LinkClus

Efficient Clustering via Heterogeneous Semantic Links

Naimeh Sadeghi

November, 28, 2007

The Paper

- Authors:
  - Xiaoxin Yin
  - Jiawei Han
  - Philip S. Yu
- Publication:
  - ACM
  - September 2006
- Number of pages:12


LinkClus: Efficient Clustering via Heterogeneous Semantic Links

Xiaoxin Yin UIUC xyin1@uiuc.edu

Jiawei Han UIUC hanj@cs.uiuc.edu

Philip S. Yu IBM T. J. Watson Res. Center psyu@us.ibm.com

ABSTRACT

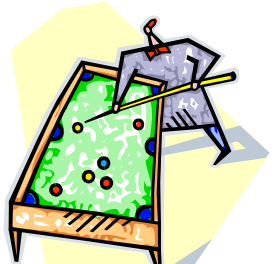
1. INTRODUCTION

Outline

- Introduction to the Problem
- SimTree Structure
- Linkclus Algorithm
- Results

Introduction

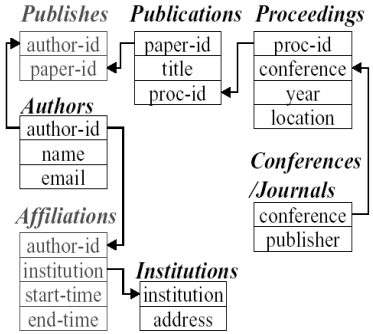
- Most of clustering methods aim to group records:
  - in a single table
  - using their own properties



Example

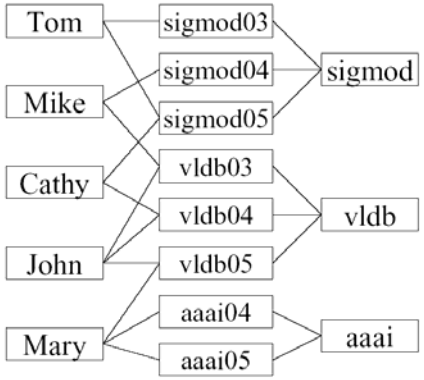
- In many real applications, linkages among objects of different types can be the most explicit information available.

Cluster authors base on the linkage between authors, papers and conferences.



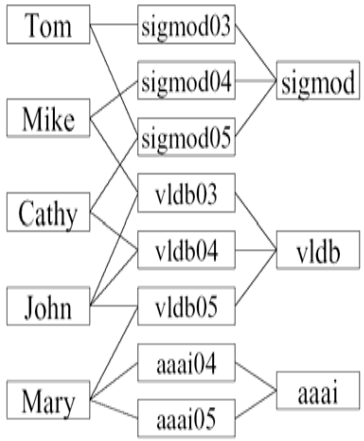
Simple Solution :direct-link

- The objects of each type are clustered based on the objects of other types linked with them.
- Tom and John will have zero similarity based on direct links!!



Problem

- If SIGMOID and VBLD are similar then SIGMOID authors and VBD authors should be similar.
- We have to consider the similarity between links.



SimRank

- Similarity between two objects is recursively defined as the average similarity between objects linked with them.
- quadratic complexity in both time and space and Impractical for larger databases

Reduce Computation

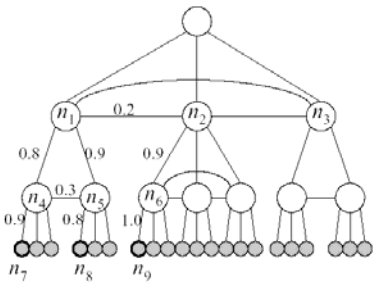
- How can we Reduce this computaion
- Is it necessary to compute and maintain pairwise similarities between objects?

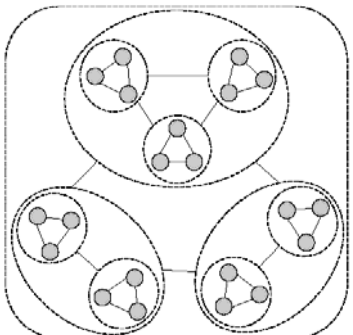
Outline

- Introduction to the Problem
- SimTree Structure
- Linkclus Algorithm
- Results

SimTree

- A hierarchical structure
- A compact representation of similarities between objects.
- No similarity storage for none sibling nodes

