Scalable Link Mining and Analysis on Information Networks

Philip S. Yu: Univ. of Illinois at Chicago

Join work with

Jiawei Han, Chen Chen, Feida Zhu: Univ. of Illinois at Urbana-Champaign

Xifeng Yan: Univ. of California at Santa Barbara

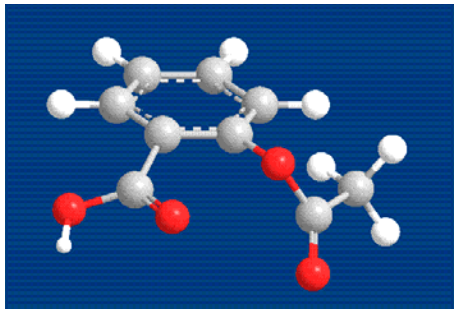
Xiaoxin Yin: Microsoft Research

Information Networks

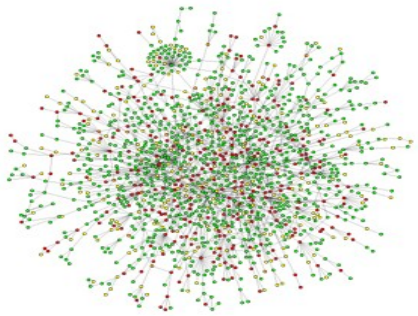
- Information network: A network where each node represents an entity (e.g., actor in a social network) and each link (e.g., tie) a relationship between entities
  - Each node/link may have attributes, labels, and weights
  - Link may carry rich semantic information
- Homogeneous vs. heterogeneous networks
  - Homogeneous networks
    - Single object type and single link type
    - Single social network (e.g., friends)
    - WWW: a collection of linked Web pages
  - Heterogeneous networks
    - Multiple object and link types
    - Medical network: patients, doctors, disease, contacts, treatments
    - Bibliographic network: publications, authors, venues

Ubiquitous Graphs and Networks

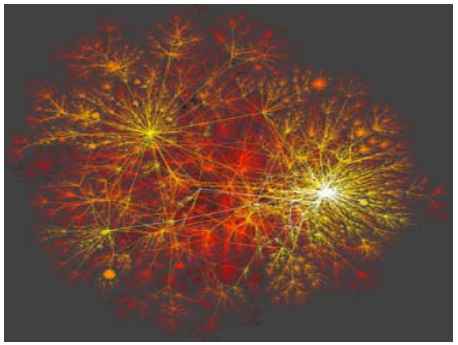
- Graphs and substructures
  - Chemical compounds, computer vision objects, circuits, XML
- Biological networks
- Bibliographic networks: DBLP, ArXiv, PubMed, ...
- Social networks: Facebook >100 million active users
- World Wide Web (WWW): > 3 billion nodes, > 50 billion arcs
- Cyber-physical networks



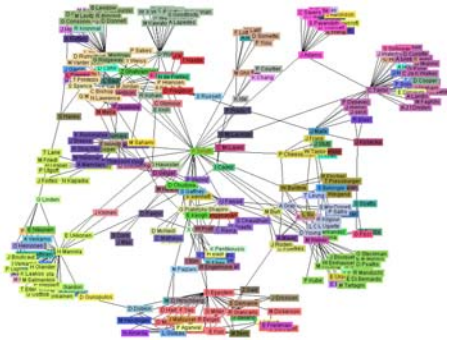
Aspirin



Yeast protein interaction network



An Internet Web



Co-author network

Talk Outline

- Introduction to Information Networks
- Data Integration, Cleaning and Validation in Information Networks
- Online Analytical Processing of Information Networks
- Mining Information Networks
- Summary

Data Integration, Cleaning and Validation in Information Networks

- Data integration in information networks
  - Object reconciliation by link analysis
  - Distinct: Distinguishing objects with identical names via link analysis
- Data cleaning and data validation (veracity analysis) in information networks
  - TruthFinder: Discovery of truth with conflicting information

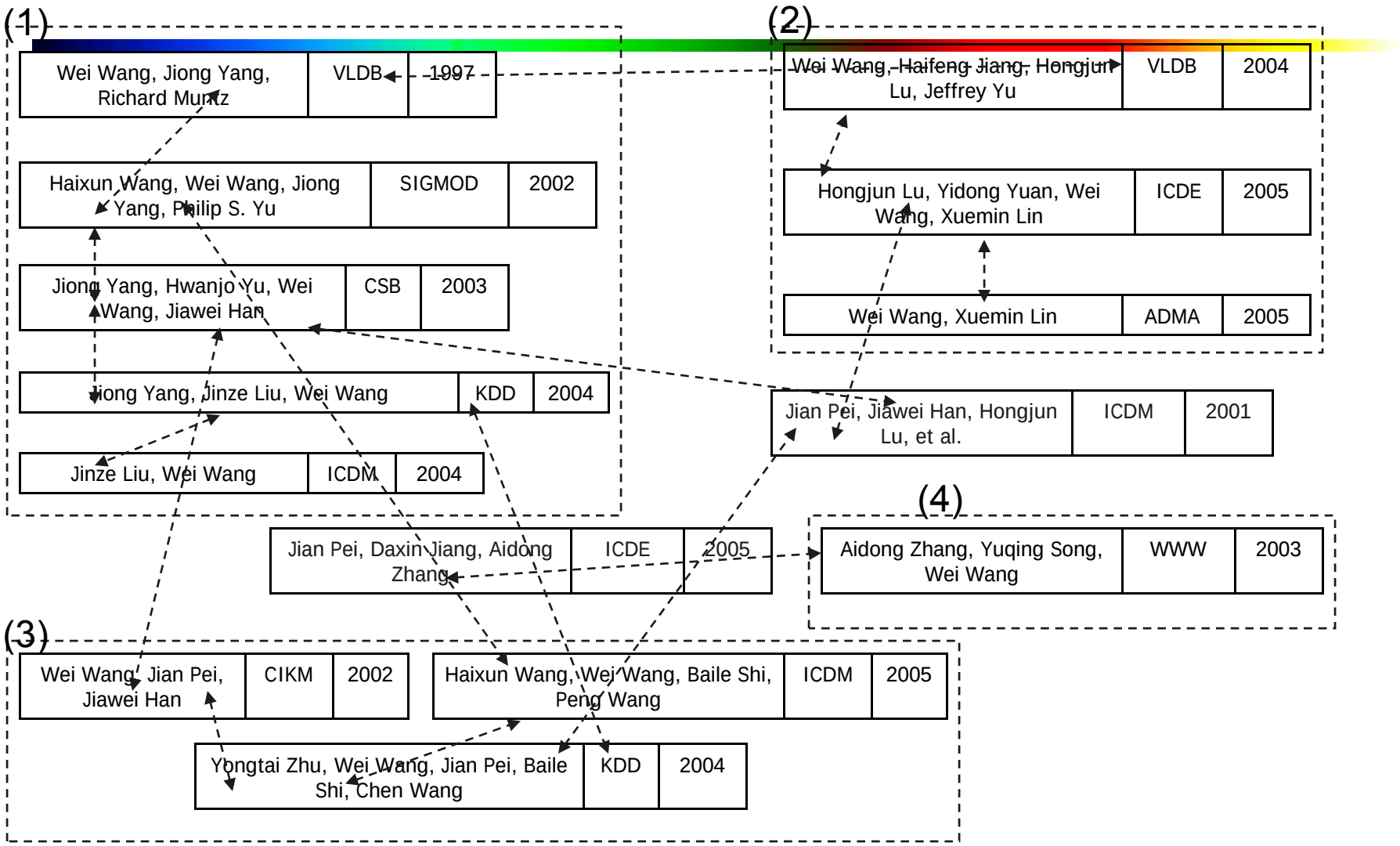
Object Reconciliation by Link Analysis

- Link makes entity cross-checking and validation easy
- Object reconciliation vs. object distinction
- Object distinction: 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”
- 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

Challenges of Object Distinction

- Related to duplicate detection, but
  - Textual similarity cannot be used
  - Different references appear in different contexts (e.g., different papers), and thus seldom share common attributes
  - Each reference is associated with limited information
- The “Wei Wang” challenge: Different Wei Wangs coauthor with the same authors and publish in the same venues!
  - Explore the links and use all the information we have

Entity Distinction: The "Wei Wang" Challenge in DBLP

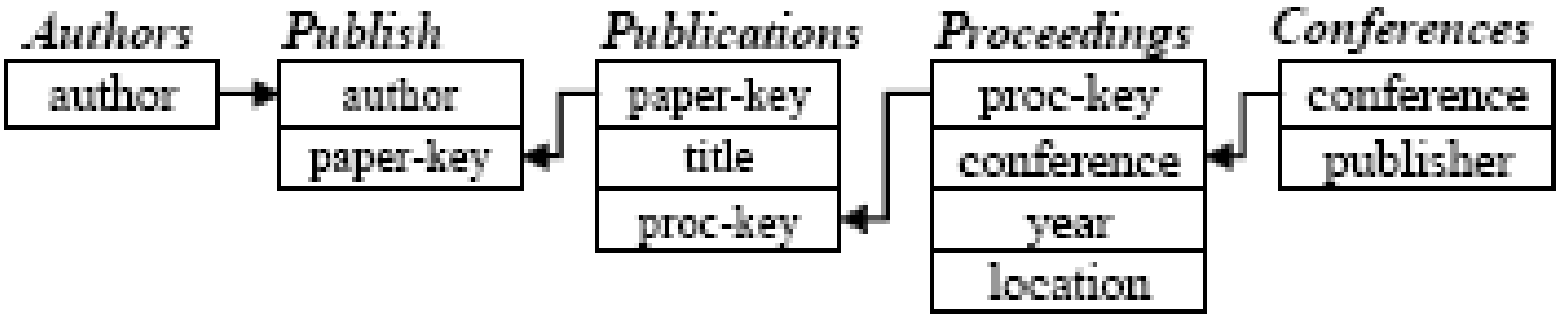


(1) Wei Wang at UNC
(2) Wei Wang at UNSW, Australia
(3) Wei Wang at Fudan Univ., China
(4) Wei Wang at SUNY Buffalo

