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

Data objects in a relational database are cross-linked with each other via multi-typed links. Links contain rich semantic information that may indicate important relationships among objects. Most current clustering methods rely only on the properties that belong to the objects per se. However, the similarities between objects are often indicated by the links, and desirable clusters cannot be generated using only the properties of objects.

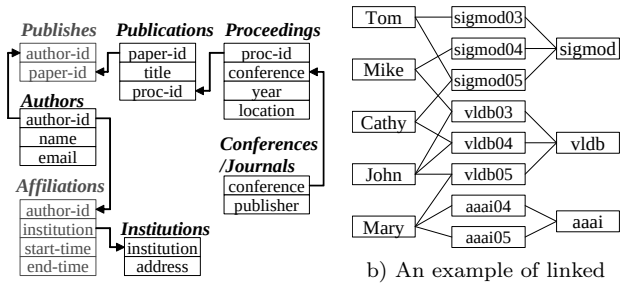
In this paper we explore linkage-based clustering, in which the similarity between two objects is measured based on the similarities between the objects linked with them. In comparison with a previous study (SimRank) that computes links recursively on all pairs of objects, we take advantage of the power law distribution of links, and develop a hierarchical structure called SimTree to represent similarities in multi-granularity manner. This method avoids the high cost of computing and storing pairwise similarities but still thoroughly explore relationships among objects. An efficient algorithm is proposed to compute similarities between objects by avoiding pairwise similarity computations through merging computations that go through the same branches in the SimTree. Experiments show the proposed approach achieves high efficiency, scalability, and accuracy in clustering multi-typed linked objects.

1. INTRODUCTION

As a process of partitioning data objects into groups according to their similarities with each other, clustering has been extensively studied for decades in different disciplines including statistics, pattern recognition, database, and data mining. There have been many clustering methods [1, 10, 14, 15, 17, 22], but most of them aim at grouping records in a *single table* into clusters using their own properties.

*The work was supported in part by the U.S. National Science Foundation NSF IIS-03-08215/05-13678 and NSF BDI-05-15813.

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the VLDB copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Very Large Data Base Endowment. To copy otherwise, or to republish, to post on servers or to redistribute to lists, requires a fee and/or special permission from the publisher, ACM.
VLDB '06, September 12-15, 2006, Seoul, Korea.
Copyright 2006 VLDB Endowment, ACM 1-59593-385-9/06/09.



a) Database schema b) An example of linked objects

Figure 1: A publication database (PubDB)

In many real applications, *linkages* among objects of different types can be the most explicit information available for clustering. For example, in a publication database (*i.e.*, PubDB) in Figure 1 (a), one may want to cluster each type of objects (authors, institutions, publications, proceedings, and conferences/journals), in order to find authors working on different topics, or groups of similar publications, *etc.* It is not so useful to cluster single type of objects (*e.g.*, authors) based only on the properties of them, as those properties often provide little information relevant to the clustering task. On the other hand, the linkages between different types of objects (*e.g.*, those between authors, papers and conferences) indicate the relationships between objects and can help cluster them effectively. Such *linkage-based clustering* is appealing in many applications. For example, an online movie store may want to cluster movies, actors, directors, reviewers, and renters, in order to improve its recommendation systems. In bioinformatics one may want to cluster genes, proteins, and their behaviors in order to discover their functions.

Clustering based on multi-typed linked objects has been studied in multi-relational clustering [13, 21], in which the objects of each type are clustered based on the objects of other types linked with them. Consider the mini example in Figure 1 (b). Authors can be clustered based on the conferences where they publish papers. However, such analysis is confined to direct links. For example, Tom publishes only SIGMOD papers, and John publishes only VLDB papers. Tom and John will have zero similarity based on direct links, although they may actually work on the same topic. Similarly, customers who have bought "*Matrix*" and those who have bought "*Matrix II*" may be considered dissimilar although they have similar interests.

The above example shows when clustering objects of one type, one needs to consider the similarities between objects of other types linked with them. For example, if it is known

that SIGMOD and VLDB are similar, then SIGMOD authors and VLDB authors should be similar. Unfortunately, similarities between conferences may not be available, either. This problem can be solved by *SimRank* [12], in which the similarity between two objects is recursively defined as the average similarity between objects linked with them. For example, the similarity between two authors is the average similarity between the conferences in which they publish papers. In Figure 1 (b) "sigmod" and "vldb" have high similarity because they share many coauthors, and thus Tom and John become similar because they publish papers in similar conferences. In contrast, John and Mary do not have high similarity even they are both linked with "vldb05".

Although SimRank provides a good definition for similarities based on linkages, it is prohibitively expensive in computation. In [12] an iterative approach is proposed to compute the similarity between every pair of objects, which has quadratic complexity in both time and space, and is impractical for large databases.

