SLANT+: A Nonlinear Model for Opinion Dynamics in Social Networks

Bhushan Kulkarni*, Sumit Agarwal*, Abir De* , Sourangshu Bhattacharya, Niloy Ganguly
Indian Institute of Technology Kharagpur, India
{kbsanjay, sumita, abir.de,sourangshu,niloy}@cse.iitkgp.ernet.in

***Abstract*—Online Social Networks (OSNs) have emerged as a global media for forming and shaping opinions on a broad spectrum of topics like politics, e-commerce, sports, etc. So, research on understanding and predicting opinion dynamics in OSNs, especially using a tractable linear model, has abound in literature. However, these linear models are too simple to uncover the actual complex dynamics of opinion flow in social networks. In this paper, we propose SLANT+, a novel nonlinear generative model for opinion dynamics, by extending our earlier linear opinion model SLANT [7]. To design this model, we rely on a network-guided recurrent neural network architecture which learns a proper temporal representation of the messages as well as the underlying network. Furthermore, we probe various signals from the real life datasets and offer a conceptually interpretable nonlinear function that not only provides concrete clues of the opinion exchange process, but also captures the coupled dynamics of message timings and opinion flow. As a result, with five real-life datasets crawled from Twitter, our proposal gives significant accuracy boost over six state-of-the-art baselines.**

I. Introduction

The social media and social networking sites often play a vital role in forming and shaping the users' opinion. In fact, in recent days, various agencies routinely use social media to probe people's opinion on the issues of interest. Hence, uncovering the dynamics of opinion flow over a network has garnered a lot of interest in recent years [3, 4, 7].

Limitations of Prior Work: Research on opinion dynamics predominantly follows two kinds of models. (i) statistical physics based models and (ii) data-driven models. The first type of models [1, 2, 8, 11, 20] is traditionally designed to capture several regulatory real-life phenomena; e.g. consensus, polarization, etc. However, these models are barely data-driven and therefore their parameters are difficult to learn from fine-grained real data. The second class of models [3, 4, 7] which are surprisingly few, attempts to overcome these limitations by learning a tractable linear model from transient opinion dynamics. Most of these approaches [3, 4] do not consider forecasting opinion at an arbitrary future time-stamp for evaluating the utility of their models. Rather, they focus on nowcasting, i.e. they attempt to predict opinion at the very next time-stamp. Only, a recent approach, SLANT [7] that incorporates the complex stochastic dynamics of the messages, can accurately forecast opinions even at a distant future.

However, all these existing approaches have looked into the opinion dynamics phenomenon through the tinted glass of two restrictive assumptions: (i) linearity of influence between past events and (ii) independent or decoupled dynamics of message timings and opinion flow. Opinion dynamics which is a complex psychological and sociological process is expected to be nonlinear in general. Therefore, while linearity is a useful notion for the model to be explainable and tractable, in reality such an imposition hinders the detection of any possible presence of nonlinearities in data, thereby ignoring the inherent complexity. Although, recently some efforts [9, 17, 19] considered nonlinear modeling of point processes in different contexts, these solutions are far from satisfactory – they don't consider the presence of the underlying network and so cannot be extended in case of opinion dynamics in general. Hence, their proposals cannot be easily extended in the context of opinion dynamics, a scenario largely propelled by the interactions between users of a social network.

Present Work and Roadmap: In this paper, we develop SLANT+, a point process based framework that captures the nonlinear dynamics of opinion propagation. At an outset, this approach extends our previously proposed model SLANT [7] to a *nonlinear joint generative model* of users' opinions and message-timings as a temporal point process that allows each user's hidden opinion (message-rate) to be modulated over time by the opinions (message-timings) of her neighbors. When a user *expresses* her opinion, her neighbors' current opinions are updated– such updates are driven by nonlinear interactions. To capture the nonlinearities, SLANT+ views the intensity function as a nonlinear function of previous opinions and time-stamps. In addition, the event history is embedded into temporal function which can be used for predicting the next message-time and opinion. In contrast to the traditional approaches that consider the intensity functions having fixed parametric forms, our proposal does not assume any restrictive form, rather it aims to learn the form of nonlinearity to uncover the underlying complexity of the process.

