposition methods for social and biological networks. In S. Barber, P. Baxter, K. Mardia, and R. Walls, editors, *Interdisciplinary Statistics and Bioinformatics*, pages 87–90. Leeds University Press, 2007.

[180] P. Pons and M. Latapy. Computing communities in large networks using random walks. In p. Yolum, T. Güngör, F. Gürgen, and C. Özturan, editors, *Computer and Information Sciences - ISCIS 2005*, volume 3733 of *Lecture Notes in Computer Science*, pages 284–293. Springer Berlin Heidelberg.

[181] A. Pothen. Graph partitioning algorithms with applications to scientific computing. Technical report, Norfolk, VA, USA, 1997.

[182] D. Radev, P. Muthukrishnan, V. Qazvinian, and A. Abu-Jbara. The acl anthology network corpus. *LREC*, pages 1–26, 2013.

[183] U. N. Raghavan, R. Albert, and S. Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Phys. Rev. E*, 76:036106, Sep 2007.

[184] W. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.

[185] M. J. Rattigan, M. Maier, and D. Jensen. Graph clustering with network structure indices. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 783–790, New York, NY, USA, 2007. ACM.

[186] J. Reichardt and S. Bornholdt. Detecting fuzzy community structures in complex networks with a potts model. *Phys. Rev. Lett.*, 93(21):218701, Nov. 2004.

[187] J. Reidy, D. A. Bader, K. Jiang, P. Pande, and R. Sharma. Detecting communities from given seeds in social networks. Technical report, Oct. 2011.

[188] W. Ren, G. Yan, X. Liao, and L. Xiao. Simple probabilistic algorithm for detecting community structure. *Phys. Rev. E*, 79:036111, Mar 2009.

[189] M. Rosvall and C. T. Bergstrom. An information-theoretic framework for resolving community structure in complex networks. *Proceedings of the National Academy of Sciences*, 104(18):7327–7331, 2007.

[190] M. Rosvall and C. T. Bergstrom. Maps of information flow reveal community structure in complex networks. In *Proceedings of the National Academy of Sciences USA*, pages 1118–1123, 2007.

[191] M. Rosvall and C. T. Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4):1118–1123, 2008.

[192] M. Sachan and R. Ichise. Using abstract information and community alignment information for link prediction. In *2010 Second International Conference on Machine Learning and Computing (ICMLC)*, pages 61–65, Feb 2010.

[193] P. Schuetz and A. Caflisch. Efficient modularity optimization by multistep greedy algorithm and vertex mover refinement. *Physical Review E*, 77(4):046112, 2008.

[194] M. Seifi, I. Junier, J.-B. Rouquier, S. Iskrov, and J.-L. Guillaume. Stable community cores in complex networks. 424:87–98.

[195] J. Shang, L. Liu, F. Xie, and C. Wu. How overlapping community structure affects epidemic spreading in complex networks. In *2014 IEEE 38th International Computer Software and Applications Conference Workshops (COMPSACW)*, pages 240–245, July 2014.

[196] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, 1948.

[197] H. Shen, X. Cheng, K. Cai, and M.-B. Hu. Detect overlapping and hierarchical community structure in networks. *Physica A: Statistical Mechanics and its Applications*, 388(8):1706–1712, 2009.

[198] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, Aug. 2000.

[199] X. Shi, B. L. Tseng, and L. A. Adamic. Information diffusion in computer science citation networks. In *ICWSM*, pages 319–322, 2009.

[200] A. J. Smola and B. Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, Aug. 2004.

[201] S.-W. Son, H. Jeong, and J. D. Noh. Random field ising model and community structure in complex networks. *The European Physical Journal B - Condensed Matter and Complex Systems*, 50(3):431–437.

[202] M. Spiliopoulou, I. Ntoutsi, Y. Theodoridis, and R. Schult. Monic: Modeling and monitoring cluster transitions. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pages 706–711, New York, NY, USA, 2006. ACM.

[203] M. Spiliopoulou, I. Ntoutsi, Y. Theodoridis, and R. Schult. Monic: modeling and monitoring cluster transitions. In *Proceedings of the 12th ACM SIGKDD*, pages 706–711, New York, USA, 2006.

[204] K. Steinhaeuser and N. V. Chawla. Identifying and evaluating community structure in complex networks. *Pattern Recognition Letters*, 31(5):413–421, 2010.

[205] A. Strehl and J. Ghosh. Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.*, 3:583–617, Mar. 2003.

[206] J. Sun, C. Faloutsos, S. Papadimitriou, and P. S. Yu. Graphscope: Parameter-free mining of large time-evolving graphs. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, pages 687–696, New York, NY, USA, 2007. ACM.

