Mining Heterogeneous Information Networks (Part 2)

JIAWEI HAN
COMPUTER SCIENCE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

MARCH 2, 2017

Outline

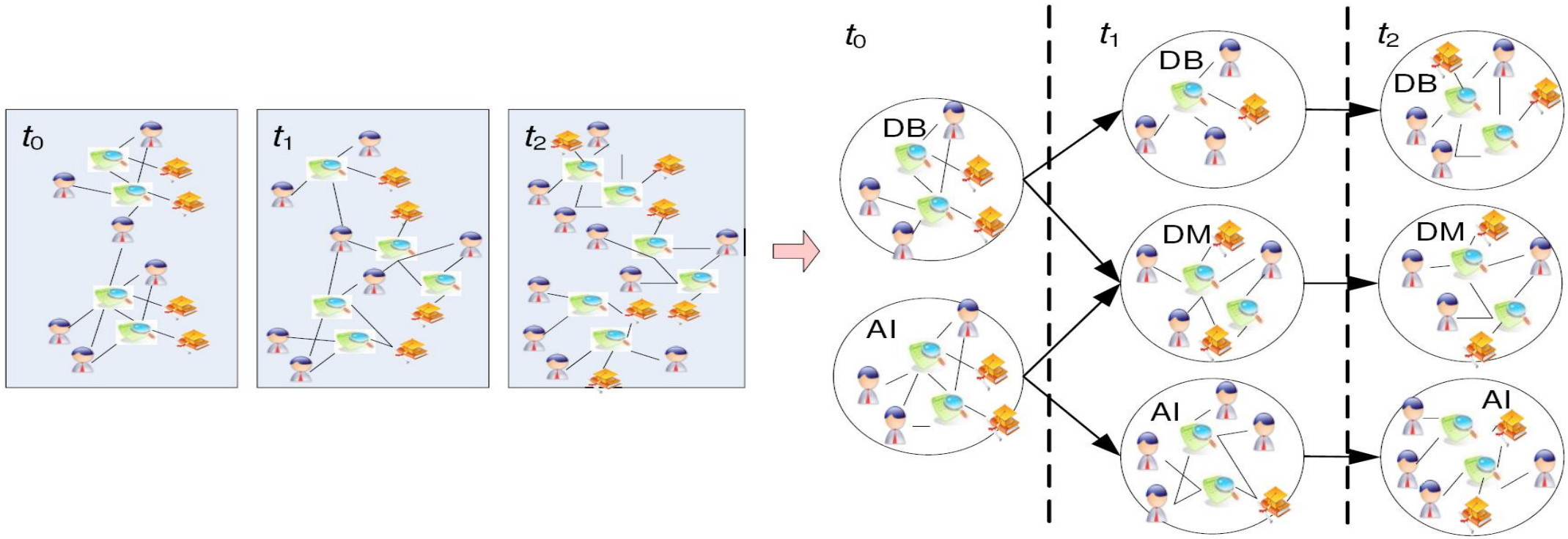
- **Motivation:** Why Mining Information Networks?
- **Part I:** Clustering and Ranking in Heterogeneous Information Networks
- **Part II:** Classification and Prediction in Heterogeneous Information Networks
- **Part III:** Other Data Mining Functions for Heterogeneous Information Networks
  - Mining Evolution and Dynamics of Information Networks
  - Role Discovery
  - OLAP in Information Networks
  - Data Cleaning: Distinct
  - CrossMine: Classification Across Multi-Relational Databases
  - EgoSet: Exploiting Word Ego-Networks for Multifaceted Set Expansion
- **Summary**

Mining Evolution and Dynamics of InfoNet

- Many networks are with time information
  - E.g., according to paper publication year, DBLP networks can form network sequences
- Motivation: Model evolution of communities in heterogeneous network
  - Automatically detect the best number of communities in each timestamp
  - Model the smoothness between communities of adjacent timestamps
  - Model the evolution structure explicitly
    - Birth, death, split
- EvoNetClus: Modeling evolution of dynamic heterogeneous networks
  - Co-evolution within a community
    - heterogeneous multi-typed object/links
  - Discovery of evolution structures among different communities
- Y. Sun, et al., "Studying Co-Evolution of Multi-Typed Objects in Dynamic Heterogeneous Information Networks", MLG'10

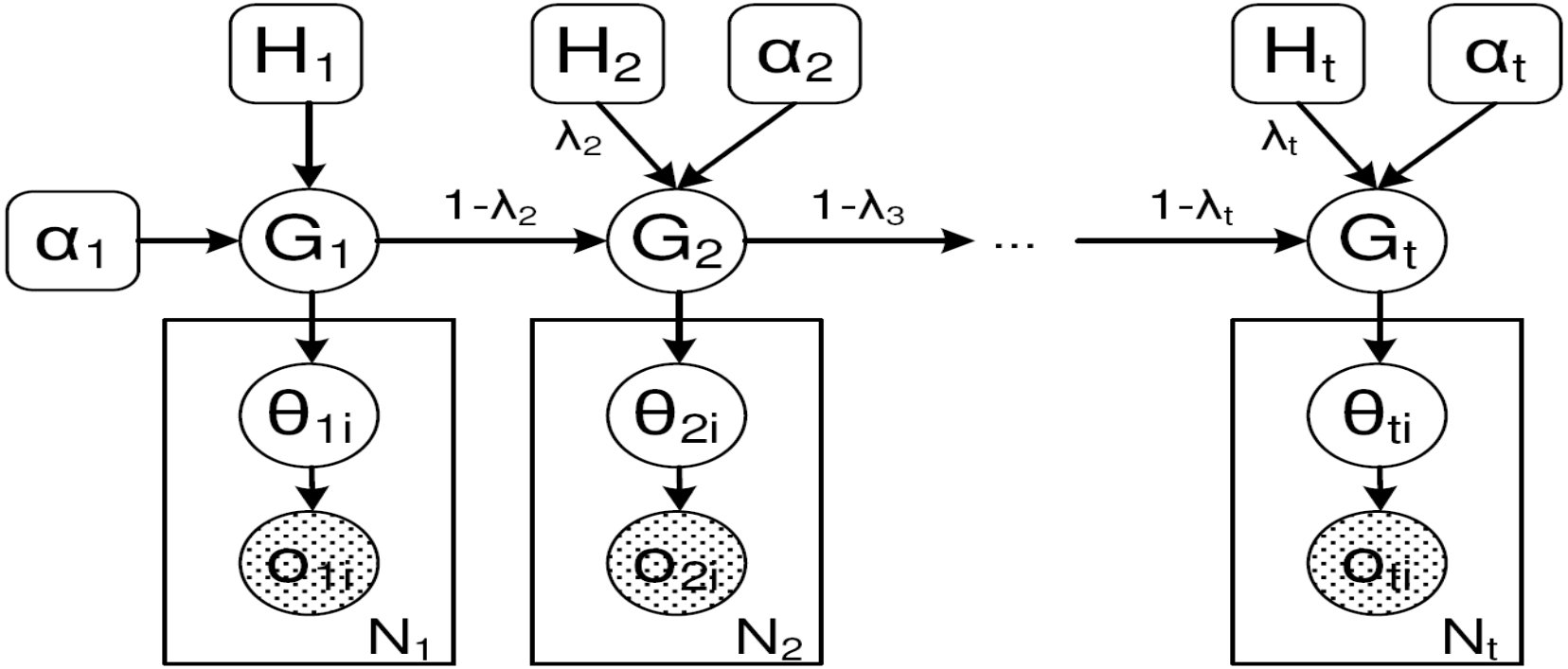
Evolution: Idea Illustration

- From network sequences to evolutionary communities



Graphical Model: A Generative Model

- **Dirichlet Process Mixture Model-based generative model**
  - At each timestamp, a community is dependent on historical communities and background community distribution



Generative Model & Model Inference

- To generate a new paper o_i
  - Decide whether to join an existing community or a new one
    - Join an existing community k with prob. $n_k/(i-1+\alpha)$
    - Join a new community k with prob. $\alpha/(i-1+\alpha)$: Decide its prior, either from a background distribution (λ) or historical communities ($(1-\lambda)\pi_k$), with different probabilities, draw the attribute distribution from the prior
  - Generate o_i according to the attribute distribution

$$
\begin{aligned}
p(o_{i,t}|z_{i,t}=k,\mathbf{\Theta}_t) &= p(o_{i,t}|\theta_{k,t}) \\
&= p(\mathbf{a}_{i,t}|\theta^A_{k,t})p(\mathbf{c}_{i,t}|\theta^C_{k,t})p(\mathbf{d}_{i,t}|\theta^D_{k,t}) \\
&= \prod_{j=1}^{|A|}\theta^A_{k,t}(j)^{a_{ij,t}}\prod_{j=1}^{|C|}\theta^C_{k,t}(j)^{c_{ij,t}}\prod_{j=1}^{|D|}\theta^D_{k,t}(j)^{d_{ij,t}}
\end{aligned}
$$

- Greedy inference for each timestamp: Collapse Gibbs sampling, which is trying to sample cluster label for each target object (e.g., paper)

Community Evolution Discovery: Algorithm

Step 1: Generating Prior Groups

- Calculation of the posterior probabilities of hidden labels

$$p(z_{t,i} = k | o_{t,i}, \mathbf{z_{t-1}}, \mathbf{O_{t-1}}, H_t) \propto n_{t-1,k} f_k^{-o_{t,i}}(o_{t,i})$$

$$p(z_{t,i} = k_{new} | o_{t,i}, H_t) \propto \gamma f_{k_{new}}(o_{t,i})$$

Step 2: Iterative Hidden Community Label Assignment

- Using an EM Algorithm

$$p(z_{ji} = k, k \notin \{K_{t-1}\} | \mathbf{z}_{-ji}, \mathbf{o}) \propto (n_{j\cdot k}^{-ji} + \alpha\eta_k) f_k^{-o_{ji}}(o_{ji})$$

$$p(z_{ji} = k, k \in \{K_{t-1}\} | \mathbf{z}_{-ji}, \mathbf{o}) \propto (n_{j\cdot k}^{-ji} + \lambda \frac{n_{t-1,k}}{N_{t-1}} + \alpha\eta_k) f_k^{-o_{ji}}(o_{ji})$$

$$p(z_{ji} = k_{new} | o_{ji}) \propto \alpha\eta_u f_{k_{new}}(o_{ij})$$

Input: Network $G_t, G_{t-1}, \mathbf{z}_{t-1}$; γ, α, β, λ ;
Output: The community assignment vector $\mathbf{z}_t$; the parameters $\mathbf{\Theta}_t = \{\theta_t^A, \theta_t^C, \theta_t^W\}$;

Assign each object into prior groups;
repeat
  for *each object* o_{ji} **do**
    1. E-step: Assign o_{ji} to the community with the maximum posterior probability in either existing community k or a new community $k+1$;
    2. M-step: Update relevant statistics ;
    3. if community k_{old} for o_i contains no objects, remove the community;
  end
until *reaches cluster change threshold*;
Estimate parameters $\mathbf{\Theta}_t$ for each community;

Algorithm 1: Parameter Estimation Algorithm.

Step 3: Community Distribution Estimation:

$$\theta_k^A(j) = \frac{\beta_A + n_j^A}{\beta_A |A| + n^A}; \theta_k^C(j) = \frac{\beta_C + n_j^C}{\beta_C |C| + n^C}; \theta_k^W(j) = \frac{\beta_D + n_j^W}{\beta_D |W| + n^W}$$

Accuracy Study

- The more types of objects used, the better accuracy
- Historical prior results in better accuracy

Year	Training Type	Testing Type	Test Size 10% (cluster number K)	Test Size 20% (cluster number K)
1992	Term	Term	1.600 (4)	1.390 (4)
1992	Term+Author	Term+Author	2.205 (8)	1.697 (6)
1992	Term+Author+Conf.	Term+Author	2.434 (8)	2.095 (8)
1992|1991	Term+Author+Conf.	Term+Author	**2.8365** (8)	**2.671** (8)

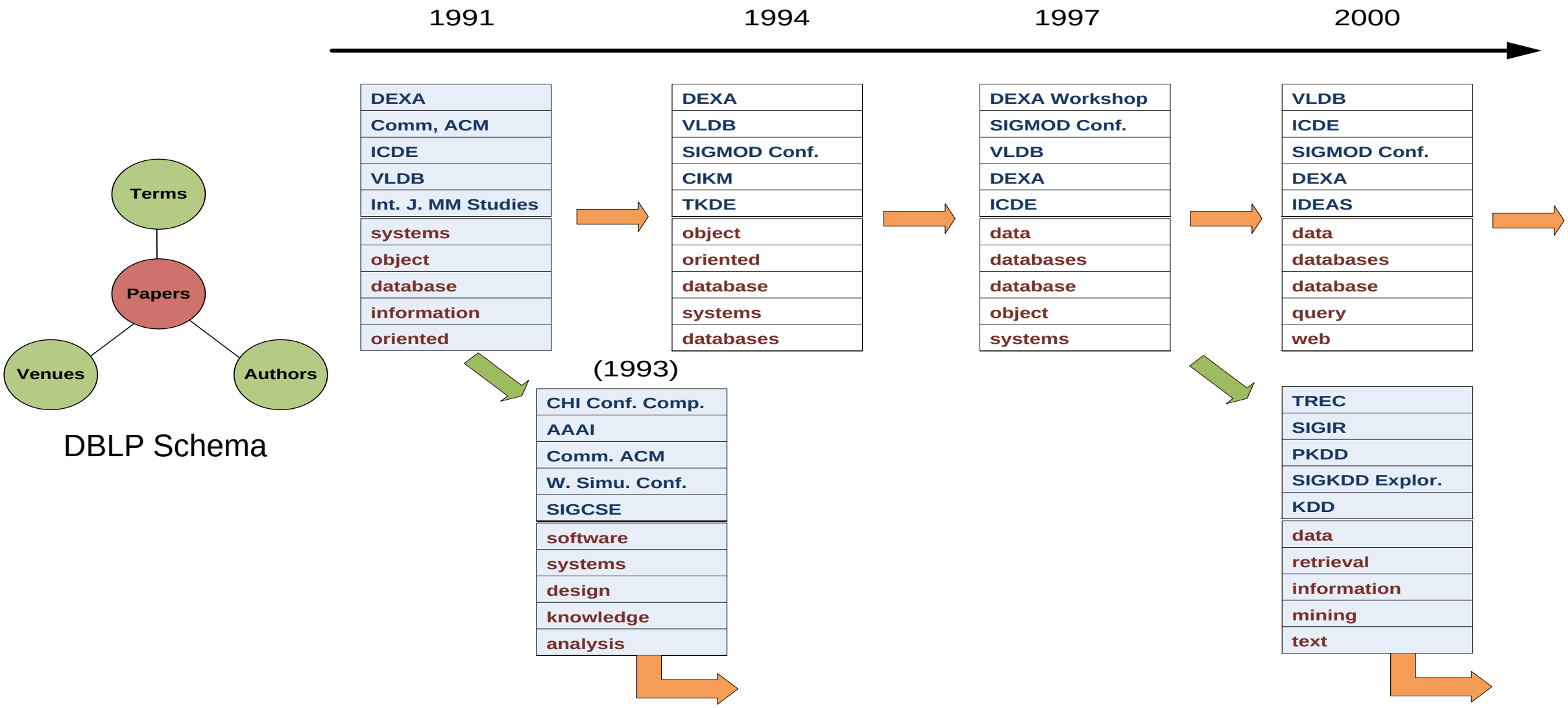
Table 1: Conference Compactness of Different Models on Test Dataset

