Algorithms for Protein Structure Analysis: Alignment and Classification

Sourangshu Bhattacharya
sourangshu@cse.iitkgp.ernet.in

Computer Science and Engineering,
IIT Kharagpur - 721302.

Outline

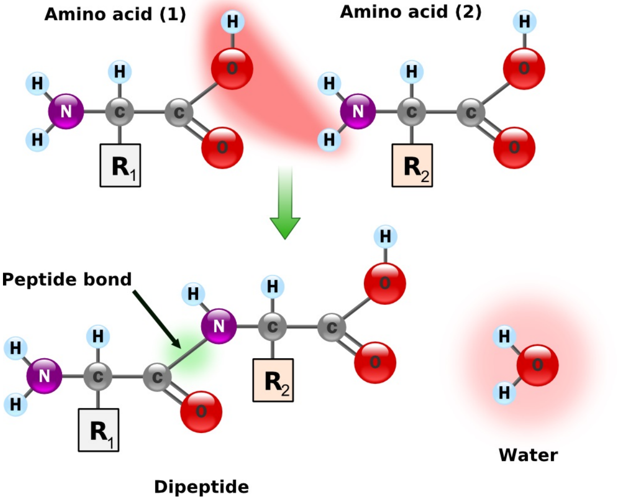
- Protein Structure Alignment
  - Introduction
  - Matrix and Graph Representation
  - Graph Matching
  - A new method
  - Results
- Protein Sub-Structure Alignment
  - Neighborhoods
  - Results
- Protein Structure Classification
  - Preliminaries
  - Kernels on Protein Structures
  - Results
- Learning with Uncertainty
  - Resolution-aware Protein Structure Classification

Outline

- Protein Structure Alignment
    - Introduction
    - Matrix and Graph Representation
    - Graph Matching
    - A new method
    - Results
- Protein Sub-Structure Alignment
    - Neighborhoods
    - Results
- Protein Structure Classification
    - Preliminaries
    - Kernels on Protein Structures
    - Results
- Learning with Uncertainty
    - Resolution-aware Protein Structure Classification

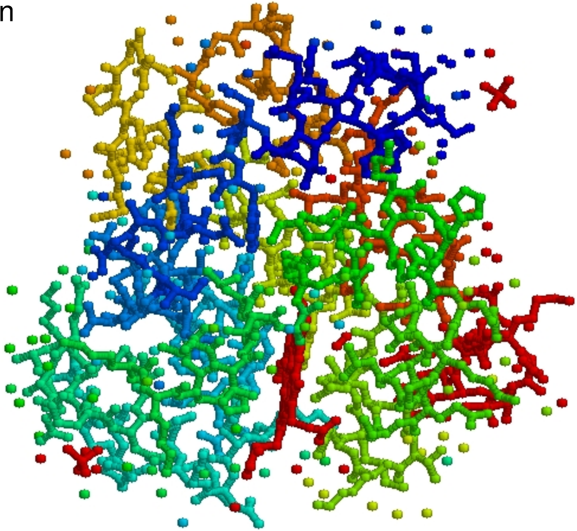
What is a Protein ?

- Amino acids form peptide bonds to polymerize.
- Proteins are poly-peptide molecules.
- Represented by sequence of residues.
- Poly-peptide chains fold to form 3D structures.



Protein Structure

Myoglobin (1DWT): Tertiary Structure



Protein Structure

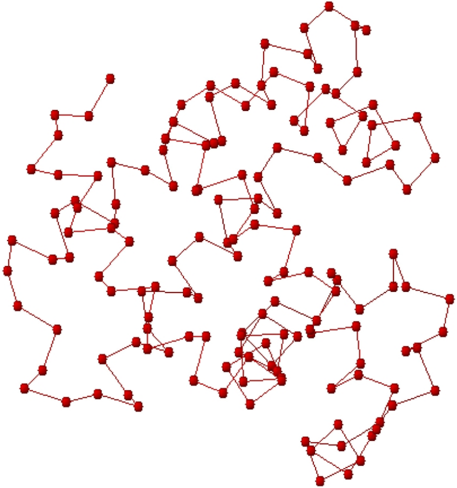
Simplification: C$^{\alpha}$ atoms and topology.

Loss:

- Side chain
- Secondary structure

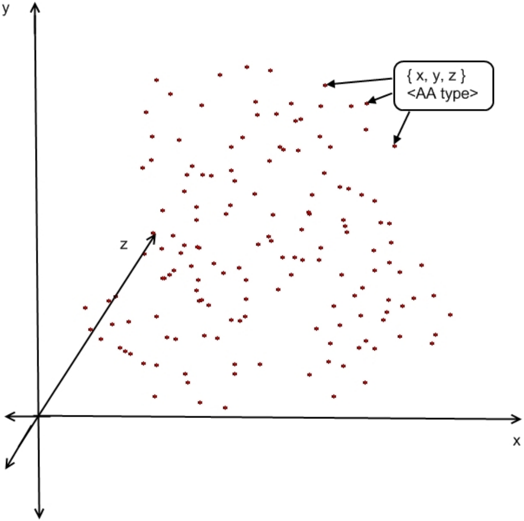
Gain: Simplicity

Past uses: SSAP, DALI, CE, etc.



Pointsets

Problem with Topology: *non-topological similarities* are not detected.
New model: Pointset.
Gain: Generality
(Active sites ?)
Past uses: C^{α} match.



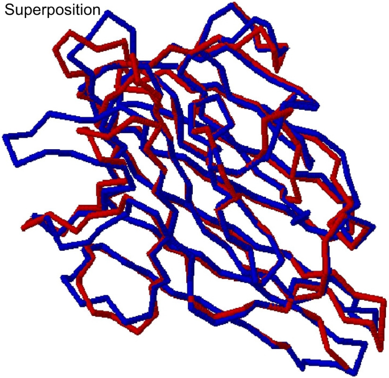
Abstraction

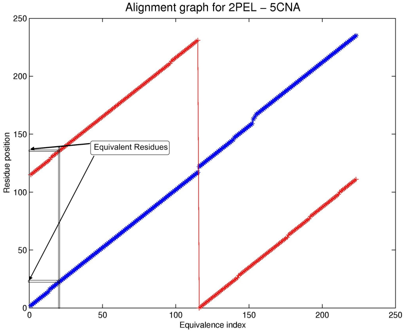
Protein Structure

A protein structure X having n residues is represented as $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ where $\mathbf{x}_i \in \mathbb{R}^3, 1 \leq i \leq n$.