In this paper, we design a network-guided recurrent neural network (RNN) architecture to understand and model the nonlinear networked-opinion dynamics. Such a neural architecture is a network of RNN cells, where each RNN

* Authors contributed equally.

cell captures the opinion and message dynamics of corresponding user and the connection between two RNN cells conforms the edge between corresponding users. Each RNN cell receives input from the current events of the neighbors as well as from the hidden units, process them to recursively construct the states (history embeddings) for the next-time step and finally generates the distribution of the next event. In a pivotal departure from prior work, our proposal also attempts to design the nonlinear form in a principled and interpretable way. To do so, we further scrutinize the crawled datasets and surprisingly observe that the extent of disagreement between two users has a strong effect on both message and opinion dynamics. Such a disagreement factor helps our model to capture the coupled dynamics between message and opinion and hence it intuitively articulates the reality very well, without drastically changing the model-setting. On the five datasets collected from Twitter, our proposal offers substantial accuracy gains, enabling the model to forecast opinions even at a distant time-stamp.

II. Problem Formulation & Temporal Point Process

We define the problem more formally, the solution of the problem is based upon temporal point process which we outline next.

A. Problem Definition

Setting: Given a directed social network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, we denote $\mathcal{N}(u)$ as the set of users followed by a user u and each post e as $e := (t, m, u)$, where the triplet means that the user $u \in \mathcal{V}$ posted a message with sentiment m at time t. We denote $\mathcal{H}_u(t) = \{e_i = (t_i, m_i, u_i) | u_i = u \text{ and } t_i < t\}$ as a collection of all messages posted by user u until time t and $\mathcal{H}(t) := \cup_{u \in \mathcal{V}} \mathcal{H}_u(t)$ as the entire history of messages posted by any user upto and excluding time t.
Given a set of messages $\mathcal{H}(T) = \{e_i = (t_i, m_i, u_i)\}_{i=1}^n$ collected in time $(0, T]$, we wish to build a suitable mathematical model for opinion dynamics which should be able to predict the value of the opinion posted at $T + \Delta t$ time for a given $\Delta t > 0$.

B. Temporal Point Process

Modeling and mining activities in social media have been addressed from various perspectives [21, 15, 16]. In this paper, we represent the timestamps of users' activities or messages by a set of counting processes. In particular, we denote the set of counting processes as a vector $\boldsymbol{N}(t)$, in which the u-th entry, $N_u(t) \in \{0\} \cup \mathbb{Z}^+$, counts the number of messages user u posted until time t. Then, we characterize the counting process using the conditional intensity function $\lambda_u^*(t)$ for user u, which is the conditional probability that u posts a message in an infinitesimal window $[t, t + dt)$ given the history $\mathcal{H}(t)$, *i.e.*,

$$\lambda_u^*(t)\, dt = \mathbb{P}\left\{\text{event in } [t, t+dt) \,|\, \mathcal{H}(t)\right\} = \mathbb{E}[dN_u(t) \,|\, \mathcal{H}(t)],$$

where $dN_u(t) := N_u(t + dt) - N_u(t) \in \{0, 1\}$, the sign * means that the intensity may depend on the history $\mathcal{H}(t)$, and the functional form of the intensity is often designed to capture the phenomena of interest [7, 12, 13, 10].

C. SLANT–A point process driven opinion model [7]

SLANT offers a semi-coupled generative dynamics of message opinion and intensity.
Stochastic process for opinion: The opinion $x_u^*(t)$ of a user u at time t is affected by the sentiments of posts posted by her neighbours as:

$$x_u^*(t) = \alpha_u + \sum_{v \in N(u)} a_{vu} \sum_{e_i \in H_v(t)} m_i \kappa(t - t_i) \qquad (1)$$