Year	Training Type	Testing Type	Test Size 10%	Test Size 20%
1992	Term+Author+Conf.	Term+Author+Conf.	3.493×10^{18}	4.673×10^{18}
1992|1991	Term+Author+Conf.	Term+Author+Conf.	$\mathbf{6.384 \times 10^{17}}$	$\mathbf{7.106 \times 10^{17}}$

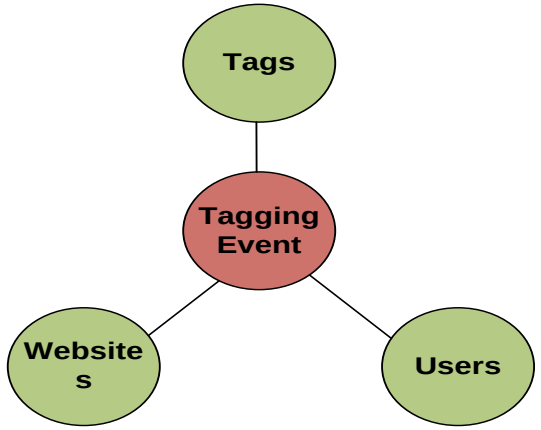
Table 2: Perplexity Comparison between Models with/without Historical Prior

Case Study on DBLP

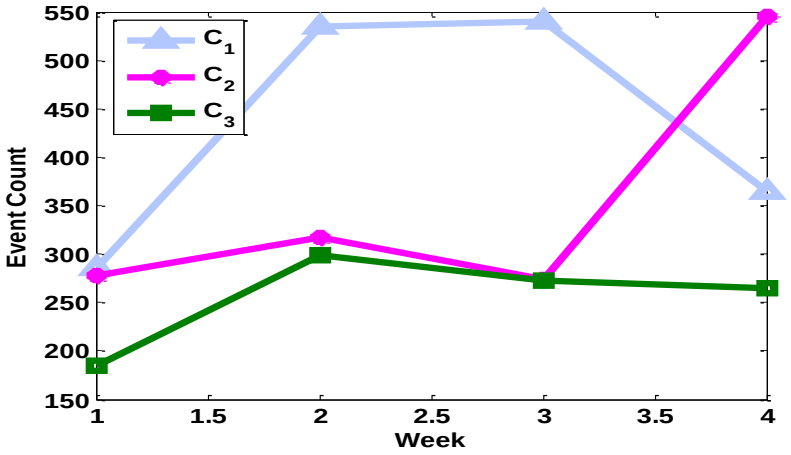
- Tracking database community evolution



Case Study on Delicious.com



Delicious Schema

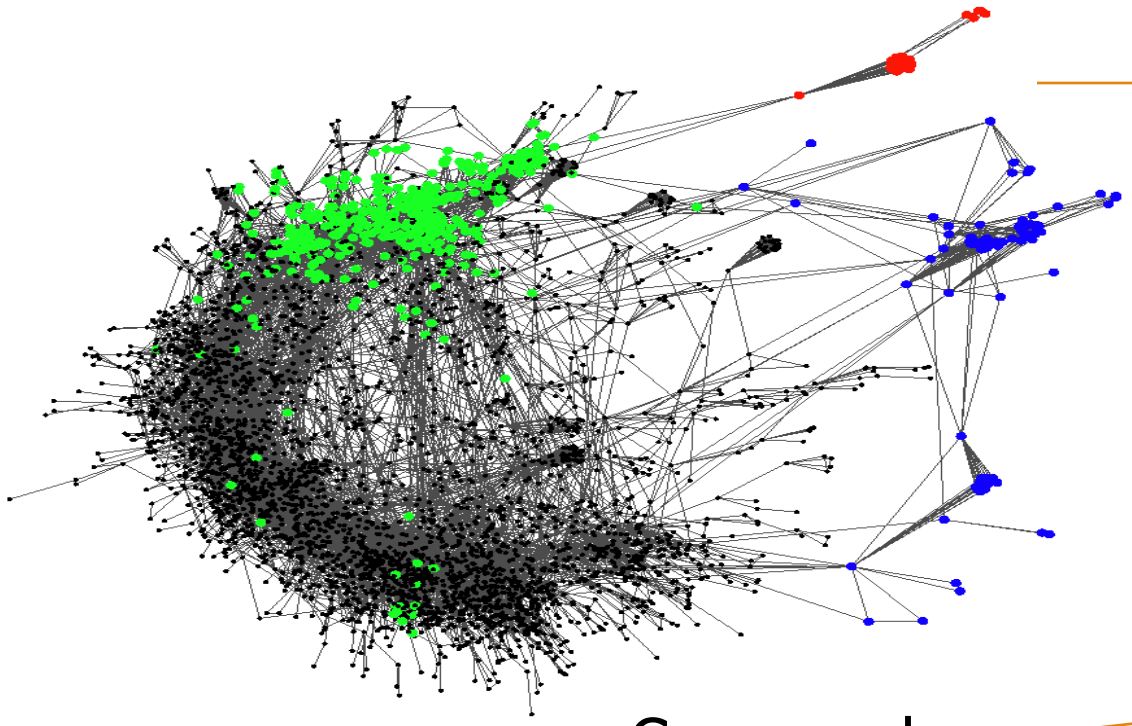


	Jan. 1 - Jan. 7	Jan. 8 – Jan. 14	Jan. 15 – Jan. 21	Jan. 22 – Jan. 28
C_1:	Security	Google	Security	Google
	Terrorism	China	Google	Security
	Politics	Security	China	China
	Travel	Internet	Internet	Internet
	Usa	Privacy	Microsoft	Privacy
	Airport	Politics	Privacy	Digg
	Israel	Censorship	Censorship	Politics
	Obama	Facebook	Politics	Datenschutz
	CIA	Business	Browser	Facebook
	Afghanistan	Terrorism	USA	USA
C_2:	Mac	Iphone	Iphone	Ipad
	Apple	Apple	Apple	Apple
	Iphone	Twitter	Mac	Iphone
	Windows	Mac	Mobile	Technology
	Tablet	Mobile	Twitter	Tablet
	Ipod	Apps	Software	Mac
	Tips	Ratio	Apps	Mobile
	Macbook	Blog	Business	Newspapers
	Tutorial	Newspapers	Osx	Kindle
	Drm	Technology	Radio	Media
C_3:	Health	Weather	Haiti	Haiti
	Depression	UK	Photography	BBC
	Sleep	Photography	BBC	Photography
	Teenagers	Photo	Earthquake	Animals
	Dubai	Haiti	Photos	Earthquake
	Tallest	Photos	UK	2010
	BBC	2010	2010	Photos
	Building	BBC	Disaster	Nature
	Architecture	Snow	Travel	Funny
	Mentalhealth	Earthquake	Wildlife	Theonion

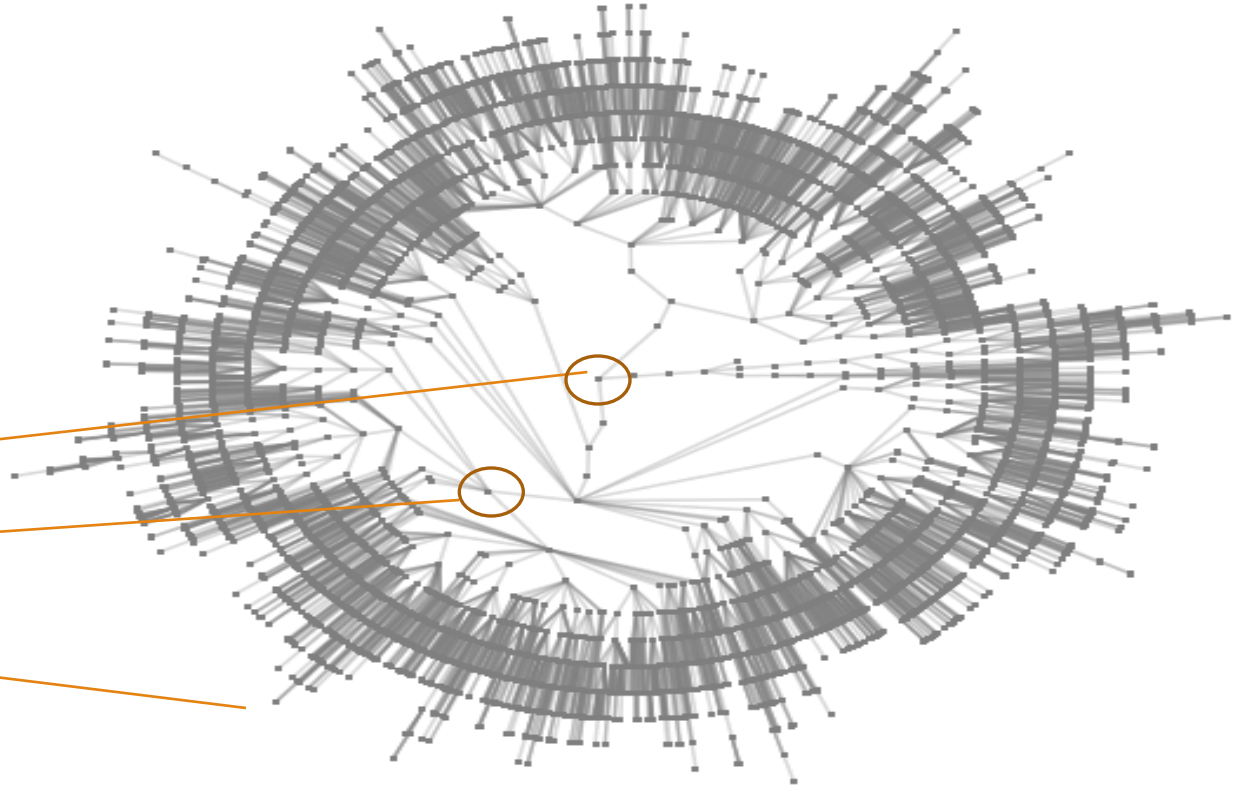
Outline

- **Motivation:** Why Mining Information Networks?
- **Part I:** Clustering and Ranking in Heterogeneous Information Networks
- **Part II:** Classification and Prediction in Heterogeneous Information Networks
- **Part III:** Other Data Mining Functions for Heterogeneous Information Networks
  - Mining Evolution and Dynamics of Information Networks
  - Role Discovery
  - OLAP in Information Networks
  - Data Cleaning: Distinct
  - CrossMine: Classification Across Multi-Relational Databases
  - EgoSet: Exploiting Word Ego-Networks for Multifaceted Set Expansion
- **Summary**

Role Discovery in Networks: Why Does It Matter?



Army communication network
(imaginary)



Automatically infer

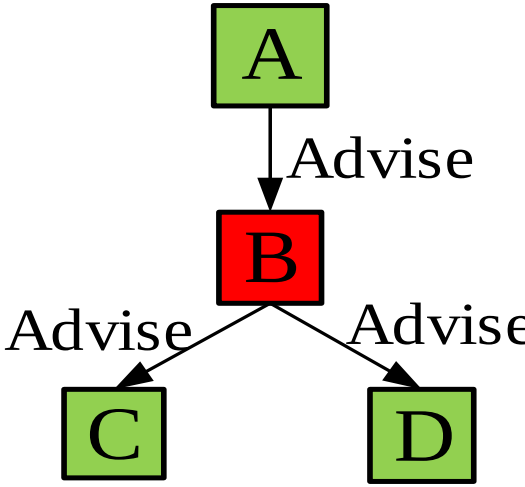
Commander

Captain

Solider

Mining Hierarchical Relationships

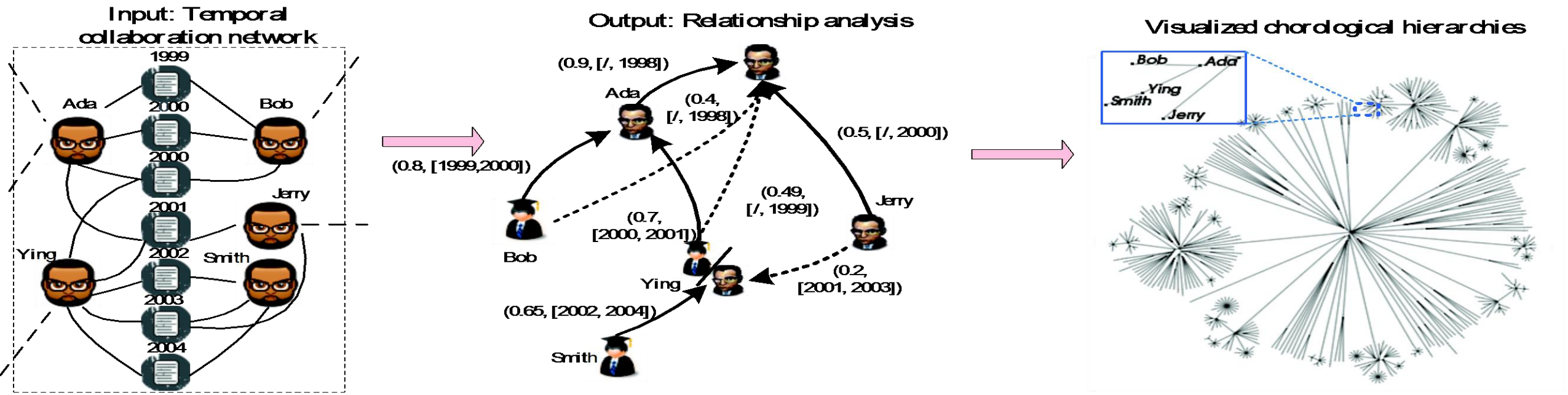
- Objective: Extract semantic meaning from plain links to finely model and better organize information networks
- Challenges: Latent semantic knowledge, interdependency, scalability
- Mining hierarchical relationships:
  - Output a tree (forest) for a given set of objects
  - Examples
    - Academic family tree
    - Company organization tree
    - Online forum discussion tree



