Machine Learning Models on Random Graphs

Haixuan Yang

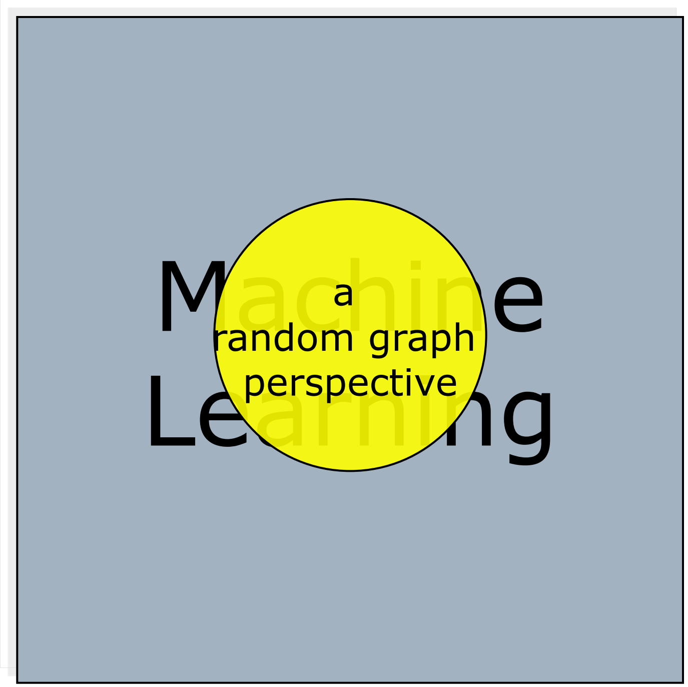
Supervisors: Prof. Irwin King and Prof. Michael R. Lyu

June 20, 2007

Outline

- Introduction
- Background
- Heat Diffusion Models on a Random Graph
- Predictive Random Graph Ranking
- Random Graph Dependency
- Conclusion and Future Work

Introduction



Machine Learning: help a computer “learn” knowledge from **data**.

Random Graph: an edge appears in a random way with a probability.

Viewpoint: **data** can be represented as random graphs in many situations.

A Formal Definition of Random Graphs

- A random graph RG=(U,P) is defined as a graph with a vertex set U in which
  - The probability of (i,j) being an edge is exactly p_{ij}, and
  - Edges are chosen independently
- Denote RG=P if U is clear in its context
- Denote RG=(U,E,P=(p_{ij})), emphasizing E={(i,j)| p_{ij} >0}
- Notes
  - Both (i,j) and (k,l) exist with a probability of p_{ij} p_{kl}
  - Remove the expectation notation, i.e., denote E(x) as x
  - Set p_{ii}=1

Random Graphs and Ordinary Graphs

- A weighted graph is different from random graphs
  - In a random graph, p_{ij} is in [0 1], the probability that (i,j) exists
  - In a random graph, there is the expectation of a variable.
- Under the assumption of independent edges, all graphs can be considered as random graphs
  - Weighted graphs can be mapped to random graphs by normalization
  - An undirected graph is a special random graph
    - $p_{ij} = p_{ji}$, p_{ij}=0 or 1
  - A directed graph is a special random graph
    - p_{ij} =0 or 1

Data Mapped to Random Graphs

- Web pages are nodes of a random graph
- Data points can be mapped to nodes of a random graph
  - A set of continuous attributes can generate a random graph by defining a probability between two data points
  - A set of discrete attributes can generate an equivalence relation

Equivalence Relations

- Definition: A binary relation ρ on a set U is called an equivalence relation if ρ satisfies

$$\begin{aligned}&\text{Reflexivity: } \forall a \in U, (a,a) \in \rho.\\ &\text{Symmetry: } \forall a,b \in U, (a,b) \in \rho \Rightarrow (b,a) \in \rho.\\ &\text{Transitivity: } \forall a,b,c \in U, (a,b) \in \rho \wedge (b,c) \in \rho \Rightarrow (a,c) \in \rho.\end{aligned}$$

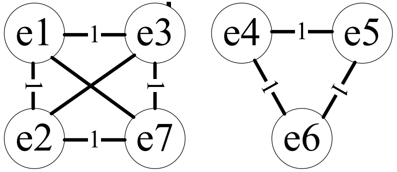
- An equivalence relation is a special random graph
  - An edge (a,b) exists with probability one if a and b have the relation, and zero otherwise
- A set P of discrete attributes can generate an equivalence relation by

$$IND(P) = \{(x,y) \in U \times U \mid (\forall a \in P)\ a(x) = a(y)\}.$$

An Example

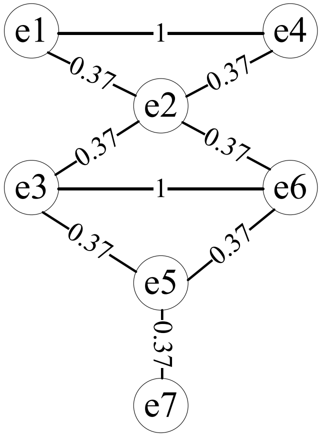
Attribute Object	Headache (a)	Muscle Pain (b)	Temperature (c)	Influenza (d)
e1	Y	Y	0	N
e2	Y	Y	1	Y
e3	Y	Y	2	Y
e4	N	Y	0	N
e5	N	N	3	N
e6	N	Y	2	Y
e7	Y	N	4	Y

{a} induces an equivalence relation



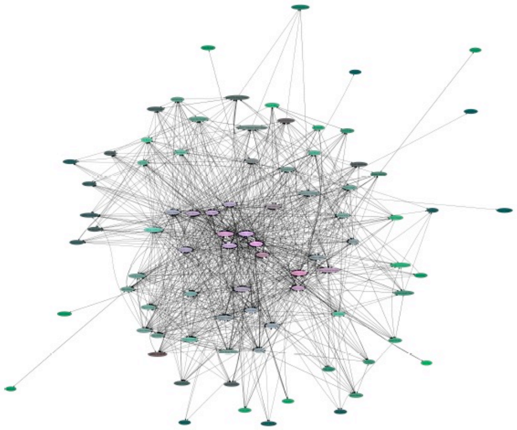
{c} generates a random graph

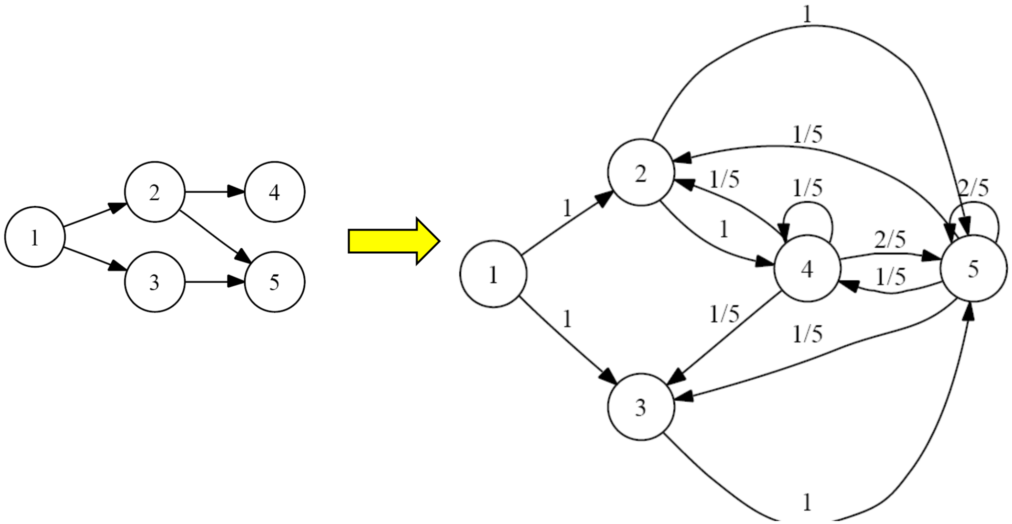
$$p(x,y) = \begin{cases} e^{-|c_1-c_2|}, & \text{if } e^{-|c_1-c_2|} > 0.2, \\ 0, & \text{elsewise.} \end{cases}$$



Another Example

- Web pages form a random graph because of the random existence of links
- A part of the whole Web pages can be predicted by a random graph





Nodes 1, 2, and 3: visited

Nodes 4 and 5: unvisited

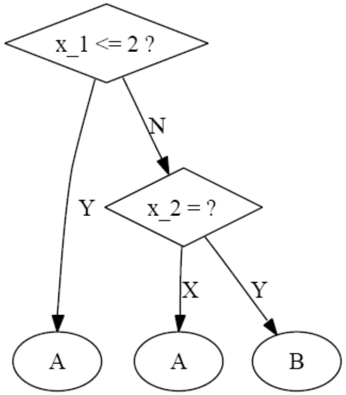
Machine Learning Background

- Three types of learning methods
  - Supervised Learning (SVM, RLS, MPM, **Decision Trees**, and etc.)
  - Semi-supervised Learning (TSVM, LapSVM, **Graph-based Methods**, and etc.)
  - Unsupervised Learning (PCA, ICA, ISOMAP, LLE, EigenMap, **Ranking**, and etc.)

Machine Learning Background

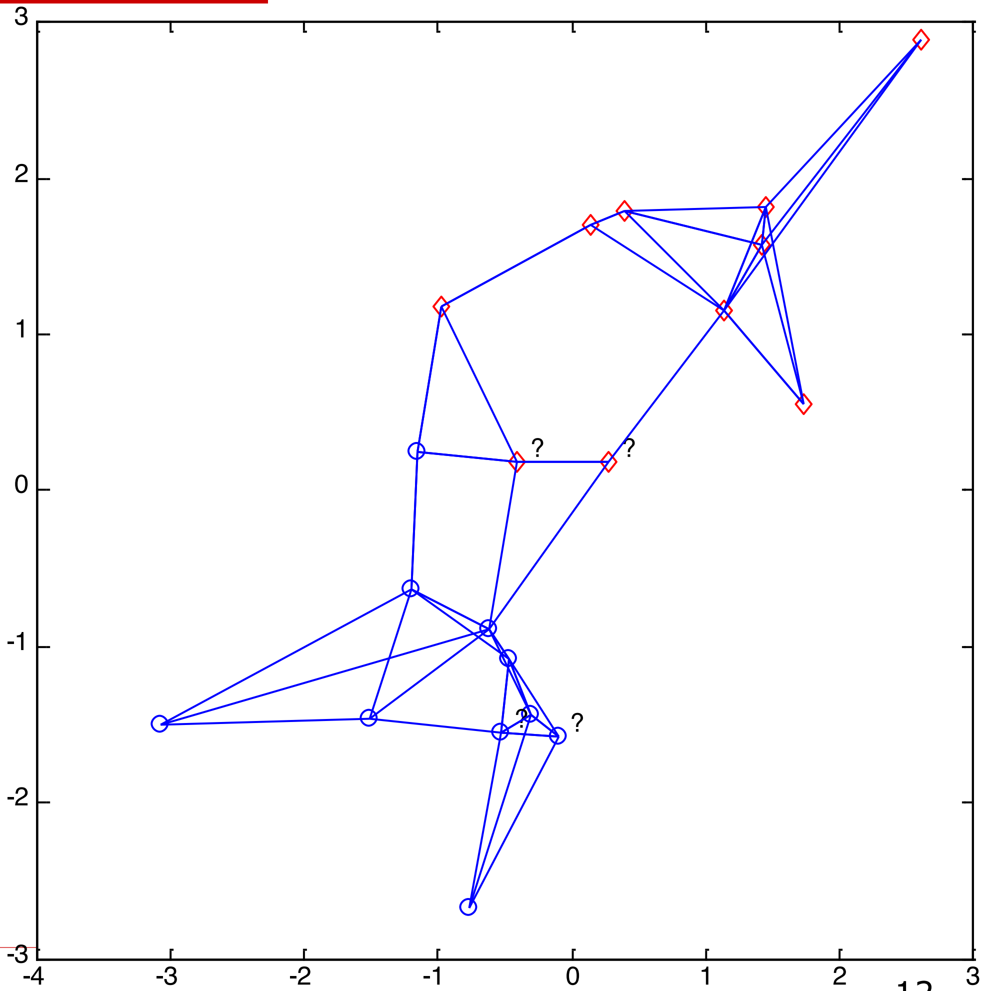
- Decision Trees
  - C4.5 employs the conditional entropy to select the most informative attribute

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
x_1	1.0	2.0	3.9	4.0	5.0	5.1	7.0	8.0
x_2	Y	X	Y	Y	X	X	Y	X
y	A	A	B	B	A	A	B	B



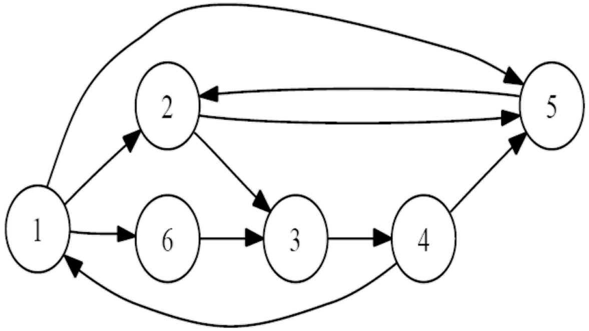
Machine Learning Background

- Graph-based Semi-supervised Learning Methods
  - Label the unlabeled examples on a graph
  - Traditional methods assuming the label smoothness over the graph



Machine Learning Background

- Ranking
  - It extracts order information from a Web graph



PageRank

$$x_i = \sum_{(j,i)\in E} 0.85 a_{ij} x_j + 0.15/n$$

$$a_{ij} = 1/d^+(j)$$

PageRank Results

1: 0.100
2 :0.255
3: 0.179
4: 0.177
5: 0.237
6: 0.053

2 > 5 > 3 >4 >1>6

Contributions

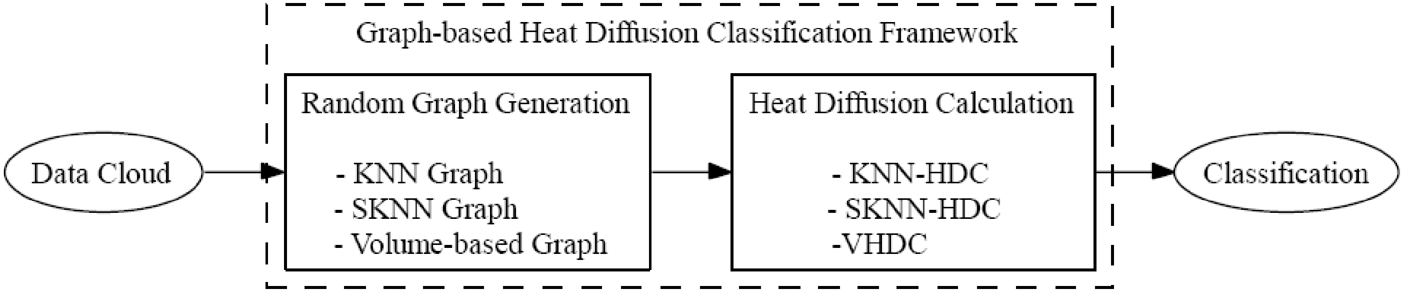
- Decision Trees
  - Improve the speed of C4.5 by one form of the proposed **random graph dependency**
  - Improve the accuracy of C4.5 by its another form
- Graph-based Semi-supervised Learning Methods
  - Establish **Heat Diffusion Models on random graphs**
- Ranking
  - Propose **Predictive Random Graph Ranking**: Predict a Web graph as a random graph, on which a ranking algorithm runs

Outline

- □ Introduction
- □ Background
- □ Heat Diffusion Models on a Random Graph
- □ Predictive Random Graph Ranking
- □ Random Graph Dependency
- □ Conclusion and Future Work

Heat Diffusion Models on Random Graphs

- An overview

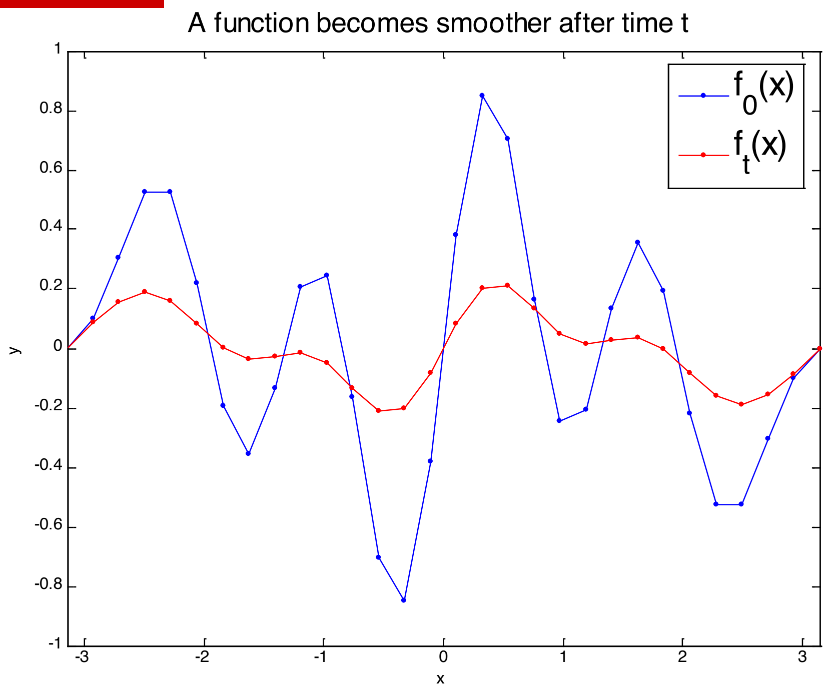


Heat Diffusion Models on Random Graphs

- Related Work
  - Tenenbaum et al. (Science 2000)
    - approximate the manifold by a KNN graph, and
    - reduce dimension by shortest paths
  - Belkin & Niyogi (Neural Computation 2003)
    - approximate the manifold by a KNN graph, and
    - reduce dimension by heat kernels
  - Kondor & Lafferty (NIPS 2002)
    - construct a diffusion kernel on an undirected graph, and
    - Apply it to SVM
  - Lafferty & Kondor (JMLR 2005)
    - construct a diffusion kernel on a special manifold, and
    - apply it to SVM

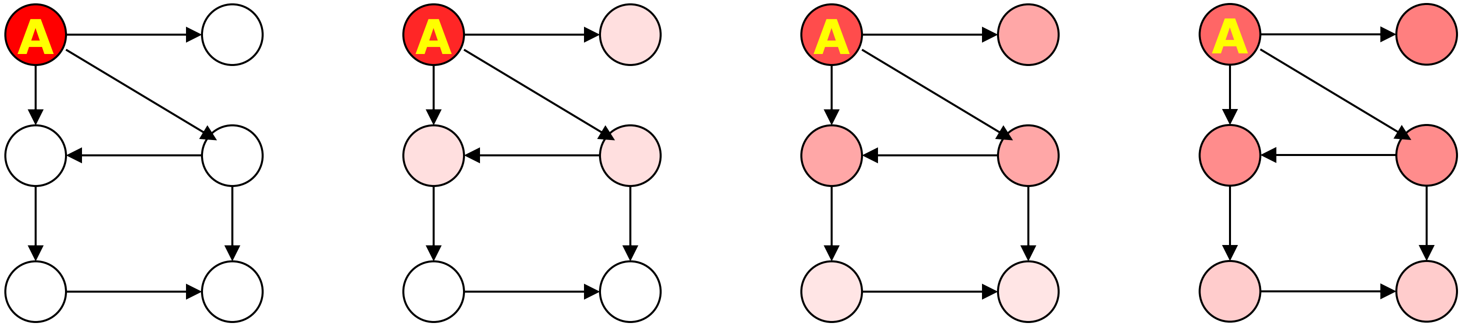
Heat Diffusion Models on Random Graphs

- Ideas we inherit
  - Local information
    - relatively accurate in a nonlinear manifold
  - Heat diffusion on a manifold
  - The approximate of a manifold by a graph
- Ideas we think differently
  - Heat diffusion imposes smoothness on a function
  - Establish the heat diffusion equation on a random graph
    - The broader settings enable its application on ranking on the Web pages
  - Construct a classifier by the solution directly



$$\begin{cases} \frac{\partial f}{\partial t} = \frac{1}{\sqrt{\det_g}} \sum_j \frac{\partial}{\partial x_j}\left(\sum_i g^{ij}\sqrt{\det g}\,\frac{\partial f}{\partial x_i}\right), \\ f(\mathbf{x},0) = f_0(\mathbf{x}). \end{cases}$$

A Simple Demonstration



Heat Diffusion Models on Random Graphs

- Notations

$G = (V, E, P)$: a directed random graph, where
$V = \{v_1, v_2, \ldots, v_n\}$,
$P = (p_{ij})$, where p_{ij} is the probability that edge (v_i, v_j) exists, and
$E = \{(v_i, v_j)\,|\text{there is an edge from } v_i \text{ to } v_j \text{ and } p_{ij} > 0\}$.