The DISTINCT Methodology

- Measure similarity between references
  - Link-based similarity: Linkages between references
    - References to the same object are more likely to be connected
  - 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

Neighbor Tuples

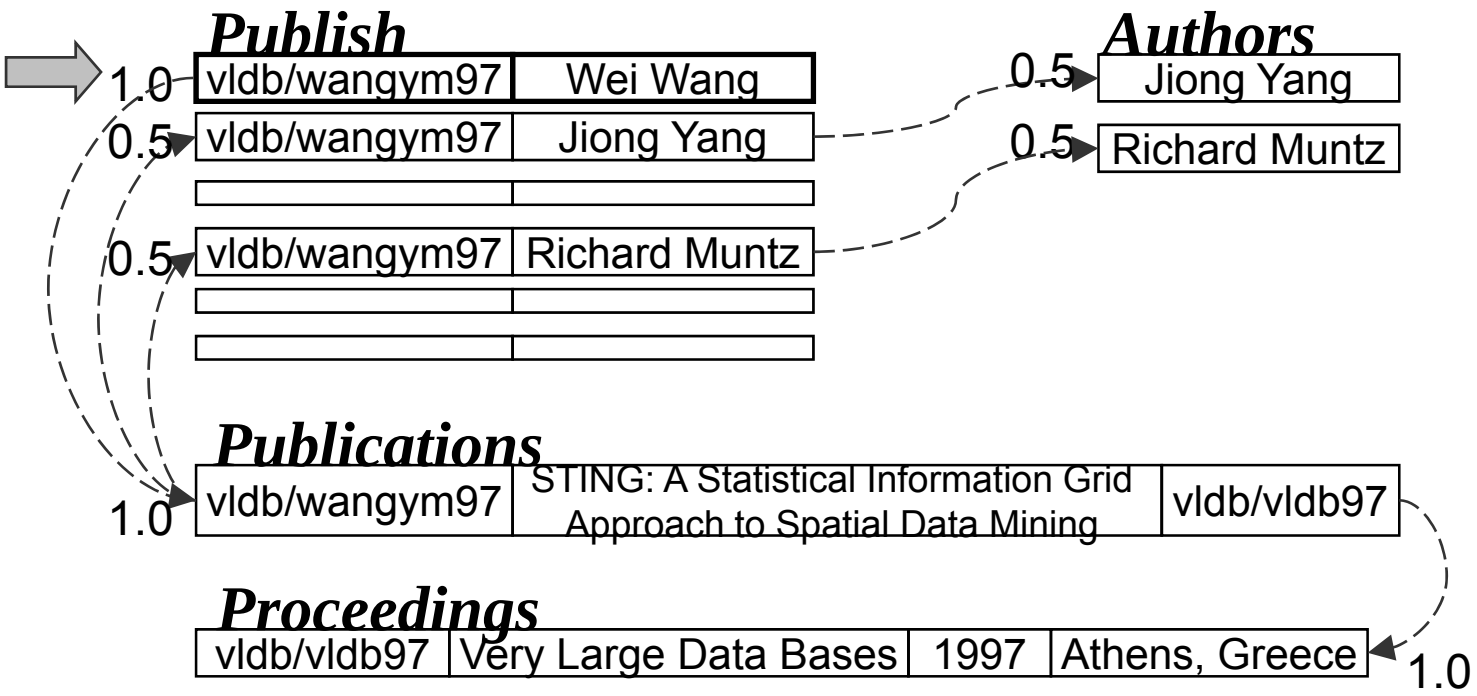


- The neighbor tuples of a reference are the tuples joinable with it.
- A reference has a set of neighbor tuples along each join path
  - Starting with the relation containing the references
- Example: Publish ⋈ Publications ⋈ Publish ⋈ Authors

Similarity 1: Link-Based Similarity

- Indicate the overall strength of connections between two references
- Use *random walk probability* between the two tuples containing the references
- Random walk probabilities along *different join paths* are handled separately
  - Because different join paths have different semantic meanings
  - Only consider join paths of length at most $2L$ (L is the number of steps of propagating probabilities)

Example of Random Walk



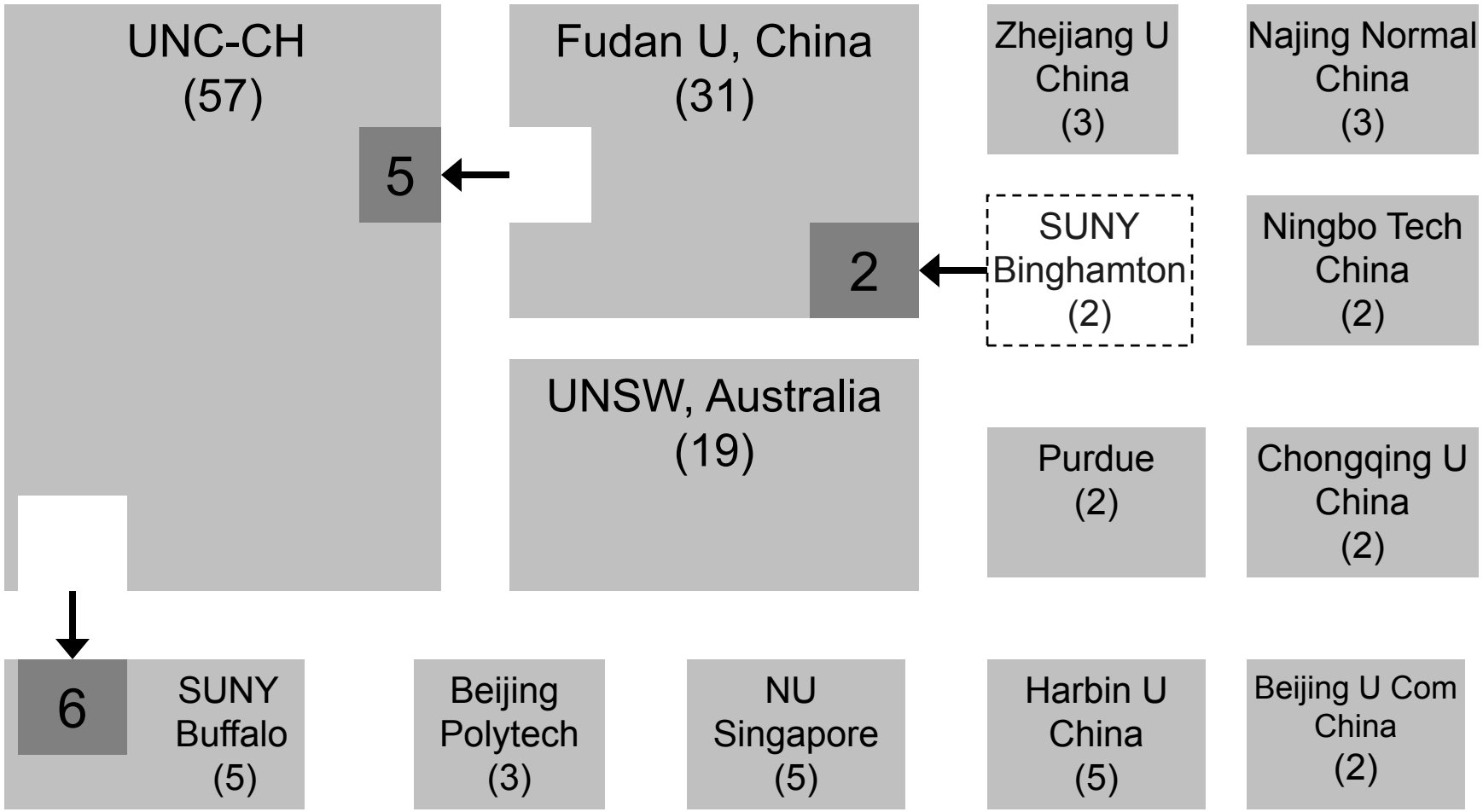
Similarity 2: Neighborhood Similarity

- Find the neighbor tuples of each reference
  - Neighbor tuples within L joins
- Weights of neighbor tuples
  - Different neighbor tuples have different connections to a reference
  - Assign each neighbor tuple a *weight*, which is the probability of walking from the reference to this tuple
- Similarity: Set resemblance between two sets of neighbor tuples

Real Cases

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



Truth Validation by Information Network Analysis

- Xiaoxin Yin, Jiawei Han, Philip S. Yu, “Truth Discovery with Multiple Conflicting Information Providers on the Web”, KDD’07
- The trustworthiness problem of the web (according to a survey):
  - 54% of Internet users trust news web sites most of time
  - 26% for web sites that sell products
  - 12% for blogs
- TruthFinder: Truth discovery on the Web by link analysis
  - Among multiple conflict results, can we automatically identify which one is likely the true fact?
- Veracity (conformity to truth):
  - Given a large amount of conflicting information about many objects, provided by multiple web sites (or other information providers), how to discover the true fact about each object?