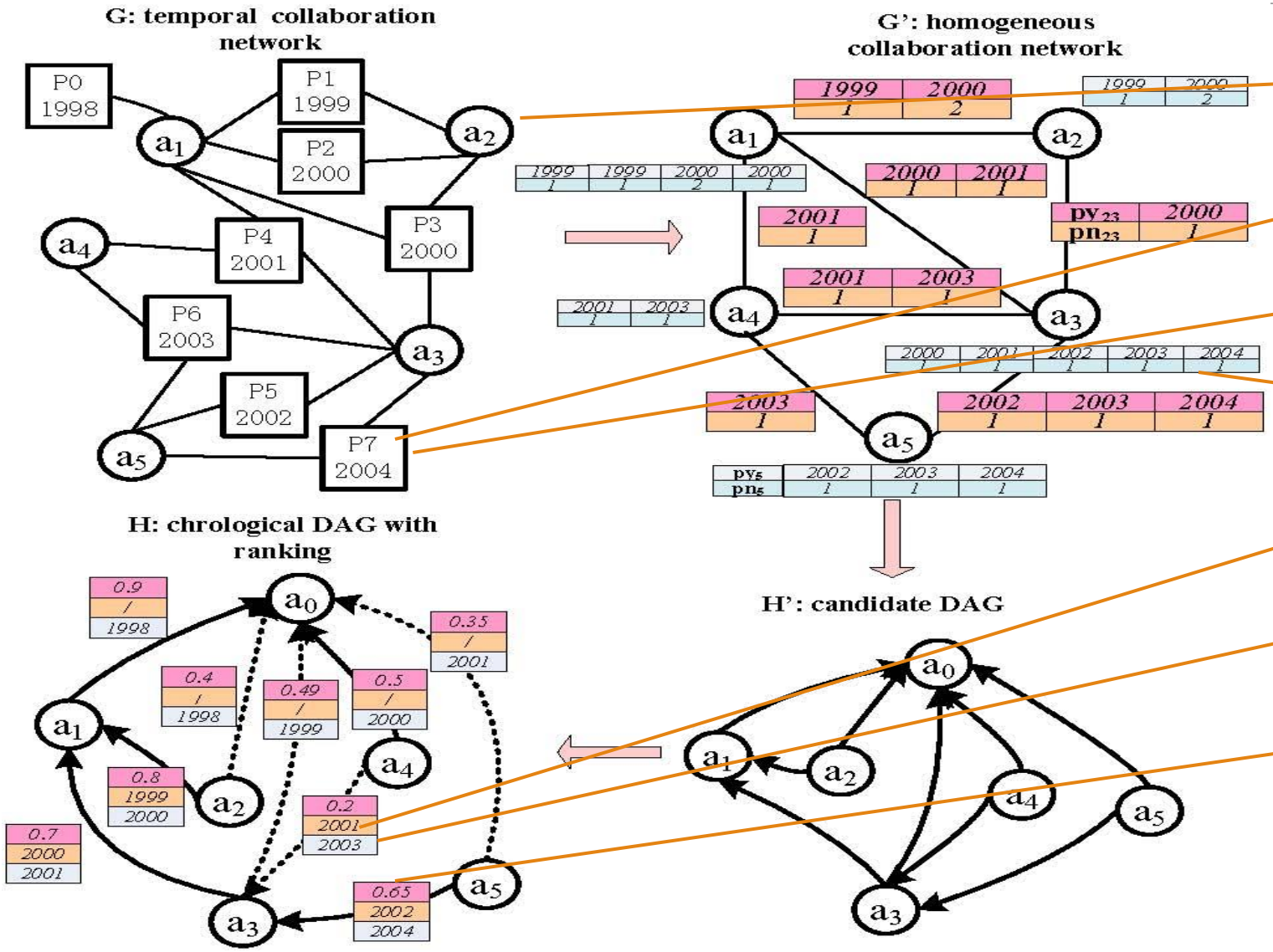
- Opportunity + heuristics: (i) Human intuition, (ii) Realistic constraint, (iii) Crosscheck with collective intelligence
- Methodology: propagate simple intuitive rules and constraints over the whole network

Discovery of Advisor-Advisee Relationships in DBLP Network

- Propagation of simple, commonly accepted constraints in Time-Constrained Probabilistic Factor Graph (TPFG)
  - *"Advisor has more publications and longer history than advisee at the time of advising"*
  - *"Once an advisee becomes advisor, s/he will not become advisee again"*

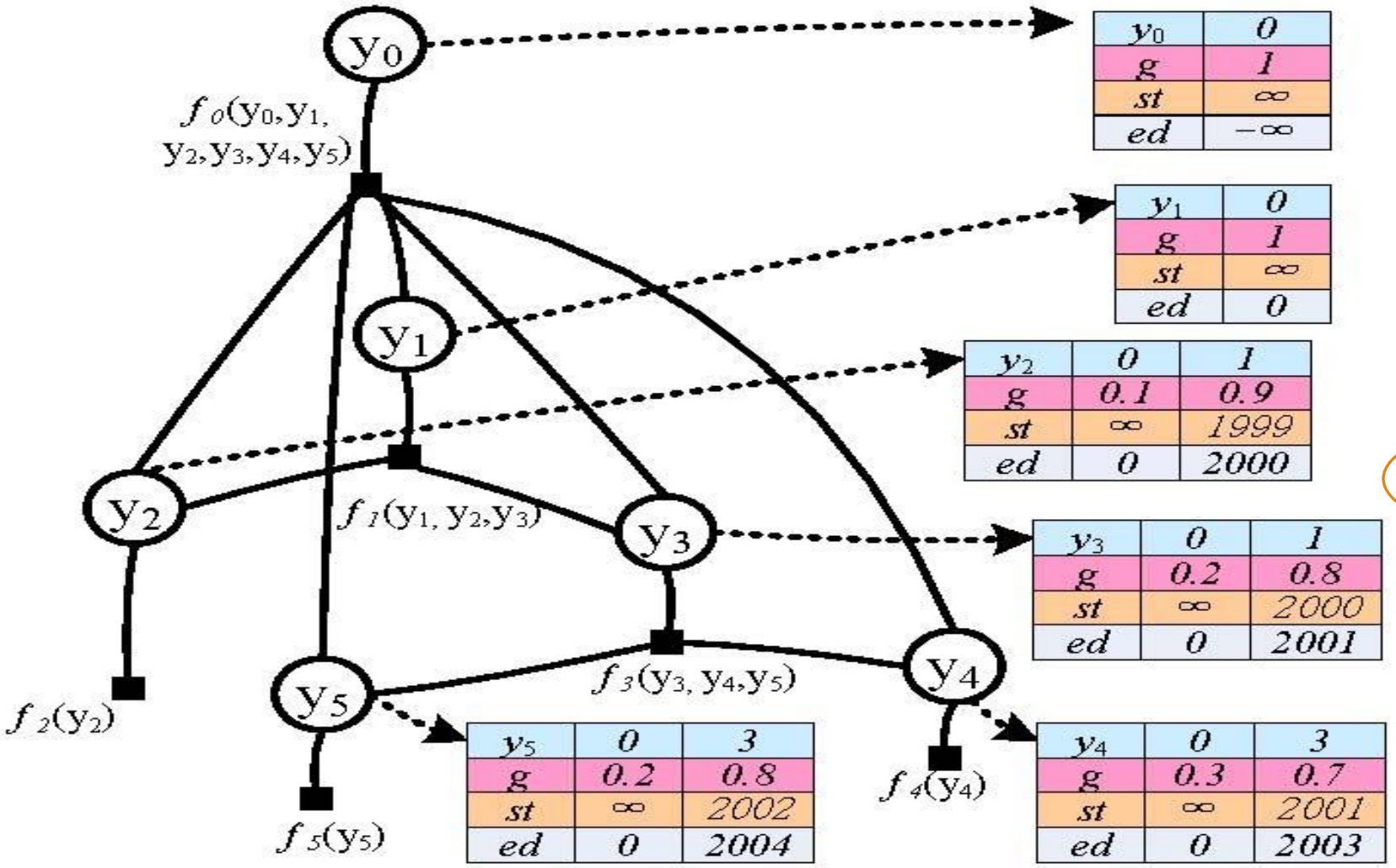


Overall Framework



- a_i: author i
- p_j: paper j
- py: paper year
- pn: paper#
- $st_{i,yi}$: starting time
- $ed_{i,yi}$: ending time
- $r_{i,yi}$: ranking score

Time-Constrained Probabilistic Factor Graph (TPFG)



y_0	0
g	1
st	∞
ed	−∞

y_1	0
g	1
st	∞
ed	0

y_2	0	1
g	0.1	0.9
st	∞	1999
ed	0	2000

y_3	0	1
g	0.2	0.8
st	∞	2000
ed	0	2001

y_5	0	3
g	0.2	0.8
st	∞	2002
ed	0	2004

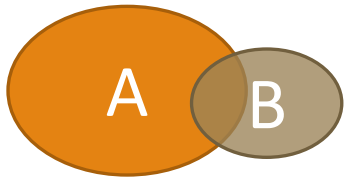
y_4	0	3
g	0.3	0.7
st	∞	2001
ed	0	2003

- y_x: a_x's advisor
- $st_{x,yx}$: starting time
  $ed_{x,yx}$: ending time
- $g(y_x, st_x, ed_x)$ is predefined local feature
- $f_x(y_x, Z_x) = \max g(y_x, st_x, ed_x)$ *under time constraint*
- Objective function $P(\{y_x\}) = \prod_x f_x(y_x, Z)$
- $Z = \{z \mid x \in Y_z\}$
- Y_x: set of potential advisors of a_x

Measures for Finding Advisor and Advisees

- Right heuristics: Advisee (B) tends to coauthor with advisor (A) during the advising period
- Kulczinski measure :

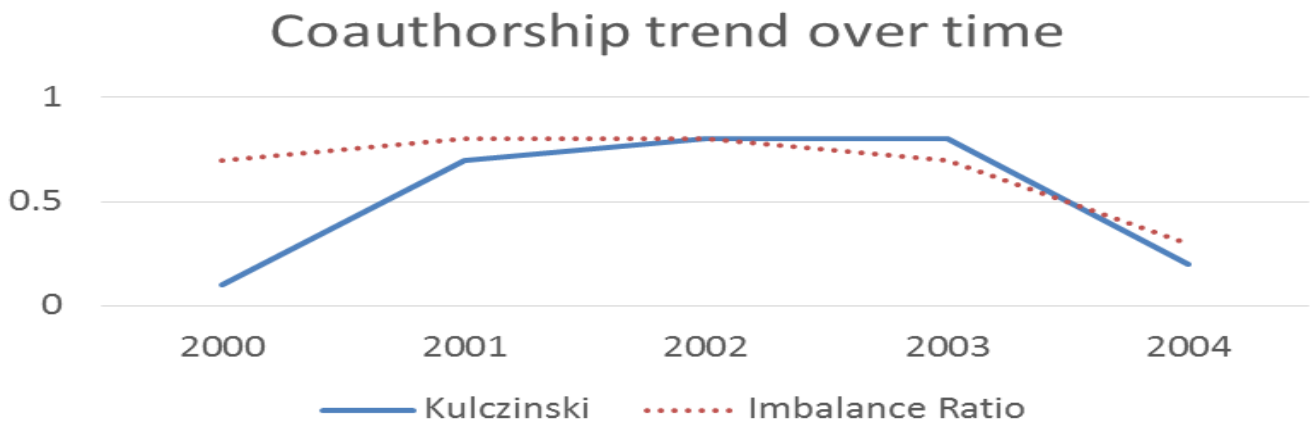
$$kulc(A,B) = \frac{|A \cap B|}{2}\left(\frac{1}{|A|} + \frac{1}{|B|}\right)$$



  - high overlap between their publications – Kulczinski measure
  - It is more often to see advisor in a publication but not advisee: Imbalance Ratio measure:

$$IR(A,B) = \frac{|A| - |B|}{|A \cup B|}$$

    - start time ≈ the year they start to collaborate
    - end time ≈ the year Kulczinski measure dropped significantly



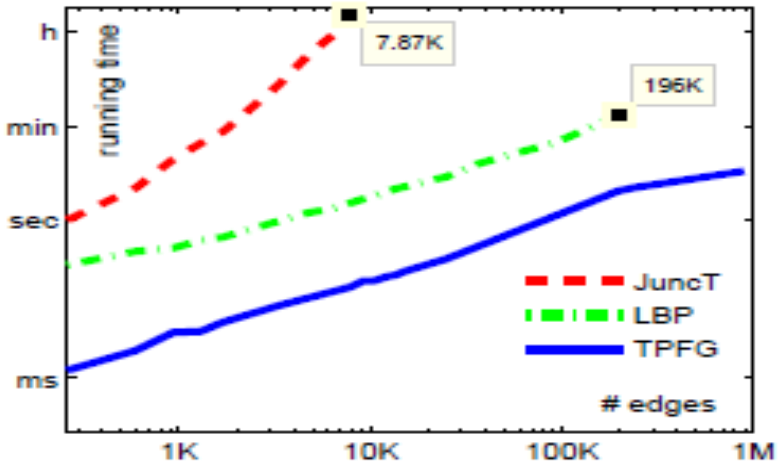
Experiment Results

- DBLP data: 654, 628 authors, 1076,946 publications, years provided
- Labeled data: MathGealogy Project; AI Gealogy Project; Homepage

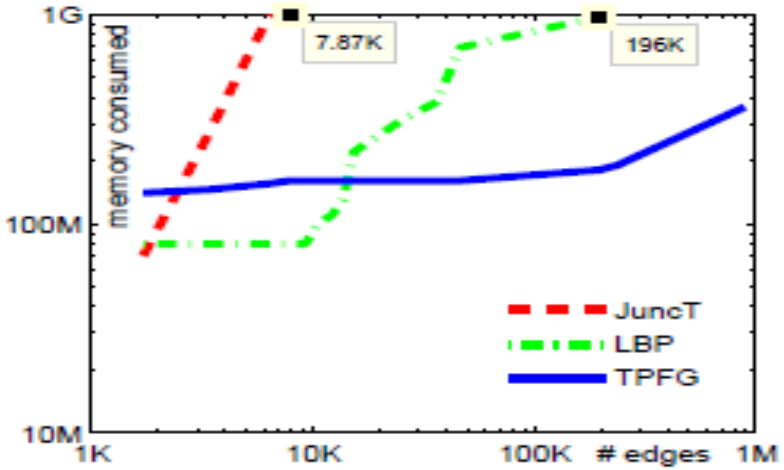
Datasets	RULE	SVM	IndMAX		TPFG	
TEST1	69.9%	73.4%	75.2%	78.9%	80.2%	**84.4%**
TEST2	69.8%	74.6%	74.6%	79.0%	81.5%	**84.3%**
TEST3	80.6%	86.7%	83.1%	90.9%	88.8%	**91.3%**