- Assumptions

1. The heat that i receives from j is proportional to the time period and the temperature difference between them

- Solution

$$f_i(t+\Delta t) - f_i(t) = \alpha \sum_{(j,i)\in E} p_{ji}(f_j(t) - f_i(t))\Delta t \Rightarrow \frac{d}{dt}\mathbf{f}(t) = \alpha H \mathbf{f}(t) \Rightarrow$$

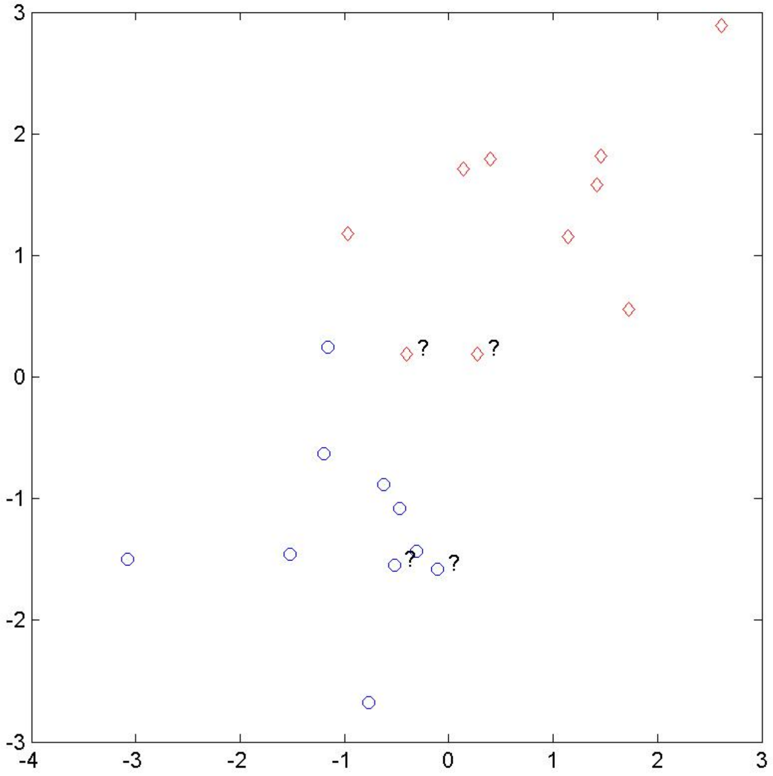
$$\mathbf{f}(t) = e^{\alpha t H}\mathbf{f}(0) = e^{\gamma H}\mathbf{f}(0)$$

$$H_{ij} = \begin{cases} -\sum_{k:(k,i)\in E} p_{ki}, & j = i; \\ p_{ji}, & (j,i) \in E; \\ 0, & \text{otherwise.} \end{cases}$$

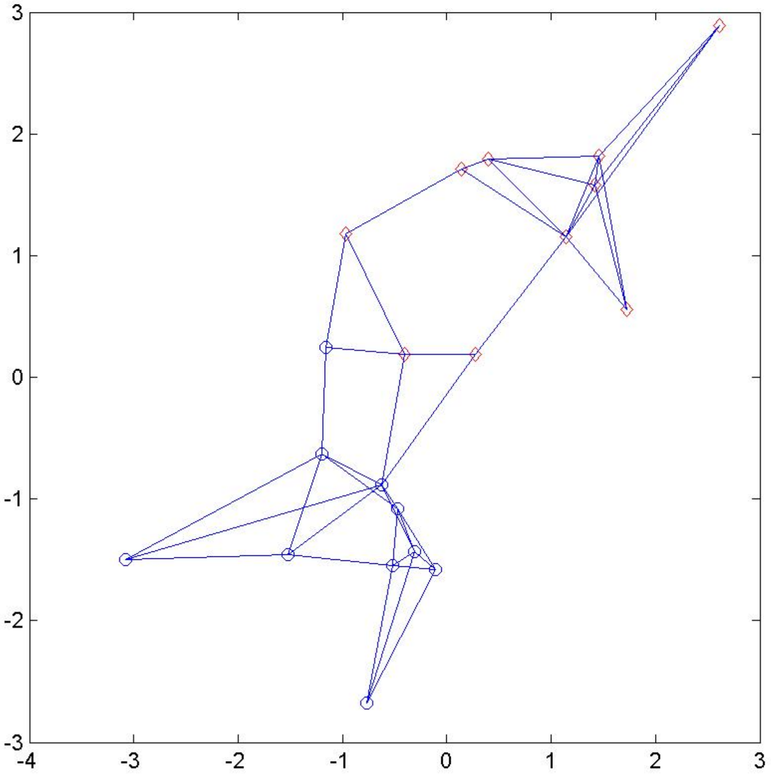
Graph-base Heat Diffusion Classifiers (G-HDC)

- Classifier
  1. Construct neighborhood graph
     - KNN Graph
     - SKNN Graph
     - Volume-based Graph
  2. Set initial temperature distribution
     - For each class k, f(i,0) is set as 1 if data is labeled as k and 0 otherwise
  3. Compute the temperature distribution for each class.
  4. Assign data j to a label q if j receives most heat from data in class q

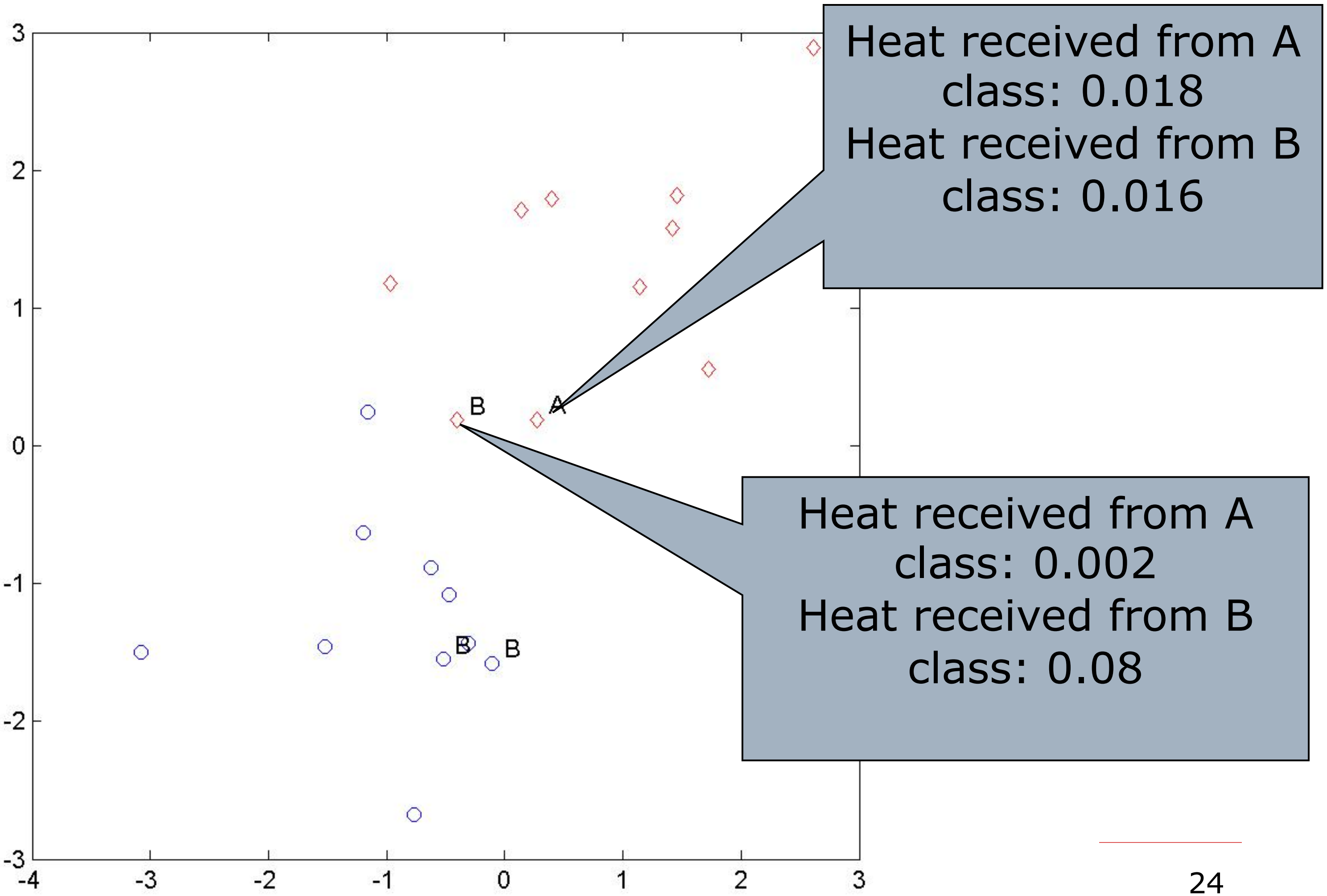
G-HDC: Illustration-1



G-HDC: Illustration-2



G-HDC: Illustration-3



Heat received from A class: 0.018
Heat received from B class: 0.016

Heat received from A class: 0.002
Heat received from B class: 0.08

Three Candidate Graphs

- KNN Graph
  - We create an edge from *j* to *i* if *j* is one of the *K* nearest neighbors of *i*, measured by the Euclidean distance

$$p_{ij} = \begin{cases} e^{-w_{ij}^2/\beta}, & \text{if } j \text{ is one } K \text{ nearest neighbors of } i\text{'s;} \\ 0, & \text{otherwise.} \end{cases}$$

- SKNN-Graph
  - We choose the smallest *K*n/2* undirected edges, which amounts to *K*n* directed edges

$$p_{ij} = \begin{cases} e^{-w_{ij}^2/\beta}, & \text{if } (j,i) \text{ is one of the shortest} K * n/2 \text{ edges;} \\ 0, & \text{otherwise.} \end{cases}$$

- Volume-based Graph

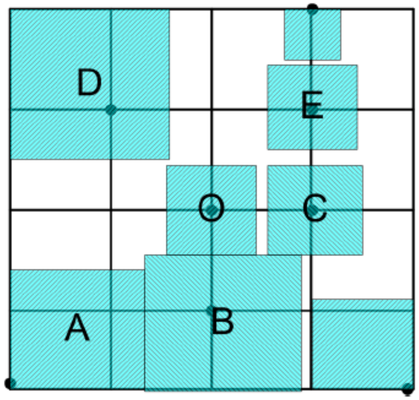
$$p_{ij} = \begin{cases} S(i)e^{-w_{ij}^2/\beta}V(j), & \text{if } j \text{ is one } K \text{ nearest neighbors of } i\text{'s;} \\ 0, & \text{otherwise.} \end{cases}$$

Volume-based Graph

- Justification by integral approximations

$$
\begin{aligned}
&(f(\mathbf{x}_i,t)-f(\mathbf{x}_i,t+\Delta t))/\Delta t\\
&\approx (f(\mathbf{x}_i,t)-\textstyle\int_M K_{\Delta t}(\mathbf{x}_i,\mathbf{y})f(\mathbf{y},t))/\Delta t\\
&\approx (f(\mathbf{x}_i,t)-(4\pi\Delta t)^{-\frac{m}{2}}\textstyle\int_M e^{-||\mathbf{x}_i-\mathbf{y}||^2/4\Delta t}f(\mathbf{y},t))/\Delta t\\
&\approx (f(\mathbf{x}_i,t)-(4\pi\Delta t)^{-\frac{m}{2}}\sum_{(j,i)\in E} e^{-||\mathbf{x}_i-\mathbf{x}_j||^2/4\Delta t}f(\mathbf{x}_j,t)V(j))/\Delta t
\end{aligned}
$$

$$V(i)=\eta\min_{j:(j,i)\in E} w_{ij}^{\nu}/2+1/2n$$

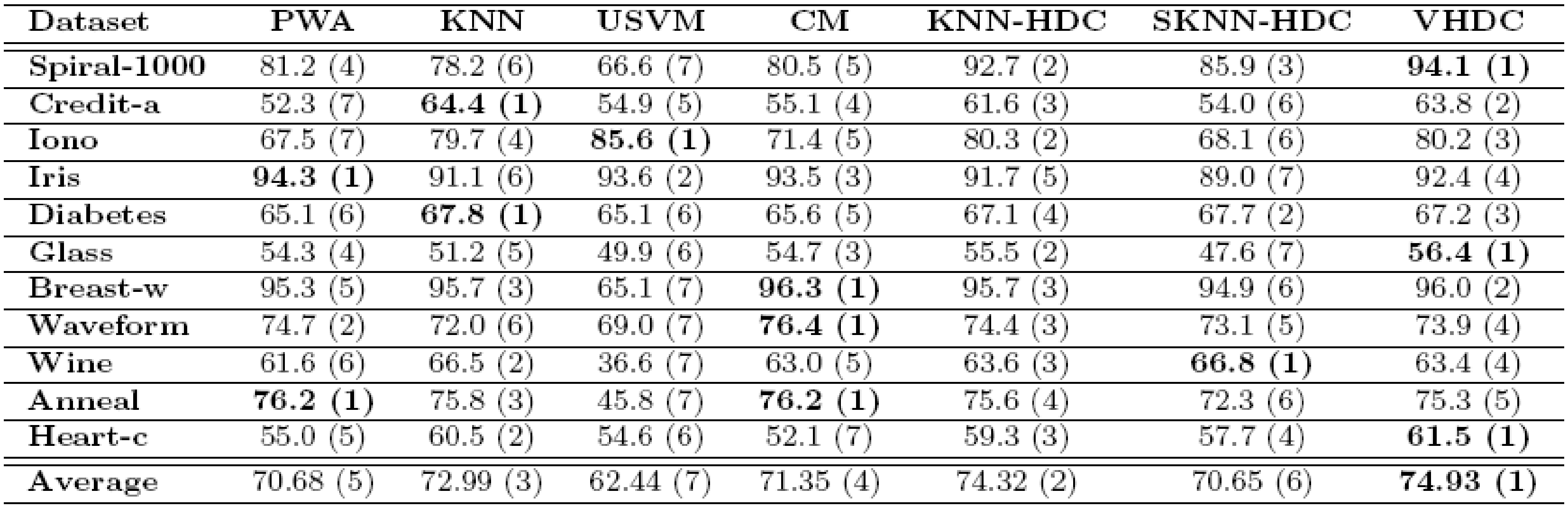


Experiments

- Experimental Setup
- Data Description
  - 1 artificial Data sets and 10 datasets from UCI
  - 10% for training and 90% for testing
- Comparison
  - Algorithms:
    - Parzen window
    - KNN
    - Transitive SVM (UniverSVM)
    - Consistency Method (CM)
    - KNN-HDC
    - SKNN-HDC
    - VHDC
  - Results: average of the ten runs

Dataset	Cases	Classes	Variable
Spiral-100	1000	2	3
Credit-a	666	2	6
Iono	351	2	34
Iris	150	3	4
Diabetes	768	2	8
Breast-w	683	2	9
Waveform	300	3	21
Wine	178	3	13
Anneal	898	5	6
Heart-c	303	2	5
Glass	214	6	9

Results



Dataset	PWA	KNN	USVM	CM	KNN-HDC	SKNN-HDC	VHDC
Spiral-1000	81.2 (4)	78.2 (6)	66.6 (7)	80.5 (5)	92.7 (2)	85.9 (3)	**94.1 (1)**
Credit-a	52.3 (7)	**64.4 (1)**	54.9 (5)	55.1 (4)	61.6 (3)	54.0 (6)	63.8 (2)
Iono	67.5 (7)	79.7 (4)	**85.6 (1)**	71.4 (5)	80.3 (2)	68.1 (6)	80.2 (3)
Iris	**94.3 (1)**	91.1 (6)	93.6 (2)	93.5 (3)	91.7 (5)	89.0 (7)	92.4 (4)
Diabetes	65.1 (6)	**67.8 (1)**	65.1 (6)	65.6 (5)	67.1 (4)	67.7 (2)	67.2 (3)
Glass	54.3 (4)	51.2 (5)	49.9 (6)	54.7 (3)	55.5 (2)	47.6 (7)	**56.4 (1)**
Breast-w	95.3 (5)	95.7 (3)	65.1 (7)	**96.3 (1)**	95.7 (3)	94.9 (6)	96.0 (2)
Waveform	74.7 (2)	72.0 (6)	69.0 (7)	**76.4 (1)**	74.4 (3)	73.1 (5)	73.9 (4)
Wine	61.6 (6)	66.5 (2)	36.6 (7)	63.0 (5)	63.6 (3)	**66.8 (1)**	63.4 (4)
Anneal	**76.2 (1)**	75.8 (3)	45.8 (7)	**76.2 (1)**	75.6 (4)	72.3 (6)	75.3 (5)
Heart-c	55.0 (5)	60.5 (2)	54.6 (6)	52.1 (7)	59.3 (3)	57.7 (4)	**61.5 (1)**
Average	70.68 (5)	72.99 (3)	62.44 (7)	71.35 (4)	74.32 (2)	70.65 (6)	**74.93 (1)**

Summary

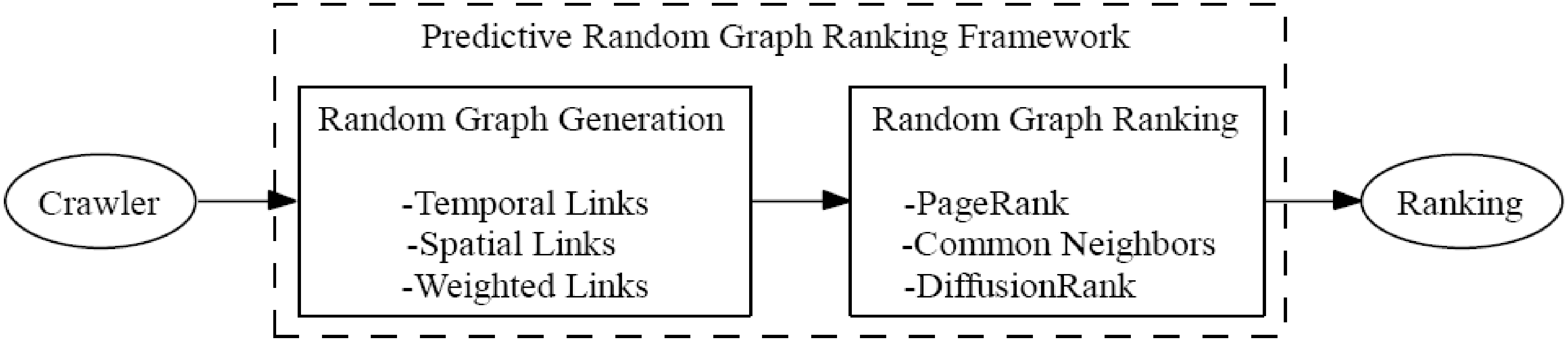
- Advantages
  - G-HDM has a closed form solution
  - VHDC gives more accurate results in a classification task
- Limitations
  - G-HDC depends on distance measures