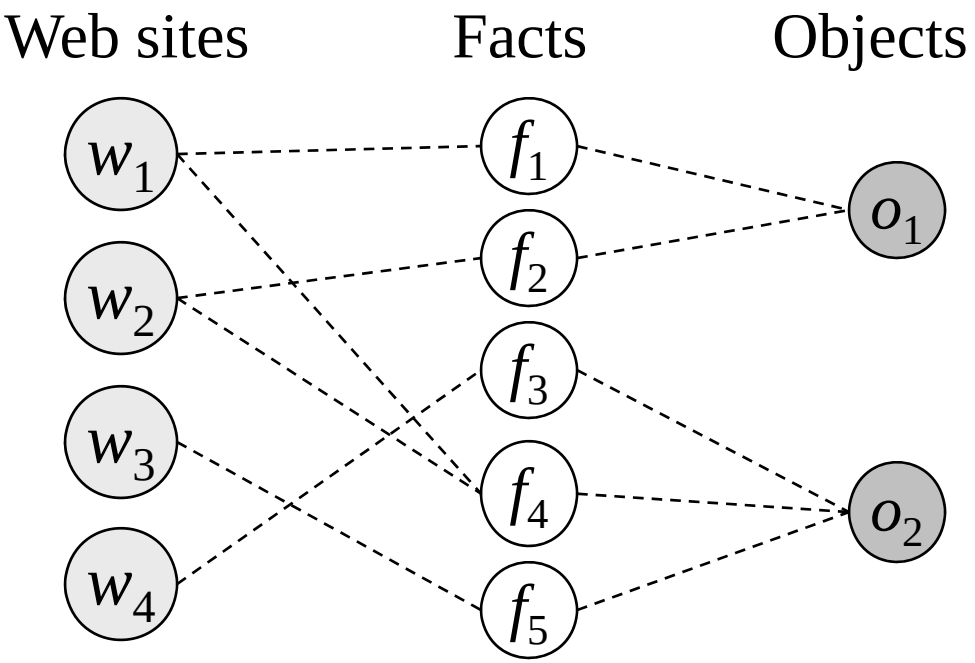
Conflicting Information on the Web

- Different websites often provide conflicting info. on a subject, e.g., Authors of *"Rapid Contextual Design"*

Online Store	***Authors***
Powell's books	Holtzblatt, Karen
Barnes & Noble	Karen Holtzblatt, Jessamyn Wendell, Shelley Wood
A1 Books	Karen Holtzblatt, Jessamyn Burns Wendell, Shelley Wood
Cornwall books	Holtzblatt-Karen, Wendell-Jessamyn Burns, Wood
Mellon's books	Wendell, Jessamyn
Lakeside books	WENDELL, JESSAMYNHOLTZBLATT, KARENWOOD, SHELLEY
Blackwell online	Wendell, Jessamyn, Holtzblatt, Karen, Wood, Shelley

Our Problem Setting

- Each object has a set of ***conflictive*** facts
  - E.g., different author names for a book
- And each web site provides some facts
- How to find the true fact for each object?



Basic Heuristics for Problem Solving

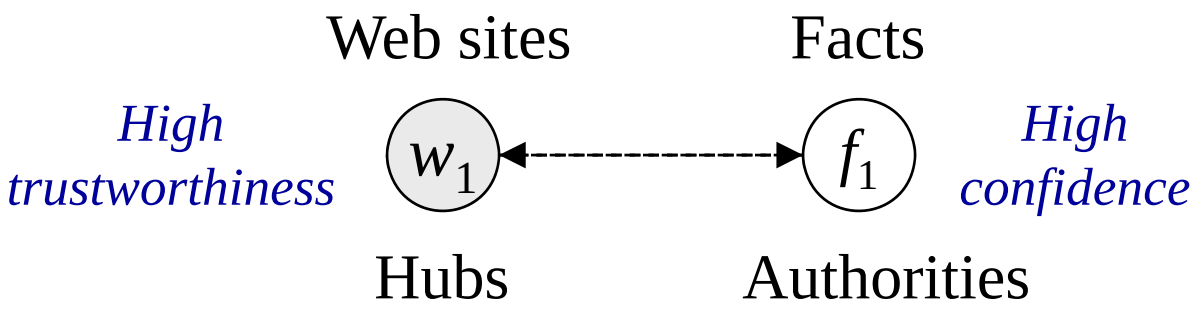
1. There is usually only one true fact for a property of an object
2. This true fact appears to be the same or similar on different web sites
   - E.g., “Jennifer Widom” vs. “J. Widom”
3. **The false facts on different web sites are less likely to be the same or similar**
   - False facts are often introduced by random factors
4. **A web site that provides mostly true facts for many objects will likely provide true facts for other objects**

Overview of the TruthFinder Method

- *Confidence of facts* ↔ *Trustworthiness of web sites*
  - A fact has *high confidence* if it is provided by (many) trustworthy web sites
  - A web site is *trustworthy* if it provides many facts with high confidence
- The TruthFinder mechanism, an overview:
  - Initially, each web site is equally trustworthy
  - Based on the above four heuristics, infer fact confidence from web site trustworthiness, and then backwards
  - Repeat until achieving stable state

Analogy to Authority-Hub Analysis

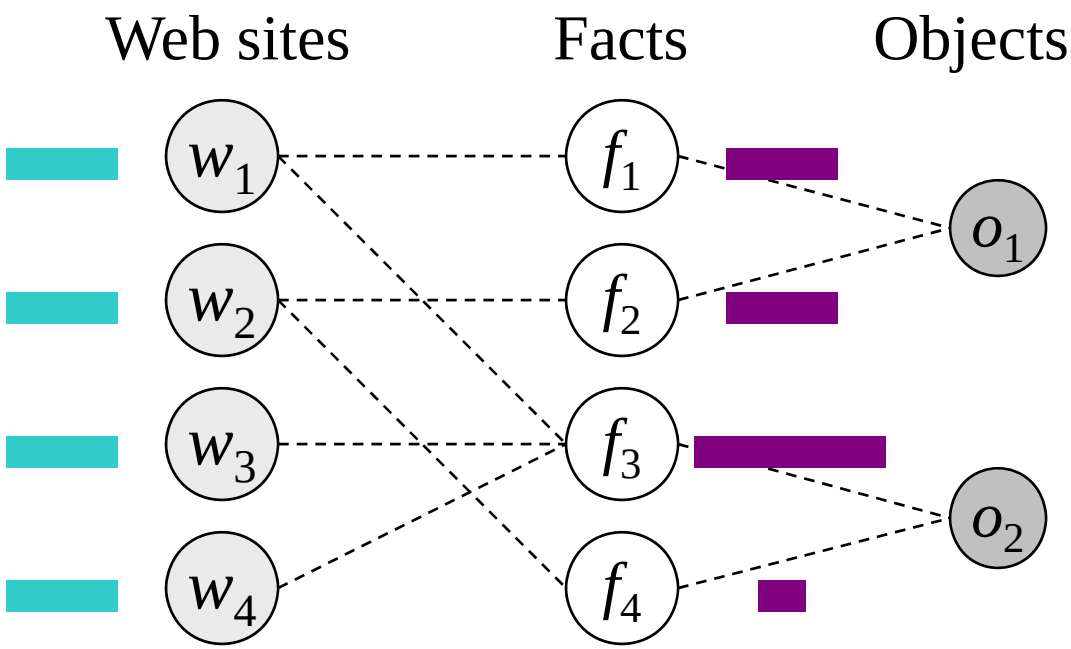
- Facts ↔ Authorities, Web sites ↔ Hubs



- Difference from authority-hub analysis
  - Linear summation cannot be used
    - A web site is trustable if it provides accurate facts, instead of many facts
    - Confidence is the probability of being true
  - Different facts of the same object influence each other

Inference on Trustworthness

- Inference of web site trustworthiness & fact confidence



True facts and trustable web sites will become apparent after some iterations

Computation Model: t(w) and s(f)

- **The trustworthiness of a web site *w*: *t*(*w*)**
  - Average confidence of facts it provides

$$t(w) = \frac{\sum_{f \in F(w)} s(f)}{|F(w)|}$$

Sum of fact confidence

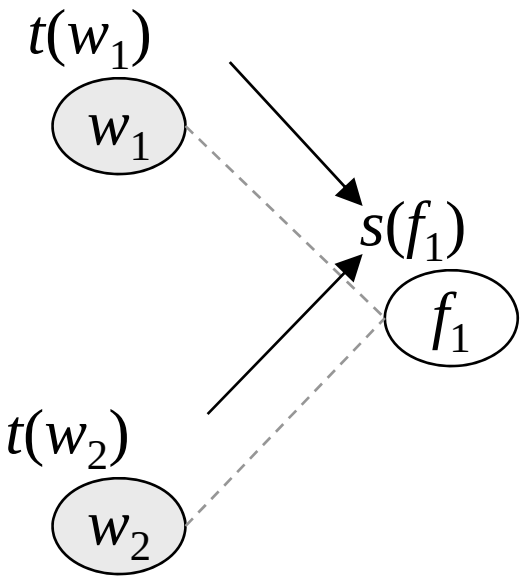
Set of facts provided by w

- **The confidence of a fact *f*: *s*(*f*)**
  - One minus the probability that all web sites providing *f* are wrong

$$s(f) = 1 - \prod_{w \in W(f)} (1 - t(w))$$

Probability that w is wrong

Set of websites providing f



Experiments: Finding Truth of Facts

- Determining authors of books
  - Dataset contains 1265 books listed on abebooks.com
  - We analyze 100 random books (using book images)