Case Study & Scalability

Advisee	Top Ranked Advisor	Time	Note
David M. Blei	1. Michael I. Jordan	01-03	PhD advisor, 2004 grad
	2. John D. Lafferty	05-06	Postdoc, 2006
Hong Cheng	1. Qiang Yang	02-03	MS advisor, 2003
	2. Jiawei Han	04-08	PhD advisor, 2008
Sergey Brin	1. Rajeev Motawani	97-98	"Unofficial advisor"



(a) Time



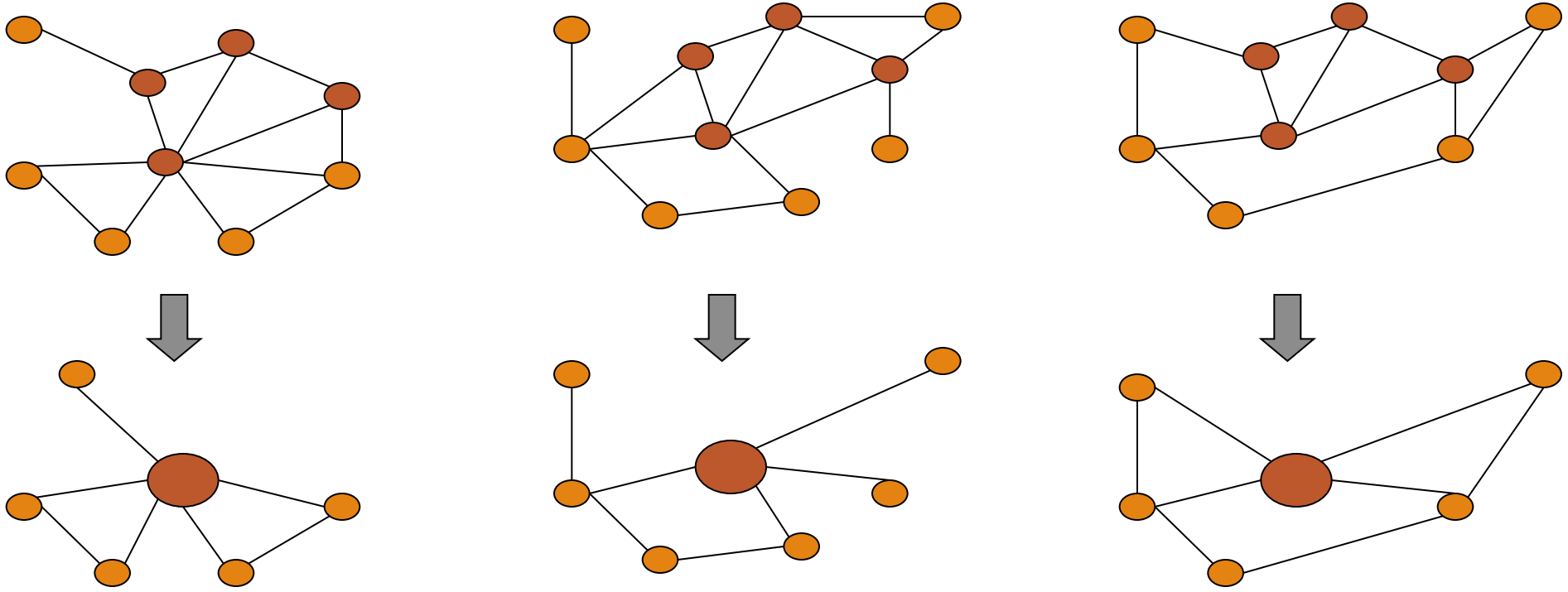
(b) Space

Outline

- **Motivation:** Why Mining Information Networks?
- **Part I:** Clustering and Ranking in Heterogeneous Information Networks
- **Part II:** Classification and Prediction in Heterogeneous Information Networks
- **Part III:** Other Data Mining Functions for Heterogeneous Information Networks
  - Mining Evolution and Dynamics of Information Networks
  - Role Discovery
  - OLAP in Information Networks
  - Data Cleaning: Distinct
  - CrossMine: Classification Across Multi-Relational Databases
  - EgoSet: Exploiting Word Ego-Networks for Multifaceted Set Expansion
- **Summary**

Graph/Network Summarization: Graph Compression

- Extract common subgraphs and simplify graphs by condensing these subgraphs into nodes

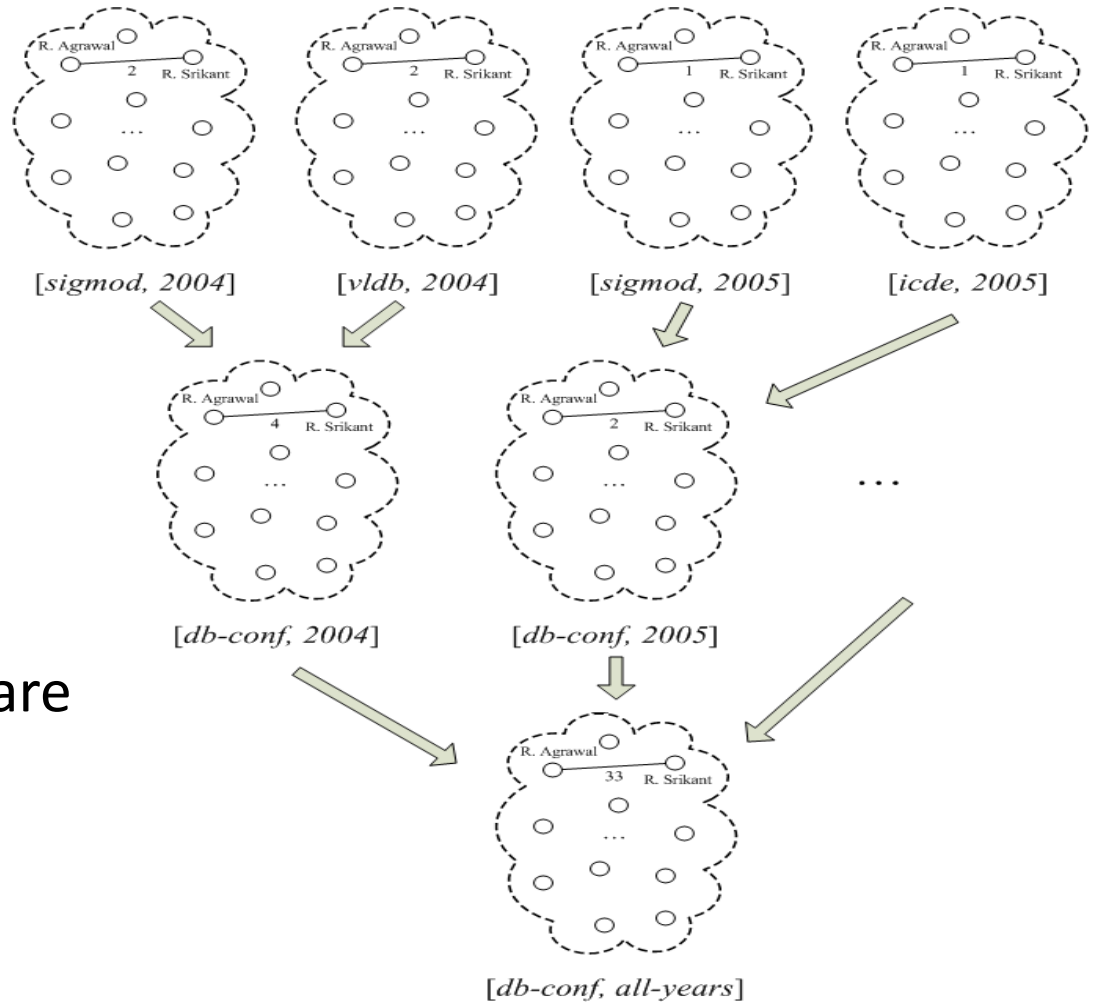


OLAP on Information Networks

- Why OLAP information networks?
- Advantages of OLAP: Interactive exploration of multi-dimensional and multi-level space in a data cube Infonet
  - Multi-dimensional: Different perspectives
  - Multi-level: Different granularities
- InfoNet OLAP: Roll-up/drill-down and slice/dice on information network data
  - Traditional OLAP cannot handle this, because they ignore links among data objects
- Handling two kinds of InfoNet OLAP
  - Informational OLAP
  - Topological OLAP

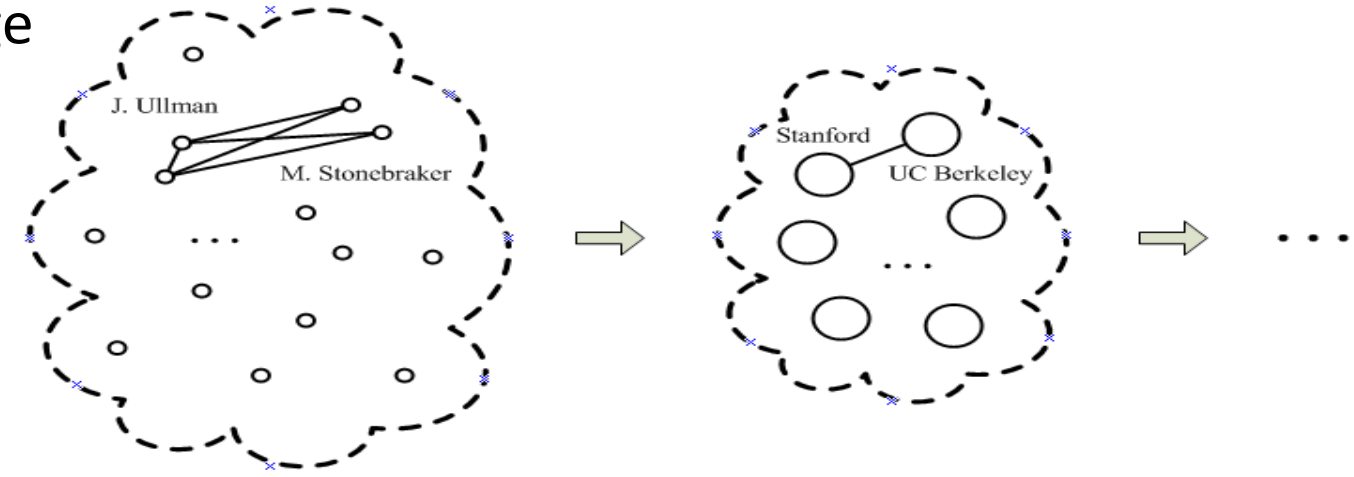
Informational OLAP

- In the DBLP network, study the collaboration patterns among researchers
- Dimensions come from informational attributes attached at the whole snapshot level, so-called *Info-Dims*
- I-OLAP Characteristics:
  - Overlay multiple pieces of information
  - No change on the objects whose interactions are being examined
    - In the underlying snapshots, each node is a researcher
    - In the summarized view, each node is still a researcher



Topological OLAP

- Dimensions come from the node/edge attributes inside individual networks, so-called *Topo-Dims*
- T-OLAP Characteristics
  - Zoom in/Zoom out
  - Network topology changed: "generalized" nodes and "generalized" edges
    - In the underlying network, each node is a researcher
    - In the summarized view, each node becomes an institute that comprises multiple researchers



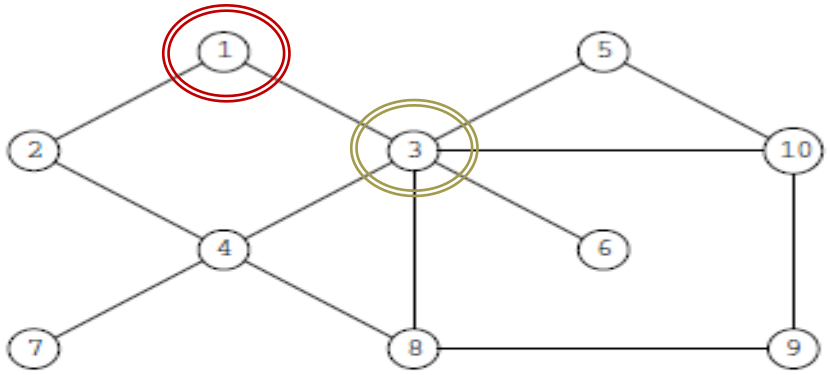
InfoNet OLAP: Operations & Framework

	InfoNet I-OLAP	InfoNet T-OLAP
Roll-up	Overlay multiple snapshots to form a higher-level summary via I-aggregated network	Shrink the topology & obtain a T-aggregated network that represents a compressed view, with topological elements (i.e., nodes and/or edges) merged and replaced by corresp. higher-level ones
Drill-down	Return to the set of lower-level snapshots from the higher-level overlaid (aggregated) network	A reverse operation of roll-up
Slice/dice	Select a subset of qualifying snapshots based on Info-Dims	Select a subnetwork based on Topo-Dims

- Measure is an aggregated graph & other measures like node count, average and degree can be treated as derived
- Graph plays a dual role: (1) data source, and (2) aggregate measure
- Measures could be complex, e.g., maximum flow, shortest path, centrality
- It is possible to combine I-OLAP and T-OLAP into a hybrid case

Graph Cube: Online Analytical Processing in Multidimensional Information Networks

A Multidimensional Information Network



(a) Graph

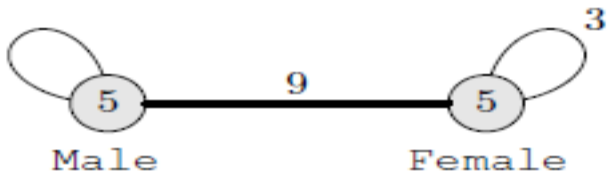
ID	Gender	Location	Profession	Income
1	Male	CA	Teacher	$70,000
2	Female	WA	Teacher	$65,000
3	Female	CA	Engineer	$80,000
4	Female	NY	Teacher	$90,000
5	Male	IL	Lawyer	$80,000
6	Female	WA	Teacher	$90,000
7	Male	NY	Lawyer	$100,000
8	Male	IL	Engineer	$75,000
9	Female	CA	Lawyer	$120,000
10	Male	IL	Engineer	$95,000

(b) Vertex Attribute Table

Figure: A Multidimensional Network Comprising a Graph Structure and a Multidimensional Vertex Attribute Table