Outline

- Introduction
- Background
- Heat Diffusion Models on a Random Graph
- Predictive Random Graph Ranking
- Random Graph Dependency
- Conclusion and Future Work

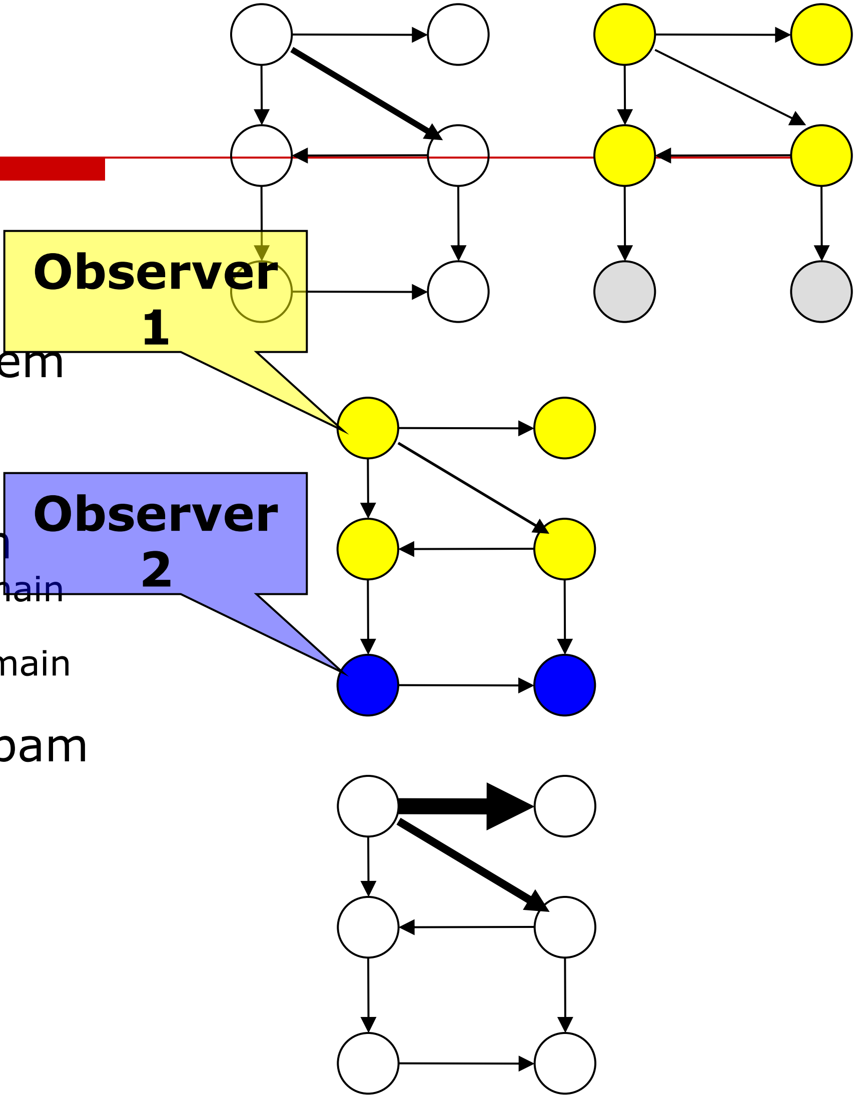
Predictive Random Graph Ranking

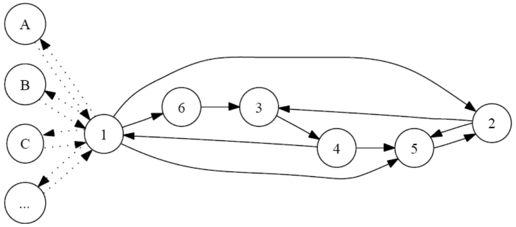
- An overview



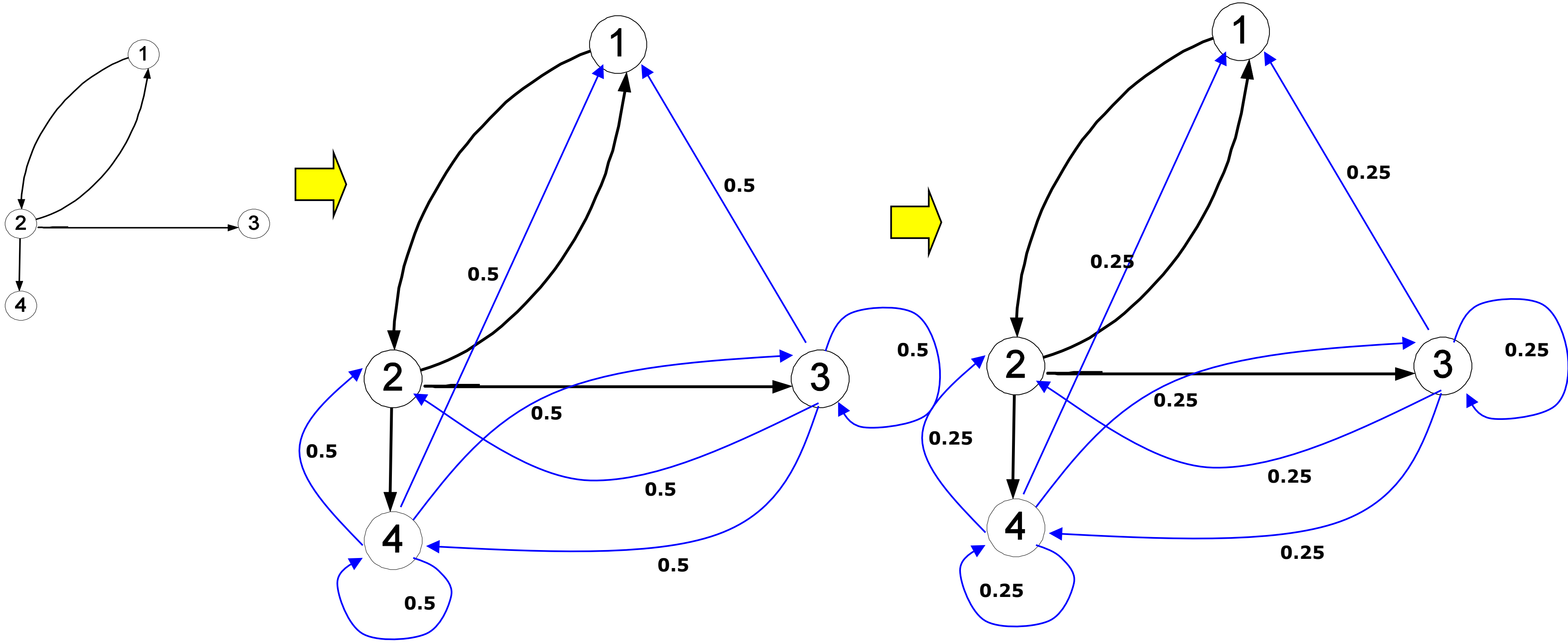
Motivations

- PageRank is inaccurate
  - The incomplete information
  - The Web page manipulations
- The incomplete information problem
  - The Web is dynamic
  - The observer is partial
  - Links are different
- The serious manipulation problem
  - About 70% of all pages in the .biz domain are spam
  - About 35% of the pages in the .us domain are spam
- PageRank is susceptible to web spam
  - Over-democratic
  - Input-independent





Random Graph Generation



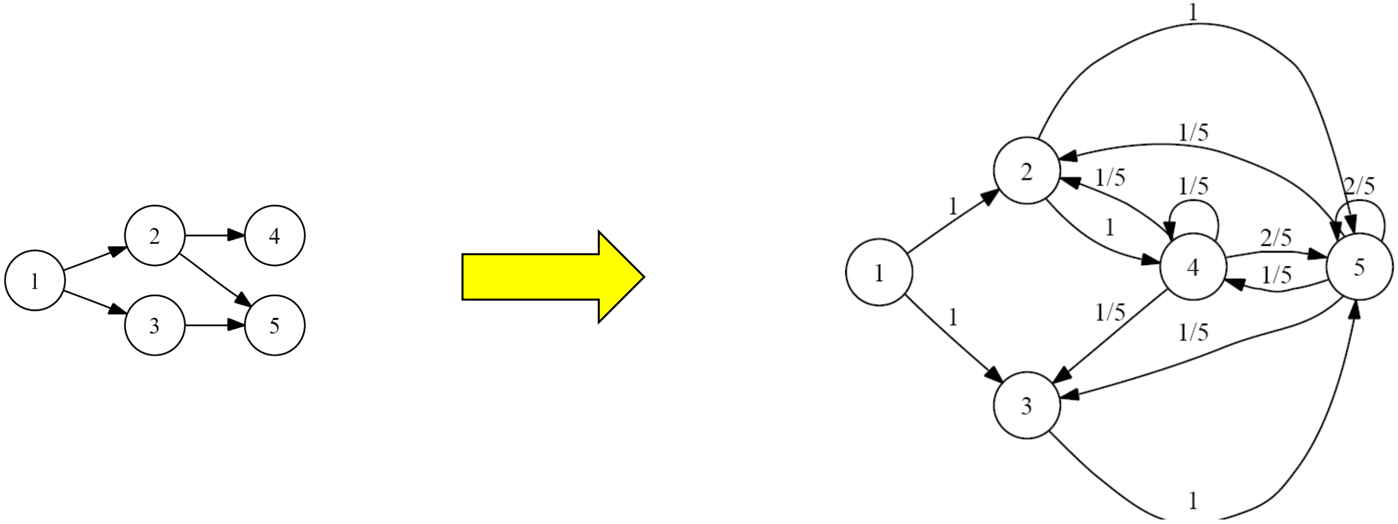
Nodes 1 and 2: visited

Nodes 3 and 4: unvisited

Estimation: Infer information about 4 nodes based on 2 true observations

Reliability: 2/4=0.5

Random Graph Generation



Nodes 1, 2, and 3: visited

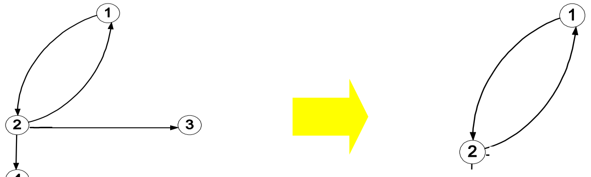
Nodes 4 and 5: unvisited

Estimation: Infer information about 5 nodes based on 3 true observations

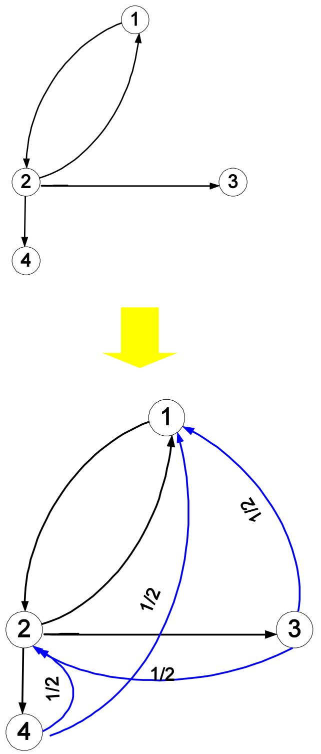
Reliability: 3/5

Related Work

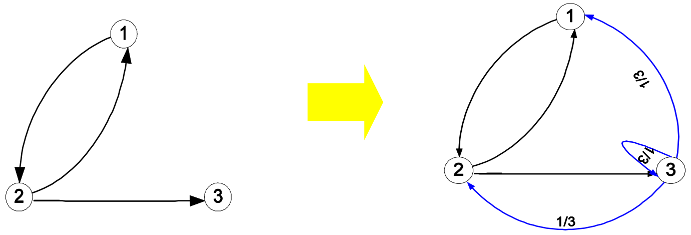
Page (1998)



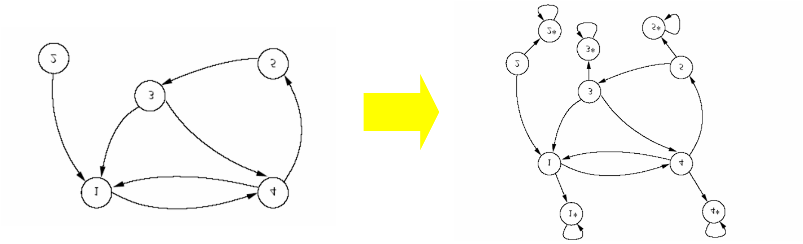
Eiron (2004)



Kamvar (2003)



Amati (2003)



Random Graph Ranking

- On a random graph RG=(V,P)
- PageRank

$$x_i = \sum_{(j,i)\in E} a_{ij} x_j \quad\Longrightarrow\quad x_i = \sum_j q_{ij} x_j$$

$$a_{ij} = 1/d^{+}(v_j) \qquad\qquad q_{ij} = p_{ji} / \textstyle\sum_k p_{jk}$$

- Common Neighbor $s(i,j) = \sum_k p_{ki} p_{kj}$
- Jaccard's Coefficient

$$\begin{aligned} s(i,j) &= |RI(v_i) \cap RI(v_j)| / |RI(v_i) \cup RI(v_j)| \\ &= \textstyle\sum_k p_{ki} p_{kj} / \sum_k (p_{ki} + p_{kj} - p_{ki} p_{kj}) \end{aligned}$$

DiffusionRank

- □ The heat diffusion model

$$\mathbf{f}(t) = e^{\alpha t H}\mathbf{f}(0) = e^{\gamma H}\mathbf{f}(0)$$

$$H_{ij} = \begin{cases} -\sum_{k:(k,i)\in E} p_{ki}, & j = i; \\ p_{ji}, & (j,i) \in E; \\ 0, & \text{otherwise.} \end{cases}$$

- □ On an undirected graph

$$\mathbf{f}(1) = e^{\gamma \mathbf{H}}\mathbf{f}(0)$$

- □ On a random directed graph

$$\mathbf{f}(1) = e^{\gamma \mathbf{R}}\mathbf{f}(0) \qquad R_{ij} = \begin{cases} -1, & j = i; \\ p_{ji}/RD^{+}(v_j), & j \neq i. \end{cases}$$

A Candidate for Web Spamming

- Initial temperature setting:
  - Select L trusted pages with highest Inverse PageRank score
  - The temperatures of these L pages are 1, and 0 for all others
- DiffusionRank is not over-democratic
- DiffusionRank is not input independent

Discuss γ

- γcan be understood as the thermal conductivity
- When $\gamma=0$, the ranking value is most robust to manipulation since no heat is diffused, but the Web structure is completely ignored
- When $\gamma= \infty$, DiffusionRank becomes PageRank, it can be manipulated easily
- When$\gamma=1$, DiffusionRank works well in practice

Computation Consideration

- Approximation of heat kernel

$$\mathbf{f}(1) = \boxed{(\mathbf{I} + \frac{\gamma}{N}\mathbf{R})^{N}}\,\mathbf{f}(0)$$

$$\Big\downarrow \text{ When N tends to infinity}$$

$$e^{\gamma\mathbf{R}}$$

- N=?
  - When γ=1, N>=30, the absolute value of real eigenvalues of $(\mathbf{I} + \frac{\gamma}{N}\mathbf{R})^{N} - e^{\gamma\mathbf{R}}$ are less than 0.01
  - When γ=1, N>=100, they are less than 0.005
  - We use N=100 in the thesis

Experiments

- Evaluate PRGR in the case that a crawler partially visit the Web
- Evaluate DiffusionRank for its Anti-manipulation effect.

Evaluation of PRGR

Data Description: The graph series are snapshots during the process of crawling pages restricted within cuhk.edu.hk in October, 2004.

Time t	1	2	3	4	5	6	7	8	9	10	11
Visited Pages	7712	78662	109383	160019	252522	301707	373579	411724	444974	471684	502610
Found Pages	18542	120970	157196	234701	355720	404728	476961	515534	549162	576139	607170

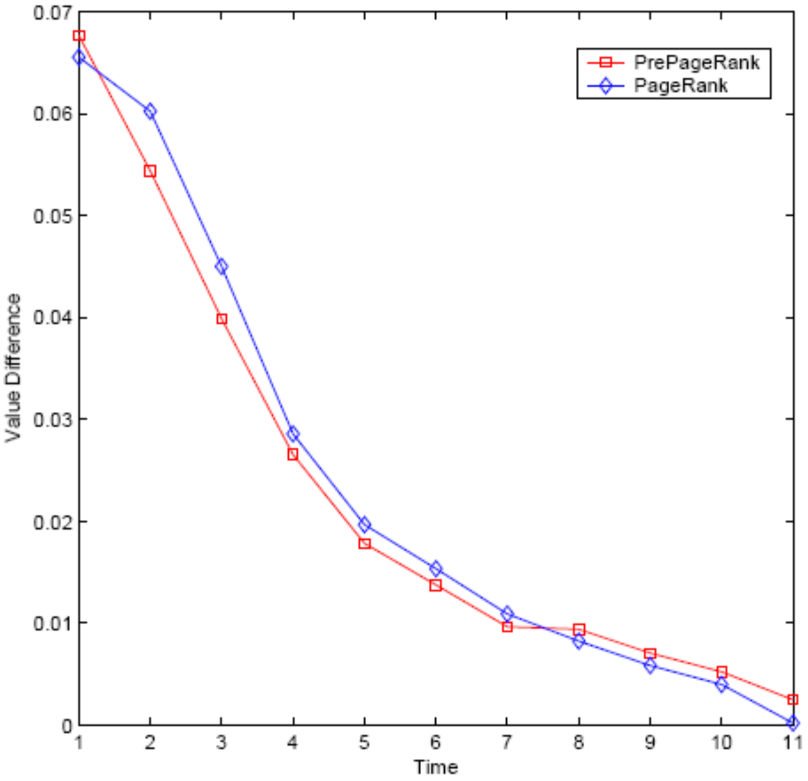
Methodology

- For each algorithm A, we have **At** and **PreAt**:

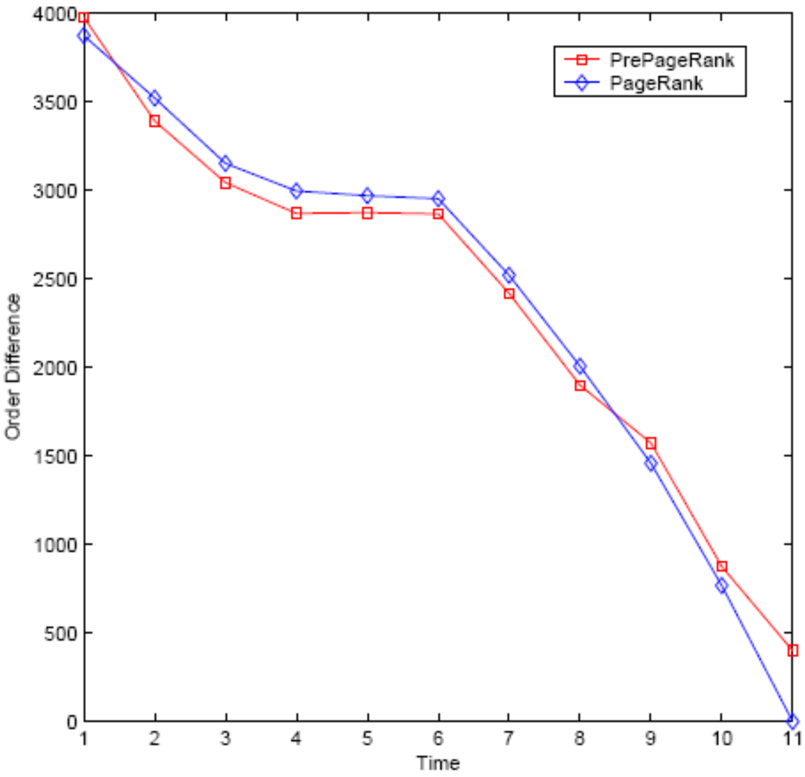
  At uses the random graph at time t generated by the Kamvar 2003.