Case	*Voting*	*TruthFinder*	*Barnes & Noble*
Correct	71	85	64
Miss author(s)	12	2	4
Incomplete names	18	5	6
Wrong first/middle names	1	1	3
Has redundant names	0	2	23
Add incorrect names	1	5	5
No information	0	0	2

Experiments: Trustable Info Providers

- Finding trustworthy information sources
  - Most trustworthy bookstores found by TruthFinder vs. Top ranked bookstores by Google (query “bookstore”)

TruthFinder

Bookstore	*trustworthiness*	*#book*	*Accuracy*
TheSaintBookstore	0.971	28	0.959
MildredsBooks	0.969	10	1.0
Alphacraze.com	0.968	13	0.947

Google

Bookstore	*Google rank*	*#book*	*Accuracy*
Barnes & Noble	1	97	0.865
Powell’s books	3	42	0.654

Summary: Data Integration, Cleaning & Truth Validation by Infonet Analysis

- Infonet contains rich semantic and statistical information that can be explored for data integration, cleaning and veracity analysis
- Distinct is an interesting example of using links and self-training to distinguish object with identical names and subtle links
- TruthFinder is an interesting example to use massive and sophisticate links to infer *trustable web sites* and *true facts*
- Future work: Put the framework into broader application scope
  - E.g., truth validation by assuming multiple versions of truth and truth evolving with time

Talk Outline

- Introduction to Information Networks
- Data Integration, Cleaning and Validation in Information Networks
- Online Analytical Processing of Information Networks
- Mining Information Networks
- Summary

Network Summary and Compression

- A network can be compressed or abstracted in different ways
  - Graph partition (min-cut, mesh transformation)
  - Selection view: KDD-pub net, 2009-pub net, ...
  - Subject-projection view: Only check some properties, e.g., evolution of co-author relationships, ...
- Need efficient summarization/compression methods
  - Do summarization/compression efficiently and dynamically (online)
  - How to pre-compute networks in multi-granularity?
  - Can we compute various types of cubes?

Why OLAP Info. Networks?

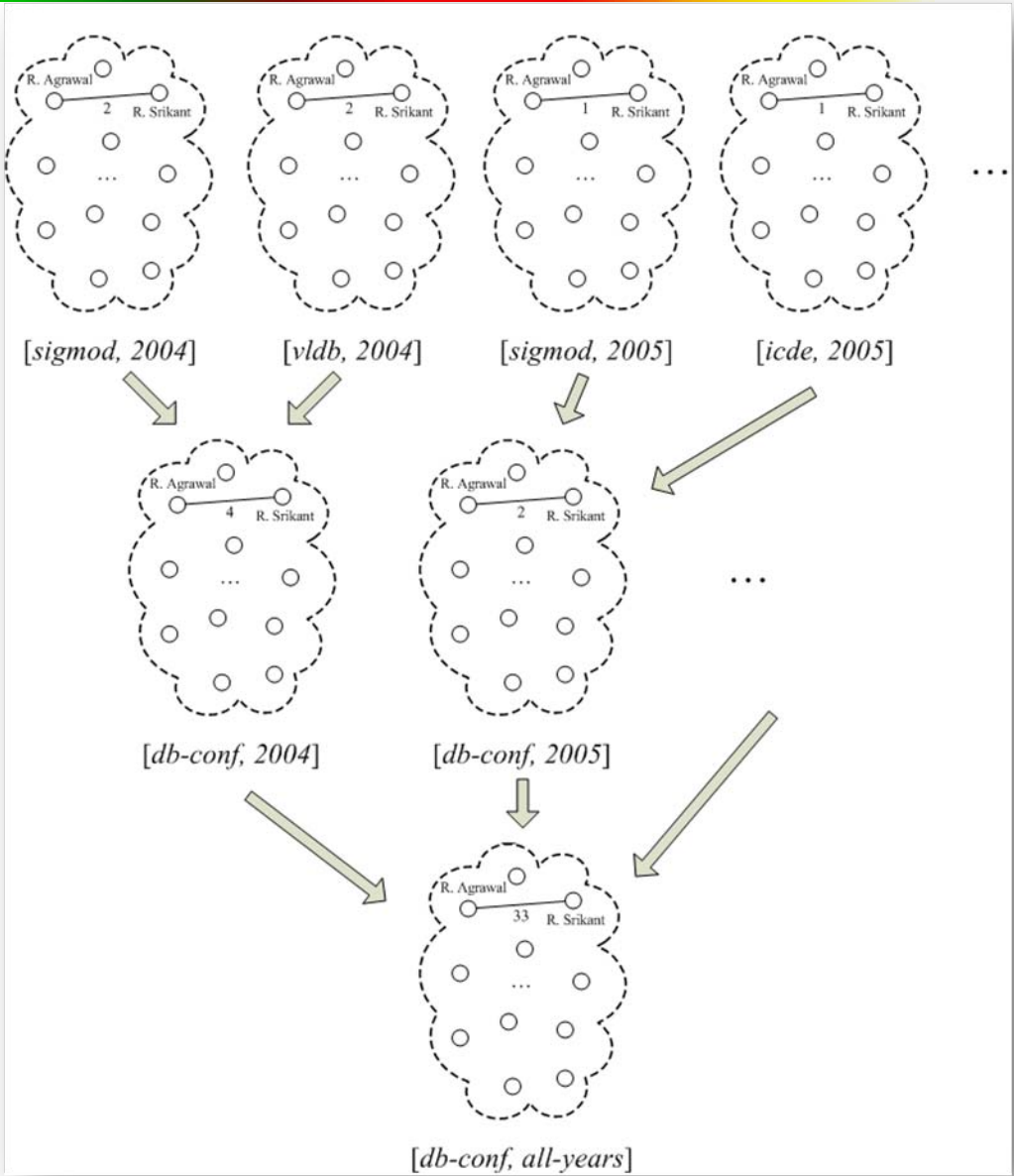
- Both homogeneous and heterogeneous info. networks are multi-dimensional in nature
  - Nodes and links contain many properties
  - E.g., DBLP paper: time, venue, author, subject
- It is highly desirable to view a large network from different angles and multiple levels of granularity
  - DBLP: author net, conf. net, field/subject net, …
- It is desirable to view network from rough to fine, drill and dice networks dynamically
- Knowledge discovery should be performed flexibly in each dimension/level combination space

Two Kinds of OLAP in Information Networks

- Two Types of OLAP
  - Informational OLAP (abbr. I-OLAP)
  - Topological OLAP (abbr. T-OLAP)
- Discovery-driven OLAP of information networks based on
  - heuristic rules, statistical analysis, pattern mining
- Discovery of network structures for InfoNet-OLAP
  - Coauthor network: advisor-advisee relationship
  - Ranking: partition network into clusters and rank nodes in each cluster (RankClus)

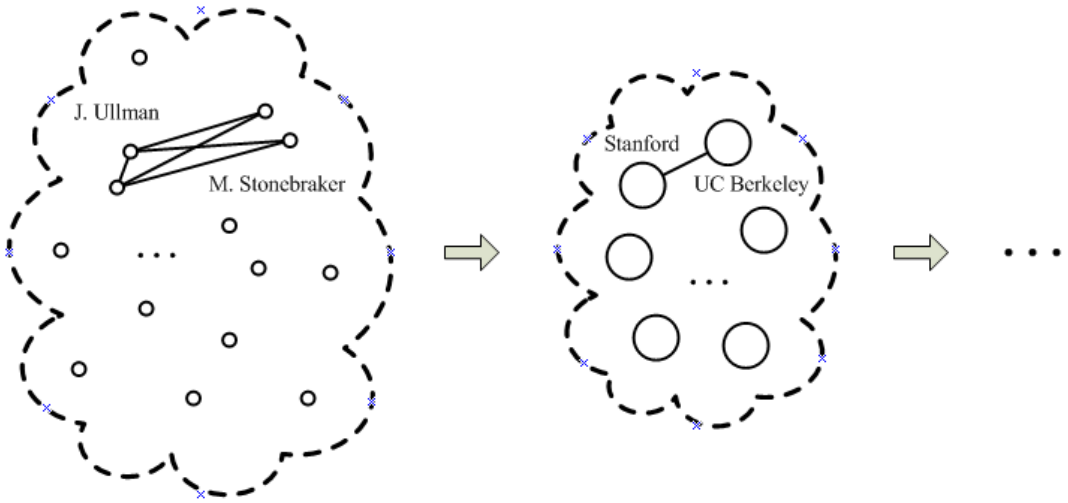
Informational OLAP

- Dimensions come from informational attributes *(venue, time)* attached at the whole snapshot level, so-called *Info-Dims*
- Overlay multiple pieces of information
- Do not change the objects whose interactions are being looked at
  - 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*
- 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



Measures in Infonet OLAP

- Measure is an aggregated infonet
  - I-aggregated infonet
  - T-aggregated infonet
  - Other measures like node count, average degree, etc. can be treated as derived
- Infonet plays a dual role
  - Data source
  - Aggregate measure
- Measures could be complex
  - e.g., maximum flow, shortest path, centrality
- Combine I-OLAP and T-OLAP into a hybrid case

Network OLAP Operations

	Network I-OLAP	Network T-OLAP
Roll-up	Overlay multiple snapshots to form a higher-level summary via I-aggregated infonet	Shrink the topology and obtain a T-aggregated infonet that represents a compressed view, whose topological elements (i.e., nodes and/or edges) have been merged and replaced by corresponding higher-level ones
Drill-down	Return to the set of lower-level snapshots from the higher-level overlaid (aggregated) infonet	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 Classification