Each $\mathbf{x}_i$ gives position of C$^\alpha$ atom of the i^{th} residue with respect to some arbitrary coordinate system.

Structural Alignment





Alignment between two proteins, 2PEL and 5CNA, showing circular permutations.
Alignment is defined by a set of *equivalences*.
Optimal superposition can be calculated easily.

Structural Alignment

Structural Alignment

A *structural alignment* between two proteins X^A and X^B is a 1-1 mapping $\phi : \{i | \mathbf{x}_i^A \in \bar{X}^A\} \to \{j | \mathbf{x}_j^B \in \bar{X}^B\}$, where $\bar{X}^A \subseteq X^A$ and $\bar{X}^B \subseteq X^B$.

Structural Alignment

Structural Alignment

A *structural alignment* between two proteins X^A and X^B is a 1-1 mapping $\phi : \{i | \mathbf{x}_i^A \in \bar{X}^A\} \rightarrow \{j | \mathbf{x}_j^B \in \bar{X}^B\}$, where $\bar{X}^A \subseteq X^A$ and $\bar{X}^B \subseteq X^B$.

Root Mean Square Deviation

$$RMSD(\phi) = \sqrt{\frac{1}{|\bar{X}^A|} \sum_{(i,j) \in \Phi} (\mathbf{x}_i^A - \mathcal{T}(\mathbf{x}_j^B))^2}$$

where $\mathcal{T}$ is the optimal transformation.

Can we use this as a score function ?

Structural Alignment

Structural Alignment

A *structural alignment* between two proteins X^A and X^B is a 1-1 mapping $\phi : \{i | \mathbf{x}_i^A \in \bar{X}^A\} \rightarrow \{j | \mathbf{x}_j^B \in \bar{X}^B\}$, where $\bar{X}^A \subseteq X^A$ and $\bar{X}^B \subseteq X^B$.

Root Mean Square Deviation

$$RMSD(\phi) = \sqrt{\frac{1}{|\bar{X}^A|} \sum_{(i,j) \in \Phi} (\mathbf{x}_i^A - \mathcal{T}(\mathbf{x}_j^B))^2}$$

where $\mathcal{T}$ is the optimal transformation.

Problem: Both ϕ and $\mathcal{T}$ are unknown and interdependent.

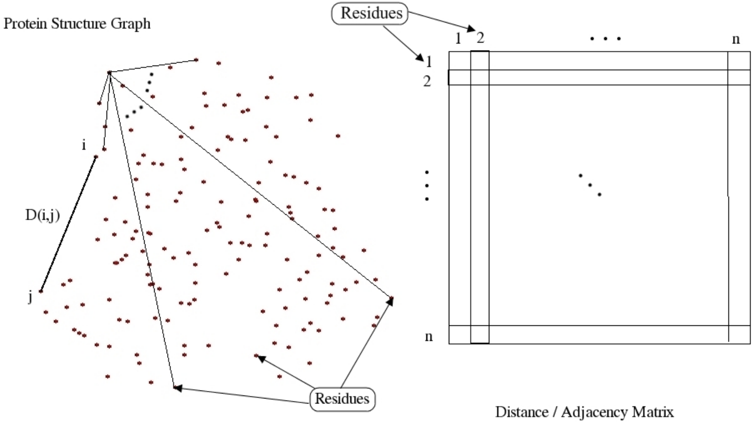
Another Score

Distance Root Mean Square Deviation

$$RMSD_D(\phi) = \sqrt{\frac{1}{|\bar{P}^A|^2} \sum_{\mathbf{x}_i^A, \mathbf{x}_j^A \in \bar{X}^A} (d_{ij}^A - d_{\phi(i)\phi(j)}^B)^2}$$

where, d_{ij}^A is the distance between residues $\mathbf{x}_i^A$ and $\mathbf{x}_j^A$.

Graph and Distance Matrix



Graph and Distance Matrix

DALI Scoring function

Known: Neighboring residues interact with greater force than far away ones.

$$S_{DALI}(\phi) = \sum_{\mathbf{x}_i^A, \mathbf{x}_j^A \in \bar{X}^A} \left(0.2 - \frac{|d_{ij}^A - d_{\phi(i)\phi(j)}^B|}{\bar{d}_{ij}}\right) \exp\left(-\left(\frac{\bar{d}_{ij}}{20}\right)^2\right)$$

Maximize S_{DALI} over all ϕ.
DALI uses heuristics which degrade it's performance. Also, not amenable to theoretical analysis.

Observation from DALI score

Neighboring residues affect the score more than far away ones.
So, use nearness instead of distance function.

Graph and Distance Matrix

Nearness matrix

The adjacency or nearness matrix $\mathcal{A}$ of a given protein $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ is defined as:

$$\mathcal{A}_{ij} = e^{\frac{-d_{ij}}{\alpha}},\ \alpha > 0$$

- An exponentially decreasing function of *d* between 0 and 1.
- A continuous and invertible function.

Scoring function

$$S(\phi) = \sum_{\mathbf{x}_i^A, \mathbf{x}_j^A \in \bar{X}^A} T - (\mathcal{A}_{ij}^A - \mathcal{A}_{\phi(i)\phi(j)}^B)^2$$

Maximize $S(\phi)$ over all ϕ. T is a known threshold.

Scoring function

Scoring function

$$S(\phi) = \sum_{\mathbf{x}_i^A, \mathbf{x}_j^A \in \bar{X}^A} T - (\mathcal{A}_{ij}^A - \mathcal{A}_{\phi(i)\phi(j)}^B)^2$$

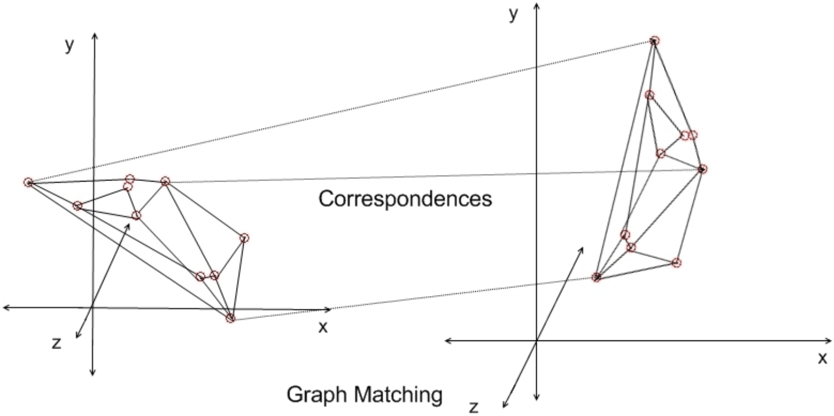
Maximize $S(\phi)$ over all ϕ. T is a known threshold.