  PreAt uses the random graph at time t generated by our method
- Compare the early results with **A11** by
  - Value Difference and
  - Order Difference

PageRank

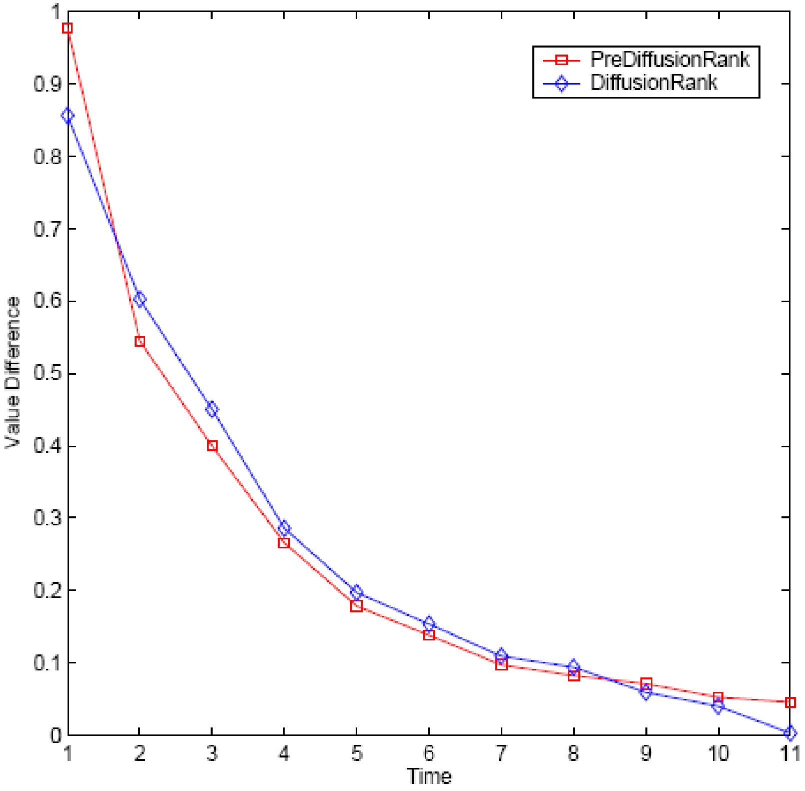


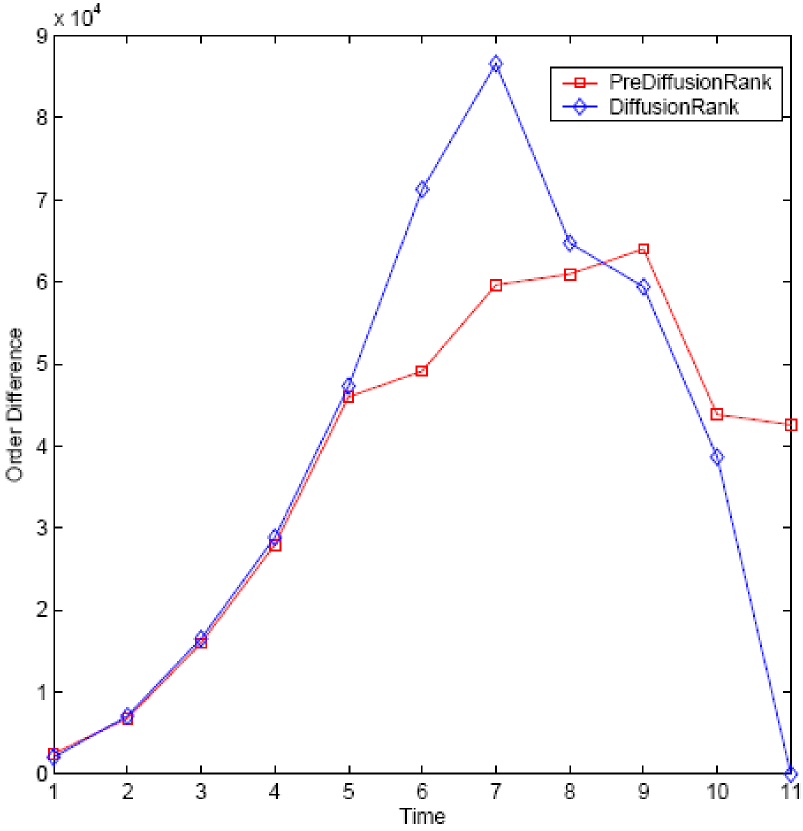
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
Value Difference
1
2
3
4
5
6
7
8
9
10
11
Time
PrePageRank
PageRank



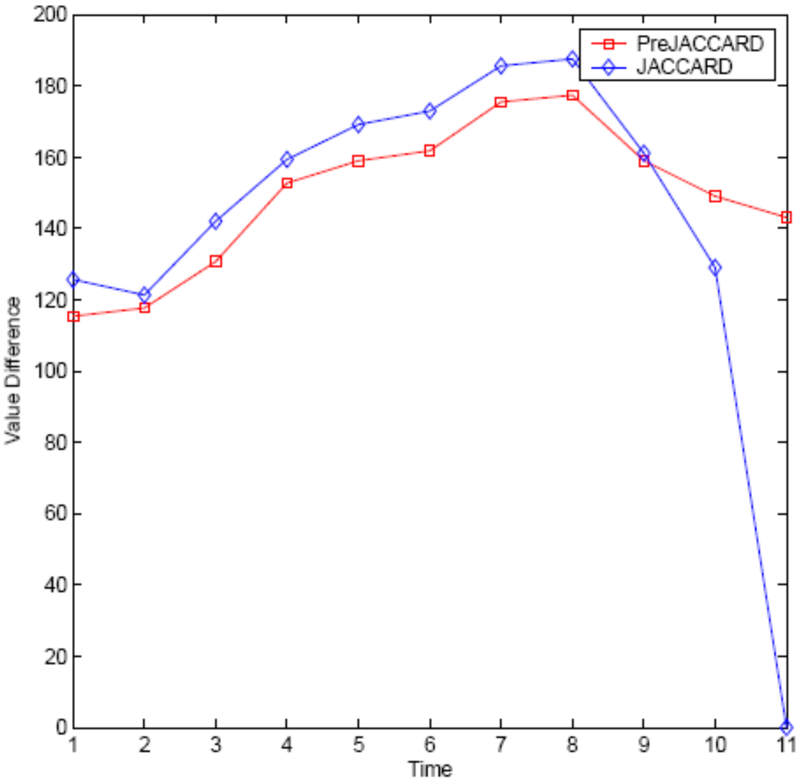
4000
3500
3000
2500
2000
1500
1000
500
0
Order Difference
1
2
3
4
5
6
7
8
9
10
11
Time
PrePageRank
PageRank

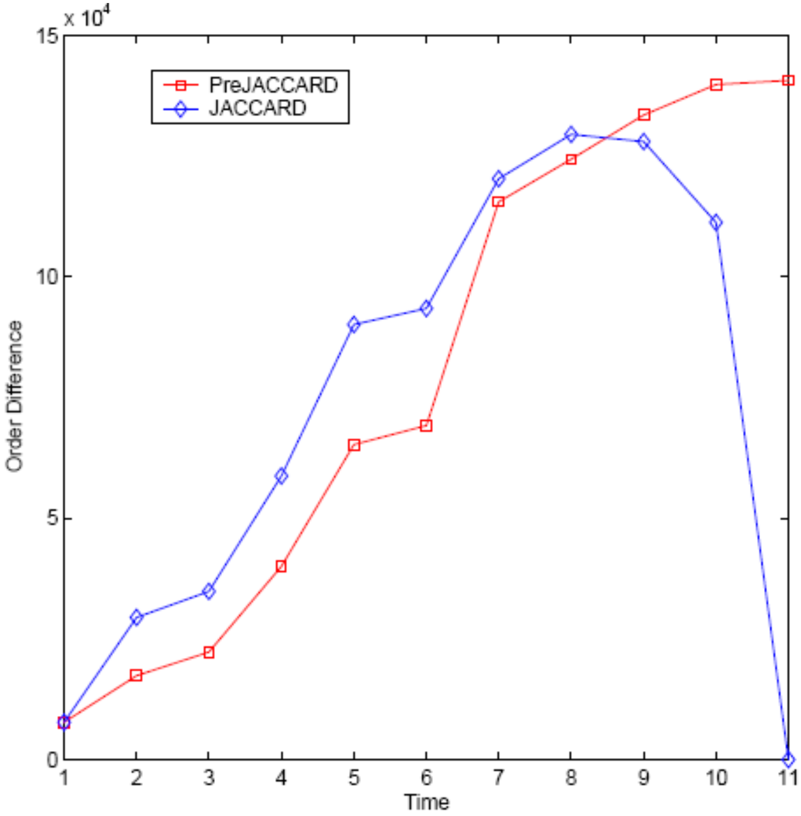
DiffusionRank



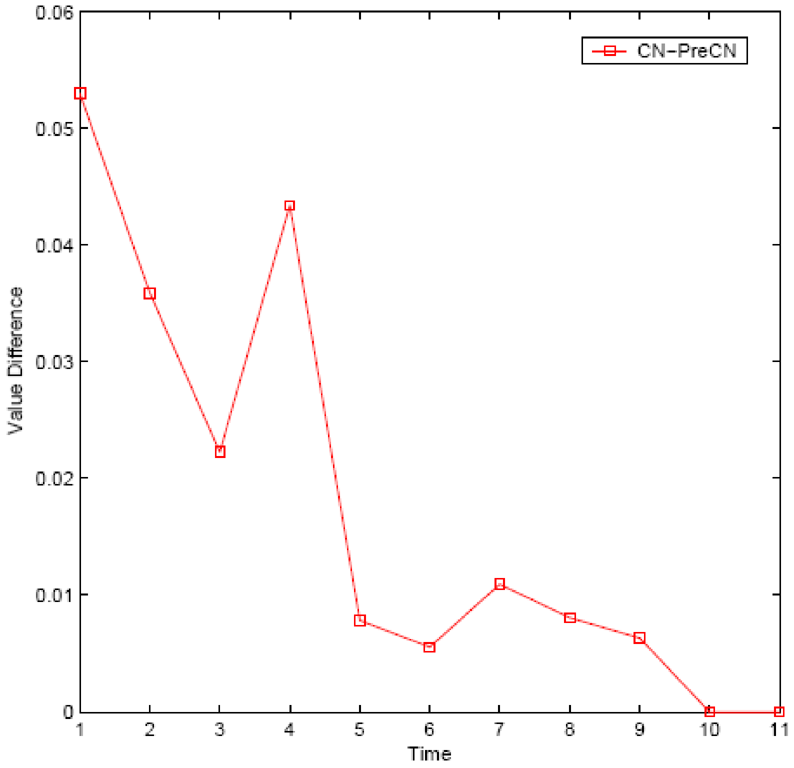


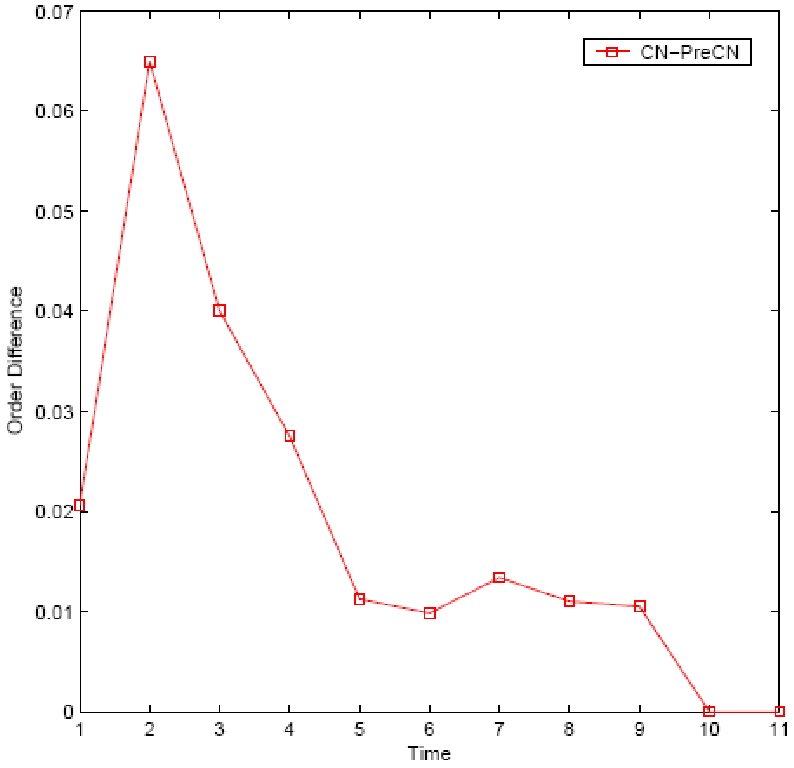
Jaccard's Coefficient





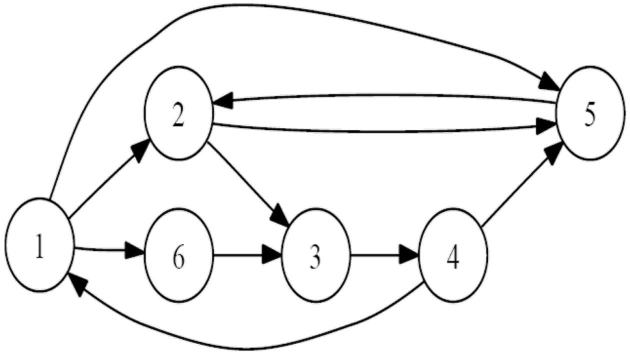
Common Neighbor

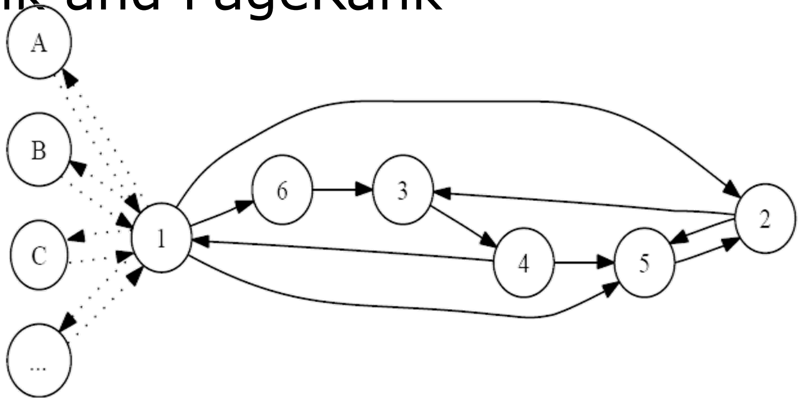




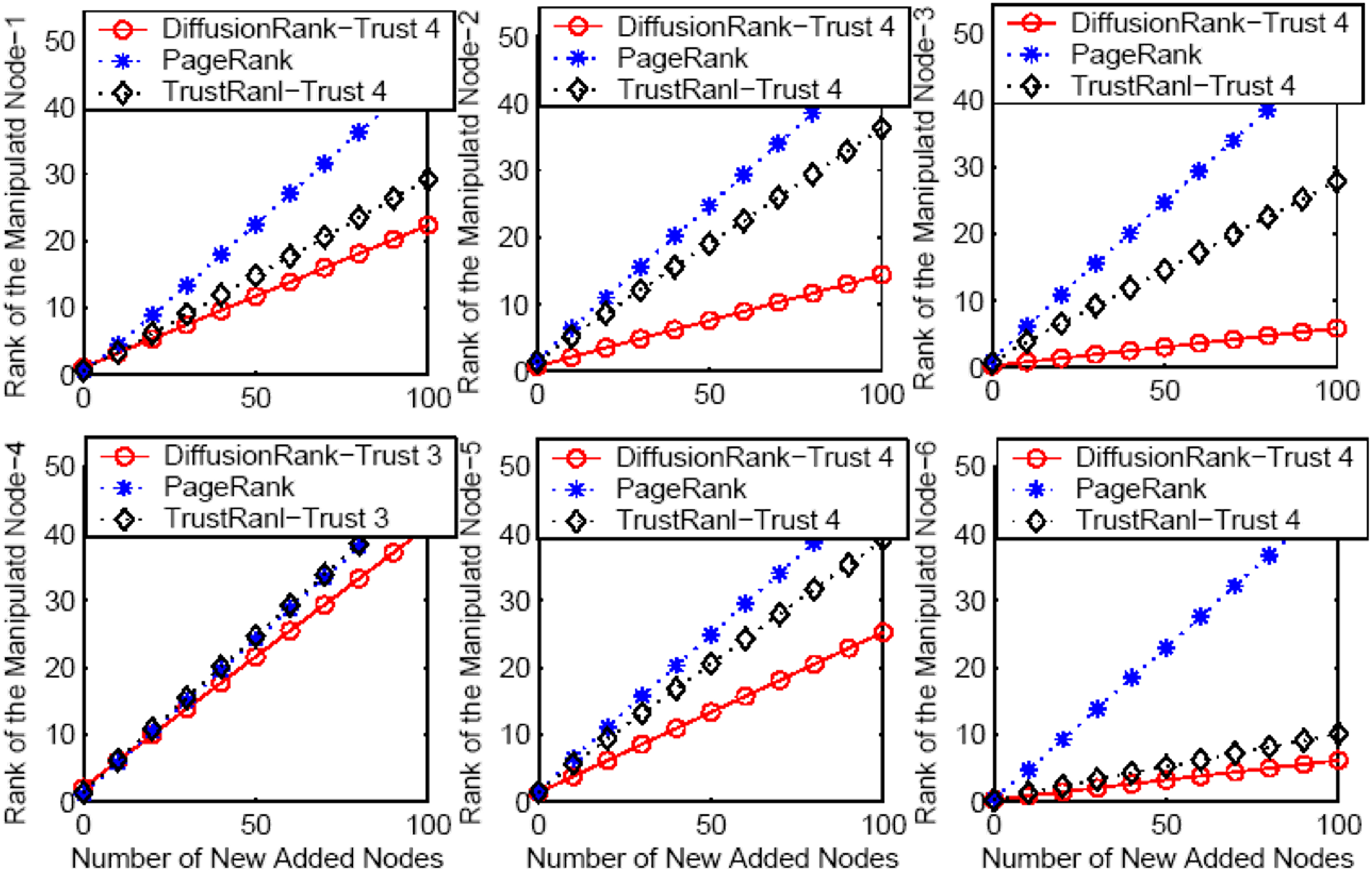
Evaluate DiffusionRank

- Experiments
  - Data:
    - a toy graph (6 nodes)
    - a middle-size real-world graph (18542 nodes)
    - a large-size real-world graph crawled from CUHK (607170 nodes)
  - Compare with TrustRank and PageRank