[207] S. R. Sundaresan, I. R. Fischhoff, J. Dushoff, and D. I. Rubenstein. Network metrics reveal differences in social organization between two fission–fusion species, Grevy's zebra and onager. *Oecologia*, 151(1):140–149, Feb. 2007.

[208] C. Tantipathananandh and T. Berger-Wolf. Constant-factor approximation algorithms for identifying dynamic communities. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 827–836, New York, NY, USA, 2009. ACM.

[209] C. Tantipathananandh, T. Berger-Wolf, and D. Kempe. A framework for community identification in dynamic social networks. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, pages 717–726, New York, NY, USA, 2007. ACM.

[210] G. Tibély and J. Kertész. On the equivalence of the label propagation method of community detection and a potts model approach. *Physica A: Statistical Mechanics and its Applications*, 387(19–20):4982–4984, 2008.

[211] J. R. Tyler, D. M. Wilkinson, and B. A. Huberman. E-mail as a spectroscopy: Automated discovery of community structure within organizations. In M. Huysman, E. Wenger, and V. Wulfs, editors, *Proceedings of the First Iternational Conference on Communities and Technologies*, 2003.

[212] J. Valverde-Rebaza and A. de Andrade Lopes. Structural link prediction using community information on twitter. In *2012 Fourth International Conference on Computational Aspects of Social Networks (CASoN)*, pages 132–137, Nov 2012.

[213] J. C. Valverde-Rebaza and A. A. Lopes. Link prediction in online social networks using group information. volume 8584, pages 31 – 45, Portugal, 2014. Springer.

[214] S. van Dongen. *Graph Clustering by Flow Simulation*. PhD thesis, University of Utrecht, 2000.

[215] N. X. Vinh, J. Epps, and J. Bailey. Information theoretic measures for clusterings comparison: Is a correction for chance necessary? In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 1073–1080, New York, NY, USA, 2009. ACM.

[216] C. S. Wagner, J. D. Roessner, K. Bobb, J. T. Klein, K. W. Boyack, J. Keyton, I. Rafols, and K. Börner. Approaches to understanding and measuring interdisciplinary scientific research (idr): A review of the literature. *J. Informetrics*, 5(1):14–26, 2011.

[217] K. Wakita and T. Tsurumi. Finding community structure in mega-scale social networks: [extended abstract]. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 1275–1276, New York, NY, USA, 2007. ACM.

[218] L. Waltman, N. J. van Eck, and E. C. M. Noyons. A unified approach to mapping and clustering of bibliometric networks. *J. Informetrics*, 4(4):629–635, 2010.

[219] X. Wang, L. Tang, H. Liu, and L. Wang. Learning with multi-resolution overlapping communities. *Knowledge and Information Systems (KAIS)*, 2012.

[220] D. Watts and S. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, (393):440–442, 1998.

[221] Y.-C. Wei and C.-K. Cheng. Towards efficient hierarchical designs by ratio cut partitioning. In *Computer-Aided Design, 1989. ICCAD-89. Digest of Technical Papers., 1989 IEEE International Conference on*, pages 298–301. IEEE, Nov. 1989.

[222] L. Weng, F. Menczer, and Y.-Y. Ahn. Virality prediction and community structure in social networks. *Sci. Rep.*, 3(2522), 2013.

[223] R. Winkler. *An Introduction to Bayesian Inference and Decision*. International series in decision processes. Holt, Rinehart and Winston, 1972.

[224] F. Y. Wu. The potts model. *Rev. Mod. Phys.*, 54(1):235–268, Jan. 1982.

[225] Z. Wu, Y. Lin, H. Wan, and S. Tian. A fast and reasonable method for community detection with adjustable extent of overlapping. In *2010 International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*, pages 376–379, Nov 2010.

[226] J. Xie, S. Kelley, and B. K. Szymanski. Overlapping community detection in networks: The state-of-the-art and comparative study. *ACM Comput. Surv.*, 45(4):43:1–43:35, Aug. 2013.

[227] J. Xie, B. Szymanski, and X. Liu. Slpa: Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process. In *2011 IEEE 11th International Conference on Data Mining Workshops (ICDMW)*, pages 344–349, Dec 2011.

[228] J. Xie and B. K. Szymanski. Towards linear time overlapping community detection in social networks. In *PAKDD*, pages 25–36, 2012.

[229] J. Xie, B. K. Szymanski, and X. Liu. SLPA: Uncovering Overlapping Communities in Social Networks via a Speaker-Listener Interaction Dynamic Process. In *ICDM Workshops*, pages 344–349, 2011.

[230] R. Yan, J. Tang, X. Liu, D. Shan, and X. Li. Citation count prediction: Learning to estimate future citations for literature. In *CIKM*, pages 1247–1252, New York, USA, 2011.