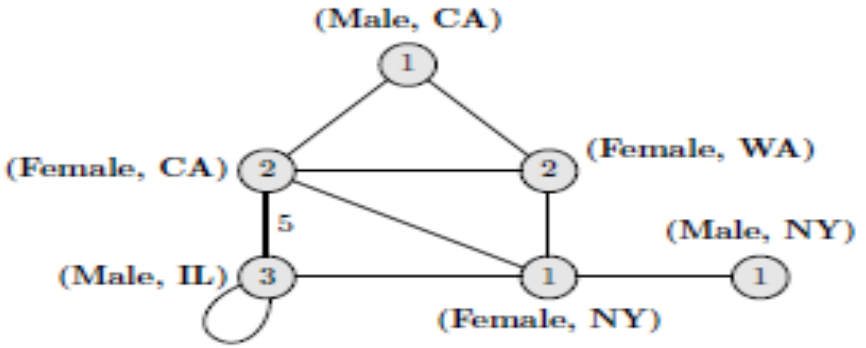
Conventional Group-by vs. Network Summarization

Gender	COUNT(*)
Male	5
Female	5



Group by "Gender"

Gender	Location	COUNT(*)
Male	CA	1
Female	CA	2
Female	WA	2
Male	IL	3
Male	NY	1
Female	NY	1



Group by "Gender" and "Location"

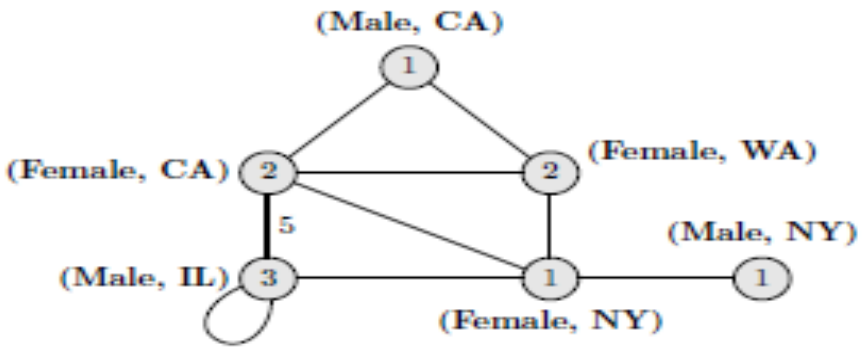
30

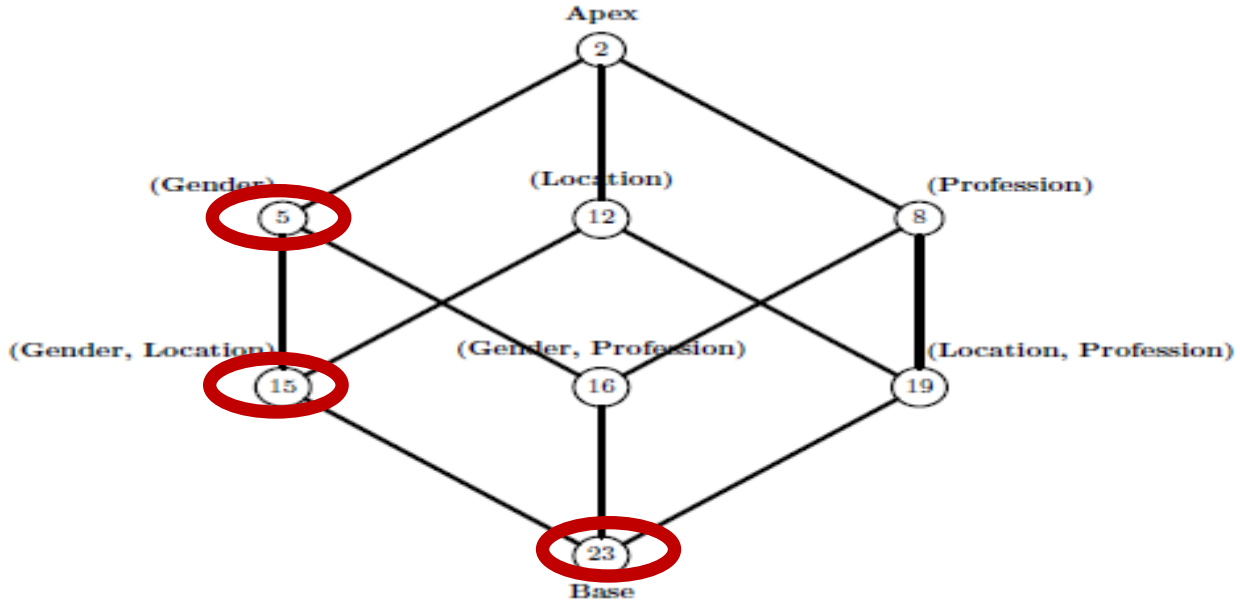
The Graph Cube Model

- Multidimensional network $N = (V, E; A)$
  - $A = \{A_1, A_2, \ldots\ldots, A_n\}$, the **dimension** of the network N, is a set of n vertex-specific attributes
    - Some (or all) dimension A_i could be *(ALL), representing a super-aggregation along A_i
    - There exist 2^n multidimensional spaces (aggregations)
  - The **measure** within each possible space is no longer a simple numeric value, but an **aggregate network!**

The Graph Cube Model

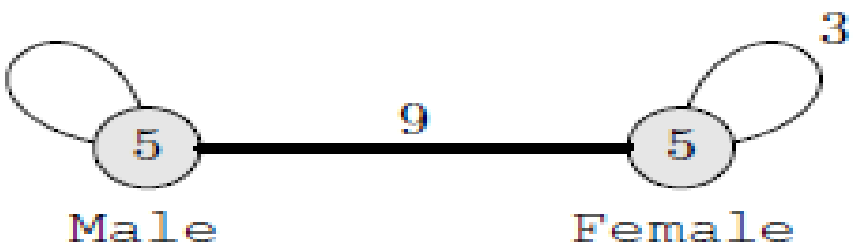
- Graph Cube
  - Restructure the network in **all possible multidimensional spaces (cuboids)** defined on *A*
  - For each multidimensional space *A'*, the measure is an **aggregate network** w.r.t. *A'*

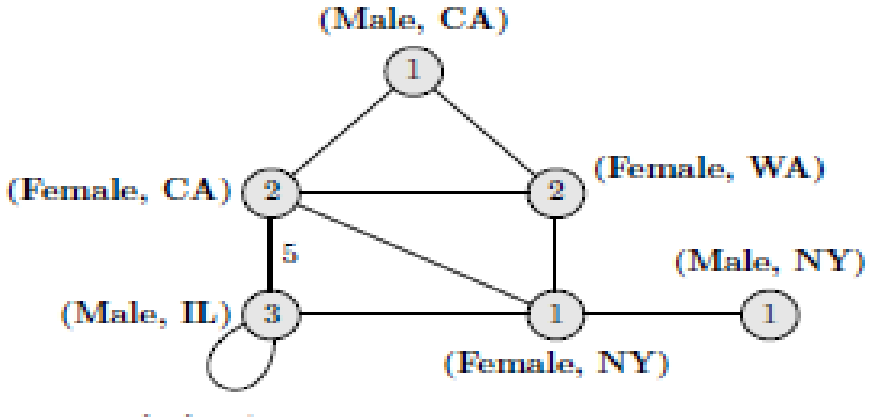




OLAP on Graph Cube

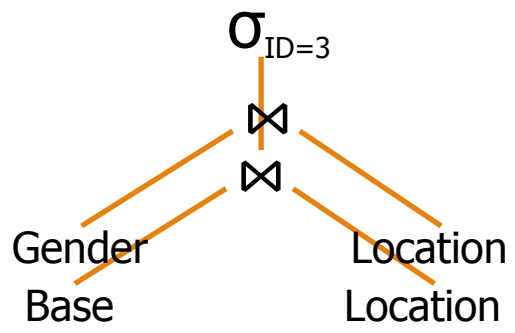
- Cuboid query
  - Return as output the aggregate network corresponding to a specific multidimensional space (**cuboid**)
    - *What is the aggregate network between various genders?*
    - *What is the aggregate network between various gender and location combinations?*

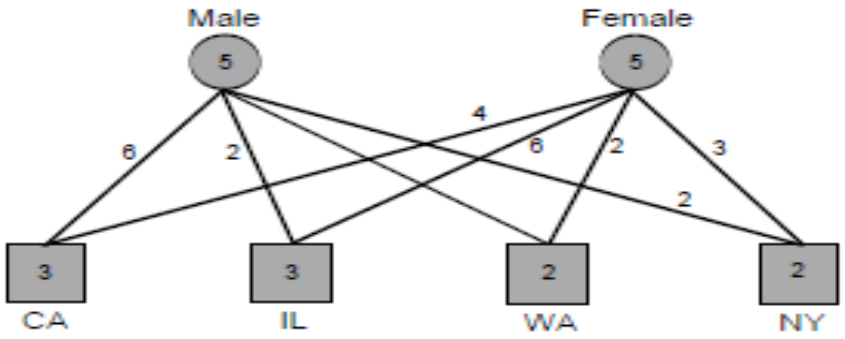




OLAP on Graph Cube

- Cuboid query
  - Within a single multidimensional space
- Crossboid query (⋈)
  - Crosses multiple multidimensional spaces of the network
    - *What is the network structure between user 3 and various locations?*
    - *What is the network structure between users grouped by gender vs. users grouped by location?*





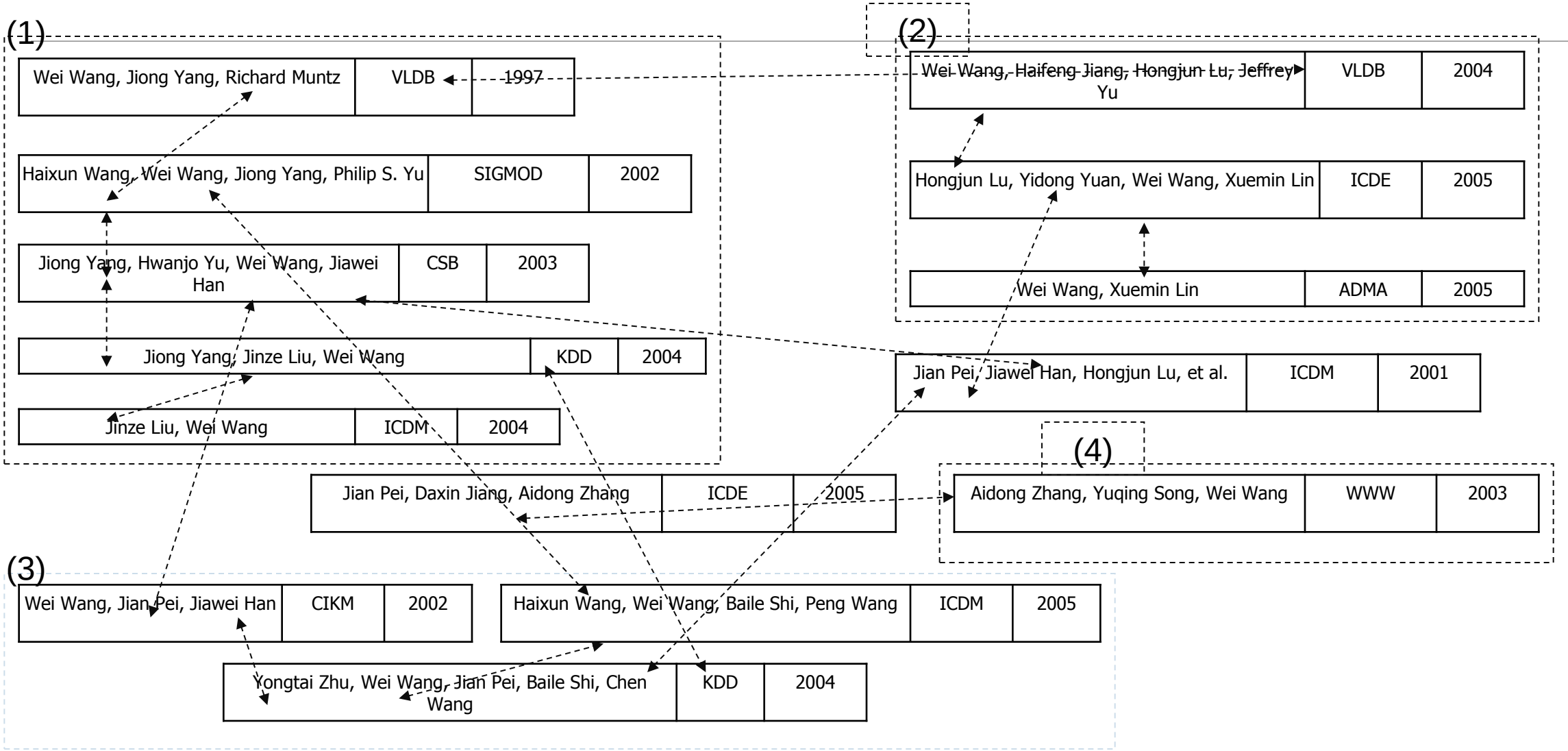
Outline

- **Motivation:** Why Mining Information Networks?
- **Part I:** Clustering and Ranking in Heterogeneous Information Networks
- **Part II:** Classification and Prediction in Heterogeneous Information Networks
- **Part III:** Other Data Mining Functions for Heterogeneous Information Networks
  - Mining Evolution and Dynamics of Information Networks
  - Role Discovery
  - OLAP in Information Networks
  - Data Cleaning: Distinct
  - CrossMine: Classification Across Multi-Relational Databases
  - EgoSet: Exploiting Word Ego-Networks for Multifaceted Set Expansion
- **Summary**

Data Cleaning by Link Analysis

- Object reconciliation vs. object distinction as data cleaning tasks
- Link analysis may take advantages of redundancy and make facilitate entity cross-checking and validation
- Object distinction: Different people/objects do share names
  - In AllMusic.com, 72 songs and 3 albums named “Forgotten” or “The Forgotten”
  - In DBLP, 141 papers are written by at least 14 “Wei Wang”
- New challenges of object distinction:
  - Textual similarity cannot be used
- Distinct: Object distinction by information network analysis
  - X. Yin, J. Han, and P. S. Yu, “Object Distinction: Distinguishing Objects with Identical Names by Link Analysis”, ICDE'07

Entity Distinction: The "Wei Wang" Challenge in DBLP



(1) Wei Wang at UNC (2) Wei Wang at UNSW, Australia

(3) Wei Wang at Fudan Univ., China (4) Wei Wang at SUNY Buffalo

The DISTINCT Methodology

- Measure similarity between references
  - Link-based similarity: Linkages between references
    - References to the same object are more likely to be connected (Using random walk probability)
  - Neighborhood similarity
    - Neighbor tuples of each reference can indicate similarity between their contexts
- Self-boosting: Training using the “same” bulky data set
- Reference-based clustering
  - Group references according to their similarities