Graph Matching

Given two weighted graphs $\mathcal{G}^A$ and $\mathcal{G}^B$, find their maximal subgraphs $\bar{\mathcal{G}}^A$ and $\bar{\mathcal{G}}^B$ and a mapping ϕ between vertices of $\bar{\mathcal{G}}^A$ and $\bar{\mathcal{G}}^B$ such that

$$|\mathcal{A}_{ij}^A - \mathcal{A}_{\phi(i)\phi(j)}^B| < T, i, j \in \bar{\mathcal{G}}^A$$

Graph Matching



Graph Matching

Graph Matching

Given two weighted graphs $\mathcal{G}^A$ and $\mathcal{G}^B$, find their maximal subgraphs $\bar{\mathcal{G}}^A$ and $\bar{\mathcal{G}}^B$ and a mapping ϕ between vertices of $\bar{\mathcal{G}}^A$ and $\bar{\mathcal{G}}^B$ such that

$$|\mathcal{A}^A_{ij} - \mathcal{A}^B_{\phi(i)\phi(j)}| < T, i, j \in \bar{\mathcal{G}}^A$$

Intractable

This is the optimization version of the well known NP-Hard problem *subgraph isomorphism*. Thus a polynomial time algorithm to find an exact solution of this problem does not exist unless $P = NP$.

Graph Matching

Assumption

Two structures have same number of residues, and all of them are aligned.

Weighted Graph Matching (Umeyama 88)[4]

$$S(P) = \|P\mathcal{A}^A P^T - \mathcal{A}^B\|^2$$

Minimize $S(P)$ over all permutation matrices P.

Spectral Solution

Motivation (Umeyama 88)[4]

Theorem 1 Let $\mathcal{A}^A$ and $\mathcal{A}^B$ be full rank adjacency matrices, with eigenvalue decompositions

$$\mathcal{A}^A = U^A \Lambda^A U^{AT}$$

$$\mathcal{A}^B = U^B \Lambda^B U^{BT}$$

$Q = U^B S U^{AT}$ minimizes $\|Q\mathcal{A}^A Q^T - \mathcal{A}^B\|^2$ for all orthogonal matrices Q. Here $S \in \mathcal{S} = \{\text{diag}(s_1, \ldots, s_n) | s_i = 1 \text{ or } -1\}$.

Theorem 2 Let $\bar{U}^A$ and $\bar{U}^B$ be matrices having absolute values of the entries in matrices U^A and U^B. Let $\hat{P}$ be the optimal permutation matrix in the case of a perfect match, then $\hat{P}$ maximizes

$$tr(P^T \bar{U}^B \bar{U}^{AT})$$

Spectral Solution

Corollary

Permutation $\hat{\pi}$ corresponding to $\hat{P}$ can be obtained by:

$$\min_{\pi \in \Pi} \sum_{i=1}^{n} \|(\bar{U}^A)_i - (\bar{U}^B)_{\pi(i)}\|^2$$

where $(A)_i$ is the i^{th} row of matrix A.

Neighborhood Preserving Projections

Projection

We are interested in projecting the residues on real line such that neighborhoods are preserved optimally.

$$\max_{\mathbf{f}\in\mathbb{R}^n} \sum_{i=1}^{n} \sum_{j=1}^{n} [\mathcal{A}_{ij}(f_i + f_j)^2 - \mathcal{A}_{ij}(f_i - f_j)^2]$$

Observations

- Second term: $|f_i - f_j|$ low whenever $\mathcal{A}_{ij}$ is high.
- First term: $|f_i + f_j|$ high whenever $\mathcal{A}_{ij}$ is low. So, f_i and f_j should be far apart.
- Unbounded solution. Constrain by adding $\|\mathbf{f}\|^2 = n$.

Neighborhood Preserving Projections

Final formulation

$$\max_{\mathbf{f}\in\mathbb{R}^n} \ \mathbf{f}^T\mathcal{A}\mathbf{f}$$

Subject to

$$\|\mathbf{f}\|^2 = n$$

This is same as finding the eigenvector corresponding to maximum eigenvalue of the matrix $\mathcal{A}$.

Absolute Value

If **f** is a eigenvector, so is $-\mathbf{f}$. Thus, we define *neighborhood preserving projections*, f_i as $|f_i^*|$.

Scoring function

Similarity score

Given two proteins X^A and X^B, and their neighborhood preserving projections $\mathbf{f}^A$ and $\mathbf{f}^B$, we define the similarity between residue i of X^A and residue j of X^B as:

$$s(i,j) = T - (f_i^A - f_j^B)^2$$

The similarity score of an alignment ϕ is:

$$S(\phi) = \sum_{\mathbf{x}_i \in \bar{X}^A} s(i, \phi(i))$$

Maximize $S(\phi)$ w.r.t. ϕ.

Connection

Spectral Similarity

By considering only the leading eigenvector, the spectral similarity score becomes:

$$\min_{\pi\in\Pi}\sum_{i=1}^{n}((\bar{U}_G^1)_i-(\bar{U}_H^1)_{\pi(i)})^2$$

or

$$\max_{\pi\in\Pi}\sum_{i=1}^{n}T-((\bar{U}_G^1)_i-(\bar{U}_H^1)_{\pi(i)})^2$$

Connection

Projection Similarity

If all residues of the two proteins are aligned, i.e. $\bar{X}^A = X^A$ and $\bar{X}^B = X^B$, we solve,

$$\max_{\phi} \sum_{\mathbf{x}_i \in X^A} T - (f_i^A - f_j^B)^2$$

Unequal residues

The above problem can be solved even in case of unequal number of residues in the two structures.

Greedy Fragment Pair Search

Topology

- The above problem is an instance of *assignment problem*. We could solve it in polynomial time.
- But we use information in protein sequence to solve the problem more efficiently.

Basic Idea

- The scoring function $s(i, j)$ is analogous to the sequence similarity function.
- Use sequence alignment algorithms, e.g. local alignment algorithm.