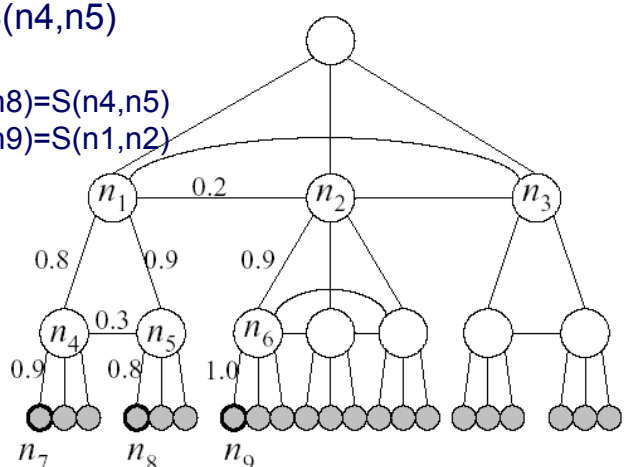




Similarity Estimation

n7 is a camera, n8 a TV, estimate their similarit as S(n4,n5)

S(n7,n8)=S(n4,n5)
S(n7,n9)=S(n1,n2)

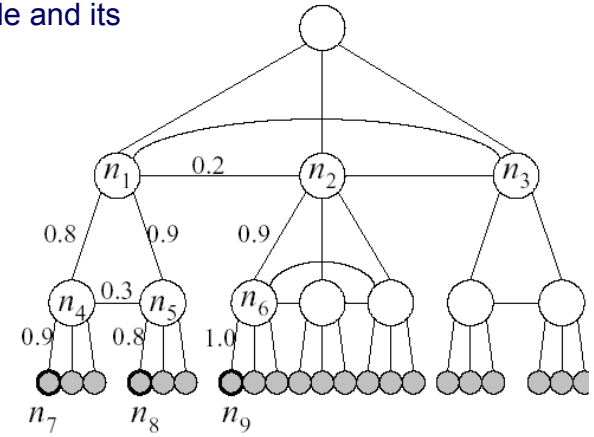


Problem

- A node and its parents may have different similarities to other nodes.

Adjustment based on the value of a node and its parents.

S(n7,n9)=
S(n7,n4).
S(n4,n1).
S(n1,n2).
S(n2,n6).
S(n6,n9)



Outline

- Introduction to the Problem
- SimTree Structure
- Linkclus Algorithm
- Results

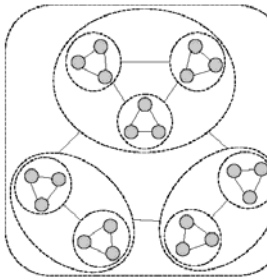
LinkClus

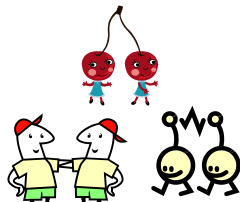
- An efficient and accurate approach for linkage-based clustering.

1. Builds an Initial SimTree
2. Improves each SimTree with an iterative method

Building the Initial SimTree

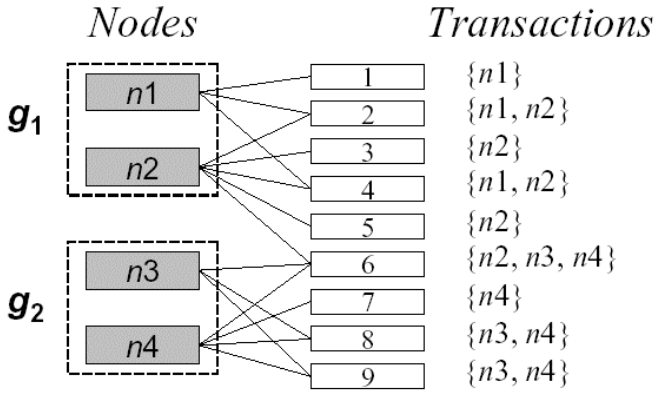
- Finding groups of nodes with high tightness
- A tight group is a set of nodes that are co-linked with many objects of other types
- Similar to a frequent pattern pattern problem: a set of items that co-appear in many transactions.