Training with the "Same" Data Set

- Build a training set automatically
  - Select distinct names, e.g., Johannes Gehrke
  - The collaboration behavior within the same community share some similarity
  - Training parameters using a typical and large set of "unambiguous" examples
- Use SVM to learn a model for combining different join paths
  - Each join path is used as two attributes (with link-based similarity and neighborhood similarity)
  - The model is a weighted sum of all attributes

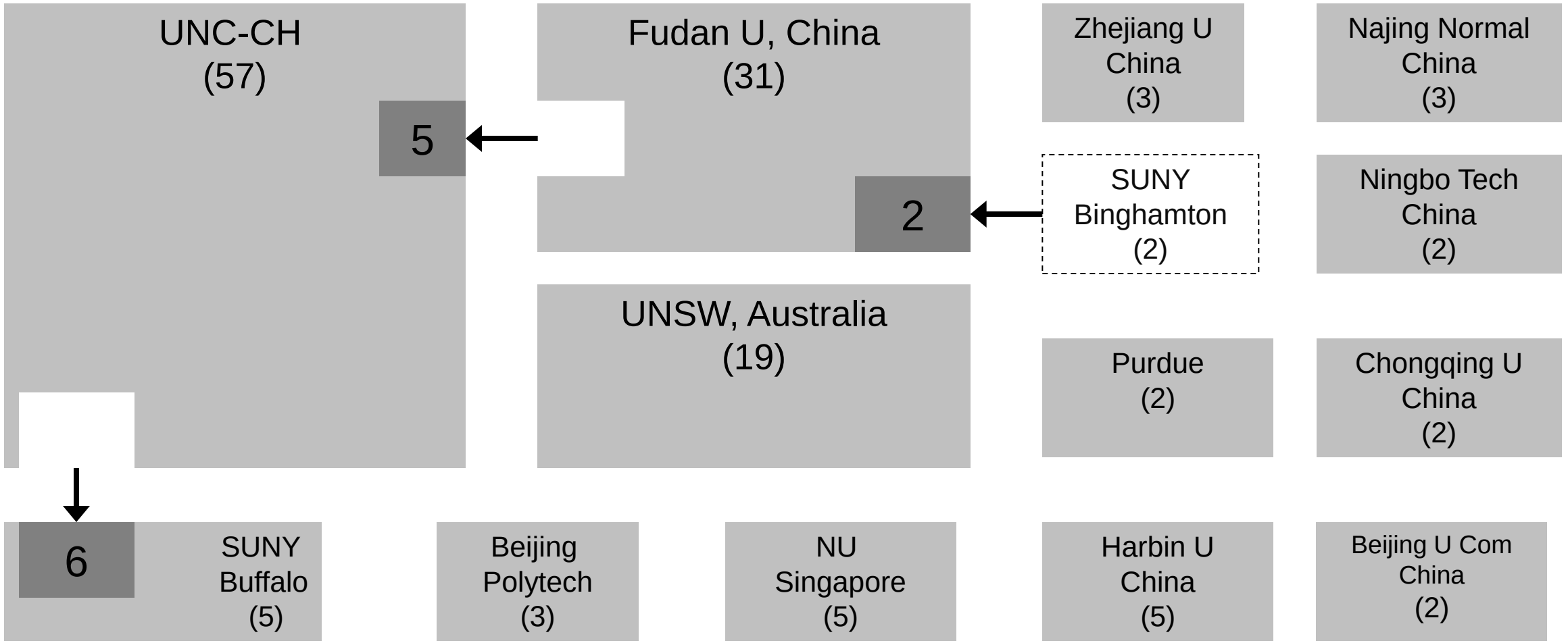
Clustering: Measure Similarity between Clusters

- Single-link (highest similarity between points in two clusters) ?
  - No, because references to different objects can be connected.
- Complete-link (minimum similarity between them)?
  - No, because references to the same object may be weakly connected.
- Average-link (average similarity between points in two clusters)?
  - A better measure
  - *Refinement: Average neighborhood similarity* and *collective random walk probability*

Real Cases: DBLP Popular Names

Name	*#author*	*#ref*	*accuracy*	*precision*	*recall*	*f-measure*
Hui Fang	3	9	1.0	1.0	1.0	1.0
Ajay Gupta	4	16	1.0	1.0	1.0	1.0
Joseph Hellerstein	2	151	0.81	1.0	0.81	0.895
Rakesh Kumar	2	36	1.0	1.0	1.0	1.0
Michael Wagner	5	29	0.395	1.0	0.395	0.566
Bing Liu	6	89	0.825	1.0	0.825	0.904
Jim Smith	3	19	0.829	0.888	0.926	0.906
Lei Wang	13	55	0.863	0.92	0.932	0.926
Wei Wang	14	141	0.716	0.855	0.814	0.834
Bin Yu	5	44	0.658	1.0	0.658	0.794
average			0.81	0.966	0.836	0.883

Distinguishing Different "Wei Wang"s



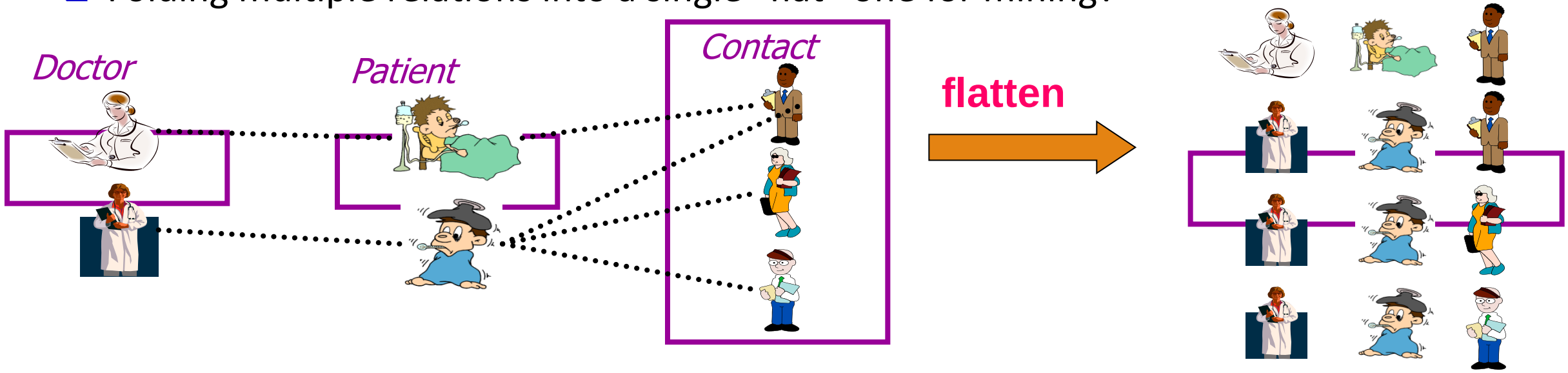

FREEDOM IS A LIGHT
FOR WHICH MANY MEN HAVE DIED IN DARKNESS
IN UNMARKED GRAVES WITHIN THIS SQUARE LIE THOUSANDS OF UNKNOWN SOLDIERS OF WASHINGTON'S ARMY WHO DIED OF WOUNDS AND SICKNESS DURING THE REVOLUTIONARY WAR
THE INDEPENDENCE AND LIBERTY YOU POSSESS ARE THE WORK OF JOINT COUNCILS AND JOINT EFFORTS-OF COMMON DANGERS, SUFFERINGS AND SUCCESS.
WASHINGTON'S FAREWELL ADDRESS SEPT. 17, 1796

Outline

- **Motivation:** Why Mining Information Networks?
- **Part I:** Clustering and Ranking in Heterogeneous Information Networks
- **Part II:** Classification and Prediction in Heterogeneous Information Networks
- **Part III:** Other Data Mining Functions for Heterogeneous Information Networks
  - Mining Evolution and Dynamics of Information Networks
  - Role Discovery
  - OLAP in Information Networks
  - Data Cleaning: Distinct
  - CrossMine: Classification Across Multi-Relational Databases
  - EgoSet: Exploiting Word Ego-Networks for Multifaceted Set Expansion
- **Summary**

Should We Flatten Relations at Multi-Relational Mining?

- Folding multiple relations into a single "flat" one for mining?

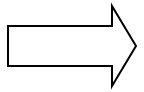


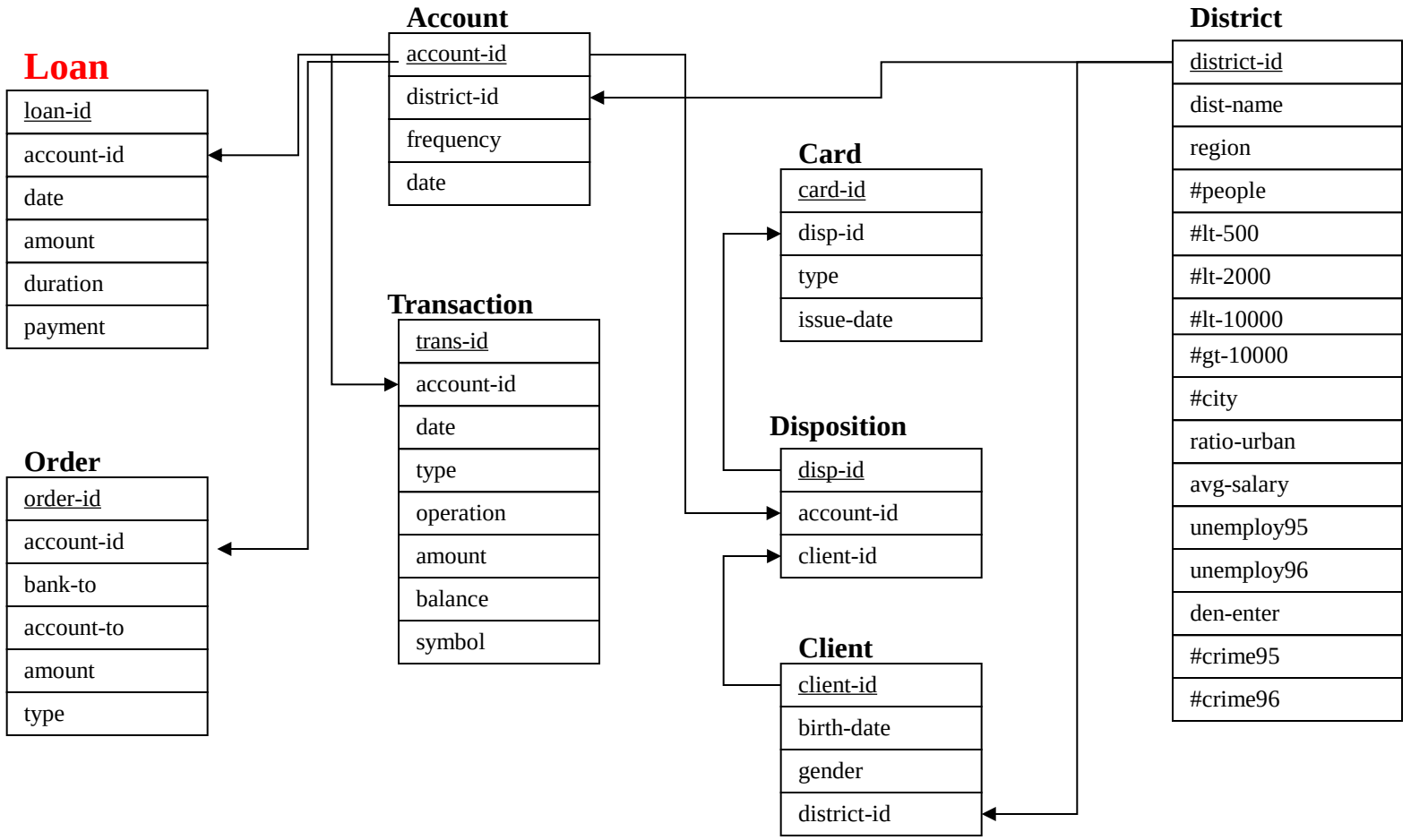
- Joining multiple relations? Cannot be a solution due to problems
  - Lose information of linkages and relationships, no semantics preservation
  - Cannot utilize information of database structures or schemas (e.g., E-R modeling)
  - Joining may distort the real quantitative relationship

Loan Applications: Backend Database

Target relation:

Each tuple has a class label, indicating whether a loan is paid on time.





How to make decisions to loan applications?

CrossMine: An Effective Multi-relational Classifier

- Methodology
  - Tuple-ID propagation: an efficient and flexible method for virtually joining relations
  - Confine the rule search process in promising directions
  - Look-one-ahead: a more powerful search strategy
  - Negative tuple sampling: improve efficiency while maintaining accuracy

Tuple ID Propagation


Applicant #1


Applicant #2


Applicant #3


Applicant #4

Loan ID	Account ID	Amount	Duration	Decision
1	124	1000	12	Yes
2	124	4000	12	Yes
3	108	10000	24	No
4	45	12000	36	No

Account ID	Frequency	Open date	Propagated ID	Labels
124	monthly	02/27/93	1, 2	2+, 0–
108	weekly	09/23/97	3	0+, 1–
45	monthly	12/09/96	4	0+, 1–
67	weekly	01/01/97	Null	0+, 0–

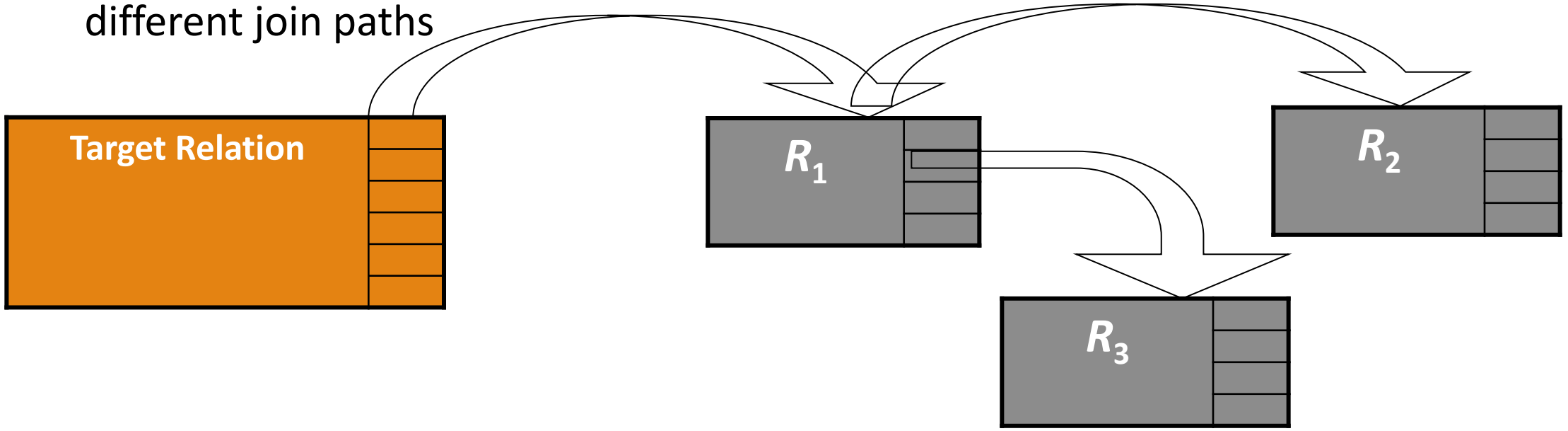
Possible predicates:

- Frequency='monthly': 2 +, 1 –
- Open date < 01/01/95: 2 +, 0 –

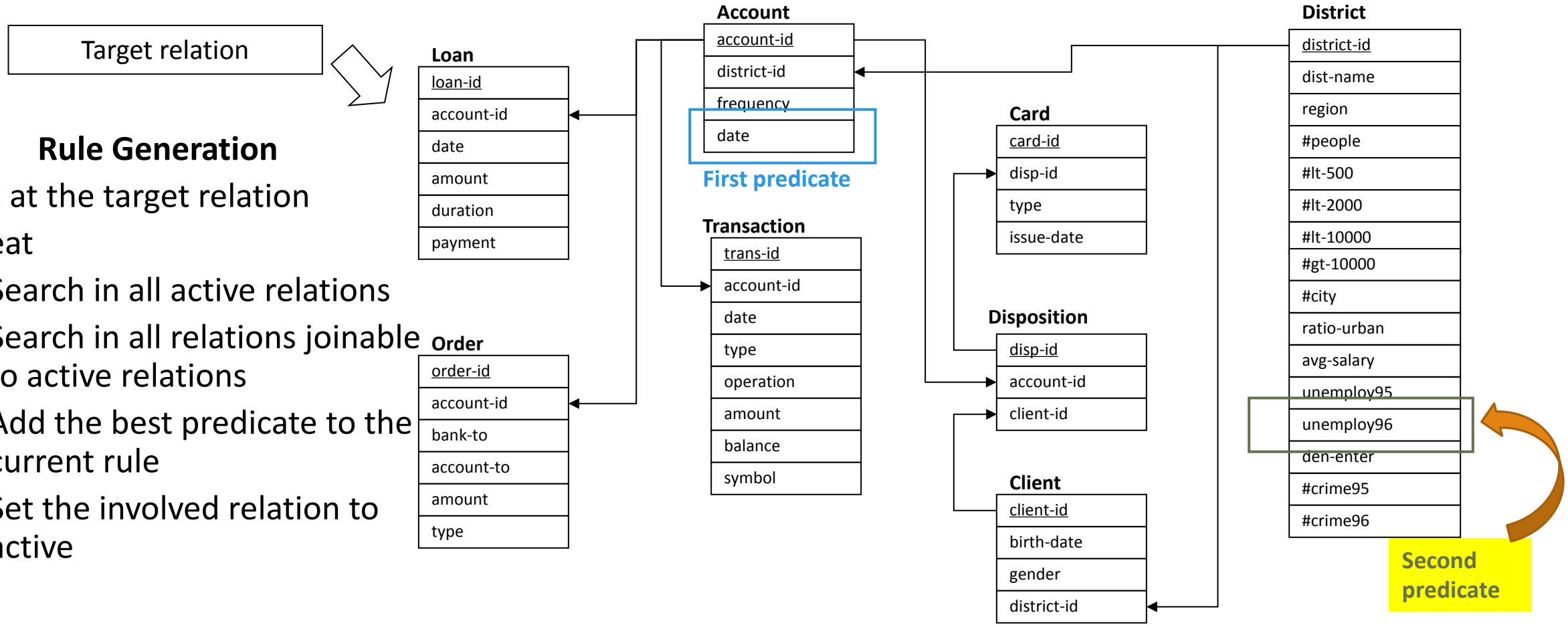
- Propagate tuple IDs of target relation to non-target relations
- Virtually join relations to avoid the high cost of physical joins

Tuple ID Propagation (Idea Outlined)

- Efficient
  - Only propagate the tuple IDs
  - Time and space usage is low
- Flexible
  - Can propagate IDs among non-target relations
  - Many sets of IDs can be kept on one relation, which are propagated from different join paths



Rule Generation: Example

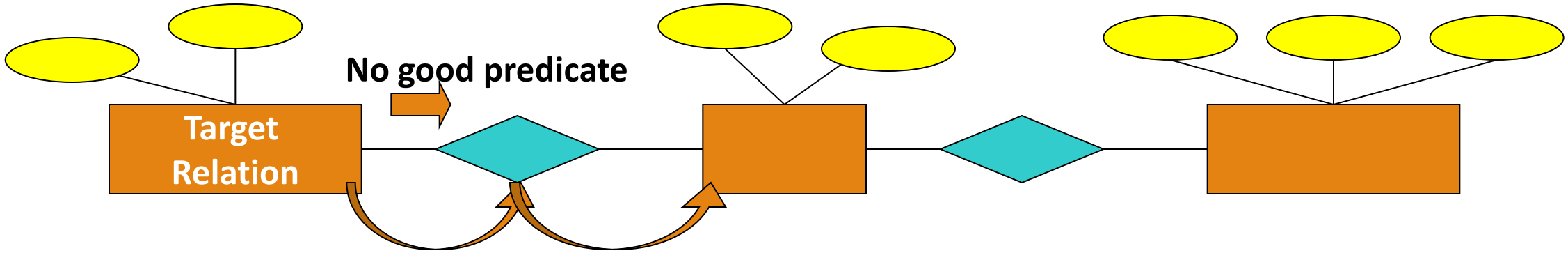


Rule Generation

- Start at the target relation
- Repeat
  - Search in all active relations
  - Search in all relations joinable to active relations
  - Add the best predicate to the current rule
  - Set the involved relation to active
- Until
  - The best predicate does not have enough gain
  - Current rule is too long

Look-One-Ahead in Rule Generation

- Two types of relations: Entity and Relationship
- Often cannot find useful predicates on relations of relationship



- Solution of CrossMine:
  - When propagating IDs to a relation of relationship, propagate one more step to next relation of entity

Rule Induction: An Inductive Logic Programming (ILP) Approach

- Find a hypothesis that is consistent with background knowledge (training data)
  - FOIL, Golem, Progol, TILDE, …
- Background knowledge
  - Relations (predicates), Tuples (ground facts)
- Inductive Logic Programming (ILP)
  - Hypothesis: The hypothesis is usually a set of rules, which can predict certain attributes in certain relations
    - Daughter(X, Y) ← female(X), parent(Y, X)

Training examples

Daughter(mary, ann)	+
Daughter(eve, tom)	+
Daughter(tom, ann)	–
Daughter(eve, ann)	–

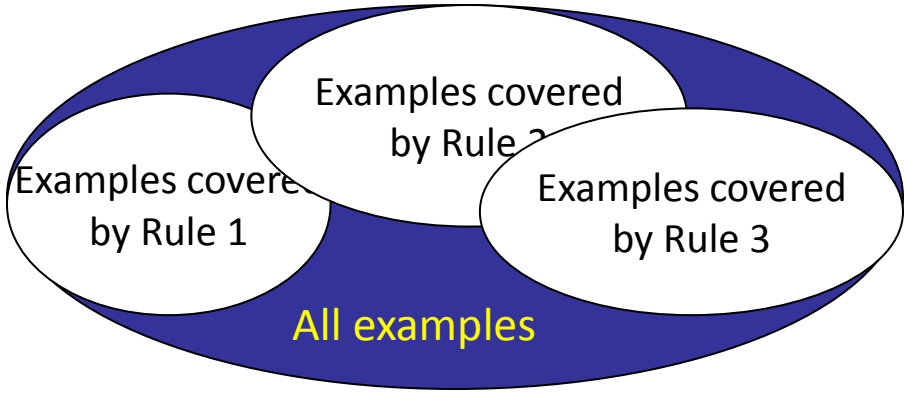
Background knowledge

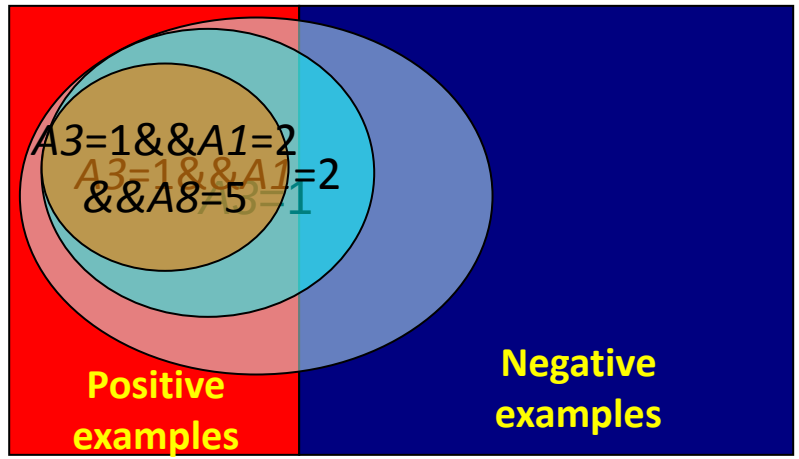
Parent(ann, mary)
Parent(ann, tom)
Parent(tom, eve)
Parent(tom, ian)

Female(ann)
Female(mary)
Female(eve)

FOIL: First-Order Inductive Learner (Rule Generation)

- Find a set of rules consistent with training data
- A top-down, sequential covering learner
- Build each rule by heuristics
  - Foil gain – a special type of information gain





- To generate a rule

while(true)

find the best predicate *p*

if foil-gain(*p*)>threshold **then** add *p* to current rule

else break

Find the Best Predicate: Predicate Evaluation

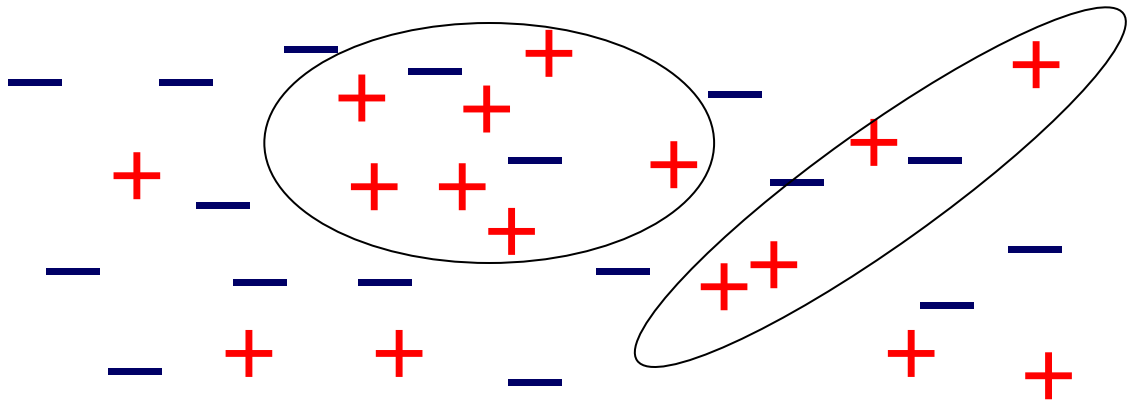
- All predicates in a relation can be evaluated based on propagated IDs
- Use *foil-gain* to evaluate predicates
  - Suppose current rule is *r*. For a predicate *p*,

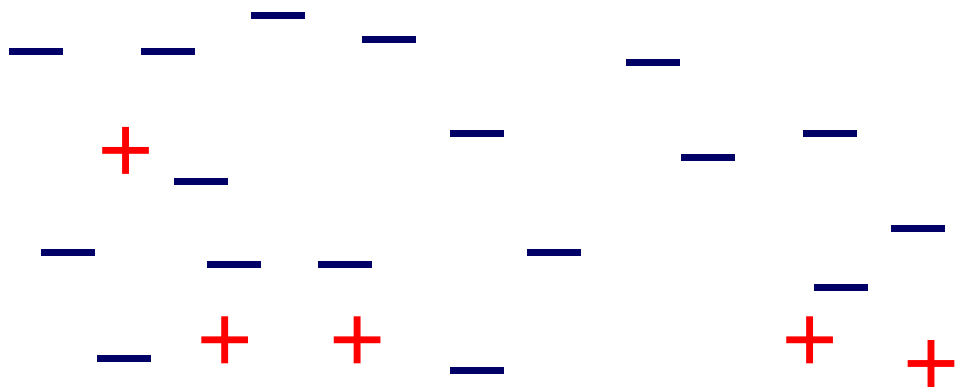
$$\textit{foil-gain}(p) = P(r+p) \times \left[-\log \frac{P(r)}{P(r)+N(r)} + \log \frac{P(r+p)}{P(r+p)+N(r+p)} \right]$$

- Categorical Attributes
  - Compute foil-gain directly
- Numerical Attributes
  - Discretize with every possible value

Negative Tuple Sampling

- Each time a rule is generated, covered positive examples are removed
- After generating many rules, there are much less positive examples than negative ones
  - Cannot build good rules (low support)
  - Still time consuming (large number of negative examples)
- Solution: Sampling on negative examples
  - Improve efficiency without affecting rule quality





CrossMine: Performance on Real Datasets

- PKDD Cup 99 dataset—Loan Application

	Accuracy	Time (per fold)
FOIL	74.0%	3338 sec
TILDE	81.3%	2429 sec
CrossMine	90.7%	15.3 sec

- Mutagenesis dataset (4 relations): Only 4 relations, so TILDE does a good job, though slow; but CrossMine is still efficient and good quality

	Accuracy	Time (per fold)
FOIL	79.7%	1.65 sec
TILDE	89.4%	25.6 sec
CrossMine	87.7%	0.83 sec

Outline

- **Motivation:** Why Mining Information Networks?
- **Part I:** Clustering and Ranking in Heterogeneous Information Networks
- **Part II:** Classification and Prediction in Heterogeneous Information Networks
- **Part III:** Other Data Mining Functions for Heterogeneous Information Networks
  - Mining Evolution and Dynamics of Information Networks
  - Role Discovery
  - OLAP in Information Networks
  - Data Cleaning: Distinct
  - CrossMine: Classification Across Multi-Relational Databases
  - EgoSet: Exploiting Word Ego-Networks for Multifaceted Set Expansion
- **Summary**