- How to combine and leverage intermediate results?
  - Distributive
    - The computation of high-level cells can be directly built on low-level cells
    - E.g., *collaboration frequency*
  - Algebraic
    - Not distributive, but can be easily derived from several distributive measures
    - E.g., *maximum flow*
  - Holistic
    - Neither distributive nor algebraic. Need to go down to the raw data and start from scratch
    - E.g., *centrality*

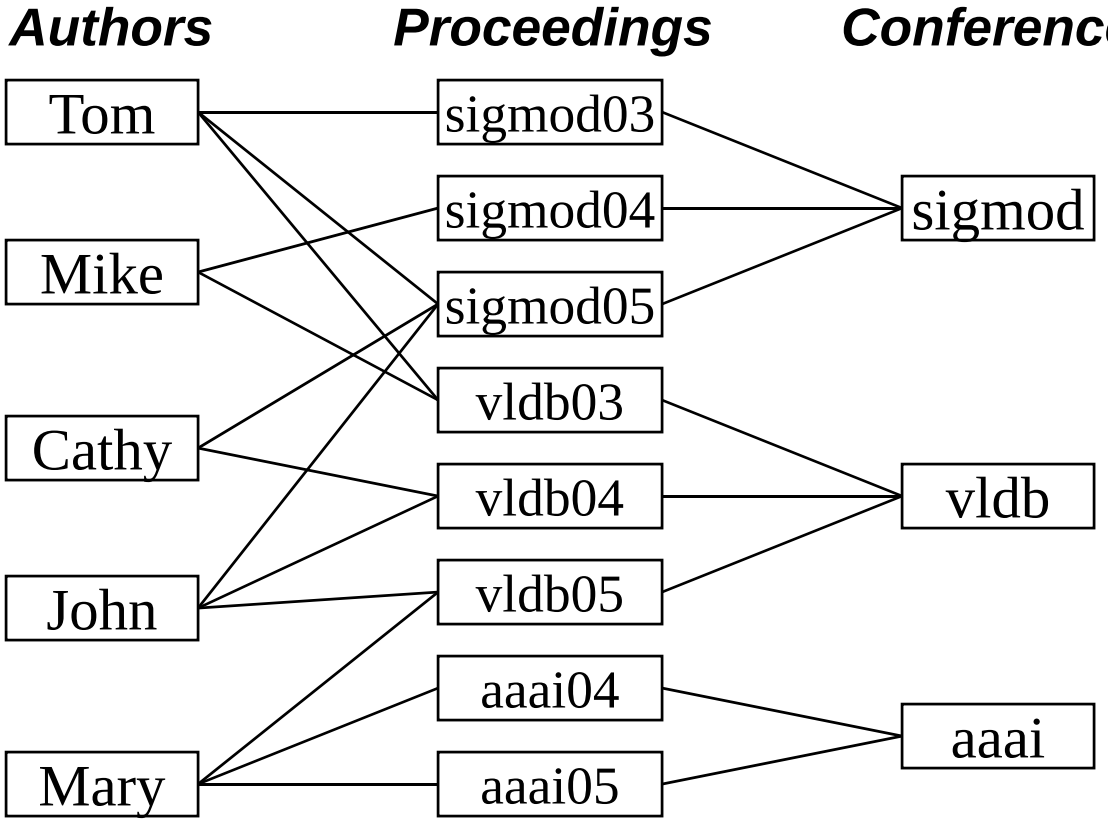
Optimizations

- Special measures may have special properties that can help optimize the calculations
- Localization
  - During computation, only a neighborhood of the networks needs to be consulted
    - e.g., the collaboration frequency of “R. Agrawal” and “R.Srikant” for [*sigmod, all-years*] only depends on their collaboration frequencies in each SIGMOD conferences
    - Perfect (i.e., 0-neighborhood) localization
    - k-neighborhood is less ideal, but still useful
      - e.g., # of common friends shared by “R. Agrawal” and “R.Srikant”

Talk Outline

- Introduction to Information Networks
- Data Integration, Cleaning and Validation in Information Networks
- Online Analytical Processing of Information Networks
- Mining Information Networks
- Summary

Link-Based Clustering: Start from Link-Based Similarity (SimRank)

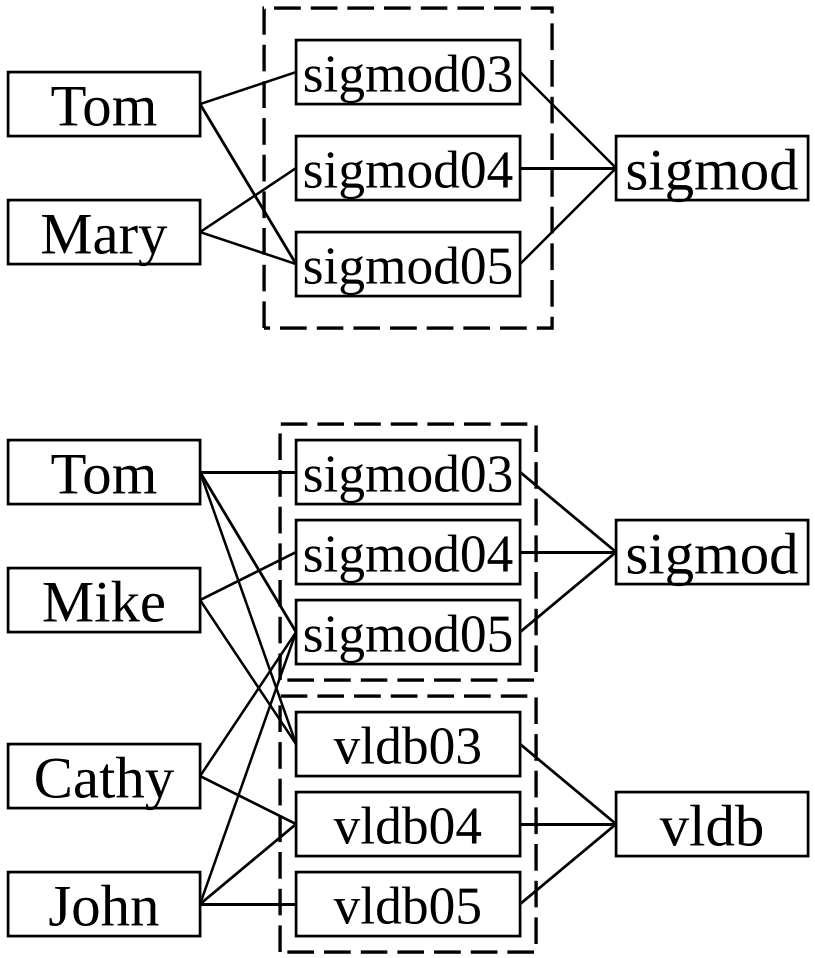


- "Two queries are similar if they are connected to similar ads"
- "Two ads are similar if they are connected to similar queries"

- Iterative procedure: at each iteration similarity propagates in the graph

Link-Based Similarities: SimRank

- Two objects are similar if they are linked with the same or similar objects



Jeh & Widom, KDD'2002 - *SimRank*

The similarity between two objects x and y is defined as the average similarity between objects linked with x and those with y:

$$\mathrm{sim}(a,b) = \frac{C}{|I(a)||I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} \mathrm{sim}(I_i(a), I_j(b))$$

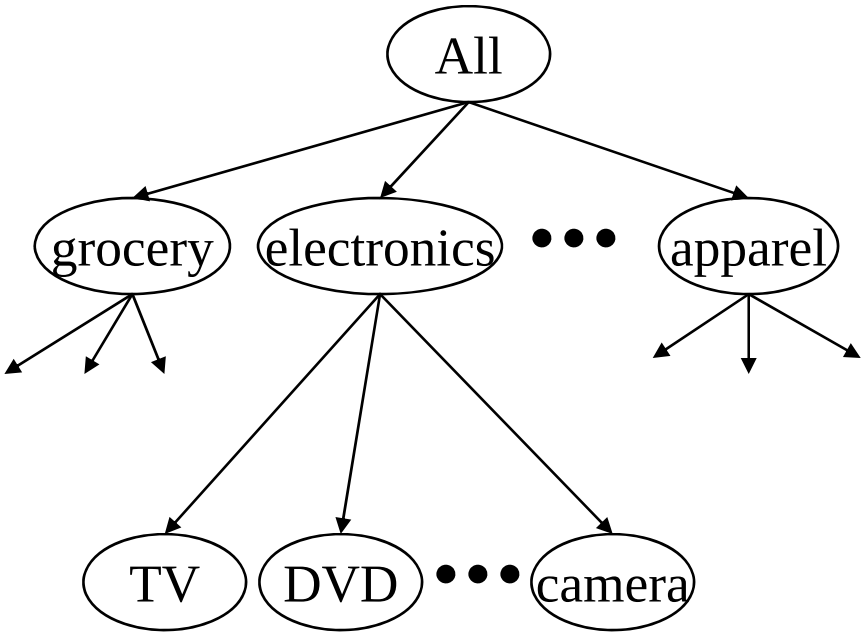
Expensive to compute:

For a dataset of N objects and M links, it takes $O(N^2)$ space and $O(M^2)$ time to compute all similarities.