where the first term, $\alpha_u \in R$, models the original opinion a user u starts with, the second term, with $a_{vu} \in R$, models updates in user u's opinion due to informational influence from her neighbors. Here, $\kappa(t)$ denotes a nonnegative triggering kernel which models the decay of informational influence over time.
Stochastic process for intensity: Slant models the message intensities using Multivariate Hawkes processes that captures a mutual excitation phenomena between message events and depends on the whole history of message events $\mathcal{H}(t)$ before t. Here, the intensity $\lambda_u^*(t)$ is governed by,

$$\lambda_u^*(t) = \mu_u + \sum_{v \in u \cup N(u)} b_{vu} \sum_{e_i \in H_v(t)} \zeta(t - t_i) \qquad (2)$$

where the first term, $\mu_u \geq 0$, models the publication of messages by user u on her own initiative, and the second term, with $b_{vu} \geq 0$, models the temporal influence of the other posts. Here, similar to $\kappa(t)$, $\zeta(t)$ is a nonnegative triggering kernel modeling the decay of temporal influence over time.

D. Limitations of the parameterized representation

In the aforesaid specifications of point processes, the form of intensity functions is already assumed (Eqs. (2), (1)). Somehow, such forms encode our prior knowledge about the latent variables of the model. However, the reality in general may or may not conform with prior intuitions. Therefore such fixed parameterizations often constrain the demonstrative power of the model. Therefore, in practice, to get a best predictive performance, one needs to try with various forms of intensity functions and as a result, most often we end up with an inaccurate model suffering certain errors due to model misspecification.
In the context of opinion propagation process, traditionally the message dynamics and opinion dynamics are assumed to be decoupled. However, in practice, the difference in opinions strongly influence the message propagation process. E.g. an open minded user may prefer an intense exchange of dialogues with her neighbors disagreeing with her. On the other hand, other users may prefer to stay apart if they

substantially differ in opinions. Therefore, opinion dynamics influences the message flow in a social network in general. **Recurrent marked temporal process:** In order to overcome the limitations of the parametric models of intensity function, Du *et al.* [9] has proposed Recurrent Marked Temporal Point Process (RMTPP) to simultaneously model the event timings and the markers. However, their method aims to capture the nonlinear dependencies in case of univariate temporal point process and hence cannot be modified to reproduce the dynamics of a multivariate Hawkes process via an easy extension. In addition, their method does not offer modeling of continuous time opinion dynamics, which is crucial in our work.

III. Our Approach

Given a directed social network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the history $\mathcal{H}(t) := \cup_{u\in\mathcal{V}}\mathcal{H}_u(t)$ of the messages, the intensity of messages $\boldsymbol{\lambda}^*(t) := (\lambda^*_u(t))_{u\in\mathcal{V}}$ and the opinion $\boldsymbol{x}(t) := (x^*_u(t))_{u\in\mathcal{V}}$, the joint nonlinear dynamics of opinion and the message intensity follow as,

$$\lambda^*_u(t) = f_u(\cup_{v\in\mathcal{N}(u)}\mathcal{H}_v(t)), \quad x^*_u(t) = g_u(\cup_{v\in\mathcal{N}(u)}\mathcal{H}_v(t))$$

More specifically, if $\mathcal{H}(t) := \{t_j, m_j, u_j\}_{j=1}^{j=i}$

$$\lambda^*_u(t_{i+1}) = f_u(\cup_{v\in\mathcal{N}(u)}\mathcal{H}_v(t_i^+)),$$
$$x^*_u(t_{i+1}) = g_u(\cup_{v\in\mathcal{N}(u)}\mathcal{H}_v(t_i^+)) \tag{3}$$

where $\boldsymbol{f}(.) = (f_u(.))_{u\in\mathcal{V}}$ and $\boldsymbol{g}(.) = (g_u(.))_{u\in\mathcal{V}}$ are arbitrary nonlinear functions in contrary to the restrictive form given in Eqs. (1) and (2). In this section, we aim to estimate these nonlinear functions $\boldsymbol{f}(.)$, $\boldsymbol{g}(.)$ with reasonable approximation exploiting the sophisticated state-of-the-art techniques of deep recurrent neural network.