Greedy Fragment Pair Search

Algorithm

1. Initialize alignment to null.
2. Calculate the local alignment matrix of incremental fragment similarity.
3. Find the maximum element in the matrix and traceback to find the high scoring fragment pair.
4. Add the currently found fragment pair to the alignment and delete the rows and columns correspodning to the currently added residues from local alignment matrix.
5. Go to step 3. If no positive scoring entry is found, terminate.

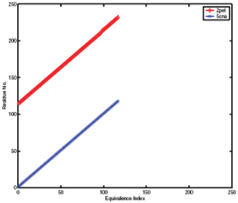
Benchmark Datasets

Comparison between Matchprot(MP) and DALI using benchmark datasets.

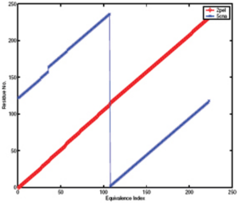
Data set / Classifn.	Total pairs	Better	Worse	Level
Fischer	68	17	18	33
Novotny et. al.				
1.10.40	21	8	1	12
1.10.164	10	2	0	8
1.25.30	21	3	0	18
2.30 110	6	1	2	3
2.40.100	28	4	3	21
2.100.10	15	5	4	6
3.10.70	10	0	2	8
3.40.91	6	6	0	0
3.70.10	15	1	3	11
2.40.20	21	1	4	16

Better: MP has lower RMSD higher length of alignment(Lali).
Worse: DALI has lower RMSD higher Lali.
Level: MP has either both higher or lower RMSD and Lali than DALI.

Non-topological Similarities

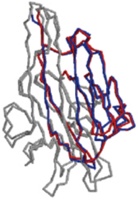


Dali

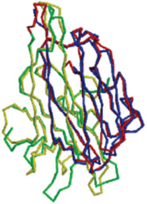


Matchprot

Alignment between 2PEL and 5CNA showing circular permutation



DALI



Matchprot

Structure Retrieval

Comparison with CE (Shindyalov and Bourne) [1]

Retrieval of domains having similar folds from ASTRAL 95% non-redundant dataset.

Query ID	Matchprot (TP/FP/prec./rec.)	CE (TP/FP/prec./rec.)
d101m__	93 / 0 / 1 / 0.95	96 / 2 / 0.97 / 0.99
d1htia_	272 / 56 / 0.82 / 0.83	307 / 29 / 0.91 / 0.93
d1jzba_	23 / 0 / 1 / 0.1	33 / 270 / 0.1 / 0.14
d2pela_	70 / 50 / 0.58 / 0.8	61 / 36 / 0.62 / 0.70
d7rsa__	18 / 0 / 1 / 1	17 / 1 / 0.94 / 0.94

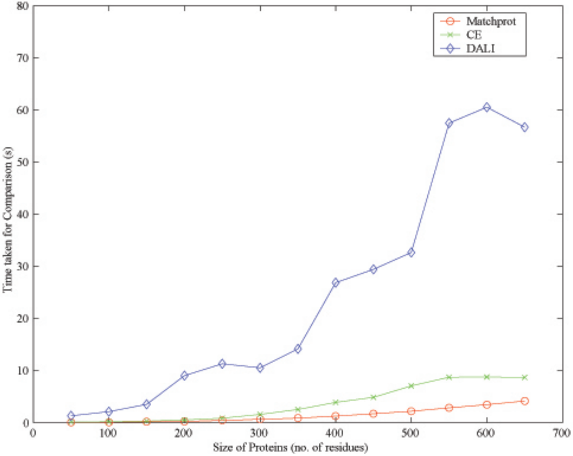
TP: True positive

FP: False positive

$$prec = \frac{TP}{TP+FP}$$

$$rec = \frac{TP}{Actual}$$

Time Comparison



Summary

- Fast $O(n^3)$ deterministic algorithm for comparing protein structure.
- New score function using neighborhood preserving projections.
- State of the art performance for structure retrieval on SCOP.

Outline

- Protein Structure Alignment
    - Introduction
    - Matrix and Graph Representation
    - Graph Matching
    - A new method
    - Results
- Protein Sub-Structure Alignment
    - Neighborhoods
    - Results
- Protein Structure Classification
    - Preliminaries
    - Kernels on Protein Structures
    - Results
- Learning with Uncertainty
    - Resolution-aware Protein Structure Classification

Problem of Indels

Problem

- The above algorithm was designed for similarly sized proteins.
- It still works for many cases with upto 40% indels.
- However, it gives wrong answers for proteins having higher indels (Roughly half of the residues are absent in the other protein).

Main Idea

Align conserved substructures called *neighborhoods*, and “grow” neighborhood alignments to entire structure.

Neighborhoods

Observation

Spatial neighborhoods are more preserved even in evolutionarily distant proteins.

Reasons

- The site crucial for functioning remains structurally preserved.
- Many a times, additions are in terms of separate domains.

Solution

- Compare spatial neighborhoods instead of entire structures using spectral method.
- “Grow” the neighborhood alignments to get a good overall alignment.

Neighborhoods

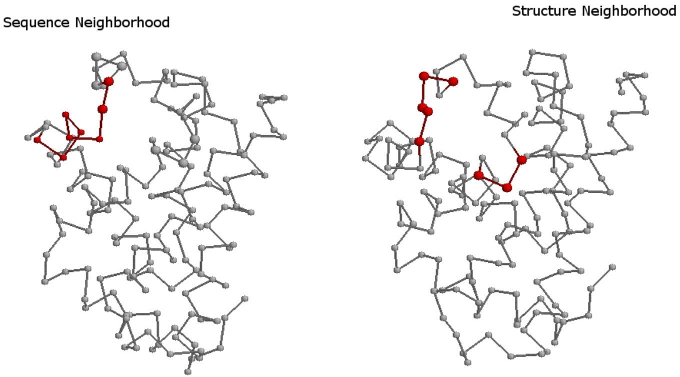
Definition

The *k-structure neighborhood* of a residue of a protein is defined as the set of k residues nearest to the given residue in 3D.

Definition

The *k-sequence neighborhood*, $N^A_{seq}(i)$ starting from residue i of structure A is defined as $N^A_{seq}(i) = \{\mathbf{x}_i, \ldots, \mathbf{x}_{i+k-1}\}$.