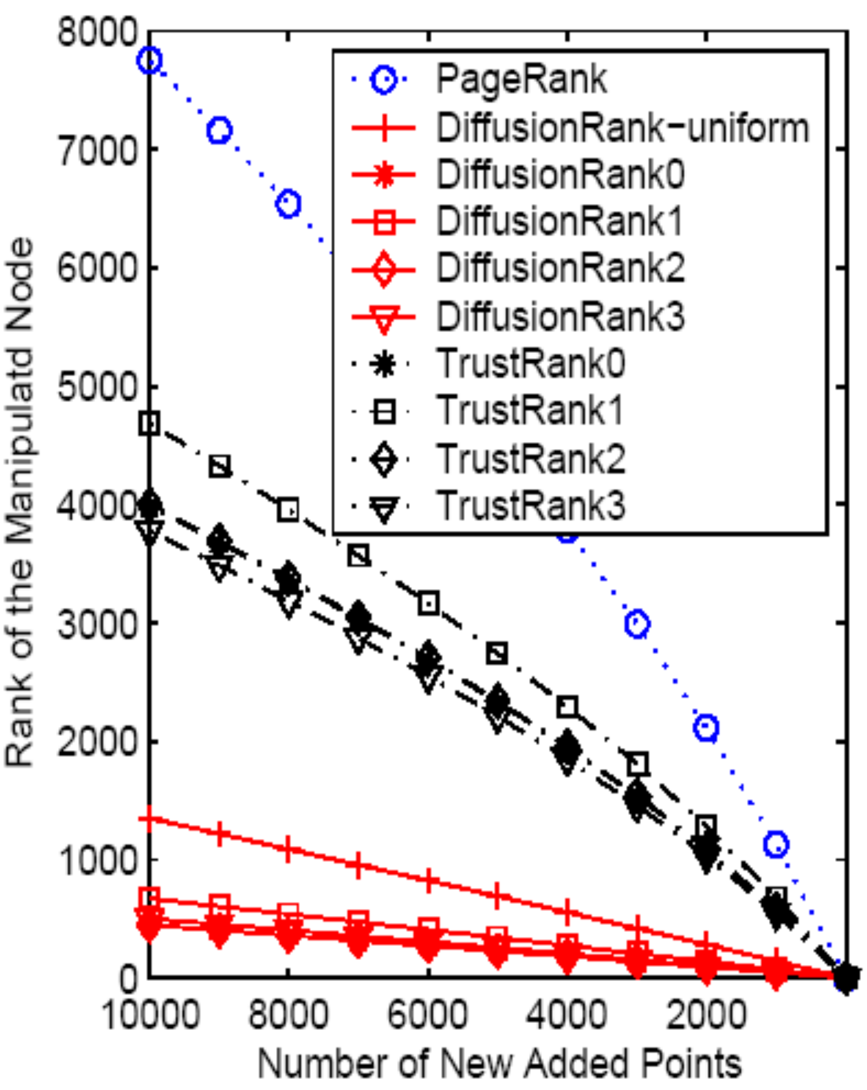


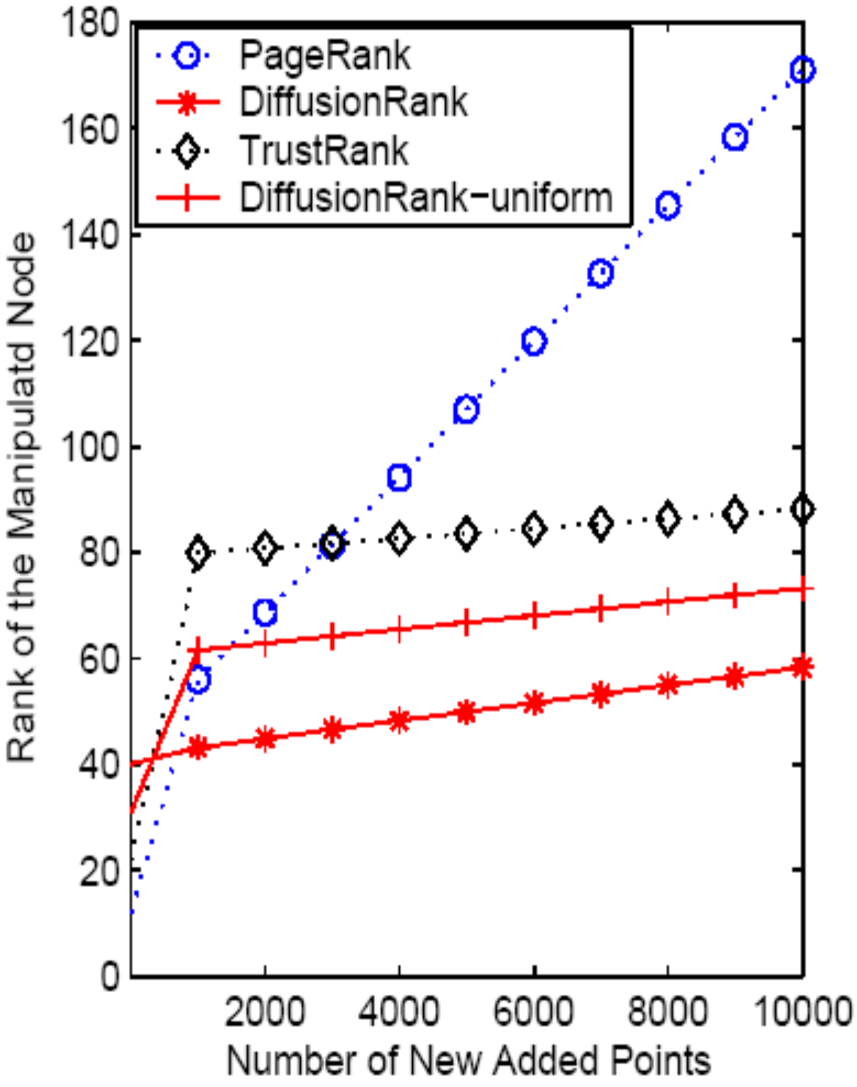


Anti-manipulation on the Toy Graph

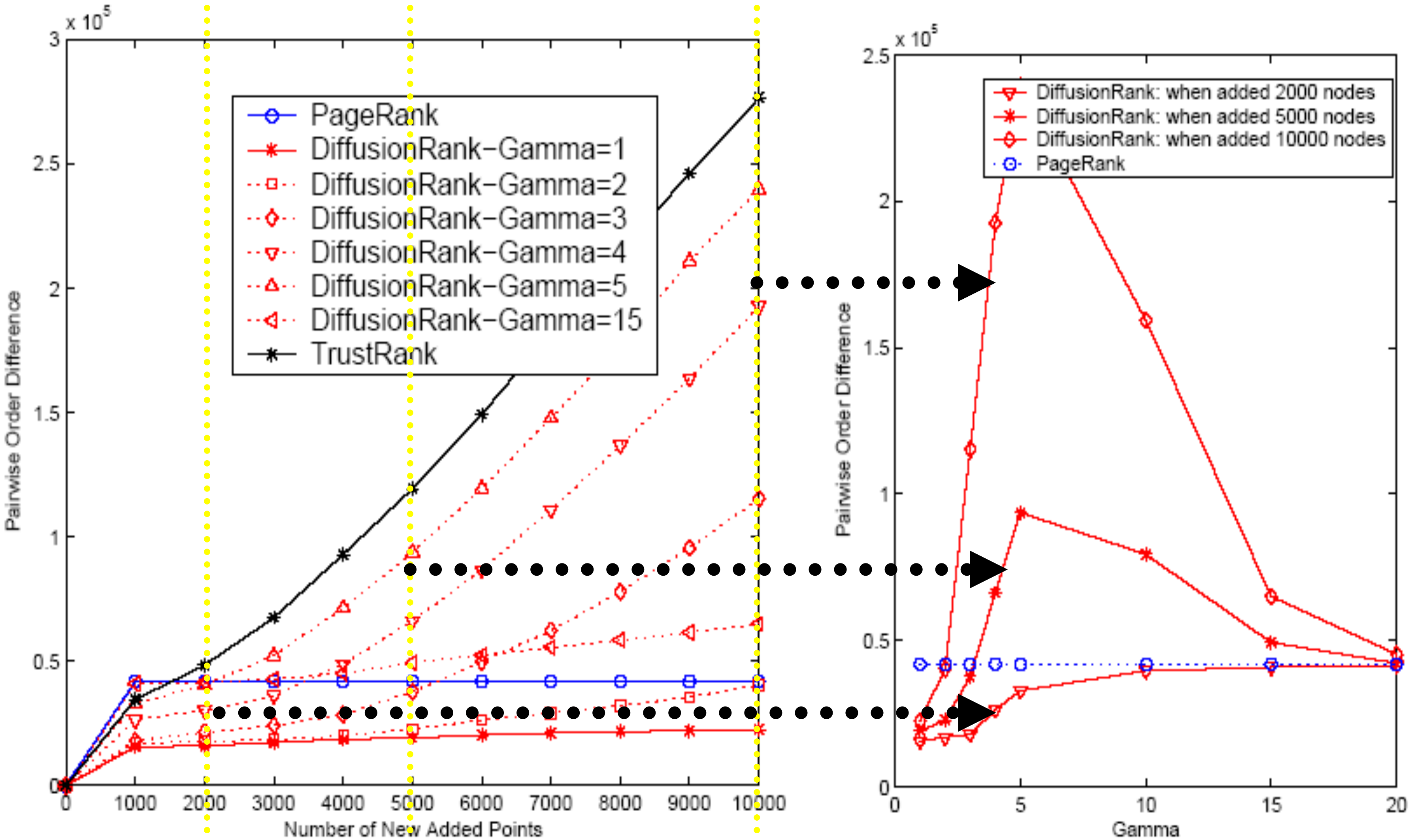


Anti-manipulation on the Middle-sized Graph and the Large-sized graph





Stability--the order difference between ranking results for an algorithm before it is manipulated and those after that



Summary

- PRGR extends the scope of some original ranking techniques, and significantly improves some of them
- DiffusionRank is a generalization of PageRank
- DiffusionRank has the effect of anti-manipulation

Outline

- □ Introduction
- □ Background
- □ Heat Diffusion Models on a Random Graph
- □ Predictive Random Graph Ranking
- □ Random Graph Dependency
- □ Conclusion and Future Work

An Overview

Theorem (Generality of $\Gamma_{\alpha}^{\varepsilon}(RG_2|RG_1)$)

$$\Gamma_{\alpha}^{\varepsilon}(RG_2|RG_1) = \begin{cases} \gamma(C,D), & \text{when } \varepsilon = 1, \alpha = -1, \\ & RG_2 = IND(D), \text{ and } RG_1 = IND(C), \\ \Gamma(C,D), & \text{when } \varepsilon = 0, \alpha = -1, \\ & RG_2 = IND(D), \text{ and } RG_1 = IND(C), \\ H(D|C), & \text{when } \varepsilon = 0, \alpha = 1, \\ & RG_2 = IND(D), \text{ and } RG_1 = IND(C), \\ H(RG_2|RG_1), & \text{when } \varepsilon = 0 \text{ and } \alpha = 1. \end{cases}$$

$\gamma(C,D)$	The measure used in Rough Set Theory
$H(D\|C)$	The measure used in C4.5 decision trees
$\Gamma(C,D)$	Employed to improve the speed of C4.5 decision trees
$H(RG_2\|RG_1)$	Employed to improve the accuracy of C4.5 decision trees
$\Gamma_{-1}^{0}(RG_2\|RG_1)$	Employed to search free parameter in KNN-HDC

Motivations

- The speed of C4.5
  - The fastest algorithm in terms of training among a group of 33 classification algorithms (Lim, 2000)
  - The speed of C4.5 will be improved from the viewpoint of information measure
    - The Computation of $\gamma(C,D)$ is fast, but it is not accurate
    - We inherit the merit of $\gamma(C,D)$ and increase its accuracy
- The prediction accuracy of the C4.5
  - Not statistically significantly different from the best among these 33 classification algorithms (Lim, 2000)
  - The accuracy will be improved
    - We will generalize $H(D|C)$ from equivalence relations to random graphs

An Overview

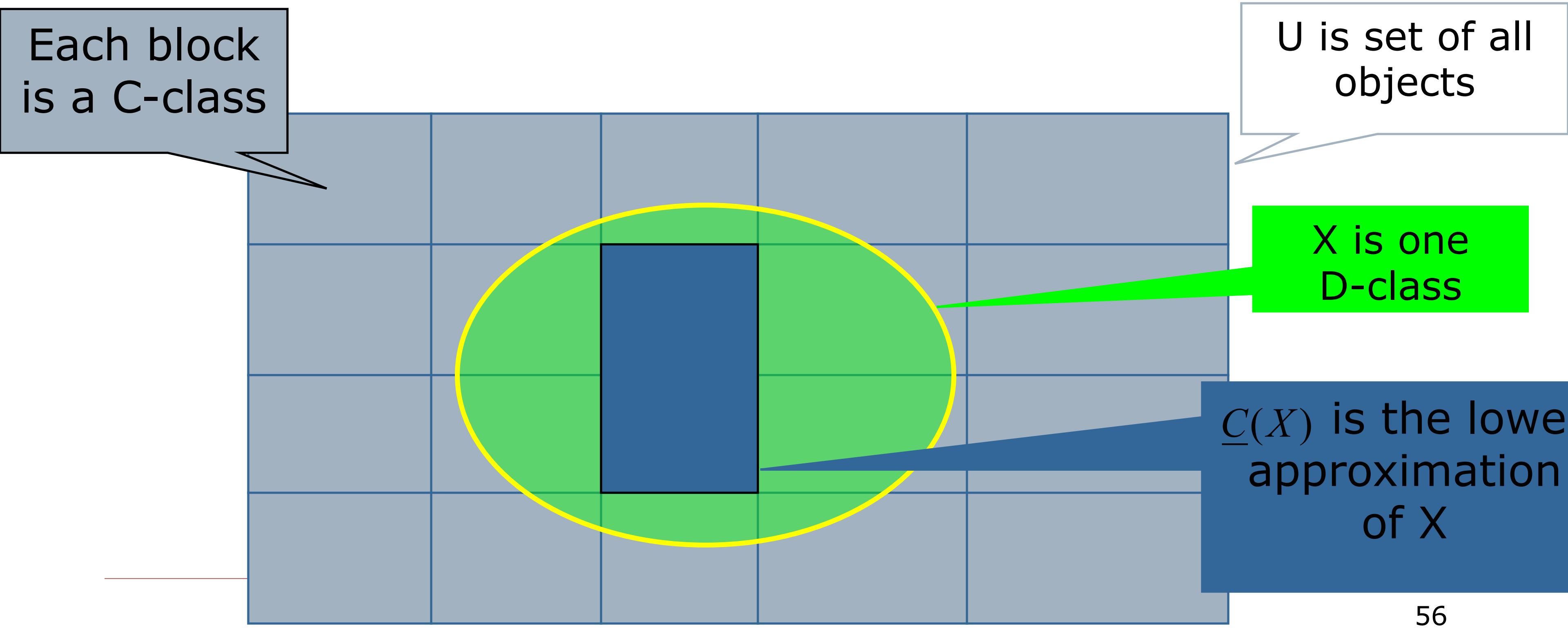
Theorem (Generality of $\Gamma_{\alpha}^{\varepsilon}(RG_2|RG_1)$)

$$\Gamma_{\alpha}^{\varepsilon}(RG_2|RG_1) = \begin{cases} \gamma(C,D), & \text{when } \varepsilon = 1, \alpha = -1, \\ & RG_2 = IND(D), \text{ and } RG_1 = IND(C), \\ \Gamma(C,D), & \text{when } \varepsilon = 0, \alpha = -1, \\ & RG_2 = IND(D), \text{ and } RG_1 = IND(C), \\ H(D|C), & \text{when } \varepsilon = 0, \alpha = 1, \\ & RG_2 = IND(D), \text{ and } RG_1 = IND(C), \\ H(RG_2|RG_1), & \text{when } \varepsilon = 0 \text{ and } \alpha = 1. \end{cases}$$

$\gamma(C,D)$	The measure used in Rough Set Theory
$H(D\|C)$	The measure used in C4.5 decision trees
$\Gamma(C,D)$	Employed to improve the speed of C4.5 decision trees
$H(RG_2\|RG_1)$	Employed to improve the accuracy of C4.5 decision trees
$\Gamma_{-1}^{0}(RG_2\|RG_1)$	Employed to search free parameter in KNN-HDC

Original Definition of γ

$$\gamma(C,D)=\frac{|POS(C,D)|}{|U|} \quad \text{where} \quad POS(C,D)=\bigcup_{X\in U/D}\underline{C}(X)$$



Each block is a C-class

U is set of all objects

X is one D-class

$\underline{C}(X)$ is the lowe approximation of X

An Example for the Inaccuracy of γ

Attribute Object	Headache (a)	Muscle Pain (b)	Temperature (c)	Influenza (d)
e1	Y	Y	0	N
e2	Y	Y	1	Y
e3	Y	Y	2	Y
e4	N	Y	0	N
e5	N	N	3	N
e6	N	Y	2	Y
e7	Y	N	4	Y

Let C={a}, D={d}, then γ(C,D)=0

An Overview

Theorem (Generality of $\Gamma^{\varepsilon}_{\alpha}(RG_2|RG_1)$)

$$\Gamma^{\varepsilon}_{\alpha}(RG_2|RG_1) = \begin{cases} \gamma(C,D), & \text{when } \varepsilon = 1, \alpha = -1, \\ & RG_2 = IND(D), \text{ and } RG_1 = IND(C), \\ \Gamma(C,D), & \text{when } \varepsilon = 0, \alpha = -1, \\ & RG_2 = IND(D), \text{ and } RG_1 = IND(C), \\ H(D|C), & \text{when } \varepsilon = 0, \alpha = 1, \\ & RG_2 = IND(D), \text{ and } RG_1 = IND(C), \\ H(RG_2|RG_1), & \text{when } \varepsilon = 0 \text{ and } \alpha = 1. \end{cases}$$

$\gamma(C,D)$	The measure used in Rough Set Theory
$\boxed{H(D\|C)}$	The measure used in C4.5 decision trees
$\Gamma(C,D)$	Employed to improve the speed of C4.5 decision trees
$H(RG_2\|RG_1)$	Employed to improve the accuracy of C4.5 decision trees
$\Gamma^{0}_{-1}(RG_2\|RG_1)$	Employed to search free parameter in KNN-HDC