EgoSet: Exploiting Word Ego-Networks for Multifaceted Set Expansion

- Xin Rong, Zhe Chen, Qiaozhu Mei, Eytan Adar, "EgoSet: Exploiting Word Ego-Networks and User-Generated Ontology for Multifaceted Set Expansion", WSDM'16
- **Entity Set Expansion**
  - Given one or few examples (i.e., seeds), find a set of "sibling" entities such that the entities and the seeds belong to the same semantic class
  - Applications: Question answering, query suggestions, consumer vocabulary construction and knowledge extraction
  - Examples
    - Given {red, blue, green} → all the colors
    - However, given "orange" → colors or fruits?
- Challenge: **Multifaceted input seeds** → significant incoherence in the result set
- Solution: **Capture multiple facets** in the input query and **generate expansions for each facet with high precision**

EgoSet: Methodology

- Ontologies and skip grams: Combine existing **user-generated ontologies** (Wikipedia) with a novel word-similarity metric based on **skip-grams**
- **Ego-network generation:** Treat words that are distributionally similar to the seed (the ego) as nodes and use the pairwise similarity between those words to create weighted edges, thereby forming an "**ego-network**"
- **EgoSet discovery**: Use the ego-network to find the initial clusters for a seed, and align those clusters with user-created ontologies

Wikipedia lists

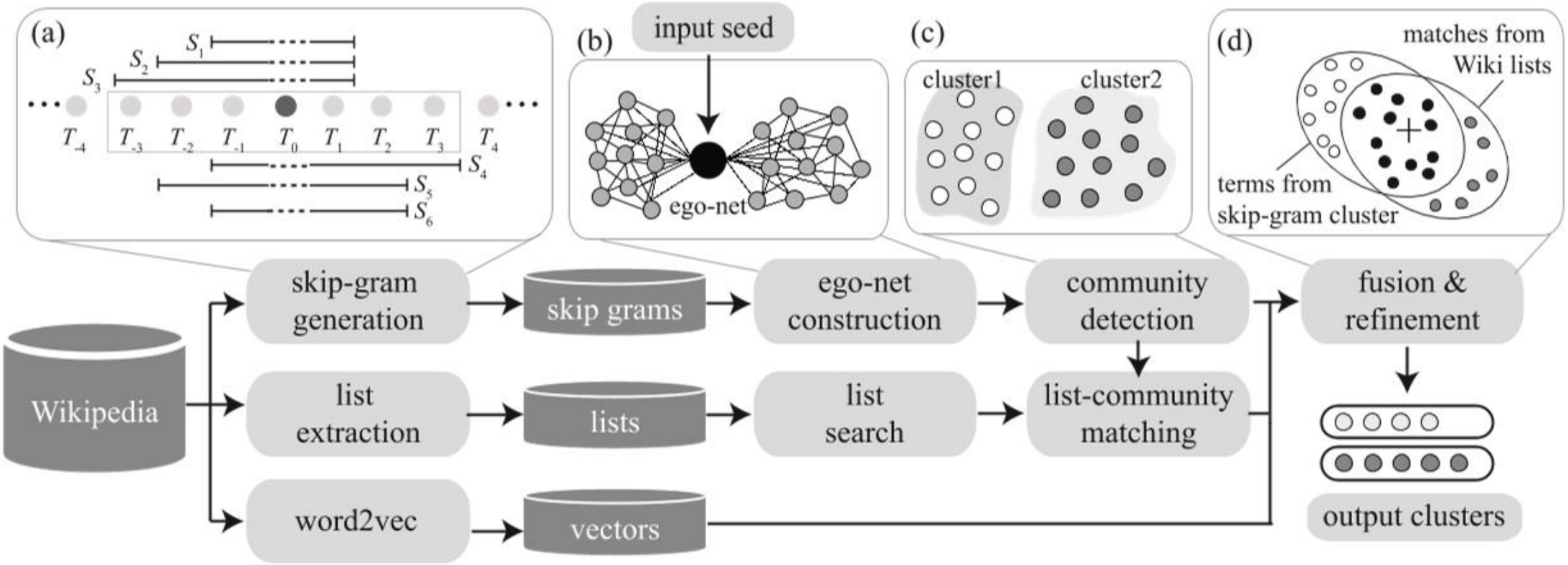
Olympic Host Cities

City	Country
Albertville	France
Amsterdam	Netherlands
Antwerp[g]	Belgium
Athens	Greece
Atlanta	United States
Barcelona	Spain
Beijing	China
Berlin	Germany

word ego-network

- Ontologies are a natural source for set expansion
- **List-of pages of Wikipedia** were found to have the right combination of being prevalent and relatively "clean"

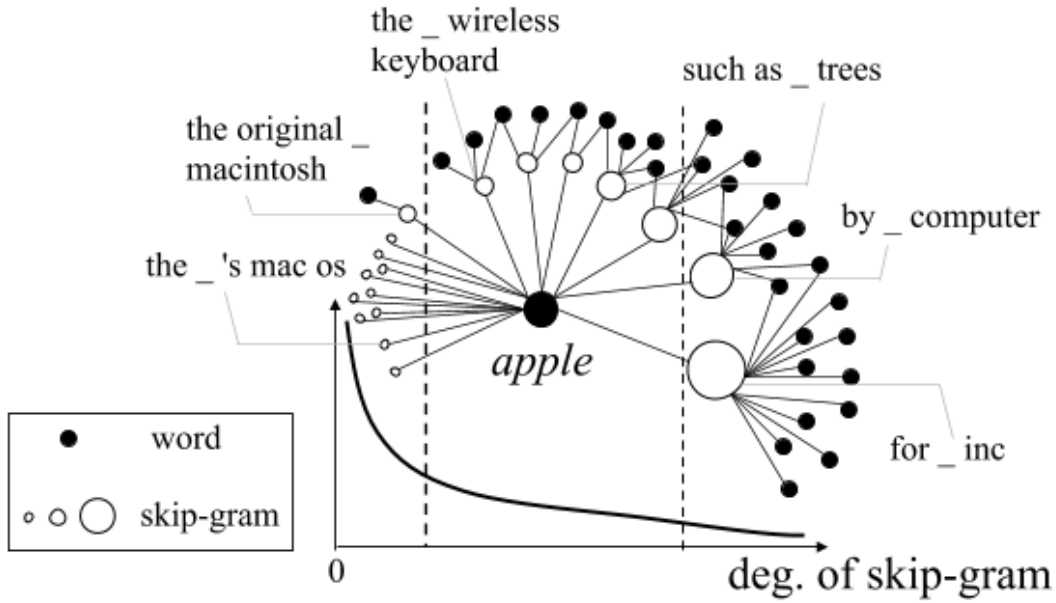
EgoSet: System Pipeline



- Fuse ego-communities with user-created ontologies, as well as word embeddings, through an ensemble model
- Provide accurate entity set expansion for multifaceted seeds that outperforms state-of-the-art baselines

Feature Extraction

- **Skip-grams** instead of unigrams or n-grams were applied because skipgrams impose **stronger positional constraints** on where contextual words may appear with regard to the target term
- In practice, most sibling relations are effectively recovered by skip-grams with 5–50 neighbors



- After this filtering process, assign **weight** for each skip-gram s and word w using:

$$f_{w,s} = \log(1 + X_{w,s})[\log|W| - \log(\sum_{w} X_{w,s})]$$

co-occurrence count between w and s

total number of words in the vocabulary

- To further improve computational efficiency, the dimensionality of the feature space was reduced by *sampling 300 skip-grams per word* using weighted minhash

Ego-Network Construction and Ego-Community Detection

- Ego-Network Construction
  - Each word has **a skip-gram vector.** For each word w, find **250 nearest neighbors**, which will become the nodes of the ego-network
  - For each pair of terms, compute a similarity score using **weighted Jaccard similarity**

$$J(w_1, w_2) = \frac{\sum_k \min(f_{w_1,s_k}, f_{w_2,s_k})}{\sum_k \max(f_{w_1,k}, f_{w_2,s_k})}$$

  - If any pair of words, wi and wj , have a similarity score higher than a threshold (e.g., 0.05), then **a link is created** between them
- Ego-Community Detection
  - A hierarchical clustering algorithm, **Louvain,** was used for community detection
  - Start by assigning a different community to each node, followed by greedily aggregating communities to optimize the modularity of the network partition until the modularity cannot be further improved, and the “optimal” number of communities is thus identified

Fusing Ego-communities and Ontologies

- For each cluster in the ego-network, **find Wikipedia lists that "match" the content of the cluster**
- Assuming there are N candidates, then for each candidate w_0, the **ensemble model** takes into account the mean distance from w_0 to all other candidates:

$$\bar{d}_{w_0} = \frac{1}{N} \sum_{w_i} f(w_0, w_i)$$

where f is one of three different **distance/similarity metrics**:

  - Hamming distance of Wikipedia list memberships;
  - Weighted Jaccard similarity of the skip-gram vectors;
  - Cosine similarity of the word embedding vectors learned by word2vec on the same corpus

- Take the "**majority vote**" of the three metrics to finally decide whether to remove w_0
- **Post-processing** to remove duplicate clusters, off-topic clusters, and too small clusters

Experiment Result and Case Study

- Tested with 150 multifaceted seeds
  - Outperforms state-of-the-art baselines:
    - SEAL
    - NeedleSeek
    - Word2vec

		1 seed			2 seeds		
		p@5	p@10	p@20	p@5	p@10	p@20
baseline	SEAL	-	-	-	0.208	0.169	0.138
	NeedleSeek	0.432	0.372	0.325	-	-	-
single	WikiList	0.369	0.331	0.292	0.313	0.295	0.250
	word2vec	0.360	0.296	0.249	0.317	0.271	0.219
fusion	EgoSet-SG & WikiList	0.465	0.413	0.358	0.357	0.316	0.272
	word2vec & WikiList	0.390	0.331	0.289	0.334	0.313	0.222
	EgoSet-ALL	**0.490**	**0.427**	**0.372**	**0.369**	**0.323**	**0.274**

- Case Study

Multi-seed Query	Identified Ego-clusters	Top Skip-gram
brown white	Cluster 1: brown white green williams roberts johnson jackson smith evans jones ... (popular English last names)	kenneth ___ ,
	Cluster 2: white red blue black yellow brown green purple light_blue ... (colors)	colors are ___ and
beijing shanghai	Cluster 1: tianjin guangzhou shanghai shenzhen chongqing beijing wuhan chengdu dalian shenyang ... (major Chinese cities)	in ___ , china .
	Cluster 2: liaoning heilongjiang hebei tianjin shanghai jilin inner_mongolia shanxi beijing hunan ... (Chinese province-level administrative regions)	of ___ , china .
beaver elk	Cluster 1: coyote moose elk cougar beaver bison opossum marten wolverine fisher ... (animals)	deer , ___ ,
	Cluster 2: westmoreland schuylkill fayette crawford beaver elk greene fulton chester ... (counties in Pennsylvania)	in ___ county , pennsylvania


bbk
bbk

Outline

- **Motivation:** Why Mining Information Networks?
- **Part I:** Clustering and Ranking in Heterogeneous Information Networks
- **Part II:** Classification and Prediction in Heterogeneous Information Networks
- **Part III:** Other Data Mining Functions for Heterogeneous Information Networks
  - Mining Evolution and Dynamics of Information Networks
  - Role Discovery
  - OLAP in Information Networks
  - Data Cleaning: Distinct
  - CrossMine: Classification Across Multi-Relational Databases
  - EgoSet: Exploiting Word Ego-Networks for Multifaceted Set Expansion
- **Summary** 

Summary

- **Heterogeneous information networks are ubiquitous**
  - Most datasets can be "organized" or "transformed" into "*structured*" multi-typed heterogeneous info. networks
  - Examples: DBLP, IMDB, Flickr, Google News, Wikipedia, …
- **Rich knowledge can be mined from structured heterogeneous info. networks**
  - Clustering, ranking, classification, path prediction, ……
- **Knowledge is power, but knowledge is hidden in massive, but "relatively structured" nodes and links!**
- **Key issue: Construction of trusted, semi-structured heterogeneous networks from unstructured data**
- **From data to knowledge: Much more to be explored but heterogeneous network mining has shown high promise!**

Future Research

- Discovering ontology and structures in information networks
- Discovering and mining hidden information networks
- Mining information networks formed by structured data linking with unstructured data (text, multimedia and Web)
- Mining cyber-physical networks (networks formed by dynamic sensors, image/video cameras, with information networks)
- Enhancing the power of knowledge discovery by transforming massive unstructured data: Incremental information extraction, role discovery, … ⇒ multi-dimensional structured info-net
- Mining noisy, uncertain, un-trustable massive datasets by information network analysis approach
- Turning Wikipedia and/or Web into structured or semi-structured databases by heterogeneous information network analysis

References: Research Papers Discussed in Part II

- Yizhou Sun, Jie Tang, Jiawei Han, Cheng Chen, and Manish Gupta, “Co-Evolution of Multi-Typed Objects in Dynamic Heterogeneous Information Networks”, IEEE Transactions on Knowledge and Data Engineering (TKDE), 26(12): 2942-2955, 2014
- Chi Wang, Jiawei Han, Yuntao Jia, Jie Tang, Duo Zhang, Yintao Yu, and Jingyi Guo, “Mining Advisor-Advisee Relationships from Research Publication Networks ", KDD'10
- Peixiang Zhao, Xiaolei Li, Dong Xin, and Jiawei Han, “Graph Cube: On Warehousing and OLAP Multidimensional Networks”, SIGMOD'11
- Peixiang Zhao and Jiawei Han, “On Graph Query Optimization in Large Networks", Proc. 2010 Int. Conf. on Very Large Data Bases (VLDB'10), Singapore, Sept. 2010
- Chen Chen, Xifeng Yan, Feida Zhu, Jiawei Han, and Philip S. Yu, "Graph OLAP: Towards Online Analytical Processing on Graphs", ICDM 2008
- Xiaoxin Yin, Jiawei Han, and Philip S. Yu, “Object Distinction: Distinguishing Objects with Identical Names by Link Analysis”, ICDE'07.
- Xiaoxin Yin, Jiawei Han, Jiong Yang and Philip S. Yu, "Efficient Classification across Multiple Database Relations:A CrossMine Approach", IEEE Trans. Knowledge and Data Engineering, 18(6): 770-783, 2006.
- Xin Rong, Zhe Chen, Qiaozhu Mei, Eytan Adar, “EgoSet: Exploiting Word Ego-Networks and User-Generated Ontology for Multifaceted Set Expansion”, WSDM’16

Inductive Logic Programming Approach to Multi-Relation Classification

- ILP Approached to Classification
  - Top-down Approaches (e.g., FOIL)

while(enough examples left)
generate a rule
remove examples satisfying this rule

  - Bottom-up Approaches (e.g., Golem)

Use each example as a rule
Generalize rules by merging rules

  - Decision Tree Approaches (e.g., TILDE)
- ILP Approach: Pros and Cons
  - Advantages: Expressive and powerful, and rules are understandable
  - Disadvantages: Inefficient for databases with complex schemas, and inappropriate for continuous attributes