A. An RNN driven approach

Theoretically, a finite-sized recurrent neural network with sigmoidal activation units can approximate a universal Turing machine [14] which is why it is considered to be a powerful tool for modeling a broad spectrum of applications (See [9] and the citations therein). At a high level, any RNN is a feedforward neural network structure with some auxiliary edges. These extra edges, referred to as the recurrent edges, connect the output signals of the hidden units at the current time into the network as the future inputs at the next time step. This type of neural structure can also be equivalently represented as a connected series of simple neural blocks, replicated at each time step to form an infinite cascade. It has been observed [9] that due to the presence of such recursive structure along with its inbuilt memory mechanism, RNN allows an efficient modeling of mutually exciting temporal processes in an accurate and tractable way. The memory factor quantifies the amount of loss of information the process allows from a past event. While such memory is often captured (in linear models) by an exponential kernel, in practice, the influence of memory can be more complicated. RNN uses the signals from the hidden units in the last time-step to generate a feedback mechanism that again creates an additional internal state in the network to memorize the effect of each past data sample.

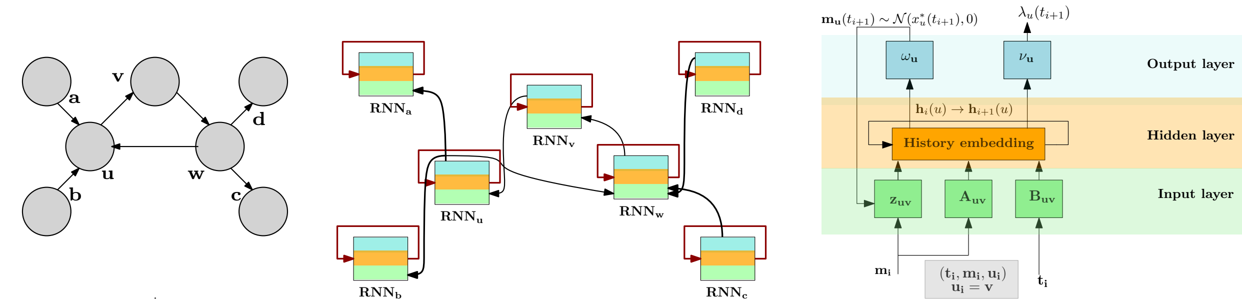
We propose a network-guided recurrent neural network architecture (Figure 1) in order to model the nonlinear dependencies between past events with memory. We assign one RNN (RNN_u) per each user u (see panel (b) in Figure 1), and RNN_u takes events $\cup_{v\in\mathcal{N}(u)}e_v$ as input and outputs the message intensity of the next candidate event posted by u along with the distribution of updated opinion m_u. Specifically, as shown in the panel (c) of Figure 1, an event (t_i, m_i, v) posted at time t_i by node v, goes as an input to RNN_u corresponding to each of her followees v, that in turn generates the distributions of the timings and the opinions of the potential next event (t_{i+1}, m_{i+1}, u) posted by u. The hidden states of each RNN_u capture the history $\mathcal{H}_u(t)$ via the embeddings $h_i(u)$ recursively computed from $h_{i-1}(u)$ using the events from her followees as input. Due to the recursive structure, such embeddings also encapsulate the memory of influence from the past timings and opinions. The recurrent neural network in each node offers a varying number of layers for different individual. As a consequence, such a network-guided recurrent neural network architecture gives an enormous advantage over the fixed parametric representations, by capturing any nonlinear dependencies between past events. As a result our proposed formulation can encapsulate a general form of representation of the intensity function.

While an intricate nonlinear modeling is often a key to uncover the complexity of an opinion propagation process, a conceptual and interpretable design of the nonlinear form is also crucial to a complete understanding of the inherent dynamics. To do that, we tap various signals from the dataset and find that the disagreement between the users play a strong role on message and opinion dynamics. More in detail, we find that the users with little or no disagreement trigger higher post-rate (dialogue in the neighborhood) than the conflicting ones, while opinion change occurs when there is a certain amount of disagreement between users – a simple, intuitive yet crucial factor that was left un-addressed in the literature. We encode this additional feature into a suitable functional form (Eq. (4)) that in turn is plugged into the nonlinearity to be learned. Such an apriori notion often helps to accurately learn the process from limited and missing data.