The Conditional Entropy Used in C4.5

$$H(D \mid C) = -\sum_{c}\sum_{d} \Pr[c] \cdot \Pr[d \mid c] \cdot \log_2(\Pr[d \mid c])$$

$$= -\sum_{c} \Pr[c] \cdot \sum_{d} \Pr[d \mid c] \cdot \log_2(\Pr[d \mid c])$$

c: vectors consisting of the values of attributes in C

d: vectors consisting of the values of attributes in D

An Overview

Theorem (Generality of $\Gamma_{\alpha}^{\varepsilon}(RG_2|RG_1)$)

$$\Gamma_{\alpha}^{\varepsilon}(RG_2|RG_1) = \begin{cases} \gamma(C,D), & \text{when } \varepsilon = 1, \alpha = -1, \\ & RG_2 = IND(D), \text{ and } RG_1 = IND(C), \\ \Gamma(C,D), & \text{when } \varepsilon = 0, \alpha = -1, \\ & RG_2 = IND(D), \text{ and } RG_1 = IND(C), \\ H(D|C), & \text{when } \varepsilon = 0, \alpha = 1, \\ & RG_2 = IND(D), \text{ and } RG_1 = IND(C), \\ H(RG_2|RG_1), & \text{when } \varepsilon = 0 \text{ and } \alpha = 1. \end{cases}$$

$\gamma(C,D)$	The measure used in Rough Set Theory
$H(D\|C)$	The measure used in C4.5 decision trees
$\boxed{\Gamma(C,D)}$	Employed to improve the speed of C4.5 decision trees
$H(RG_2\|RG_1)$	Employed to improve the accuracy of C4.5 decision trees
$\Gamma_{-1}^{0}(RG_2\|RG_1)$	Employed to search free parameter in KNN-HDC

Generalized Dependency DegreeΓ

$$\Gamma(C,D) = \frac{1}{|U|}\sum_{x\in U}\frac{|D(x)\cap C(x)|}{|C(x)|}$$

$$\gamma(C,D) = \frac{1}{|U|}\sum_{x\in U\wedge C(x)\subseteq D(x)}\frac{|D(x)\cap C(x)|}{|C(x)|}$$

U: universe of objects

C, D: sets of attributes

C(x): C-class containing x

D(x): D-class containing x

$\frac{|D(x)\cap C(x)|}{|C(x)|}$:

the percentage that common neighbors of x in C and D occupy in the neighbors of x in C

Properties of Γ

Γ can be extended to equivalence relations R_1 and R_2.

$$\Gamma(R_1, R_2) = \frac{1}{|U|} \sum_{x \in U} \frac{|R_2(x) \cap R_1(x)|}{|R_1(x)|}$$

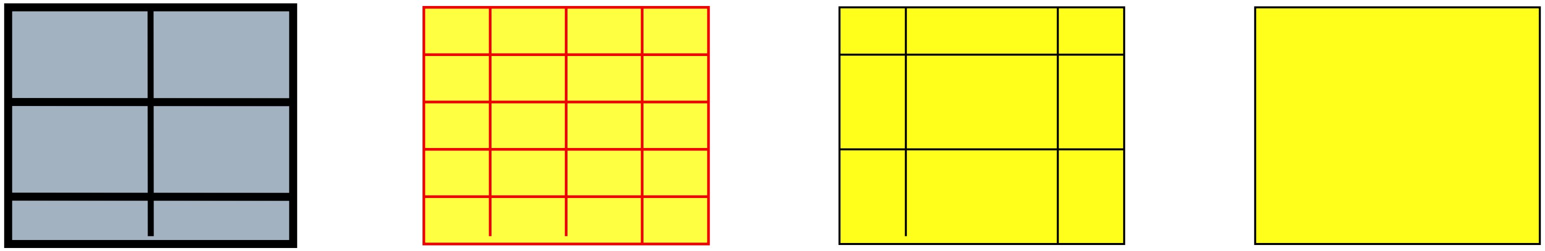
Property 1. $0 \leq \Gamma(C, D) \leq 1$

Property 2. $R_2 \subseteq R \Longrightarrow \Gamma(R_1, R_2) \leq \Gamma(R_1, R)$

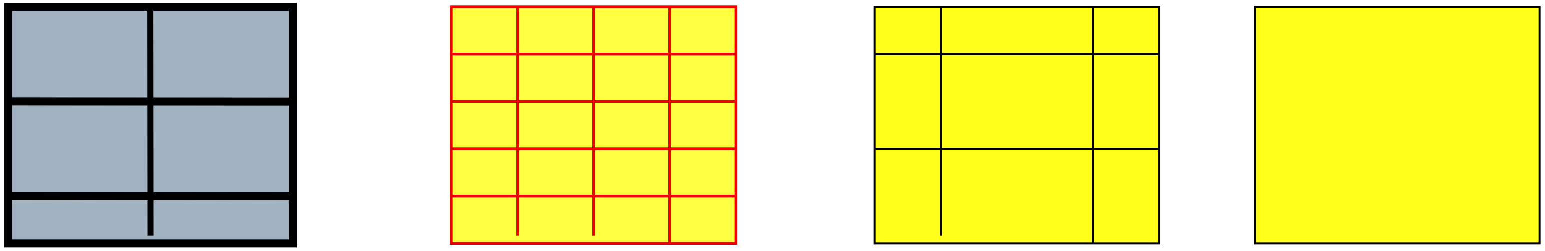
Property 3. $R_1 \subseteq R \Longrightarrow \Gamma(R_1, R_2) \geq \Gamma(R, R_2)$

Property 4. $\min\limits_{R_1 \in \mathcal{ER}(U)} \Gamma(R_1, R_2) = \Gamma(U \times U, R_2)$

Illustrations



R_1 $R_2 \subseteq R$ $U \times U$



R_2 $R_1 \subseteq R$ $U \times U$

Evaluation of Γ

- Comparison with H(D|C) in C4.5
  - Change the information gain $H(D)-H(D|\{a\})$

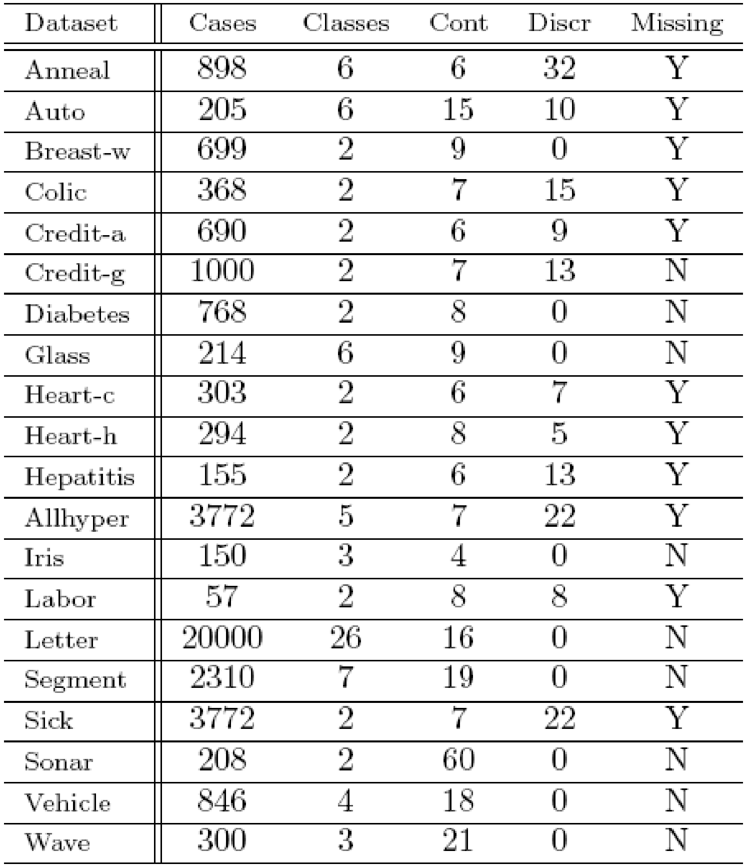
$$G(D,\{a\}) = \Gamma(\{a\},D) - \Gamma(U \times U, IND(D))$$

  - Stop the procedure of building trees when

$$N * G(D,\{a\}) < 0.75$$

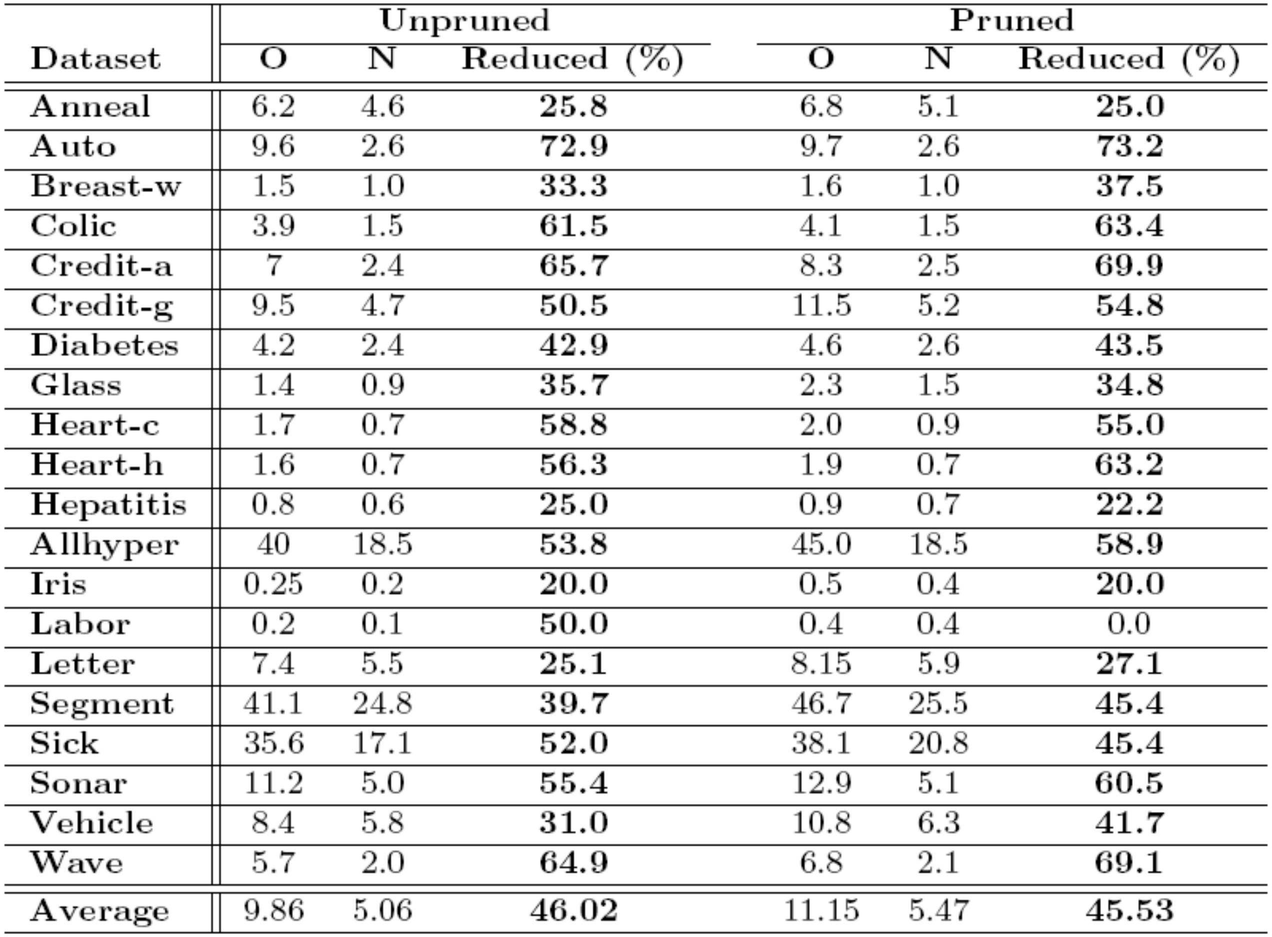
- Comparison with γ in attribute selection
  - For a given k, we will select C such that |C|=k, and Γ(C,D) [γ(C,D)] is maximal
  - We will compare the accuracy using the selected attributes by C4.5

Data



Dataset	Cases	Classes	Cont	Discr	Missing
Anneal	898	6	6	32	Y
Auto	205	6	15	10	Y
Breast-w	699	2	9	0	Y
Colic	368	2	7	15	Y
Credit-a	690	2	6	9	Y
Credit-g	1000	2	7	13	N
Diabetes	768	2	8	0	N
Glass	214	6	9	0	N
Heart-c	303	2	6	7	Y
Heart-h	294	2	8	5	Y
Hepatitis	155	2	6	13	Y
Allhyper	3772	5	7	22	Y
Iris	150	3	4	0	N
Labor	57	2	8	8	Y
Letter	20000	26	16	0	N
Segment	2310	7	19	0	N
Sick	3772	2	7	22	Y
Sonar	208	2	60	0	N
Vehicle	846	4	18	0	N
Wave	300	3	21	0	N

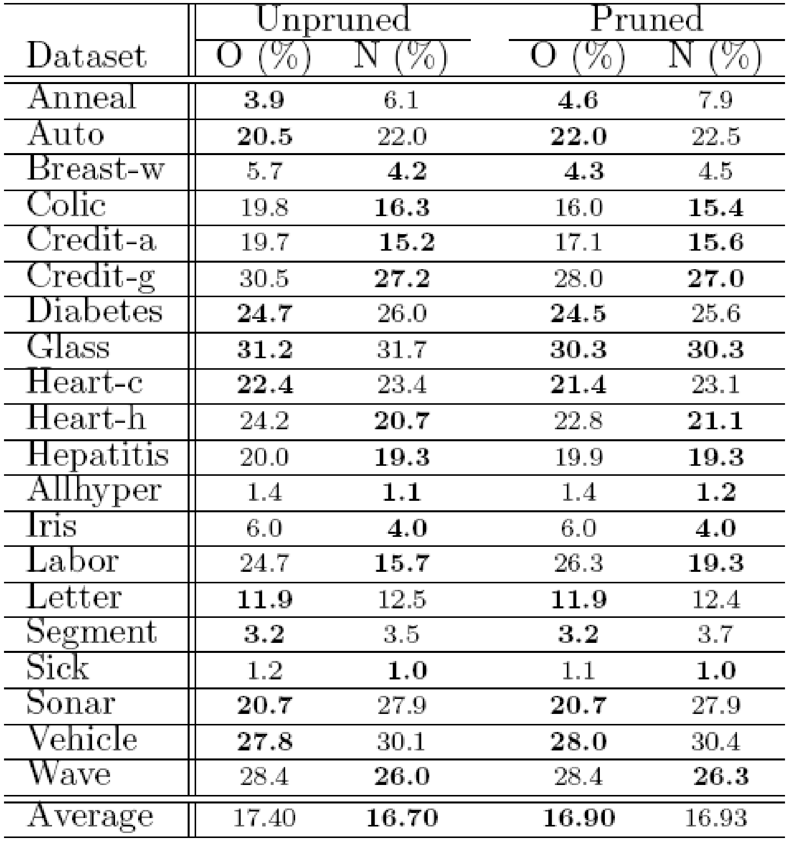
Speed



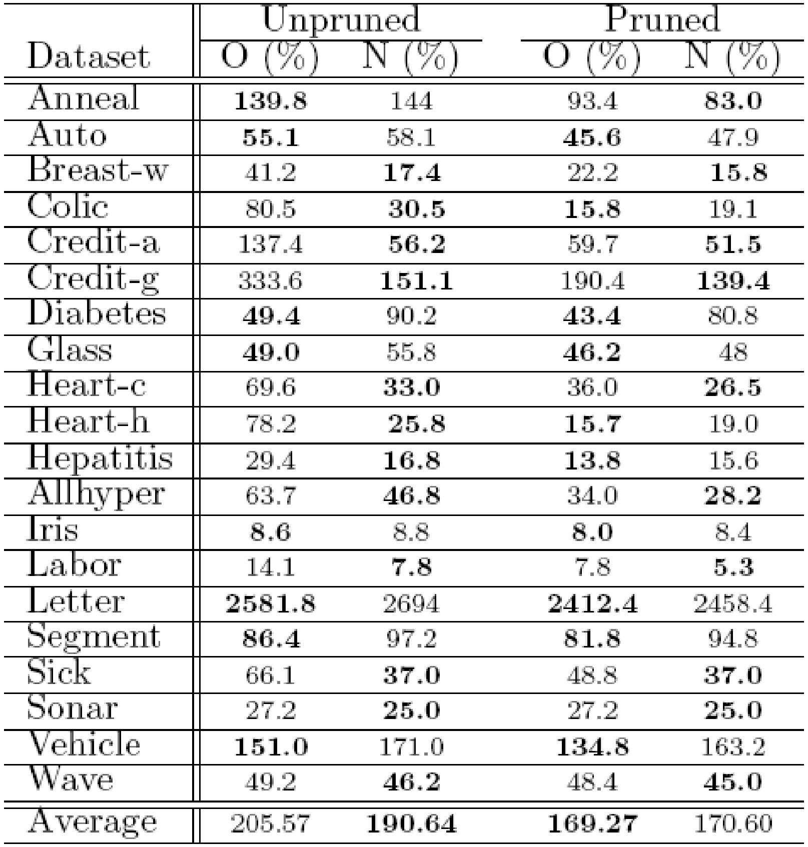
Dataset	Unpruned			Pruned		
	O	N	Reduced (%)	O	N	Reduced (%)
Anneal	6.2	4.6	**25.8**	6.8	5.1	**25.0**
Auto	9.6	2.6	**72.9**	9.7	2.6	**73.2**
Breast-w	1.5	1.0	**33.3**	1.6	1.0	**37.5**
Colic	3.9	1.5	**61.5**	4.1	1.5	**63.4**
Credit-a	7	2.4	**65.7**	8.3	2.5	**69.9**
Credit-g	9.5	4.7	**50.5**	11.5	5.2	**54.8**
Diabetes	4.2	2.4	**42.9**	4.6	2.6	**43.5**
Glass	1.4	0.9	**35.7**	2.3	1.5	**34.8**
Heart-c	1.7	0.7	**58.8**	2.0	0.9	**55.0**
Heart-h	1.6	0.7	**56.3**	1.9	0.7	**63.2**
Hepatitis	0.8	0.6	**25.0**	0.9	0.7	**22.2**
Allhyper	40	18.5	**53.8**	45.0	18.5	**58.9**
Iris	0.25	0.2	**20.0**	0.5	0.4	**20.0**
Labor	0.2	0.1	**50.0**	0.4	0.4	0.0
Letter	7.4	5.5	**25.1**	8.15	5.9	**27.1**
Segment	41.1	24.8	**39.7**	46.7	25.5	**45.4**
Sick	35.6	17.1	**52.0**	38.1	20.8	**45.4**
Sonar	11.2	5.0	**55.4**	12.9	5.1	**60.5**
Vehicle	8.4	5.8	**31.0**	10.8	6.3	**41.7**
Wave	5.7	2.0	**64.9**	6.8	2.1	**69.1**
Average	9.86	5.06	**46.02**	11.15	5.47	**45.53**

O: Original C4.5, N: The new C4.5.