Tightly related Nodes

Number of co-links of objects of a group g
=support of group g in transactional database



Building the Initial SimTree

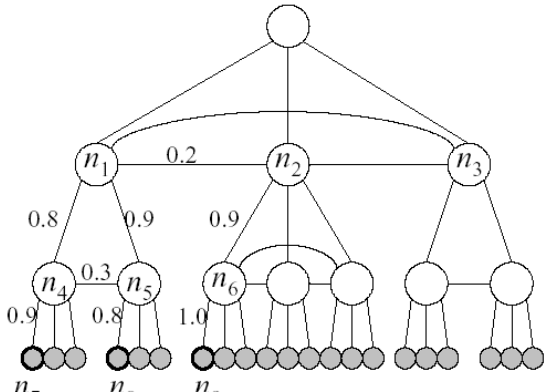
- Using a frequent closed pattern mining
- N(l) nodes at level l
- Each node should have at most c children
- Leave some space so
  N(l)/c<N(l+1)<a*N(l)/c and 1<a<2
  Thus the height of SimTree is $O(\log_c N)$
- Select groups with highest support that has no overlap with the previously selected group.
- Assign level-l node n that does not belong to any group to a group with highest connection to n . (number of objects that linked with both n and some members of g)

Improving SimTree

- LinkClus improves each SimTree with an iterative method
  1. Compute the edge values of childs and Parents
  2. Update the similarities
  3. Update the structure of the SimTree

Updating values Between Childs-Parents

$$s(n7, n4) = \frac{\text{Average similarity between all n7 and all leaf nodes except n4's descendants}}{\text{Average similarity between n4 and those nodes}}$$

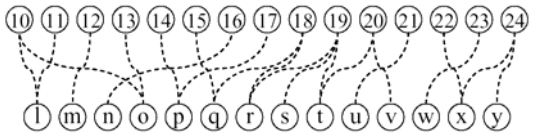


Updating Similarities

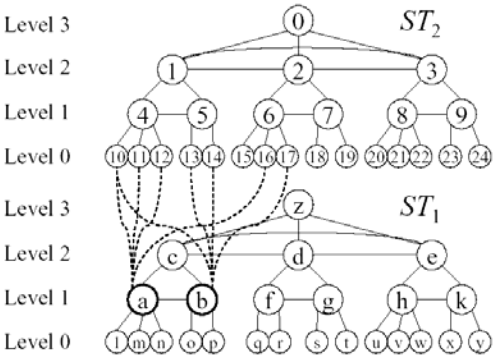
S(n 4,n 5)= the average similarity between objects linked with these two groups of leaf nodes

It can be shown that the procedure will take $O(mc \log_c N)$

Linkage for Updating Similarity Values



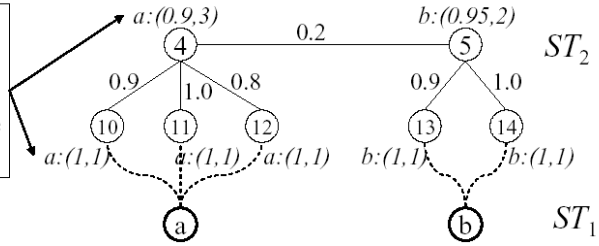
Assign a weight to each linkage that represents the number of links between descendant leaf nodes of n and n'



Updating Similarity

A simplified version of previous graph:

The two numbers in a bracket represent the average similarity and total weight of a linkage between two nodes

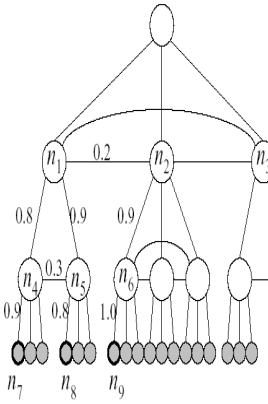


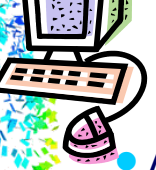
siml (na,nb) is the average path –based similarity between each node in {n10,n11,n12} and each node {n13, n14}

$$sim_l(n_a, n_b) = \frac{\sum_{k=10}^{12} s(n_k, n_4)}{3} \cdot s(n_4, n_5) \cdot \frac{\sum_{l=13}^{14} s(n_l, n_5)}{2}$$

Restructuring SimTree

- After computing similarities
- If n has higher similarity with a sibling n' of parent(n)
- Then n will become a child of n', if n' have less than c children
- If more than c nodes are most similar to n', keep only the top c nodes
- Assign remaining to other nodes





Complexity of Each Iteration

- At each node n in ST1,
  - Compute its similarity to its parents and parents' siblings $O(mc \log_c N)$
- Each parent have at most c siblings. $O(mc^2 \log_c N)$
- For all nodes there is M linkage in each level: $O(Mc^2(\log_c N)^2)$

Outline

- Introduction to the Problem
- SimTree Structure
- Linkclus Algorithm
- Results

Experimental Results

- Accuracy of LinkClus is either very close or sometimes even better than that of SimRank
- It has Much higher efficiency and scalability than SimRank.
- Much higher accuracy than other approaches on linkage-based clusterings Such as approaches that approximating SimRank for high efficiency