B. SLANT+

Now, we describe the implementation of SLANT+.

Input layer: On the arrival of event e_i, the input layer in RNN_u converts the message-sentiments m_i into $\boldsymbol{y}^i_u = \boldsymbol{A}_{uu_i}m_i + \boldsymbol{\alpha}_u$. Here $\boldsymbol{A}_{uu_i}$ is a vector indicating influence between user u and u_i and $\boldsymbol{\alpha}_u$ gives the biases. $\boldsymbol{A}_{uu_i}$ usually depends on the attributes of nodes u and u_i [6, 5]. Simi-



(a) A sample network (b) Network-guided RNN architecture (c) Architecture for RNN_u

Figure 1: The neural network architecture of SLANT+. Panel (b) shows the high level representation of the network-guided RNN formation corresponding to the sample network in panel (a). In this network-guided architecture, the output from RNN_x goes as input to RNN_y if y follows x. Panel (c) shows the architecture for RNN_u.

larly, the message-timings undergo a similar transformation $\boldsymbol{\theta}_u^i = \boldsymbol{B}_{uu_i} t_i + \boldsymbol{\mu}_u$. In contrast to the existing works that assume influence to be a scalar, here we model the inter-user influence as a vector capturing influence in various dimensions.

In a notable addition, the input layer makes an additional nonlinear transformation (z_{uv}) on the message sentiments which capture the effect of disagreement between users. As mentioned previously, the consensual users would more likely to interact with each other in contrast to the conflicting ones. On the other hand, users with certain level of disagreement can mutually influence each other to change their opinion. To model these traits, we design $z_{uv}(t)$ as,

$$z_{uv}(t_i) = (w_1^{z_{uv}} \sigma(w_{uv}^{(1)}.|\Delta m_{uv}|) + w_2^{z_{uv}} \sigma(w_{uv}^{(2)}.|\Delta m_{uv}|))_+ \quad (4)$$

Here $\sigma(x) = 1/(1+\exp(-x))$ is the usual sigmoid function. However, parameters of this curve in general are different across users, since some users may be open to discussion with the users with different opinions and others may not be. For example, if $w_1^{z_{uv}}.w_2^{z_{uv}} > 0$, then u is open to discussion (and perhaps change opinion) even with an intensely conflicting user v. If $w_1^{z_{uv}}.w_2^{z_{uv}} < 0$, the disagreement function is simply a bell shaped curve. That said, a reasonable amount disagreement between two users would intensify their conversions in which they may like to resolve their conflicts. However if either of the weights is zero, it indicates that the users with similar opinion prefer to carry on the conversions.

Hidden Layer: The purpose of the layer is to appropriately embed the history. In order to do that, RNN_u, the RNN cell corresponding to the node u, updates $h_i(u)$ recursively using $h_{i-1}(u)$ and the signals of current events from appropriate nodes. In this paper, the generative dynamics of the embedding function $h_i(u)$ is governed by the following.

$$h_{i+1}(u) = \sigma(w_u^h h_i(u) + \boldsymbol{w}_u^\theta.\boldsymbol{\theta}_u^i + \boldsymbol{w}_u^y.\boldsymbol{y}_u^i + w_u^z \sum_y z_{uv}) \quad (5)$$

Output Layer: The nonlinear embeddings $h_i(u)$ thus obtained are fed as inputs in the output layer and converted into the intensities and opinions. The intensity λ_u^* is,

$$\lambda_u^*(t) = \exp(\omega_u^h h_{i+1} + \omega_u^t (t - t_i) + b^t) \quad (6)$$

Here, the term $\omega_u^h h_{i+1}$ indicates the past influence of the events and $\omega_u^t (t - t_i)$ indicates the current influences.