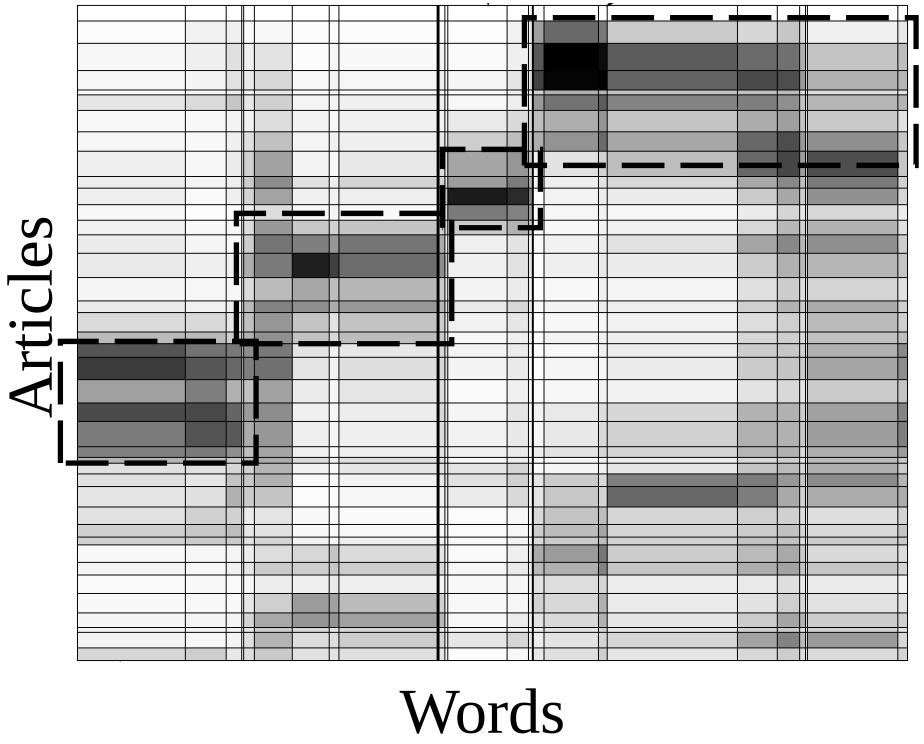
Observation 1: Hierarchical Structures

- Hierarchical structures often exist naturally among objects (e.g., taxonomy of animals)

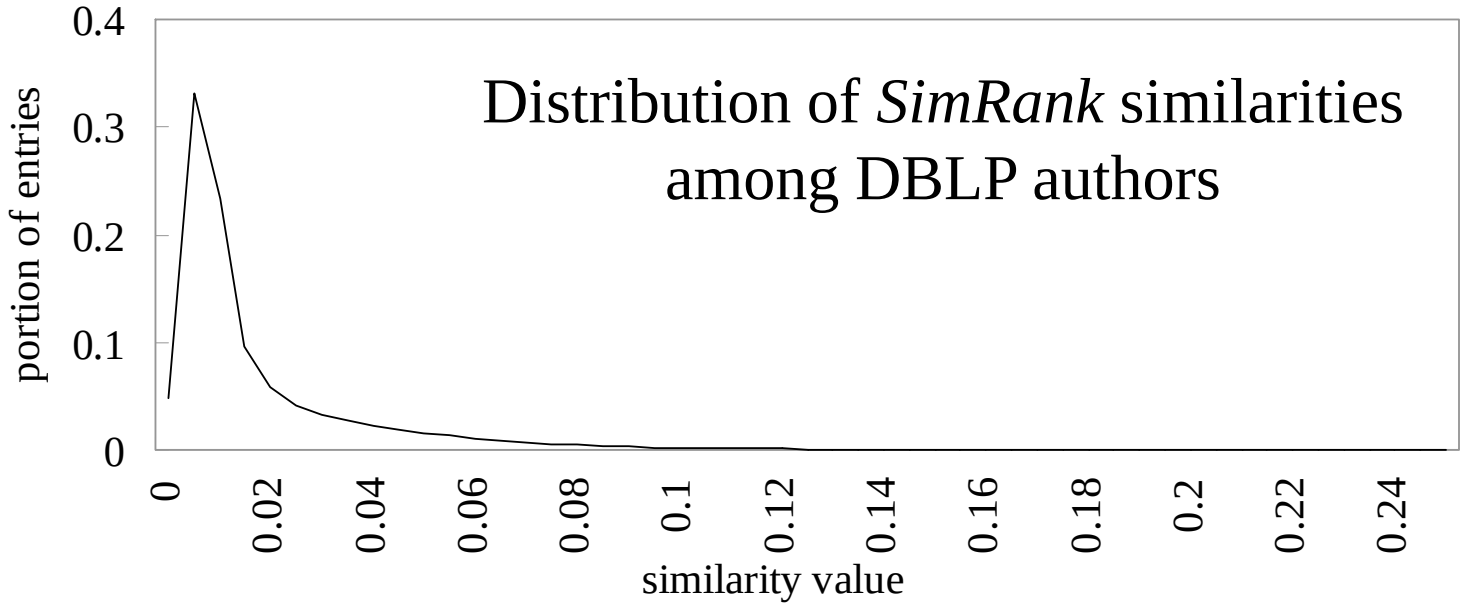
A hierarchical structure of products in Walmart



Relationships between articles and words (Chakrabarti, Papadimitriou, Modha, Faloutsos, 2004)

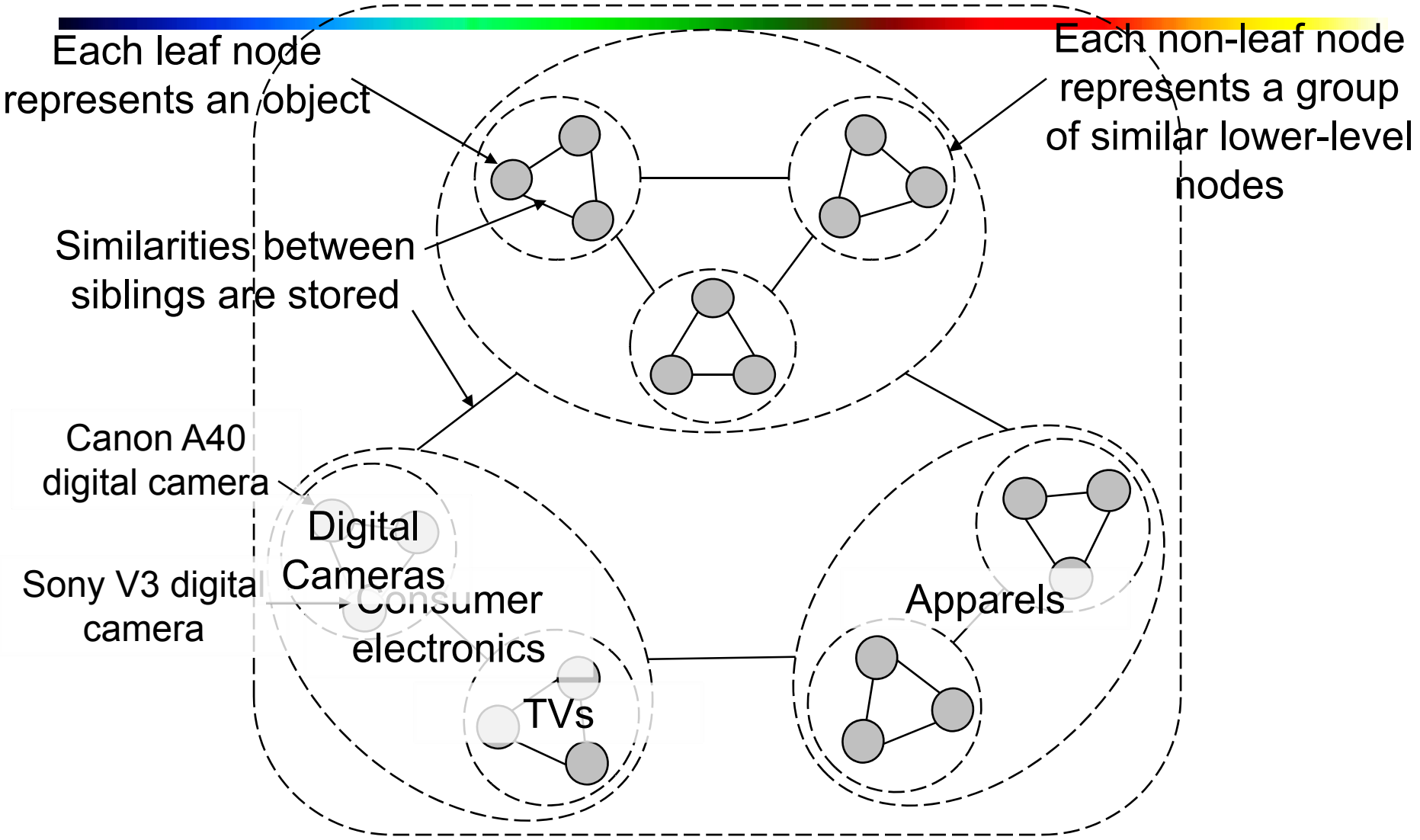


Observation 2: Distribution of Similarity



- Power law distribution exists in similarities
  - 56% of similarity entries are in [0.005, 0.015]
  - 1.4% of similarity entries are larger than 0.1
  - Our goal: Design a data structure that stores the significant similarities and compresses insignificant ones