Neighborhoods



Sequence Neighborhood

Structure Neighborhood

Myoglobin (1DWT)

Alignment using Neighborhoods

Overall Scheme

1. Calculate a spanning set of neighborhoods.
2. Align all pairs of neighborhoods.
3. Grow neighborhood alignments to entire structure.

Spanning set of Neighborhoods

- Set of neighborhoods should span the entire protein, and should not be very high.
- For structure neighborhoods, choose one around every residue.
- For sequence neighborhoods, choose one starting at every residue.

Alignment using Neighborhoods

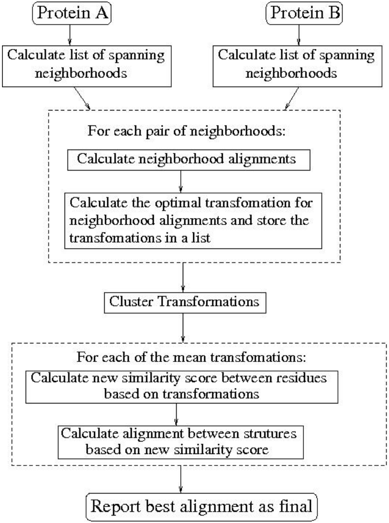
Neighborhood Alignment

- For sequence neighborhoods, use the spectral algorithm developed above.
- For structure neighborhoods, solve *maximal common subgraph*.
- Restrict sizes of structure neighborhoods.

Growing Neighborhood Alignments

- Calculate optimal transformation based on neighborhood alignment.
- Re-calculate similarity measure based using transformed coordinates.
- Calculate final alignment using revised similarity measure.

Algorithm



Protein A

Protein B

Calculate list of spanning neighborhoods

Calculate list of spanning neighborhoods

For each pair of neighborhoods:

Calculate neighborhood alignments

Calculate the optimal transfomation for neighborhood alignments and store the transfomations in a list

Cluster Transformations

For each of the mean transfomations:

Calculate new similarity score between residues based on transformations

Calculate alignment between strutures based on new similarity score

Report best alignment as final

Comparison with existing methods: Difficult Cases

10 difficult pairs mentioned in (Shindyalov and Bourne)[1]

PDBid1(size) -	Seq Nbhd	Struct Nbhd	DALI	CE	SSM
PDBid2(size)	LAli / RMS	LAli / RMS	Len / RMS	Len / RMS	Len / RMS
1fxiA(96) - 1ubq(76)	54 / 2.18	56 / 2.16	60 / 2.6	100 / 3.82	60 / 2.86
1ten(90) - 3hhrB(185)	84 / 1.58	82 / 1.39	86 / 1.9	87 / 1.90	73 / 2.09
3hlaB(270) - 2rhe(114)	70 / 2.26	68 / 2.26	75 / 3	85 / 3.46	78 / 3.08
2azaA(129) - 1paz(120)	72 / 2.46	79 / 2.20	81 / 2.5	85 / 2.90	79 / 2.41
1cewl(108) - 1molA(94)	68 / 1.80	79 / 1.94	81 / 2.3	81 / 2.34	79 / 2.12
1cid(177) - 2rhe(114)	91 / 2.05	91 / 2.06	97 / 3.2	98 / 2.97	89 / 2.32
1crl(534) - 1ede(310)	160 / 2.50	174 / 2.49	211 / 3.5	220 / 3.91	188 / 3.81
2sim(381) - 1nsbA(390)	262 / 2.72	262 / 2.63	222 / 3.8	276 / 2.99	271 / 2.86
1bgeB(159) - 2gmfA(121)	85 / 2.48	87 / 2.22	94 / 3.3	102 / 4.02	44 / 2.49
1tie(166) - 4fgf(124)	105 / 2.20	106 / 2.27	114 / 3.1	115 / 2.86	114 / 2.85

Overall Results on Benchmark Datasets

Comparison with DALI [3]

Data set/	Align. sequence nbhd.	Align. structure nbhd.
classifn.	Better / Worse / Level	Better / Worse / Level
Fischer's	4 / 4 / 60	5 / 2 / 61
Novotny's		
1.10.164	1 / 0 / 9	2 / 0 / 8
1.10.40	11 / 0 / 10	5 / 0 / 16
1.25.30	10 / 0 / 11	5 / 0 / 16
2.30.110	0 / 0 / 6	0 / 0 / 6
2.40.100	0 / 0 / 28	0 / 0 / 28
2.100.10	5 / 3 / 7	5 / 0 / 10
3.10.70	0 / 0 / 10	0 / 0 / 10
3.40.91	0 / 0 / 6	0 / 0 / 6
3.70.10	0 / 0 / 15	2 / 0 / 13
2.40.20	0 / 3 / 18	0 / 0 / 21

Overall Results on Benchmark Datasets

Comparison with CE [1]

Data set/ classifn.	Align. sequence nbhd. Better / Worse / Level	Align. structure nbhd. Better / Worse / Level
Fischer's	2 / 1 / 65	2 / 0 / 66
Novotny's		
1.10.164	0 / 0 / 10	0 / 0 / 10
1.10.40	0 / 0 / 21	0 / 0 / 21
1.25.30	1 / 0 / 20	0 / 0 / 21
2.30.110	0 / 0 / 6	0 / 0 / 6
2.40.100	6 / 0 / 22	4 / 0 / 24
2.100.10	4 / 0 / 11	4 / 0 / 11
3.10.70	1 / 0 / 9	1 / 0 / 9
3.40.91	0 / 0 / 6	0 / 0 / 6
3.70.10	0 / 0 / 15	0 / 1 / 14
2.40.20	1 / 1 / 19	0 / 0 / 21

Overall Results on Benchmark Datasets

Comparison with SSM

Data set/ classifn.	Align. sequence nbhd. Better / Worse / Level	Align. structure nbhd. Better / Worse / Level
Fischer's	13 / 10 / 45	23 / 5 / 40
Novotny's		
1.10.164	3 / 1 / 6	4 / 0 / 6
1.10.40	9 / 0 / 12	8 / 0 / 13
1.25.30	9 / 0 / 12	3 / 0 / 18
2.30.110	1 / 1 / 4	1 / 1 / 4
2.40.100	1 / 0 / 27	2 / 1 / 25
2.100.10	1 / 4 / 10	3 / 1 / 11
3.10.70	2 / 0 / 8	3 / 0 / 7
3.40.91	2 / 0 / 4	1 / 0 / 5
3.70.10	0 / 1 / 14	1 / 2 / 12
2.40.20	0 / 6 / 15	3 / 2 / 16