The opinion is generated as follows:

$$x_u^*(t) = \tanh(\nu_u^h h_{i+1} + \nu_u^t (t - t_i) + k^t) \quad (7)$$

C. Parameter Estimation

Given a collection of messages $\mathcal{H}(T)$ recorded during a time period $[0, T)$ we find the optimal parameters of the RNNs by maximizing the following likelihood

$$\mathcal{L} = \sum_{e_i \in \mathcal{H}(T)} \left[\frac{(x_{u_i}(t_i) - m_i)^2}{\sigma^2} + \log \lambda_{u_i}^*(t_i) \right] - \sum_{u \in \mathcal{V}} \int_0^T \lambda_u^*(\tau) d\tau$$

To train the model, we exploit the Back Propagation Through Time (BPTT) approach [18] that we implement in TensorFlow [1] with a few lines of code.

IV. Experiments

In this section we provide the details of the experiments beginning with a brief description of the datasets, the metrics used, the evaluation protocol, and finally give a comparative sketch of the performances of various models on various datasets.

A. Datasets & Baselines

We used the five real datasets to validate our proposal. Among these datasets, the first four datasets are collected from [7]. For the new one (Series), we followed the similar way adopted in our previous work [7], on the tweets related to the discussion of Season 6 of *The Game of Thrones*, from April 8 to 15, 2015. We compare our results with several state-of-the-art baseline models for example, DeGroot [8], Voter Model [20], Biased Voter Model [3], Asynchronous Linear Model [4], and SLANT [7].

[1] https://www.tensorflow.org/

	Mean Squared Error						
Dataset	SLANT+	SLANT	BVoter	Voter	AsLM	DeGroot	Flocking
Movie	0.007 (90.79)	0.076	0.755	0.822	1.367	0.499	0.69
Politics	0.038 (82.16)	0.213	0.771	0.670	1.023	0.875	0.76
Fight	0.045 (79.82)	0.223	1.351	1.477	1.514	0.963	1.31
Bollywood	0.049 (88.71)	0.434	2.015	2.132	3.579	1.724	1.94
Series	0.049 (32.88)	0.073	0.287	0.536	0.796	0.533	0.49
	Failure Rate						
Movie	0.00 (–)	0.0	0.0	0.0	0.0	0.0	0.0
Politics	0.03 (80.0)	0.15	0.51	0.51	0.51	0.46	0.58
Fight	0.06 (53.85)	0.13	0.59	0.59	0.54	0.43	0.54
Bollywood	0.01 (93.33)	0.15	0.43	0.44	0.50	0.42	0.43
Series	0.01 (66.67)	0.03	0.31	0.41	0.33	0.47	0.48

Table I: Opinion forecasting performance with forecasting at 6 hrs., using a 10% held-out set for each dataset. The first half of the table dissects forecasting error in terms of MSE and the second half shows FR. In each cell, The cells with light orange (blue) color indicates the best (second best) predictor. Numbers in the bracket denote percentage improvement over the nearest baseline. Numbers in the italics indicate the best performer among the six state-of-the-art baselines.

Dataset	$\|\mathcal{V}\|$	$\|\mathcal{E}\|$	$\|\mathcal{H}(T)\|$	$\mathbb{E}[\mathbf{m}]$	$\mathbf{std}[m]$
Politics	548	5271	20026	0.0169	0.1780
Movie	567	4886	14016	0.5969	0.1358
Fight	848	10118	21526	-0.0123	0.2577
Bollywood	1031	34952	46845	0.5101	0.2310
Series	947	10253	20026	-0.0216	0.3177

Table II: Real datasets statistics

B. Metrics

We report the utility of our proposal by the following metrics. Suppose that, given at any time t_i, the expressed sentiment of a user u is m_u and our proposal forecasts the corresponding opinion to be x_u^i, then we define the following measures of the errors:

Mean Square Error (MSE): MSE at the time t_i is defined as follows,

$$\text{MSE} \ = \frac{1}{|\mathcal{V}|} \sum_{u \in \mathcal{V}} (x_u^i - m_u)^2 \tag{8}$$

Failure Rate (FR): Failure rate indicates how accurately we can detect the polarity of users at future. It is defined as,

$$\text{Failure Rate} \ = \frac{1}{|\mathcal{V}|} \sum_{u \in \mathcal{V}} \mathbb{1}(\text{sign}(x_u^i) \neq \text{sign}(m_u)) \tag{9}$$

C. Evaluation protocol

Given a stream of message $\mathcal{H}$, we first split it into training and test set where training set comprises of the first 90% of the total number of messages ($|\mathcal{H}|$). These messages are used to train our model for estimating the parameters. The estimated model is thereafter used to predict the opinions of the messages in the test set. For the discrete-time baselines, we simulate N_T times in $(t - T, t)$, where N_T is the no. of posts in time T.