Accuracy and Tree Size



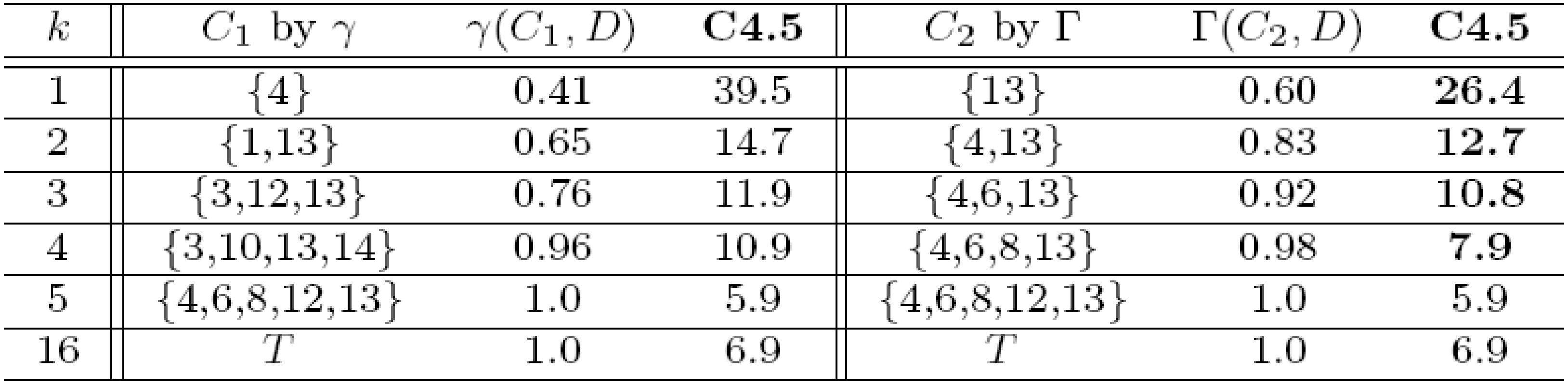
Dataset	Unpruned		Pruned	
	O (%)	N (%)	O (%)	N (%)
Anneal	**3.9**	6.1	**4.6**	7.9
Auto	**20.5**	22.0	**22.0**	22.5
Breast-w	5.7	**4.2**	**4.3**	4.5
Colic	19.8	**16.3**	16.0	**15.4**
Credit-a	19.7	**15.2**	17.1	**15.6**
Credit-g	30.5	**27.2**	28.0	**27.0**
Diabetes	**24.7**	26.0	**24.5**	25.6
Glass	**31.2**	31.7	**30.3**	**30.3**
Heart-c	**22.4**	23.4	**21.4**	23.1
Heart-h	24.2	**20.7**	22.8	**21.1**
Hepatitis	20.0	**19.3**	19.9	**19.3**
Allhyper	1.4	**1.1**	1.4	**1.2**
Iris	6.0	**4.0**	6.0	**4.0**
Labor	24.7	**15.7**	26.3	**19.3**
Letter	**11.9**	12.5	**11.9**	12.4
Segment	**3.2**	3.5	**3.2**	3.7
Sick	1.2	**1.0**	1.1	**1.0**
Sonar	**20.7**	27.9	**20.7**	27.9
Vehicle	**27.8**	30.1	**28.0**	30.4
Wave	28.4	**26.0**	28.4	**26.3**
Average	17.40	**16.70**	**16.90**	16.93



Dataset	Unpruned		Pruned	
	O (%)	N (%)	O (%)	N (%)
Anneal	**139.8**	144	93.4	**83.0**
Auto	**55.1**	58.1	**45.6**	47.9
Breast-w	41.2	**17.4**	22.2	**15.8**
Colic	80.5	**30.5**	**15.8**	19.1
Credit-a	137.4	**56.2**	59.7	**51.5**
Credit-g	333.6	**151.1**	190.4	**139.4**
Diabetes	**49.4**	90.2	**43.4**	80.8
Glass	**49.0**	55.8	**46.2**	48
Heart-c	69.6	**33.0**	36.0	**26.5**
Heart-h	78.2	**25.8**	**15.7**	19.0
Hepatitis	29.4	**16.8**	**13.8**	15.6
Allhyper	63.7	**46.8**	34.0	**28.2**
Iris	**8.6**	8.8	**8.0**	8.4
Labor	14.1	**7.8**	7.8	**5.3**
Letter	**2581.8**	2694	**2412.4**	2458.4
Segment	**86.4**	97.2	**81.8**	94.8
Sick	66.1	**37.0**	48.8	**37.0**
Sonar	27.2	**25.0**	27.2	**25.0**
Vehicle	**151.0**	171.0	**134.8**	163.2
Wave	49.2	**46.2**	48.4	**45.0**
Average	205.57	**190.64**	**169.27**	170.60

O: Original C4.5   N: The new C4.5

Feature Selection



k	C_1 by γ	$\gamma(C_1, D)$	**C4.5**	C_2 by Γ	$\Gamma(C_2, D)$	**C4.5**
1	{4}	0.41	39.5	{13}	0.60	**26.4**
2	{1,13}	0.65	14.7	{4,13}	0.83	**12.7**
3	{3,12,13}	0.76	11.9	{4,6,13}	0.92	**10.8**
4	{3,10,13,14}	0.96	10.9	{4,6,8,13}	0.98	**7.9**
5	{4,6,8,12,13}	1.0	5.9	{4,6,8,12,13}	1.0	5.9
16	T	1.0	6.9	T	1.0	6.9

Summary

- Γ is an informative measure in decision trees and attribute selection
- C4.5 using Γ is faster than that using the conditional entropy
- $\Gamma$is more accurate than γ in feature selection

An Overview

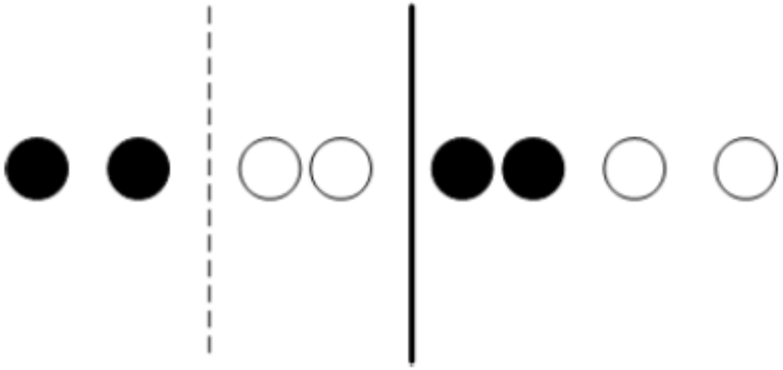
Theorem (Generality of $\Gamma_{\alpha}^{\varepsilon}(RG_2|RG_1)$)

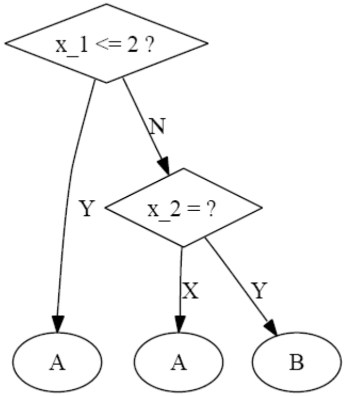
$$\Gamma_{\alpha}^{\varepsilon}(RG_2|RG_1) = \begin{cases} \gamma(C,D), & \text{when } \varepsilon = 1, \alpha = -1, \\ & RG_2 = IND(D), \text{ and } RG_1 = IND(C), \\ \Gamma(C,D), & \text{when } \varepsilon = 0, \alpha = -1, \\ & RG_2 = IND(D), \text{ and } RG_1 = IND(C), \\ H(D|C), & \text{when } \varepsilon = 0, \alpha = 1, \\ & RG_2 = IND(D), \text{ and } RG_1 = IND(C), \\ H(RG_2|RG_1), & \text{when } \varepsilon = 0 \text{ and } \alpha = 1. \end{cases}$$

$\gamma(C,D)$	The measure used in Rough Set Theory
$H(D\|C)$	The measure used in C4.5 decision trees
$\Gamma(C,D)$	Employed to improve the speed of C4.5 decision trees
$H(RG_2\|RG_1)$	Employed to improve the accuracy of C4.5 decision trees
$\Gamma_{-1}^{0}(RG_2\|RG_1)$	Employed to search free parameter in KNN-HDC

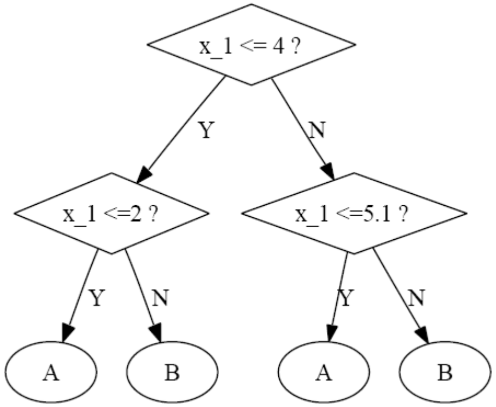
An example showing the inaccuracy of H(C,D)

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
x_1	1.0	2.0	3.9	4.0	5.0	5.1	7.0	8.0
x_2	Y	X	Y	Y	X	X	Y	X
y	A	A	B	B	A	A	B	B



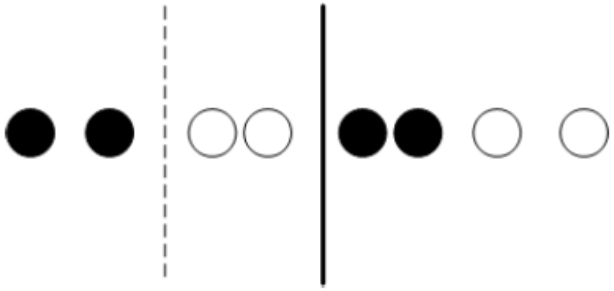


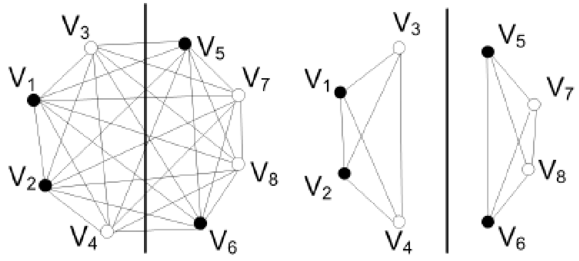
Generated by C4.5 using H(C,D)



The ideal one

Reasons





1. **The middle cut in C4.5 means a condition** x_1 <= 4
2. **After the middle cut, the distance information in the part is ignored, and so is that in the right part**
3. **The information gain is underestimated**

Random Graph Dependency Measure $H(RG_2|RG_1)$

$$H(RG_2|RG_1) \quad = \frac{1}{|U|} \sum_{x \in U} \log_2 \frac{|RG_2(x) \cap RG_1(x)|}{|RG_1(x)|}$$

U: universe of objects

RG_1: a random graph on U

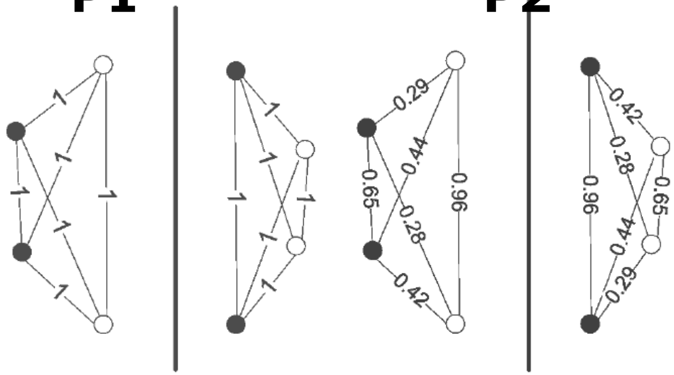
RG_2: another random graph on U

$RG_1(x)$: random neighbors of x in RG_1

$RG_2(x)$: random neighbors of x in RG_2

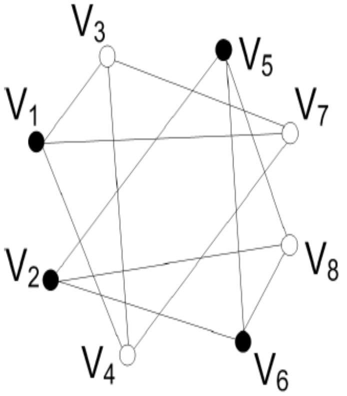
Representing a feature as a random graph

P1 **P2**

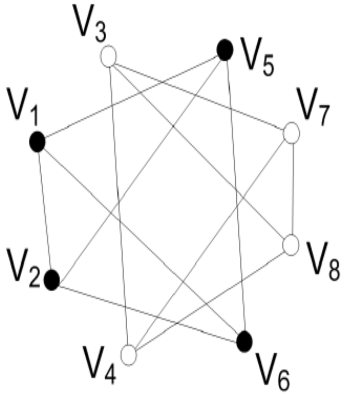


(a) $\sigma = 0$ (b) $\sigma = 3$

P3

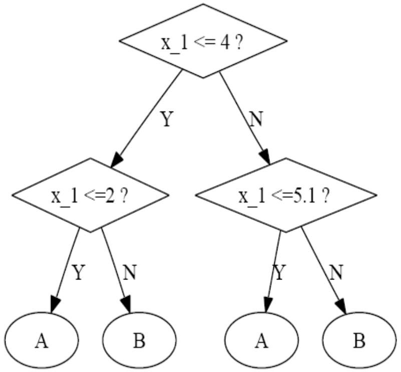


P4



Generated by x_1 using $p_{ij} = e^{-\sigma * |a_i - a_j|}$ **Generated by x_2** **Generated by y**

H(P4|P1)=-1 H(P4|P2)=-0.48 H(P4|P3)=-0.81

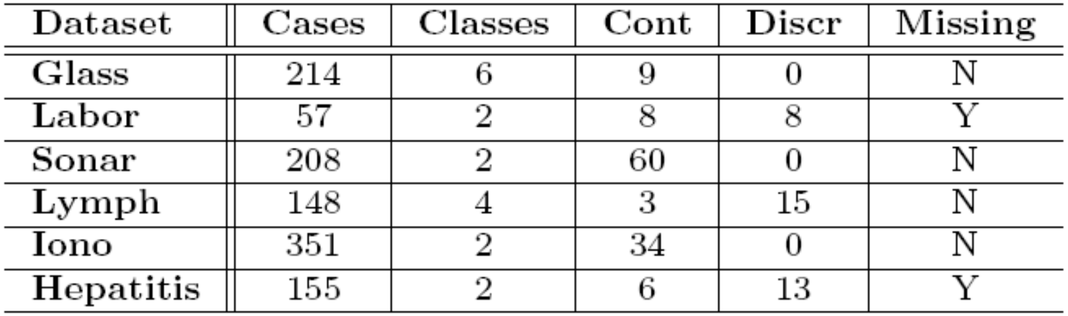


Evaluation of $H(RG_2|RG_1)$

- Comparisonwith H(D|C) in C4.5
  - Change the information measure $H(D|C)$

$$H(RG_2|RG_1)$$

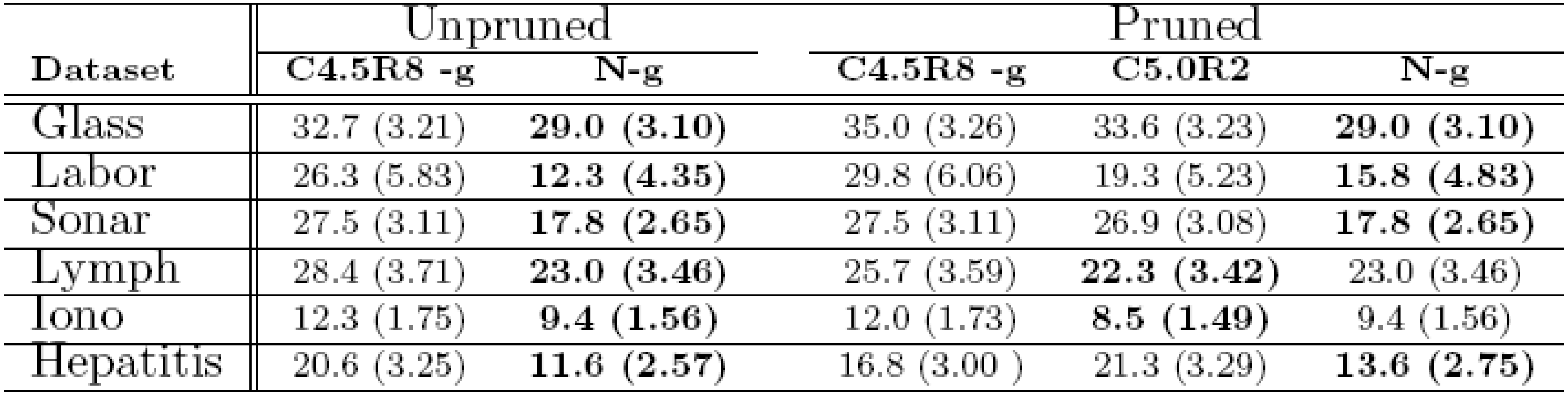
- Comparison with C5.0R2
  - C5.0 is a commercial development of C4.5.
  - The number of samples is limited to 400 in the evaluation version
- Data



Dataset	Cases	Classes	Cont	Discr	Missing
Glass	214	6	9	0	N
Labor	57	2	8	8	Y
Sonar	208	2	60	0	N
Lymph	148	4	3	15	N
Iono	351	2	34	0	N
Hepatitis	155	2	6	13	Y

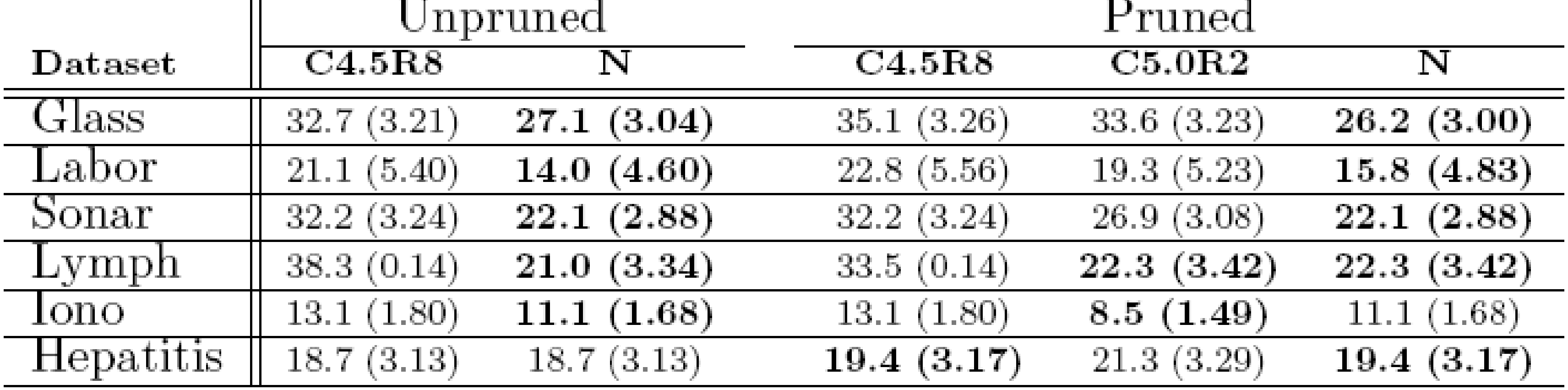
Accuracy

Information Gain



Dataset	Unpruned		Pruned		
	C4.5R8 -g	N-g	C4.5R8 -g	C5.0R2	N-g
Glass	32.7 (3.21)	**29.0 (3.10)**	35.0 (3.26)	33.6 (3.23)	**29.0 (3.10)**
Labor	26.3 (5.83)	**12.3 (4.35)**	29.8 (6.06)	19.3 (5.23)	**15.8 (4.83)**
Sonar	27.5 (3.11)	**17.8 (2.65)**	27.5 (3.11)	26.9 (3.08)	**17.8 (2.65)**
Lymph	28.4 (3.71)	**23.0 (3.46)**	25.7 (3.59)	**22.3 (3.42)**	23.0 (3.46)
Iono	12.3 (1.75)	**9.4 (1.56)**	12.0 (1.73)	**8.5 (1.49)**	9.4 (1.56)
Hepatitis	20.6 (3.25)	**11.6 (2.57)**	16.8 (3.00)	21.3 (3.29)	**13.6 (2.75)**