Is it necessary to compute and maintain pairwise similarities between objects? Our answer is no for the following two reasons. First, hierarchy structures naturally exist among objects of many types, such as the taxonomy of animals and hierarchical categories of merchandise. Consider the example of clustering authors according to their research. There are groups of authors working on the same research topic (*e.g.*, data integration or XML), who have high similarity with each other. Multiple such groups may form a larger group, such as the authors working on the same research area (*e.g.*, database vs. AI), who may have weaker similarity than the former. As a similar example, the density of linkages between clusters of articles and words is shown in Figure 2 (adapted from Figure 5 (b) in [4]). We highlight four dense regions with dashed boxes, and in each dense region there are multiple smaller and denser regions. The large dense regions correspond to high-level clusters, and the smaller denser regions correspond to low-level clusters within the high-level clusters.

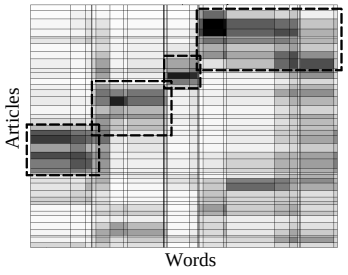


Figure 2: Density of linkages between articles and words

Second, recent studies show that there exist power law distributions among the linkages in many domains, such as Internet topology and social networks [8]. Interestingly, based on our observation, such relationships also exist in the similarities between objects in interlinked environments. For example, Figure 3 shows the distribution of pairwise SimRank similarity values between 4170 authors in DBLP database (the plot shows portion of values in each 0.005 range of similarity value). It can be seen that majority of similarity entries have very small values which lie within a small range (0.005 – 0.015). While only a small portion of similarity entries have significant values, — 1.4% of similarity entries (about 123K of them) are greater than 0.1, and these values will play the major role in clustering. Therefore, we want to

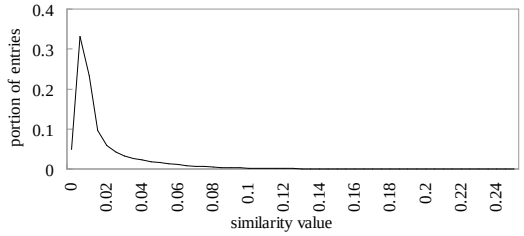


Figure 3: Portions of similarity values

design a data structure that stores the significant similarity values, and compresses those insignificant ones.

Based on the above two observations, we propose a new hierarchical strategy to effectively prune the similarity space, which greatly speedups the identification of similar objects. Taking advantage of the power law distribution of linkages, we substantially reduce the number of pairwise similarities that need to be tracked, and the similarity between less similar objects will be approximated using aggregate measures.

We propose a hierarchical data structure called SimTree as a compact representation of similarities between objects. Each leaf node of a SimTree corresponds to an object, and each non-leaf node contains a group of lower-level nodes that are closely related to each other. SimTree stores similarities in a multi-granularity way by storing similarity between each two objects corresponding to sibling leaf nodes, and storing the overall similarity between each two sibling non-leaf nodes. Pairwise similarity is not pre-computed or maintained between objects that are not siblings. Their similarity, if needed, is derived based on the similarity information stored in the tree path. For example, consider the hierarchical categories of merchandise in Walmart. It is meaningful to compute the similarity between every two cameras, but not so meaningful to compute that for each camera and each TV, as an overall similarity between cameras and TVs should be sufficient.

Based on SimTree, we propose LinkClus, an efficient and accurate approach for linkage-based clustering. At the beginning LinkClus builds a SimTree for each type of objects in a bottom-up manner, by finding groups of objects (or groups of lower level nodes) that are similar to each other. Because inter-object similarity is not available yet, the similarity between two nodes are measured based on the intersection size of their neighbor objects. Thus the initial SimTrees cannot fully catch the relationships between objects (*e.g.*, some SIGMOD authors and VLDB authors have similarity 0).

LinkClus improves each SimTree with an iterative method, following the recursive rule that *two nodes are similar if they are linked with similar objects*. In each iteration it measures the similarity between two nodes in a SimTree by the average similarity between objects linked with them. For example, after one iteration SIGMOD and VLDB will become similar because they share many authors, which will then increase the similarities between SIGMOD authors and VLDB authors, and further increase that between SIGMOD and VLDB. We design an efficient algorithm for updating SimTrees, which merges the expensive similarity computations that go through the same paths in the SimTree. For a problem involving N objects and M linkages, LinkClus only takes $O(M(\log N)^2)$ time and $O(M+N)$ space (SimRank takes $O(M^2)$ time and $O(N^2)$ space).

Comprehensive experiments on both real and synthetic datasets are performed to test the accuracy and efficiency of

LinkClus. It is shown that the accuracy of LinkClus is either very close or sometimes even better than that of SimRank, but with much higher efficiency and scalability. LinkClus also achieves much higher accuracy than other approaches on linkage-based clustering such as *ReCom* [20], and approach for approximating SimRank with high efficiency [9].