Structure Retrieval

5 SCOP Folds

SCOPid	Method	cutoff	True +ve	False +ve
(tot. num.)		(% / Z)		
d101m__	seq nbhd	50%	34	9
(37)	seq nbhd	45%	35	38
	CE	4.0	35	95
d1htia_	seq nbhd	50%	190	4
(253)	seq nbhd	45%	231	13
	CE	4.0	233	224
d1jzba_	seq nbhd	50%	28	56
(119)	seq nbhd	45%	48	172
	CE	4.0	2	0
d2pela_	seq nbhd	50%	41	9
(48)	seq nbhd	45%	45	21
	CE	4.0	36	8
d7rsa__	seq nbhd	50%	4	0
(4)	seq nbhd	45%	4	13
	CE	4.0	4	0

Summary

- A robust algorithm for for protein structure alignment.
- Idea of neighborhood alignments and growing of neighborhood alignments to entire structures.
- Outperformed state of the art techniques on benchmark datasets.

Outline

Protein Structure Alignment
- Introduction
- Matrix and Graph Representation
- Graph Matching
- A new method
- Results

Protein Sub-Structure Alignment
- Neighborhoods
- Results

Protein Structure Classification
- Preliminaries
- Kernels on Protein Structures
- Results

Learning with Uncertainty
- Resolution-aware Protein Structure Classification

Automatic Structure Classification

Problem

Classify given protein structures into SCOP superfamilies.

Approach

Define kernels on protein structures and use kernel methods.

Motivation

- Support vector machines (SVMs) are one of the most popular classifiers.
- SVMs cannot be directly used with protein structures.
- Kernels on protein structures will allow SVMs and many other methods to applied.

Kernel Methods

Definition

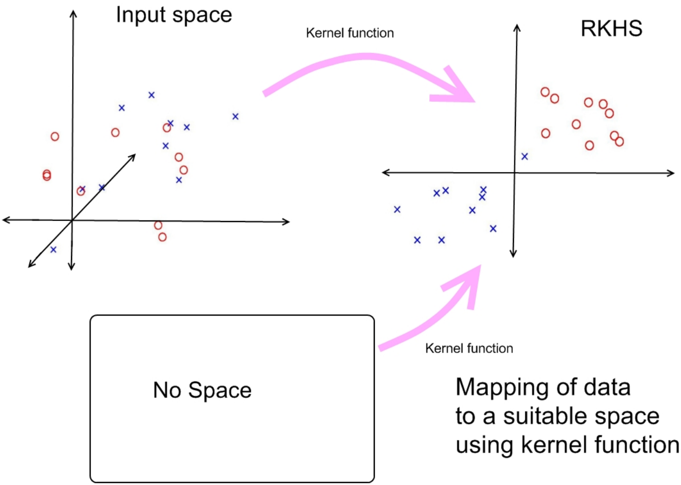
A kernel $\mathcal{K}$ on a set $\mathcal{X}$ is a real valued function on $\mathcal{X} \times \mathcal{X}$ satisfying the following properties:

- $\mathcal{K}(x, y) = \mathcal{K}(y, x)$ (Symmetric)
- $\mathcal{K}(x, x) \geq 0,$ and 0 only if $x = 0$
- $\sum_{i,j} c_i c_j \mathcal{K}(x_i, x_j) \geq 0 \forall c_i, c_j \in \mathbb{R}$ (Positive semidefinite)

RKHS

Kernels can be thought of as dot products in a higher dimensional space called *reproducing kernel hilbert space* (RKHS).

Kernel Methods



Mapping of data
to a suitable space
using kernel functions

Kernel Methods

Geometry

- Kernel functions define a geometry in the RKHS.
- Angles can be measured using the kernels.
- Distances can be defined as $d(x_i, x_j) = \sqrt{\mathcal{K}(x_i, x_i) + \mathcal{K}(x_j, x_j) - 2 * \mathcal{K}(x_i, x_j)}$.

Kernelized Algorithms

Many machine learning techniques can be modified to be used with kernels rather than vectorial data.

- Support Vector Machines.
- K-means clustering.
- Gaussian process regression, Principal component analysis, etc

Building New kernels

If $k_1(x,y)$ and $k_2(x,y)$ are two valid kernels, then the following kernels are valid:

- Linear Combination:

$$k(x,y) = c_1 k_1(x,y) + c_2 k_2(x,y)$$

- Exponentiation:

$$k(x,y) = exp(k_1(x,y))$$

- Product:

$$k(x,y) = k_1(x,y) k_2(x,y)$$

- Polynomial Transformation:

$$k(x,y) = Q(k_1(x,y))$$

- Function product:

$$k(x,y) = f(x) k_1(x,y) f(y)$$

Structured data

Motivation

- Many types of data processed by learning algortihms cannot be naturally represented as vectors.
- Kernelized learning algorithms can be used, if appropriate kernels are defined on those data.

Examples

- Strings, trees, graphs, etc.
- Protein structures.

Kernels on Structured Data

Intuition

- Kernels can be thought of as similarity measures since $d(x_i, x_j)$ is a decreasing function of $\mathcal{K}(x_i, x_j)$.
- Define similarity measures on structured data satisfying properties of kernels.
- Generally, positive-semidefiniteness is most difficult to ensure.

Other Kernels on Proteins

- Graph Kernels.
- Sequence based kernels.
- Alignment Kernels using empirical kernel maps.

Scheme

Problem

Define kernels capturing similarity between protein structures.

Ideas

- Kernels should capture the notion of structural alignment.
- Define kernels on neighborhoods and extend them to entire protein structures.

Kernels on Neighborhoods

Convolution Kernels (Haussler 99)

- $x \in X$ is a composite object, parts from $X_1, \dots, X_m$.
- R is a relation over $X_1 \times \cdots \times X_m \times X$ such that $R(x_1, \dots, x_m, x)$ is true if x is composed of $x_1, \dots, x_m$
- $K^1, \dots, K^m$ be kernels on $X_1, \dots, X_m$, respectively.

It can be showed that K is a kernel on X.

$$K(x, y) = \sum_{(x_1,\dots,x_m) \in R^{-1}(x), (y_1,\dots,y_m) \in R^{-1}(y)} \prod_{i=1}^{m} K^i(x_i, y_i)$$

where
$R^{-1}(x) = (x_1, \dots, x_m) \in X_1 \times \cdots \times X_m | R(x_1, \dots, x_m, x) =$ true.