D. Comparison with Baselines

Table I dissects a comparative analysis of the prediction-error (MSE and Failure Rate) of six state-of-the-art algorithms along with three additional variations of our algorithms. The top-half of the table reports the Mean Squared Error (actual opinion prediction error) while the rest reports the Failure Rate (polarity prediction error). We observe that for all the datasets, the overall performance of our proposal is substantially better than all the baselines. Among the baselines, we find that SLANT performs best. However, the variants of SLANT+ show a significant performance boost w.r.t. SLANT (upto 92%) which conforms the utility of our proposal.

Linear models: We observe that the forecasting performance of AsLM and DeGroot are substantially poor. During the forecasting phase, these models were iterated multiple rounds to compute the predicted opinion. However, since they are discrete-step Markov linear models, their training only permits a reasonably good prediction at the very next-step. As a result, the predictive performance degrades appallingly despite its modest performance in nowcasting.

Voter and Biased Voter Model: Voter model allows a user to update her opinion randomly from one of her neighbors. As a result it cannot sense the signals from the actual dynamics. Moreover, such an update strategy restricts the set of opinions in a network invariant throughout the opinion exchange process. This inherent property of voter model constrains the opinion-values to remain in a smaller space, as opposed to the spirit of continuous opinion-model. This assists the model to perform better than AsLM but worse than DeGroot. Biased Voter Model attempts to overcome these by introducing node weights but succeeds only partially.

Flocking model: It is interesting that MSE for flocking is substantially lower than other baselines in most cases. Recall that this model updates the opinion of a node by averaging those of her neighbours, that are very close to her. Such a selective averaging strategy makes it functionally similar with the DeGroot model. As a result the performance of this model is better than AsLM as well as Voter model and its variants.

SLANT: We observe that SLANT performs better than all other baselines. Since SLANT takes the message dynamics into consideration, which was left unaddressed by its older counterparts, it can properly leverage the effect of message dynamics on opinion exchange and as a consequence, it can properly anticipate the dynamics of conversation in future. While the earlier approaches rely on discrete-time modeling of opinion flow which is a continuous-time process, SLANT adopts a principled approach to exploit the continuous dynamics for forecasting opinion. However, the model offers a linear dynamics of opinion flow. In addition, it considers a fixed parameterized representation of the dynamics which often does not match with the reality. As a result, it performs poorer than SLANT+.

SLANT+: SLANT only considers the effect of message dynamics on opinion propagation, but not vice-versa. However, SLANT+ also considers the effect of opinion propagation on message dynamics. As a result, it captures the coupled dynamics of both message and opinion dynamics. More importantly, SLANT+ captures the nonlinear influences of the past events for each individual user. Hence it is able to accurately capture the intrinsic complexity of the process, which none of the existing baselines could do. Consequently it gains a much higher mileage in terms of forecasting opinions even at a distant future.

Failure Rate: In general, we observe that SLANT+ outperforms the other baselines in predicting polarity of the users. However, for Movie dataset, all the users have positive polarity across time, as a result all the algorithms can perfectly predict polarity of users. Apart from this, we also observe that Flocking model fares poorly in polarity prediction as compared to its performance in MSE. This is because it associates more weights to opinions "close" to one's own opininion. However, the notion of "closeness" is absolute and hence polarity difference is ignored.

V. Conclusion

In this paper, we build SLANT+, a novel nonlinear generative model for opinion dynamics, that extends our previous linear model SLANT [7]. To devise such a nonlinear model, we propose a novel network-guided recurrent neural network architecture which captures a generic form of nonlinear dependencies between the past events and the underlying social network. Moreover, our proposal also emphasizes a principled design of the nonlinearity. As a result, our approach outperforms the existing approaches in terms of both actual opinion prediction error (MSE) as well as polarity prediction error (FR).