The rest of the paper is organized as follows. We discuss related work in Section 2, and give an overview in Section 3. Section 4 introduces SimTree, the hierarchical structure for representing similarities. The algorithms for building SimTrees and computing similarities are described in Section 5. Our performance study is reported in Section 6, and this study is concluded in Section 7.

2. RELATED WORK

Clustering has been extensively studied for decades in different disciplines including statistics, pattern recognition, database, and data mining, with many approaches proposed [1, 10, 14, 15, 17, 22]. Most existing clustering approaches aim at grouping objects in a single table into clusters, using properties of each object. Some recent approaches [13, 21] extend previous clustering approaches to relational databases and measures similarity between objects based on the objects joinable with them in multiple relations.

In many real applications of clustering, objects of different types are given, together with linkages among them. As the attributes of objects often provide very limited information, traditional clustering approaches can hardly be applied, and linkage-based clustering is needed, which is based on the principle that two objects are similar if they are linked with similar objects.

This problem is related to bi-clustering [5] (or co-clustering [7], cross-association [4]), which aims at finding dense submatrices in the relationship matrix of two types of objects. A dense submatrix corresponds to two groups of objects of different types that are highly related to each other, such as a cluster of genes and a cluster of conditions that are highly related. Unlike bi-clustering that involves no similarity computation, LinkClus computes similarities between objects based on their linked objects. Moreover, LinkClus works on a more general problem as it can be applied to a relational database with arbitrary schema, instead of two types of linked objects. LinkClus also avoids the expensive matrix operations often used in bi-clustering approaches.

A bi-clustering approach [7] is extended in [3], which performs agglomerative and conglomerative clustering simultaneously on different types of objects. However, it is very expensive, — quadratic complexity for two types and cubic complexity for more types.

Jeh and Widom propose SimRank [12], a linkage-based approach for computing the similarity between objects, which is able to find the underlying similarities between objects through iterative computations. Unfortunately SimRank is very expensive as it has quadratic complexity in both time and space. The authors also discuss a pruning technique for approximating SimRank, which only computes the similarity between a small number of preselected object pairs. In the extended version of [12] the following heuristic is used: Only similarities between pairs of objects that are linked with same objects are computed. With this heuristic, in Figure 1 (b) the similarity between SIGMOD and VLDB will never be computed. Neither will the similarity between Tom and John, Tom and Mike, *etc.* In general, it is very challenging to identify the right pairs of objects at the beginning, because many pairs of similar objects can only be identified after computing similarities between other objects. In fact this is the major reason that we adopt the recursive definition of similarity and use iterative methods.

A method is proposed in [9] to perform similarity searches by approximating SimRank similarities. It creates a large sample of random walk paths from each object and uses them to estimate the SimRank similarity between two objects when needed. It is suitable for answering similarity queries. However, very large samples of paths are needed for making accurate estimations for similarities. Thus it is very expensive in both time and space to use this approach for clustering a large number of objects, which requires computing similarities between numerous pairs of objects.

Wang et al. propose ReCom [20], an approach for clustering inter-linked objects of different types. ReCom first generates clusters using attributes and linked objects of each object, and then repeatedly refines the clusters using the clusters linked with each object. Compared with SimRank that explores pairwise similarities between objects, ReCom only explores the neighbor clusters and does not compute similarities between objects. Thus it is much more efficient but much less accurate than SimRank.

LinkClus is also related to hierarchical clustering [10, 17]. However, they are fundamentally different. Hierarchical clustering approaches use some similarity measures to put objects into hierarchies. While LinkClus uses hierarchical structures to represent similarities.

3. OVERVIEW

Linkage-based clustering is based on the principle that two objects are similar if they are linked with similar objects. For example, in a publication database (Figure 1 (b)), two authors are similar if they publish similar papers. The final goal of linkage-based clustering is to divide objects into clusters using such similarities. Figure 4 shows an example of three types of linked objects, and clusters of similar objects which are inferred from the linkages. It is important to note that objects 12 and 18 do not share common neighbors, but they are linked to objects 22 and 24, which are similar because their common linkages to 35, 37 and 38.

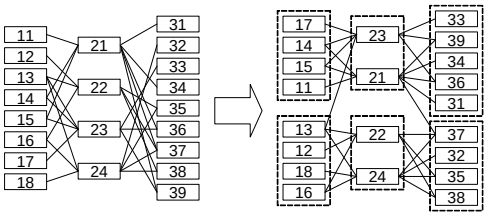


Figure 4: Finding groups of similar objects

In order to capture the inter-object relationships as in the above example, we adopt the recursive definition of similarity in SimRank [12], in which the similarity between two objects x and y is defined as the average similarity between the objects linked with x and those linked with y.

As mentioned in the introduction, a hierarchical structure can capture the hierarchical relationships among objects, and can compress the majority of similarity values which are insignificant. Thus we use SimTree, a hierarchical structure for storing similarities in a multi-granularity way. It stores detailed similarities between closely related objects, and overall similarities between object groups. We