[231] J. Yang and J. Leskovec. Defining and evaluating network communities based on ground-truth. In *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics*, MDS '12, pages 3:1–3:8, New York, USA, 2012.

[232] J. Yang and J. Leskovec. Overlapping community detection at scale: a nonnegative matrix factorization approach. In *WSDM*, pages 587–596, 2013.

[233] J. Yang and J. Leskovec. Overlapping communities explain core-periphery organization of networks. *Proceedings of IEEE*, 102:1892 – 1902, 2014.

[234] M. Zarei, D. Izadi, and K. A. Samani. Detecting overlapping community structure of networks based on vertex–vertex correlations. *Journal of Statistical Mechanics: Theory and Experiment*, 2009(11):P11013, 2009.

[235] M. Zarei and K. A. Samani. Eigenvectors of network complement reveal community structure more accurately. *Physica A: Statistical Mechanics and its Applications*, 388(8):1721–1730, 2009.

[236] S. Zhang, R.-S. Wang, and X.-S. Zhang. Identification of overlapping community structure in complex networks using fuzzy -means clustering. *Physica A: Statistical Mechanics and its Applications*, 374(1):483–490, 2007.

[237] S. Zhang, R.-S. Wang, and X.-S. Zhang. Uncovering fuzzy community structure in complex networks. *Phys. Rev. E*, 76:046103, Oct 2007.

[238] Y. Zhang, J. Wang, Y. Wang, and L. Zhou. Parallel community detection on large networks with propinquity dynamics. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 997–1006, New York, NY, USA, 2009. ACM.

[239] H. Zhou. Distance, dissimilarity index, and network community structure. *Physical Review E*, 67(6):061901, 2003.

[240] L. Zhuhadar, R. Yang, and O. Nasraoui. Toward the design of a recommender system: Visual clustering and detecting community structure in a web usage network. In *2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, volume 1, pages 354–361, Dec 2012.

Appendix A

List of all Publications by the Candidate

Following is a list of all the publications by the candidate including those on and related to the work presented in the thesis. The publications are arranged in chronological order and in four sections – (i) book chapter, (ii) journals, (iii) conferences, and (iv) workshops/research colloquiums.

Book Chapter

1. S. Srinivasan, **T. Chakraborty**, and S. Bhowmick. "Identifying Base Clusters and Their Application to Maximizing Modularity". *Contemporary Mathematics. Graph partitioning and Graph Clustering. (D. A. Bader, H. Meyerhenke, P. Sanders and D. Wagner eds.), AMS-DIMACS*, pp. 141-156, 2012.

Journals

1. **T. Chakraborty**, N. Ganguly, A. Mukherjee, S. Bhowmick. "Overlapping Permanence: A New Vertex-based Metric to Analyze Overlapping Communities" (communicated to *IEEE TKDE*).

2. **T. Chakraborty**, A. Krishna, M. Singh, N. Ganguly, P. Goyal, A. Mukherjee. "FeRoSA: A Faceted Recommendation System for Scientific Articles" (communicated to *ACM TIST*).

3. **T. Chakraborty**, R. Narayanam. DiSCern: A Trade-off between Relevance and Diversity in Scientific Paper Recommendation (communicated to *DMKD* journal).

4. **T. Chakraborty**, N. Ganguly, A. Mukherjee, S. Bhowmick. “Permanence and Community Analysis in Complex Networks” (communicated to *ACM TKDD* journal).

5. **Tanmoy Chakraborty**. Leveraging disjoint communities for detecting overlapping community structure, *Journal of Statistical Mechanics: Theory and Experiment* (*JSTAT*), 5, ISSN 1742-5468, pp. P05017, May 2015.

6. **Tanmoy Chakraborty**, Niloy Ganguly and Animesh Mukherjee. An author is known by the context she keeps: significance of network motifs in scientific collaborations, *Social Network Analysis and Mining* (*SNAM*), 5:16, Springer Vienna, ISSN 1869-5450, pp. 1-21, May 2015.

7. **Tanmoy Chakraborty**, Suhansanu Kumar, Pawan Goyal, Niloy Ganguly, Animesh Mukherjee. On the categorization of scientific citation profiles in computer sciences, *Communications of the ACM* (CACM), 58: 9, ISSN 0001-0782, pp. 82-90.

8. **T. Chakraborty**, V. Tammana, N. Ganguly, A. Mukherjee. “Understanding and Modeling Diverse Scientific Careers of Researchers”, *Journal of Informetrics*, 9:1, ISSN 1751-1577, pp. 69-78, Jan 2015.

9. **T. Chakraborty**, S. Srinivasan, N. Ganguly, S. Bhowmick, A. Mukherjee. “Constant Communities in Complex Networks”, *Nature Scientific Reports 3*, 1825, ISSN 2045-2322, 2013.