Acknowledgement: A.De is supported by Google India under Google India PhD Fellowship 2013.

References

[1] R. Axelrod. The dissemination of culture a model with local convergence and global polarization. *Journal of conflict resolution*, 41(2):203–226, 1997.

[2] D. Bindel, J. Kleinberg, and S. Oren. How bad is forming your own opinion? In *FOCS*, 2011.

[3] A. Das, S. Gollapudi, and K. Munagala. Modeling opinion dynamics in social networks. In *WSDM*, 2014.

[4] A. De, S. Bhattacharya, P. Bhattacharya, N. Ganguly, and S. Chakrabarti. Learning a linear influence model from transient opinion dynamics. In *CIKM*, 2014.

[5] Abir De, Sourangshu Bhattacharya, Sourav Sarkar, Niloy Ganguly, and Soumen Chakrabarti. Discriminative link prediction using local, community, and global signals. *IEEE Transactions on Knowledge and Data Engineering*, 28(8):2057–2070, 2016.

[6] Abir De, Niloy Ganguly, and Soumen Chakrabarti. Discriminative link prediction using local links, node features and community structure. In *ICDM*, 2013.

[7] Abir De, Isabel Valera, Niloy Ganguly, Sourangshu Bhattacharya, and Manuel Gomez-Rodriguez. Learning and forecasting opinion dynamics in social networks. In *NIPS, 2016*.

[8] M. H. DeGroot. Reaching a consensus. *Journal of the American Statistical Association*, 69(345), 1974.

[9] Nan Du, Hanjun Dai, Rakshit Trivedi, Utkarsh Upadhyay, Manuel Gomez-Rodriguez, and Le Song. Recurrent marked temporal point processes: Embedding event history to vector. In *KDD*, 2016.

[10] M. Farajtabar, Y. Wang, M. Gomez-Rodriguez, S. Li, H. Zha, and L. Song. Coevolve: A joint point process model for information diffusion and network co-evolution. In *NIPS*, 2015.

[11] R. Hegselmann and U. Krause. Opinion dynamics and bounded confidence models, analysis, and simulation. *Journal of Artificial Societies and Social Simulation*, 5(3), 2002.

[12] Bidisha Samanta, Abir De, Abhijnan Chakraborty, and Niloy Ganguly. Lmpp: A large margin point process combining reinforcement and competition for modeling hashtag popularity. In *IJCAI*, 2017.

[13] Bidisha Samanta, Abir De, and Niloy Ganguly. Strm: A sister tweet reinforcement process for modeling hashtag popularity. In *INFOCOM*, 2017.

[14] Hava T Siegelmann and Eduardo D Sontag. Turing computability with neural nets. *Applied Mathematics Letters*, 4(6):77–80, 1991.

[15] Niket Tandon, Gerard de Melo, Abir De, and Gerhard Weikum. Knowlywood: Mining activity knowledge from hollywood narratives. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM, 2015.

[16] Niket Tandon, Gerhard Weikum, Gerard de Melo, and Abir De. Lights, camera, action: Knowledge extraction from movie scripts. In *WWW*. ACM, 2015.

[17] Yongqing Wang, Shenghua Liu, Huawei Shen, Jinhua Gao, and Xueqi Cheng. Marked temporal dynamics modeling based on recurrent neural network. In *PAKDD*, 2017.

[18] Paul J Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

[19] Shuai Xiao, Junchi Yan, Xiaokang Yang, Hongyuan Zha, and S Chu. Modeling the intensity function of point process via recurrent neural networks. AAAI, 2017.

[20] M. E. Yildiz, R. Pagliari, A. Ozdaglar, and A. Scaglione. Voting models in random networks. In *Information Theory and Applications Workshop*, pages 1–7, 2010.

[21] A. Zarezade, A. De, H. Rabiee, and M. G. Rodriguez. Cheshire: An online algorithm for activity maximization in social networks. *arXiv:1703.02059*, 2017.