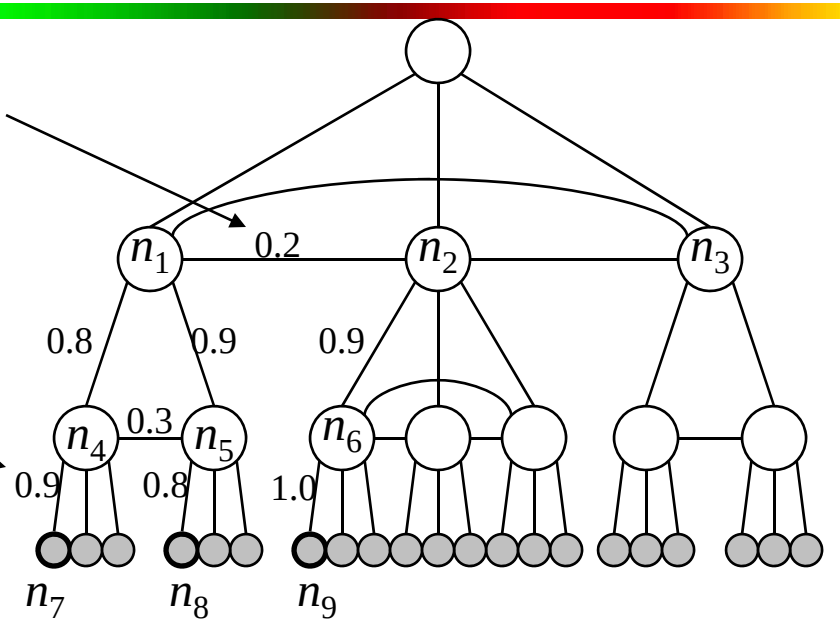
Our Data Structure: SimTree



Similarity Defined by SimTree

Similarity between two sibling nodes n_1 and n_2

Adjustment ratio for node n_7



- $sim_p(n_7, n_8) = s(n_7, n_4) \times s(n_4, n_5) \times s(n_5, n_8)$
  - Path-based node similarity

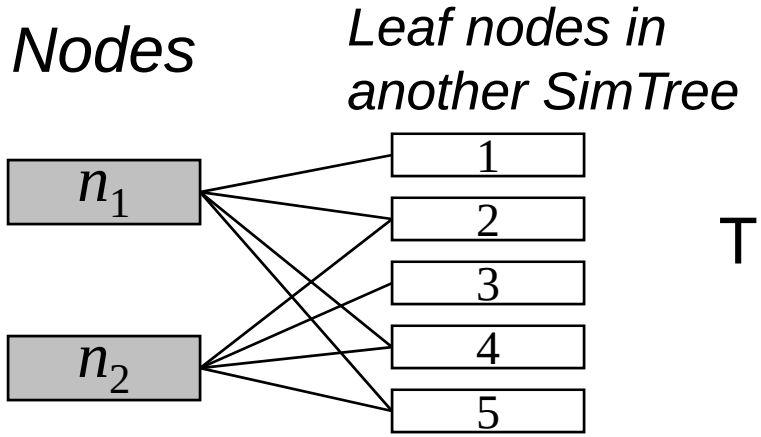
- Adjustment ratio for x = $\dfrac{\text{Average similarity between } x \text{ and all other nodes}}{\text{Average similarity between } x\text{'s parent and all other nodes}}$

Overview of LinkClus

- Initialize a SimTree for objects of each type
- Repeat
  - For each SimTree, update the similarities between its nodes using similarities in other SimTrees
    - Similarity between two nodes x and y is the average similarity between objects linked with them
  - Adjust the structure of each SimTree
    - Assign each node to the parent node that it is most similar to
- X. Yin, J. Han, and P. S. Yu, “LinkClus: Efficient Clustering via Heterogeneous Semantic Links”, VLDB'06

Initialization of SimTrees

- Initializing a SimTree
  - Repeatedly find groups of tightly related nodes, which are merged into a higher-level node
- Tightness of a group of nodes
  - For a group of nodes $\{n_1, \ldots, n_k\}$, its tightness is defined as the number of leaf nodes in other SimTrees that are connected to all of $\{n_1, \ldots, n_k\}$



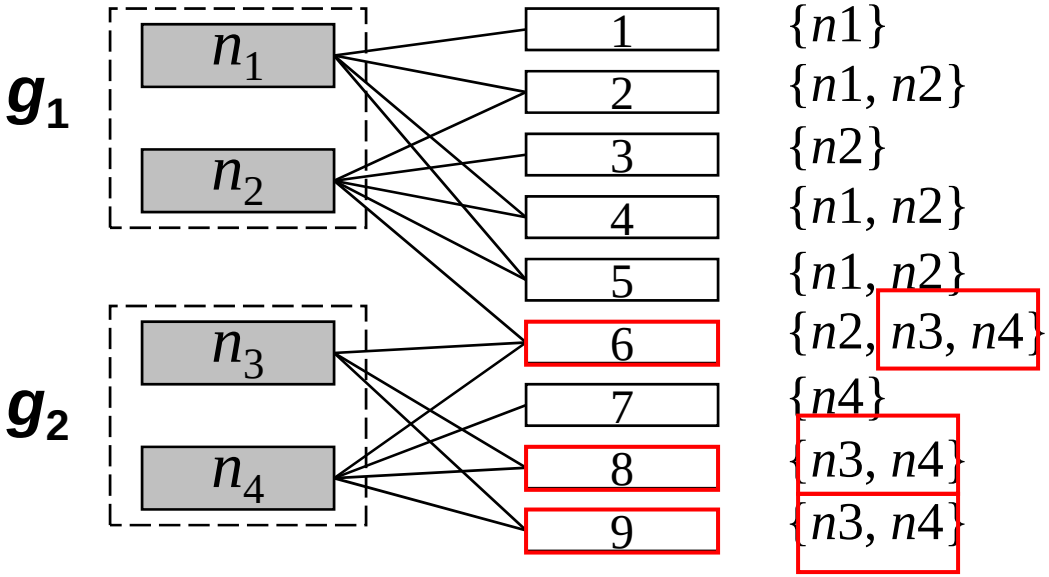
The tightness of $\{n_1, n_2\}$ is 3

(continued)

- Finding tight groups ⟶ Frequent pattern mining

Reduced to

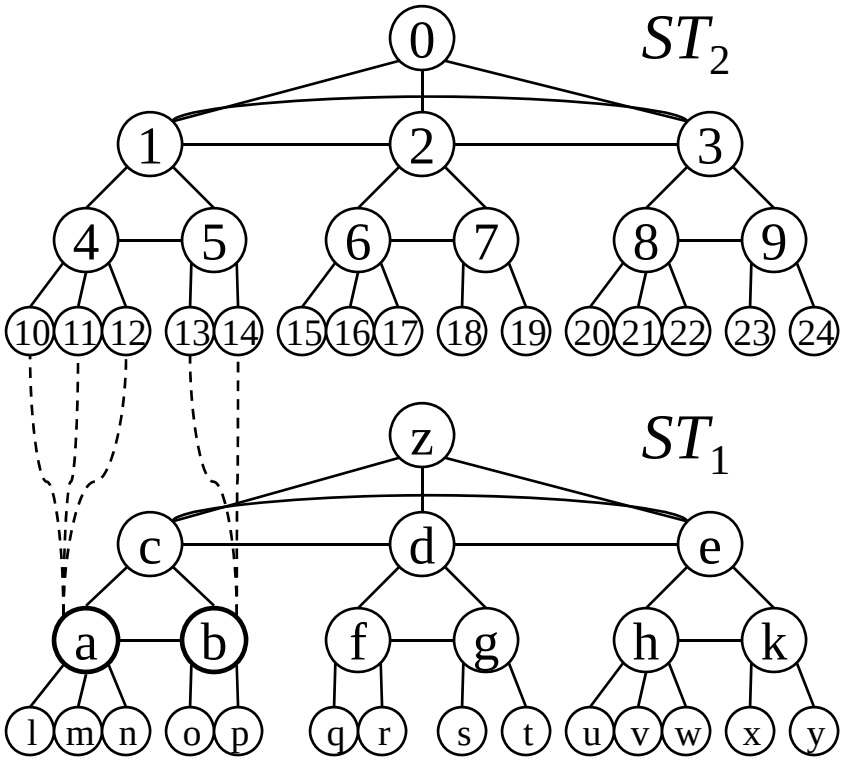
The tightness of a group of nodes is the support of a frequent pattern



- Procedure of initializing a tree
  - Start from leaf nodes (level-0)
  - At each level /, find non-overlapping groups of similar nodes with frequent pattern mining

Updating Similarities Between Nodes

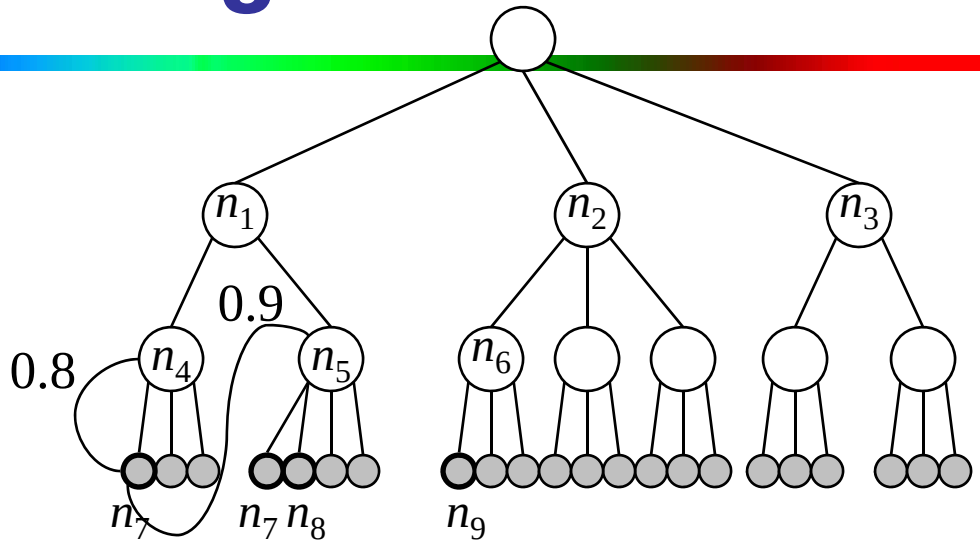
- The initial similarities can seldom capture the relationships between objects
- Iteratively update similarities
  - Similarity between two nodes is the average similarity between objects linked with them



$sim(n_a,n_b)$ = average similarity between (10, 11, 12) and (13, 14)

takes $O(3x2)$ time

Adjusting SimTree Structures



- After similarity changes, the tree structure also needs to be changed
  - If a node is more similar to its parent's sibling, then move it to be a child of that sibling
  - Try to move each node to its parent's sibling that it is most similar to, under the constraint that each parent node can have at most *c* children

Complexity

For two types of objects, N in each, and M linkages between them.