10. **T. Chakraborty**, S. Sikdar, N. Ganguly and A. Mukherjee. “Citation Interactions among Computer Science Fields: A Quantitative Route to the Rise and Fall of scientific Research”, *Social Network Analysis and Mining* (*SNAM*), 4:1, Springer Vienna, ISSN 1869-5450, pp. 1-18, 2014.

Conferences

1. M. Singh, V. Patidar, S. Kumar, **T. Chakraborty**, A. Mukherjee, P. Goyal. The role of citation context in predicting long-term citation profiles: an experimental study based on a massive bibliographic text dataset, In *24th ACM International Conference on Information and Knowledge Management (CIKM 2015)*, Melbourne, Australia, October 19-23, 2015. (Accepted)

2. **T. Chakraborty**, S. Patranabis, P. Goyal, A. Mukherjee. “On the formation circles in co-authorship networks”. In *Proceedings of 21th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Sydney, August 10 - 13, 2015, pp. 109-118.

3. M. Singh, **T. Chakraborty**, P. Goyal and A. Mukherjee. “ConfAssist: A Conflict resolution framework for assisting the categorization of Computer Science conferences”. In *Joint Conference on Digital Libraries* (*JCDL*), Tennessee, USA, June 21 -25, 2015, pp. 257-258.

4. **T. Chakraborty**, N. Modani, R. Narayanam, S. Nagar. “DiSCern: A Diversified Citation Recommendation System for Scientific Queries”, In *31st IEEE International Conference on Data Engineering* (*ICDE*), Seoul, Korea, April 13-17, 2015, pp. 555-566.

5. **T. Chakraborty**, N. Ganguly, A. Mukherjee. “Automatic Classification of Scientific Groups as Productive: An Approach based on Motif Analysis”, In *Proceedings of IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining* (*ASONAM*), Beijing, China August 17-20, 2014, pp. 130-137.

6. **T. Chakraborty**, S. Kumar, P. Goyal, N. Ganguly, A. Mukherjee. “Towards a Stratified Learning Approach to Predict Future Citation Counts”, In *Proceedings of ACM/IEEE Digital Libraries* (jointly with *JCDL* and *TPDL*), London, United Kingdom, September 8-12, 2014, pp. 351-360.

7. **T. Chakraborty**, S. Srinivasan, N. Ganguly, A. Mukherjee, S. Bhowmick. “On the permanence of vertices in network communities”, In *Proceedings of 20th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, New York City, August 24 - 27, 2014, pp. 1396-1405.

8. **T. Chakraborty**, V. Tammana, N. Ganguly, A. Mukherjee. “Analysis and Modeling of Lowest Unique Bid Auctions”, In *Proceedings of Sixth ASE International Conference on Social Computing* (*SocialCom*), Stanford, CA, USA, May 27-31, 2014.

9. **T. Chakraborty**, S. Kumar, M. D. Reddy, S. Kumar, N. Ganguly, A. Mukherjee. “Automatic Classification and Analysis of Interdisciplinary Fields in Computer Sciences”, In *Proceedings of 2013 ASE/IEEE International Conference on Social Computing* (*SocialCom*), Washington D.C., USA, September 8- 14, 2013, pp. 180 - 187.

10. **T. Chakraborty**, S. Sikdar, V. Tammana, N. Ganguly, A. Mukherjee. “Computer Science Fields as Ground-truth Communities: Their Impact, Rise and Fall”, In *Proceedings of IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining* (*ASONAM*), Niagara Falls, Canada, August 25-28, 2013, pp. 426-433.

11. **T. Chakraborty**, A. Chakraborty. “OverCite: Finding Overlapping Communities in Citation Network”, In *Proceedings of IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining* (*ASONAM*), Niagara Falls, Canada, August 25-28, 2013, pp. 1124 - 1131.

Workshops/ Research Colloquium

1. **T. Chakraborty**, S. Srinivasan, N. Ganguly, A. Mukherjee, S. Bhowmick. “On the permanence of vertices in network communities”, In *Microsoft TechVista-2015, Microsoft Research India’s annual research symposium*, Bangalore, India, January 23, 2015. **(Received first prize)**

2. M. Singh, S. Pramanik, **T. Chakraborty**. PubIndia: A Framework for Analyzing Indian Research Publications in Computer Sciences, In *XRCI Open Research Symposium*, Bangalore, India, January 22-23, 2015.

3. **T. Chakraborty**, N. Ganguly, A. Mukherjee. Rising Popularity of Interdisciplinary Research - an Analysis of Citation Networks, *Workshop on Science and Engineering of Social Networks, 6th International Conference on Communication System and Networks* (*COMSNETS)*, Bangalore, India, January 10, 2014. **(Best presentation award)**