Kernels on Neighborhoods

Spectral Kernel

- X is set of all neighborhoods.
- $X_1, \ldots, X_m$ are sets of residues.
- $R(x_1, \ldots, x_m, N)$ is true if $\{x_1, \ldots, x_m\} \in N$.
- $K^1, \ldots, K^m$ are RBF kernels comparing spectral projections.

Spectral kernel is defined as:

$$\mathcal{K}_{SS}(N_i, N_j) = \sum_{\pi \in \Pi} e^{\frac{-\|\mathbf{f}^i - \pi(\mathbf{f}^j)\|^2}{\beta}}$$

Kernels on Neighborhoods

Pairwise Distance Kernel

- X is set of all neighborhoods.
- $X_1, \ldots, X_m$ are sets of all pairs of residues.
- $R(d_1, \ldots, d_m, N)$ is true if $d_1, \ldots, d_m$ are pariwise distances in N.
- $K^1, \ldots, K^m$ are RBF kernels comparing pariwise distances.

Pairwise distance kernel is defined as:

$$\mathcal{K}_{PDS}(N_i, N_j) = \sum_{\pi \in \Pi} e^{\frac{-\|\mathbf{d}^i - \pi(\mathbf{d}^j)\|^2}{\sigma^2}}$$

Connection with Spectral Score

Theorem

Let N_i and N_j be two sub-structures with spectral projection vectors f^i and f^j. Let $S(N_i, N_j)$ be the score of alignment of N_i and N_j, obtained by solving assignment problem. For large enough value of T such that all residues are matched.

$$\lim_{\beta \to 0} \mathcal{K}_{SS}(N_i, N_j))^{\beta} = e^{S(N_i,N_j)-kT}$$

Non-psd Kernel

$$\mathcal{K}_{LSS}(N_1, N_2) = \lim_{\beta \to 0} (\mathcal{K}_{SS}(N_1, N_2))^{\beta}$$

Kernels on Protein Structures

Kernels on Structures

For a set of proteins $X^1, \ldots, X^n$, define kernels:

$$\mathcal{K}_1(X^i, X^j) = \sum_{a=1}^{n_i} \sum_{b=1}^{n_j} \mathcal{K}_{SS}(N_a^i, N_b^j)$$

$$\mathcal{K}_2(X^i, X^j) = \sum_{a=1}^{n_i} \sum_{b=1}^{n_j} \mathcal{K}_{PDS}(N_a^i, N_b^j)$$

Kernels on Protein Structures

More Accurate Kernels

$$\mathcal{K}_3(X^i, X^j) = \sum_{a,b=1}^{n_i} \sum_{c,d=1}^{n_j} \mathcal{K}_{SS}(N_a^i, N_b^i) \times \mathcal{K}_{SS}(N_c^j, N_d^j) \times \mathcal{K}_{norm}((N_a^i, N_b^i), (N_c^j, N_d^j))$$

$$\mathcal{K}_4(X^i, X^j) = \sum_{a,b=1}^{n_i} \sum_{c,d=1}^{n_j} \mathcal{K}_{PDS}(N_a^i, N_b^i) \times \mathcal{K}_{SS}(N_c^j, N_d^j) \times \mathcal{K}_{norm}((N_a^i, N_b^i), (N_c^j, N_d^j))$$

where, $\mathcal{K}_{norm}((N_a^i, N_b^i), (N_c^j, N_d^j)) = e^{-\frac{(\|x_a^i - x_b^i\| - \|x_c^j - x_d^j\|)^2}{\sigma^2}}$.

Kernels on Protein Structures

Alignment Kernels

- Increase the accuracy of these kernels by using alignment information.
- Add neighborhood kernels for aligned residues:

$$\mathcal{K}_1^{Al}(X^i, X^j; \phi_{ij}) = \sum_{a | x_a^i \in \bar{X}^i} \mathcal{K}_{SS}(N_a^i, N_{\phi_{ij}(a)}^j)$$

- $\mathcal{K}_2^{Al}$ and $\mathcal{K}_3^{Al}$ are defined analogously using $\mathcal{K}_{LSS}$ and $\mathcal{K}_{PDS}$.

Kernels on Protein Structures

Alignment Kernels

- Make alignment kernels positive semidefinite:

$$\mathcal{K}_4^{Al}(P^i, P^j) = \begin{cases} \displaystyle\sum_{a \mid p_a^i \in \bar{P}^i} \mathcal{K}_{SS}(N_a^i, N_{\phi_{ij}(a)}^j) & \text{if } i \neq j \\ \displaystyle\sum_{b=1}^{M} \sum_{a \mid p_a^i \in \mathrm{dom}(\phi_{ib})} \mathcal{K}_{SS}(N_a^i, N_{\phi_{ib}(a)}^i) & \text{if } i = j \end{cases}$$

- $\mathcal{K}_5^{Al}$ and $\mathcal{K}_6^{Al}$ are defined analogously using $\mathcal{K}_{LSS}$ and $\mathcal{K}_{PDS}$.

Structure Kernels

Kernel	Positive Acc.	Negative Acc.	Total Acc.
K_1	69.67%	54.87%	62.27%
K_2	68.73%	61.33%	65.03%
K_3	56.13%	54.93%	55.53%
K_4	64.00%	60.93%	62.45%
CE	96.47%	63.33%	79.90%

Alignment Kernels

Kernel	Positive Acc.	Negative Acc.	Total Acc.
K_1^{Al}	74.33%	83.47%	78.90%
K_2^{Al}	79.13%	86.47%	82.80%
K_3^{Al}	73.87%	82.67%	78.27%
K_4^{Al}	91.87%	75.93%	83.90%
K_5^{Al}	80.67%	76.07%	78.37%
K_6^{Al}	88.53%	80.20%	84.37%
CE (NN)	96.47%	63.33%	79.90%

Outline

- Protein Structure Alignment
  - Introduction
  - Matrix and Graph Representation
  - Graph Matching
  - A new method
  - Results
- Protein Sub-Structure Alignment
  - Neighborhoods
  - Results
- Protein Structure Classification
  - Preliminaries
  - Kernels on Protein Structures
  - Results
- Learning with Uncertainty
  - Resolution-aware Protein Structure Classification

Summary

Problem

Build a protein structure classifier which takes resolution information into account.