	Time	Space
Updating similarities	$O(M(\log N)^2)$	$O(M+N)$
Adjusting tree structures	$O(N)$	$O(N)$
LinkClus	$O(M(\log N)^2)$	$O(M+N)$
SimRank	$O(M^2)$	$O(N^2)$

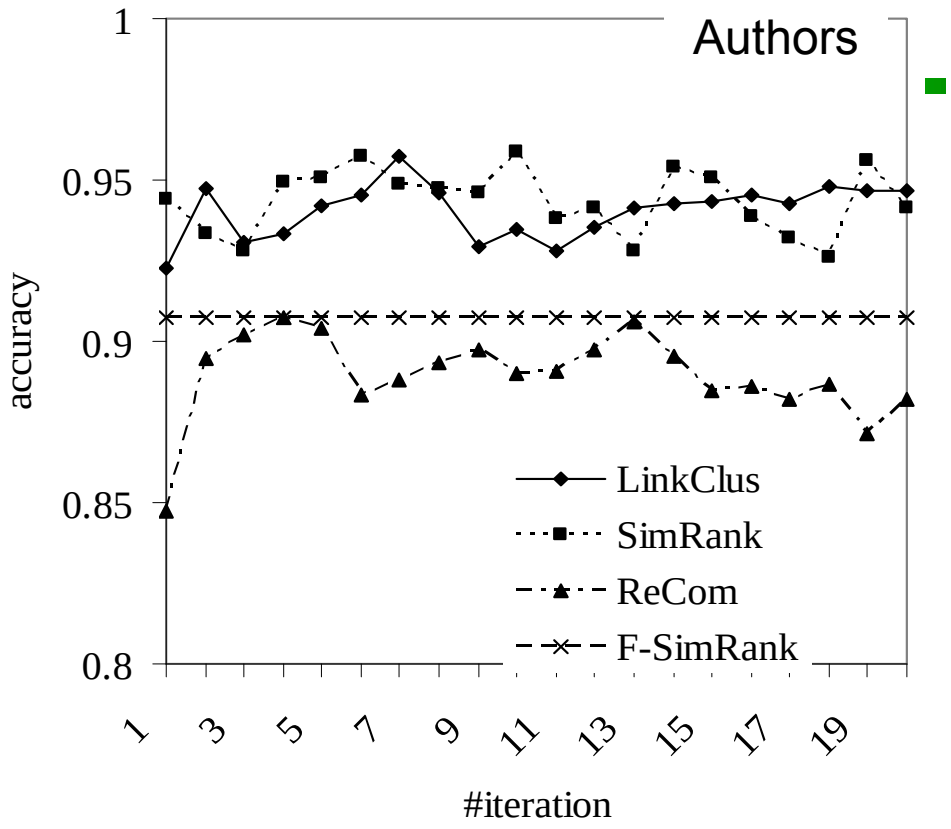
Empirical Study

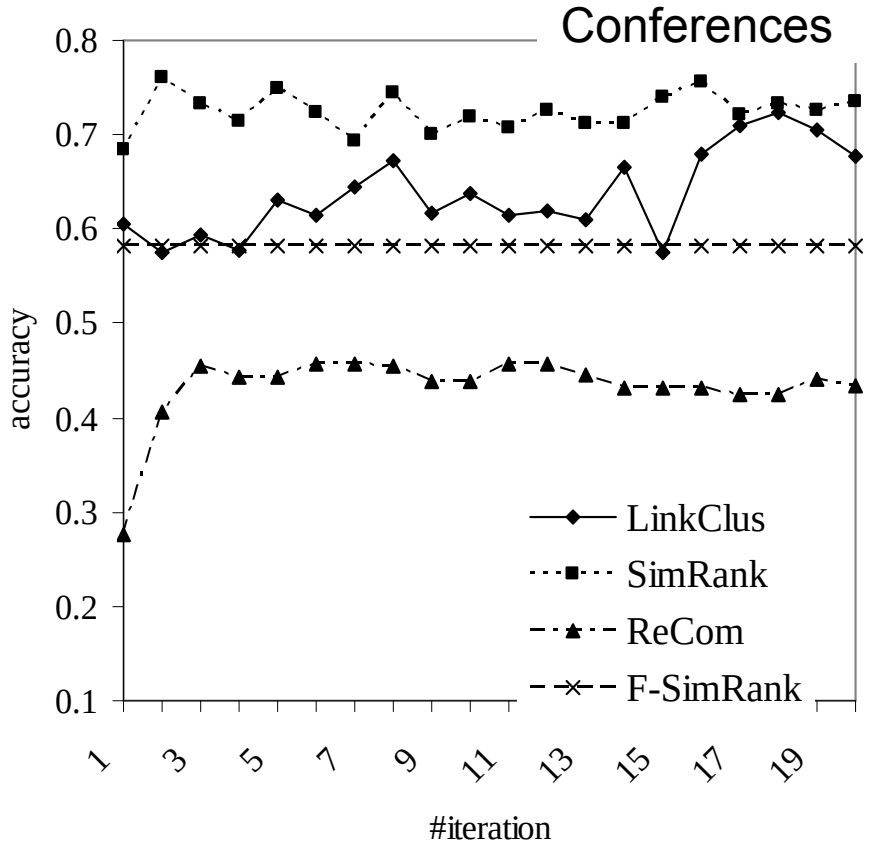
- Generating clusters using a SimTree
  - Suppose *K* clusters are to be generated
  - Find a level in the SimTree that has number of nodes closest to *K*
  - Merging most similar nodes or dividing largest nodes on that level to get *K* clusters
- Accuracy
  - Measured by manually labeled data
  - Accuracy of clustering: Percentage of pairs of objects in the same cluster that share common label
- Efficiency and scalability
  - Scalability w.r.t. number of objects, clusters, and linkages

Experiment Setup

- DBLP dataset: 4170 most productive authors, and 154 well-known conferences with most proceedings
  - Manually labeled research areas of 400 most productive authors according to their home pages (or publications)
  - Manually labeled areas of 154 conferences according to their call for papers
- Approaches Compared:
  - SimRank (Jeh & Widom, KDD 2002)
    - Computing pair-wise similarities
  - SimRank with FingerPrints (F-SimRank)
    - Fogaras & R´acz, WWW 2005
    - pre-computes a large sample of random paths from each object and uses samples of two objects to estimate SimRank similarity
  - ReCom (Wang et al. SIGIR 2003)
    - Iteratively clustering objects using cluster labels of linked objects

Accuracy





Approaches	*Accr-Author*	*Accr-Conf*	*average time*
LinkClus	0.957	0.723	76.7
SimRank	0.958	0.760	1020
ReCom	0.907	0.457	43.1
F-SimRank	0.908	0.583	83.6

Email Dataset

- F. Nielsen. Email dataset. http://www.imm.dtu.dk/~rem/data/Email-1431.zip
- 370 emails on conferences, 272 on jobs, and 789 spam emails

Approach	*Accuracy*	*Total time (sec)*
LinkClus	0.8026	1579.6
SimRank	0.7965	39160.0
ReCom	0.5711	74.6
F-SimRank	0.3688	479.7
CLARANS	0.4768	8.55

Summary

- Scalable OLAP and mining information networks: An essential new task
- Data integration, cleaning and validation in information networks
  - Distinct and TruthFinder
- Online Analytical Processing of Information Networks
  - Infonet I-OLAP vs. T-OLAP models
- Mining Information Networks
  - Clustering heterogeneous networks: SimRank and LinkClus
- Knowledge is power, but knowledge is hidden in massive links
- Much more to be explored!

Major References of in This Talk

- C. Chen, X. Yan, F. Zhu, J. Han, and P. S. Yu, "*Graph OLAP: Towards Online Analytical Processing on Graphs*", ICDM'08.
- G. Jeh and J. Widom, "*Simrank: A measure of structural-context similarity*", KDD'02
- X. Yin, J. Han, and P. S. Yu, "*LinkClus: Efficient Clustering via Heterogeneous Semantic Links*", VLDB'06
- X. Yin, J. Han, and P. S. Yu, "*Object Distinction: Distinguishing Objects with Identical Names by Link Analysis*", ICDE'07
- X. Yin, J. Han, and P. S. Yu, "*Truth Discovery with Multiple Conflicting Information Providers on the Web*", KDD'07/TKDE'08