Information Gain Ratio



Dataset	Unpruned		Pruned		
	C4.5R8	N	C4.5R8	C5.0R2	N
Glass	32.7 (3.21)	**27.1 (3.04)**	35.1 (3.26)	33.6 (3.23)	**26.2 (3.00)**
Labor	21.1 (5.40)	**14.0 (4.60)**	22.8 (5.56)	19.3 (5.23)	**15.8 (4.83)**
Sonar	32.2 (3.24)	**22.1 (2.88)**	32.2 (3.24)	26.9 (3.08)	**22.1 (2.88)**
Lymph	38.3 (0.14)	**21.0 (3.34)**	33.5 (0.14)	**22.3 (3.42)**	**22.3 (3.42)**
Iono	13.1 (1.80)	**11.1 (1.68)**	13.1 (1.80)	**8.5 (1.49)**	11.1 (1.68)
Hepatitis	18.7 (3.13)	18.7 (3.13)	**19.4 (3.17)**	21.3 (3.29)	**19.4 (3.17)**

An Overview

Theorem (Generality of $\Gamma_{\alpha}^{\varepsilon}(RG_2|RG_1)$)

$$\Gamma_{\alpha}^{\varepsilon}(RG_2|RG_1) = \begin{cases} \gamma(C,D), & \text{when } \varepsilon = 1, \alpha = -1, \\ & RG_2 = IND(D), \text{ and } RG_1 = IND(C), \\ \Gamma(C,D), & \text{when } \varepsilon = 0, \alpha = -1, \\ & RG_2 = IND(D), \text{ and } RG_1 = IND(C), \\ H(D|C), & \text{when } \varepsilon = 0, \alpha = 1, \\ & RG_2 = IND(D), \text{ and } RG_1 = IND(C), \\ H(RG_2|RG_1), & \text{when } \varepsilon = 0 \text{ and } \alpha = 1. \end{cases}$$

$\gamma(C,D)$	The measure used in Rough Set Theory
$H(D\|C)$	The measure used in C4.5 decision trees
$\Gamma(C,D)$	Employed to improve the speed of C4.5 decision trees
$H(RG_2\|RG_1)$	Employed to improve the accuracy of C4.5 decision trees
$\Gamma_{-1}^{0}(RG_2\|RG_1)$	Employed to search free parameter in KNN-HDC

A General Form

$$\Gamma_{\alpha}^{\varepsilon}(RG_2|RG_1) = f_{\alpha}^{-1}(\frac{1}{|U|}\sum_{x\in U^{\varepsilon}(RG_1,RG_2)} f_{\alpha}(\frac{|RG_2(x)\cap RG_1(x)|}{|RG_1(x)|}))$$

$$U^{\varepsilon}(RG_1, RG_2) = \{x|x\in U \wedge |RG_2(x)\cap RG_1(x)|/|RG_1(x)| \geq \varepsilon\}$$

$$f_{\alpha}(u) = \begin{cases} u^{\frac{1-\alpha}{2}}, & \alpha \neq 1, \\ \log u, & \alpha = 1. \end{cases}$$

Motivations

- In KNN-HDC, a naive method to find (K, β, γ) is the cross-validation (CV), but
  - Knp multiplications are needed at each fold of CV
- Find (K, β) by the random graph dependency because
  - Only Kn multiplications and n divisions are needed
- Leave γ by cross-validation, because
  - nn multiplications are needed by the random graph dependency measure

n: the number of data
K: the number of neighbors
p: the number of iterations

Methods

- For given (K, β), a random graph is generated

$$R^1(K,\beta)_{ij} = \begin{cases} 1, & j = i; \\ e^{-w_{ij}^2/\beta}, & \text{if } j \text{ is one of the neighbors of } i; \\ 0, & \text{otherwise.} \end{cases}$$

- Label information forms another random graph

$$R_{ij}^2 = \begin{cases} 1, & j = i; \\ 1, & \text{if } j \text{ and } i \text{ are labeled same;} \\ 0, & \text{if } j \text{ and } i \text{ are labeled differently;} \\ p_l, & \text{if one of } i \text{ and } j \text{ is labeled as } l \text{ while the other is not labeled;} \\ r, & \text{if both } j \text{ and } i \text{ are not labeled.} \end{cases}$$

$$r = \sum_{i=1}^{c} p_i^2$$

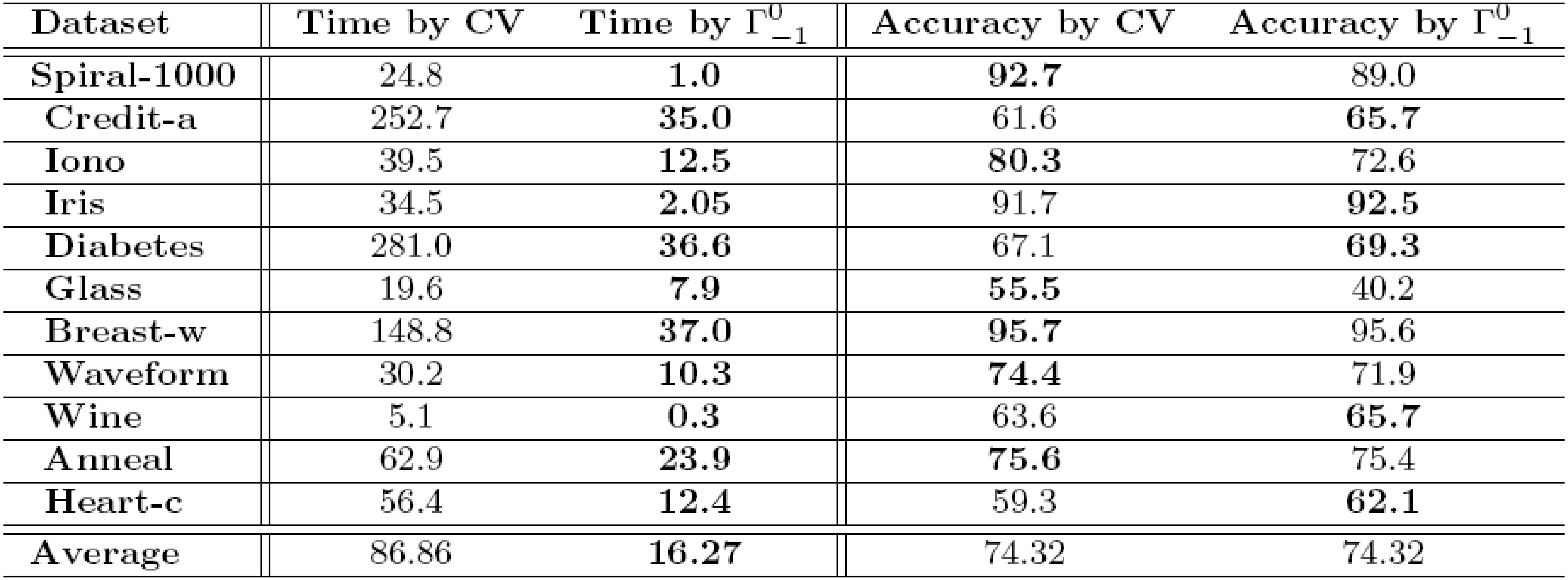
$$\max_{K,\beta} \Gamma^0_{-1}(R^2 | R^1(K,\beta)) \boxed{\Gamma^0_{-1}(R^1(K,\beta) | U \times U)}$$

P_l : the frequency of label l in the labeled data

c : the number of classes

r : the probability that two randomly chosen points share the same label

Results



Dataset	Time by CV	Time by Γ^0_{-1}	Accuracy by CV	Accuracy by Γ^0_{-1}
Spiral-1000	24.8	**1.0**	**92.7**	89.0
Credit-a	252.7	**35.0**	61.6	**65.7**
Iono	39.5	**12.5**	**80.3**	72.6
Iris	34.5	**2.05**	91.7	**92.5**
Diabetes	281.0	**36.6**	67.1	**69.3**
Glass	19.6	**7.9**	**55.5**	40.2
Breast-w	148.8	**37.0**	**95.7**	95.6
Waveform	30.2	**10.3**	**74.4**	71.9
Wine	5.1	**0.3**	63.6	**65.7**
Anneal	62.9	**23.9**	**75.6**	75.4
Heart-c	56.4	**12.4**	59.3	**62.1**
Average	86.86	**16.27**	74.32	74.32

Summary

- A general information measure is developed
  - Improve C4.5 decision trees in speed by one special case
  - Improve C4.5 decision trees in accuracy by another special case
  - Help to find free parameter in KNN-HDC

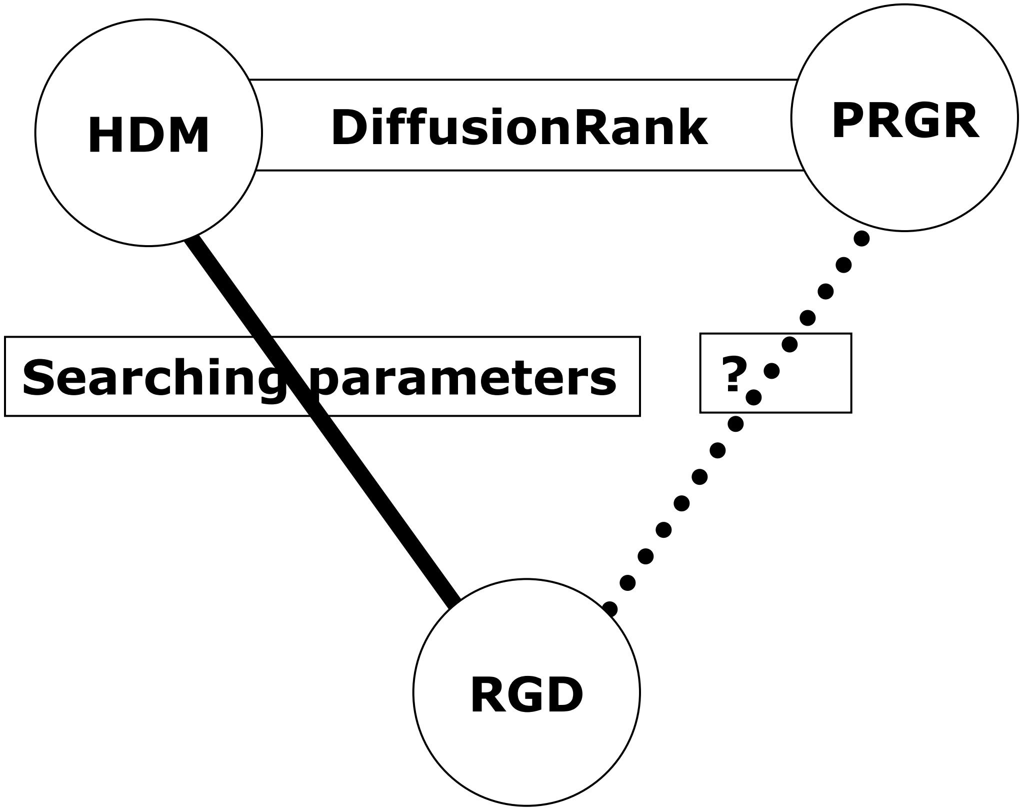
Outline

- □ Introduction
- □ Background
- □ Heat Diffusion Models on a Random Graph
- □ Predictive Random Graph Ranking
- □ Random Graph Dependency
- □ Conclusion and Future Work

Conclusion

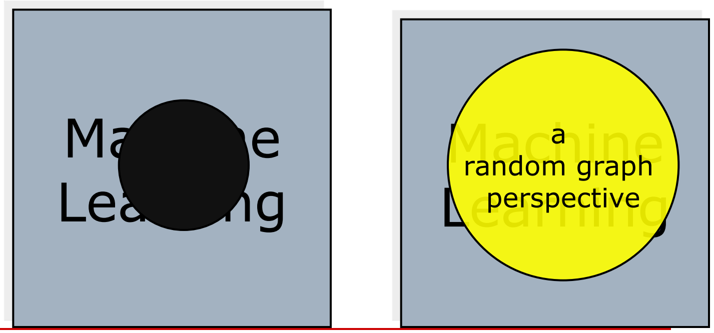
- With a viewpoint of a random graph, three machine learning models are successfully established
  - G-HDC can achieve better performance in accuracy in some benchmark datasets
  - PRGR extends the scope of some current ranking algorithms, and improve the accuracy of ranking algorithms such as PageRank and Common Neighbor
  - DiffusionRank can achieve the ability of anti-manipulation
  - Random Graph Dependency can improve the speed and accuracy of C4.5 algorithms, and can help to search free parameters in G-HDC

Future Work



$\max \Gamma^{0}_{-1}(RG_2|RG_1)$ s.t. RG_2 is a total ordered graph.

Future Work



- Deepen
  - Need more accurate random graph generation methods
  - For G-HDC, try a better initial temperature setting
  - For PRGR, investigate page-makers' preference on link orders
  - For random graph dependency, find more properties and shorten the computation time
- Widen
  - For G-HDC, try to apply it to inductive learning
  - For PRGR, try to make SimRank work, and include other ranking algorithms
  - For random graph dependency, apply it to ranking problem and apply it to determining kernels

Publication list

1. Haixuan Yang, Irwin King, and Michael R. Lyu. NHDC and PHDC: Non-propagating and Propagating Heat Diffusion Classifiers. In Proceedings of the 12th International Conference on Neural Information Processing (ICONIP), pages 394—399, 2005
2. Haixuan Yang, Irwin King, and Michael R. Lyu. Heat Diffusion Classifiers on Graphs. Pattern Analysis and Applications, Accepted, 2006
3. Haixuan Yang, Irwin King, and Michael R. Lyu. Predictive ranking: a novel page ranking approach by estimating the web structure. In Proceedings of the 14th international conference on World Wide Web (WWW) - Special interest tracks and posters, pages 944—945, 2005
4. Haixuan Yang, Irwin King, and Michael R. Lyu. Predictive random graph ranking on the Web. In Proceedings of the IEEE World Congress on Computational Intelligence (WCCI), pages 3491—3498, 2006
5. Haixuan Yang, Irwin King, and Michael R. Lyu. DiffusionRank: A Possible Penicillin for Web Spamming. In Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), Accepted, 2007
6. Haixuan Yang, Irwin King, and Michael R. Lyu. The Generalized Dependency Degree Between Attributes. Journal of the American Society for Information Science and Technology, Accepted, 2007

G-HDC except VHDC: 1 and 2

PRGR: 3,4,5

Random Graph Dependency about Γ: 6

Thanks

MPM

$$\begin{aligned}
\max_{\alpha,\mathbf{a}\neq\mathbf{0},b} \quad & \alpha \quad \text{s.t.} \\
& \inf_{\mathbf{x}\sim(\overline{\mathbf{x}},\Sigma_{\mathbf{x}})} \mathbf{Pr}\{\mathbf{a}^T\mathbf{x} \geq b\} \geq \alpha, \\
& \inf_{\mathbf{y}\sim(\overline{\mathbf{y}},\Sigma_{\mathbf{y}})} \mathbf{Pr}\{\mathbf{a}^T\mathbf{y} \leq b\} \geq \alpha,
\end{aligned}$$

Volume Computation

- □ Define *V*(*i*) to be the volume of the hypercube whose side length is the average distance between node *i* and its neighbors.

$$V(i) = \eta \min_{j:(j,i)\in E} w_{ij}^{\nu}/2 + 1/2n$$

$$\nu = \frac{1}{n}\sum_{i=1}^{n} \hat{m}_K(x_i)$$

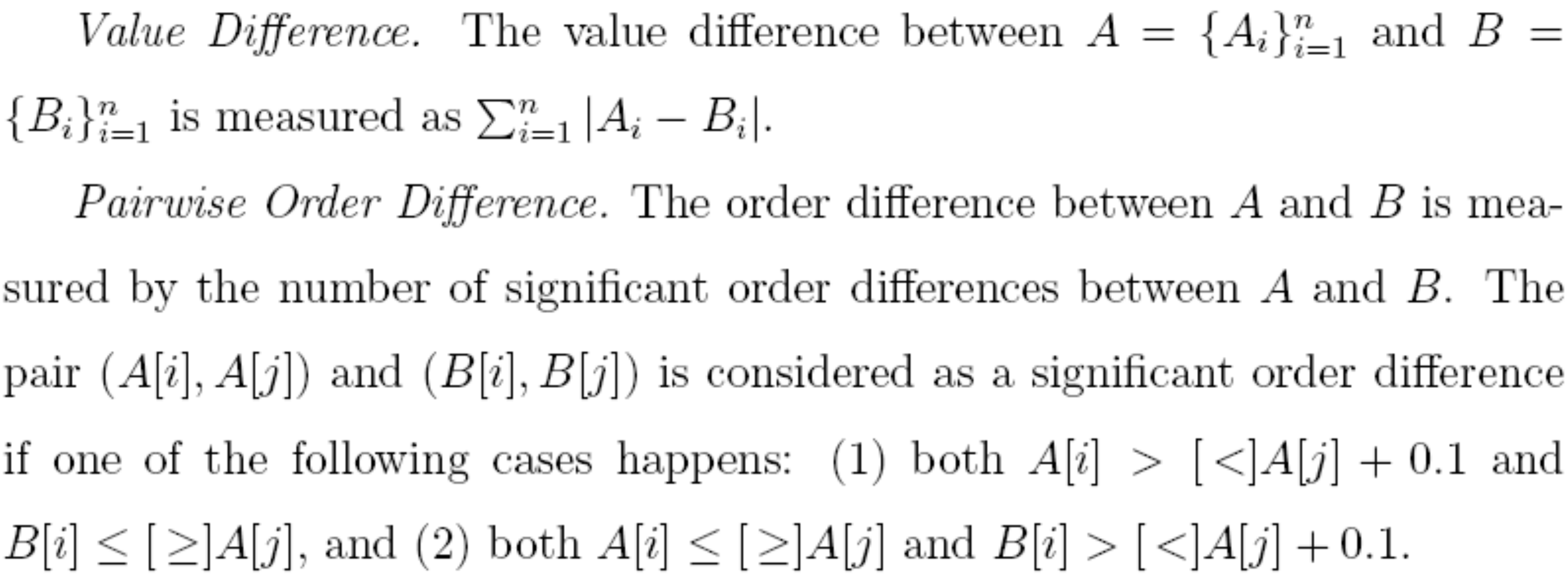
a maximum likelihood estimation

$$\hat{m}_K(x) = \left[\frac{1}{K-1}\sum_{j=1}^{K-1}\log\frac{T_K(x)+\epsilon}{T_j(x)+\epsilon}\right]^{-1}$$

Problems

- POL?
- When to stop in C4.5
  - /* If all cases are of the same class or there are not enough cases to divide, the tree is a leaf */
- PCA
- Why HDC can achieve a better result?
- MPM?
- Kernel?

Value Difference



Value Difference. The value difference between $A = \{A_i\}_{i=1}^n$ and $B = \{B_i\}_{i=1}^n$ is measured as $\sum_{i=1}^n |A_i - B_i|$.

Pairwise Order Difference. The order difference between A and B is measured by the number of significant order differences between A and B. The pair $(A[i], A[j])$ and $(B[i], B[j])$ is considered as a significant order difference if one of the following cases happens: (1) both $A[i] > [<]A[j] + 0.1$ and $B[i] \leq [\geq]A[j]$, and (2) both $A[i] \leq [\geq]A[j]$ and $B[i] > [<]A[j] + 0.1$.