Motivation

- Coordinates of atoms in protein structures are **resolved** to a particular accuracy.
- For example 1biaa1: 2.3Å, 1repc1: 2.6Å
- RMSDs of alignment between proteins are sometimes lower than the resolution.
- Example: 1biaa1 - 1repc1: 2.2Å
- Kernel values are perturbed due to perturbation in structure within resolution.

SVM Classification with uncertain kernels

SVM dual form

$$\max_{\alpha \in S_n, t} \alpha^\top e - \frac{1}{2} t \quad \text{s.t.} \quad \alpha^\top Y \mathbf{K} Y \alpha \leq t$$

where $S_n = \{\alpha | 0 \leq \alpha_i \leq C, \; \sum_{i=1}^n \alpha_i y_i = 0\}$ and $Y = diag(y_i)$.

SVM chance constrained form

$$\max_{t, \alpha \in S_n} \alpha^\top e - \frac{1}{2} t$$

$$\text{s.t. } Prob\left(\alpha^\top Y(\overline{\mathbf{K}} + Z) Y \alpha \leq t\right) \geq 1 - \epsilon$$

where Z is a matrix of random noise.

Gaussian Uncertainty

Theorem

- Z is an $n \times n$ random matrix.
- $Z_{ij} \sim N(0, \sigma_{ij}^2)$.
- $\mathbf{K} = \overline{\mathbf{K}} + Z$, where $\overline{\mathbf{K}}$ is kernel matrix.

For such a **K**, the chance constraint in previous formulation is satisfied if the following holds.

$$\sum_{ij} y_i y_j \alpha_i \alpha_j \overline{K}_{ij} - \Phi^{-1}(\epsilon) \|\Sigma * (\alpha\alpha^\top)\|_F \leq t$$

Interval Uncertainty

Theorem

- Z be a $n \times n$ random matrix with $E(Z_{ij}) = 0$.
- $P(a_{ij} \leq Z_{ij} \leq b_{ij}) = 1$.
- $\mathbf{K} = \overline{\mathbf{K}} + Z$, where $\overline{\mathbf{K}}$ is kernel matrix.

For such a $\mathbf{K}$, the chance constraint in previous formulation is satisfied if the following holds.

$$\sum_{ij} y_i y_j \alpha_i \alpha_j \overline{K}_{ij} + \sqrt{2\log(1/\epsilon)} \sqrt{\sum_{ij} \beta_{ij} \alpha_i^2 \alpha_j^2} \leq t$$

where, $\beta_{ij} = l_{ij}^2 \gamma_{ij}^2$, $l_{ij} = \frac{b_{ij} - a_{ij}}{2}$, γ_{ij} is a function of a_{ij} and b_{ij}.

Robust SVM

Deterministic Optimization Problem

The chance constraint program proposed earlier for learning SVMs with uncertain kernels can be posed as:

$$\begin{aligned} \min_{t,\alpha \in S_n} \quad & \tfrac{1}{2}t - \textstyle\sum_i \alpha_i \\ \text{s.t.} \quad & \textstyle\sum_{ij} y_i y_j \alpha_i \alpha_j \overline{K}_{ij} + \kappa \sqrt{\textstyle\sum_{ij} \beta_{ij} \alpha_i^2 \alpha_j^2} \leq t \end{aligned}$$

where κ depends on ϵ. This problem is applicable for both Gaussian and interval uncertainties.

Solution of the above problem

- The solution method depends on the matrix $\beta = [\beta_{ij}]$.
- When β is rank one, the solution boils down to solving SVM with modified kernel.
- When β is PSD, the problem is a second order cone program (SOCP), and can be solved using SOCP solver.
- In the general case, the problem in non-convex and can be solved using a standard descent algorithm.

Results

	RSVM			SVM			MI
	QP	SOCP	QN	nominal	M	R	
	MajErr						
TA	72.67	73.56	82.78	62.89	71.11	71.67	72.11
F1	73.49	74.35	82.95	63.50	71.87	72.58	72.17
	RobustErr						
TA	27.11	50.33	66.44	34.56	22.00	61.56	20.11
F1	26.81	50.28	66.36	34.07	21.70	61.26	19.63
	NomErr						
TA	66.50	66.65	76.00	61.02	65.00	70.44	x
F1	65.13	65.16	75.80	60.86	64.48	67.58	x

Results

Observations

- Robust SVM performs better than nominal SVM on synthetic datasets generated using Gaussian, Uniform and Beta noise.
- Robust error RSVM-SOCP and RSVM-QN increases less rapidly than nominal SVM, as the uncertainty is increased.
- For resolution-aware protein structure classification, RSVM-QN outperforms nominal SVM, and SVM with multiple instance kernel on 15 SCOP superfamilies.

Bibliography

P E Bourne and I N Shindyalov.
Protein structure alignment by incremental combinatorial extension of optimal path.
Protein Engineering, 11(9):739–747, 1998.

David Haussler.
Convolution kernels on discrete structures.
Technical report, University of California, Santa Cruz, 1999.

Liisa Holm and Chris Sander.
Protein structure comparison by alignment of distance matrices.
Journal of Molecular Biology, 233:123–138, 1993.

Shinji Umeyama.
An eigendecomposition approach to weighted graph matching problems.
IEEE transactions on pattern analysis and machine intelligence, 10(5):695–703, 1988.

Publications

Discussed here

- **Projections for fast protein structure retrieval. Sourangshu Bhattacharya, Chiranjib Bhattacharyya and Nagasuma R. Chandra.** *BMC Bioinformatics. 2006 Dec 18;7 Suppl 5:S5.*
- **Comparison of protein structures by growing neighborhood alignments. Sourangshu Bhattacharya, Chiranjib Bhattacharyya and Nagasuma R Chandra.** *BMC Bioinformatics. 2007 Mar 6;8:77.*
- **Structural Alignment based Kernels for Protein Structure Classification. Sourangshu Bhattacharya, Chiranjib Bhattacharyya and Nagasuma R Chandra. In Proceedings of** *24th International Conference on Machine Learning (ICML), 2007.*
- ***Robust Formulations for Handling Uncertainty in Kernel Matrices.*** *Sahely Bhadra, Sourangshu Bhattacharya, Chiranjib Bhattacharyya, Aharon Ben-Tal. International Conference on Machine Learning (ICML), 2010.*

Thank you !

Questions ?