A Bipartite Graph Co-Clustering Approach to Ontology Mapping

Yiling Chen **Frederico Fonseca**

School of Information Sciences and Technology
The Pennsylvania State University
University Park, PA 16802
ychen@ist.psu.edu ffonseca@ist.psu.edu

Abstract

The necessity of mapping concepts of one ontology to concepts in a second ontology is an important research topic due to the requirements brought by the Semantic Web. Most ontology mapping techniques available today do not allow the existence of many-to-many correspondences among concepts. To overcome this problem we propose to model two ontologies as a weighted bipartite graph. We assign weights to graph edges using current similarity measure techniques and apply graph partitioning techniques in order to co-cluster the vertex sets of the bipartite graph. Then we use the resulting concept clusters to establish mappings between concepts of the two ontologies. The approach combines mapping methods that rely solely on similarity measures with an unsupervised learning technique called bipartite graph co-clustering. The advantage of the combination is that it allows mappings to have many-to-many concept correspondences.

Introduction

The rapid development of the web technology has brought along an increasing interest in research on knowledge sharing in a distributed environment. The Semantic Web (Berners-Lee, Hendler, and Lassila 2001) envisions a world of software agents that understand documents semantically in a decentralized architecture. Ontologies have been recognized as a crucial component for knowledge sharing and the realization of this vision. However, it is unlikely that a global ontology can be developed for distributed systems. In practice, ontologies for different systems are developed independently by different communities. Thus, if knowledge and data are to be shared, it is essential to establish semantic mappings between ontologies.

Manually creating mappings between different sources is a labor-intensive and error-prone process, which is a major bottleneck when scaling up systems to a large number of sources. Many methods for creating mappings between ontologies have been proposed. These methods usually either rely on similarity measures, which are based on linguistic and/or structural characteristics of ontologies, to establish mappings (Noy and Musen 2000; Noy and Musen 2001; Rahm and Bernstein 2001; and Rodríguez and Egenhofer 2003), or use machine learning techniques to implicitly "learn" semantic relationships between ontologies (Lacher and Groh 2001; Doan, Domingos, and Halevy 2001; Doan, Domingos, and Halevy 2002; and Berlin and Motro 2002). Most of these methods attempt to establish one-to-one relationships between concepts of different ontologies.

However, establishing one-to-one semantic mappings is not an easy task. An ontology reflects the worldview of the community who built it. There is nothing to constrain worldviews of independent communities that later would lead to establishing one-to-one correspondences among concepts. It is very likely that several concepts in one ontology are semantically equivalent to several concepts in another ontology. Therefore, in many applications it is desirable to have ontology mapping that allows many-to-many concept correspondences (Wache et al. 2001).

In this paper, we propose an ontology mapping method by which concepts in both ontologies are grouped into concept clusters. A cluster only consists of concepts from the same ontology. Similarity of a pair of concepts, each coming from a different ontology, is measured by some existing similarity assessment technique. Similarity of a pair of concept clusters, each coming from a different ontology, can be measured by the sum of similarities of all concept pairs (with each concept of a pair from a different ontology) within the cluster pair. Roughly speaking, our method seeks to map a cluster of concepts in one ontology to a similar cluster of concepts in the second ontology. This establishes many-to-many concept correspondences between the two ontologies. The method proposed here models an ontology mapping problem as a weighted bipartite graph partitioning problem (Dhillon 2001).

Bipartite Graph Co-Clustering for Ontology Mapping

A bipartite graph has two disjoint vertex sets. Co-clustering a bipartite graph is to simultaneously group several vertices into similar clusters for each vertex set. We use this technique here in order to be able to establish many-to-many mappings between concepts of two ontologies.

Rationale of the Method

An ontology represents a worldview of a community. Even if two ontologies are about the same subject, they may be

different because communities who built the ontologies may have different worldviews. Assume an ideal situation, in which we can rearrange concepts in one ontology according to the other ontology's worldview. This reorganization is similar to a clustering process, which is to group concepts of one ontology into clusters. Concepts in a cluster are similar because they all semantically associate with some concept in the other ontology. In an ideal situation, ontology mapping can be achieved by mapping each concept cluster in one ontology with a corresponding single concept in the second ontology.

However, such ideal situation usually does not exist. It is more likely that a concept in one ontology associates with several concepts in another ontology and vice versa (Rishe 1992). Thus, to some degree concepts of both ontologies need to be rearranged according to the other's perspective. The need of co-clustering concepts in both ontologies hence arises.

Bipartite Graph Modeling of Ontologies

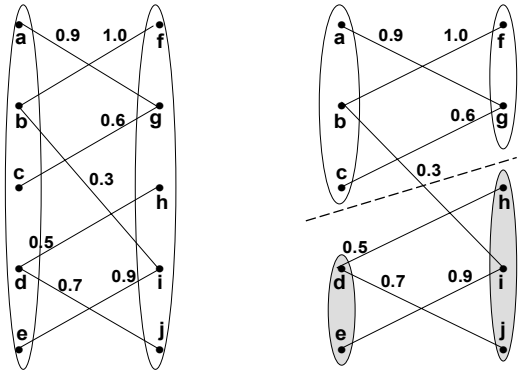
A bipartite graph is an undirected graph whose vertices are divided into two disjoint sets such that no two vertices within the same set are adjacent. Edges only connect vertices from different sets. Many real world data types can be modeled as bipartite graphs, including terms and documents in a text corpus, customers and purchasing items in market basket analysis, and reviewers and movies in a movie recommender system (Zha et al. 2001).

We choose to model two ontologies on the same subject as a weighted bipartite graph. The graph has two vertex sets. The first includes concept nodes of the first ontology while the second consists of concept nodes of the second ontology. Edges in the graph connect concepts of one ontology with concepts of another ontology. Concepts of the same ontology are not connected. The weight of an edge represents the similarity between the two concepts that are connected by the edge and it is calculated through the use of current techniques for similarity assessment such as that in (Rodríguez and Egenhofer 2003). In order to reduce the computational complexity, only edges with weight greater than a predefined threshold are included in the graph.

Bipartite Graph Co-Clustering

After modeling two ontologies as a bipartite graph, we apply bipartite graph co-clustering technique to establish mappings between two ontologies. Co-clustering in a bipartite graph can be naturally formulated as a graph-partitioning problem, which aims at getting the vertex partition with minimum cut (Dhillon 2001; and Zha et al. 2001). In order to better understand the technique, we present an example in Figure 1.

Figure 1 has two parts that illustrate a bipartite graph before and after the bi-partition respectively. Part A depicts the situation before the partition, when each vertex set of the bipartite graph can be viewed as a single cluster. In part B, a partition (intuitively shown by the dotted line) breaks one edge and separates each vertex set into two parts. Each part can be viewed as a cluster.



A. Before the Partition **B. After the Partition**

Figure 1: A Bipartite Graph Partitioning Example

The graph has ten vertices. Vertices a, b, c, d, and e, which are concepts of the first ontology, form one vertex set. Vertices f, g, h, i, and j, which are concepts of the second ontology, make the other vertex set. Numbers on the edges are the weights, which are calculated through the use of a similarity measure technique. Part A of Figure 1 shows the graph before any partition. We can view each vertex set as a single cluster. A bi-partition of a bipartite graph is the result of cutting through the vertex sets of the graph. The cut of a partition is defined as the sum of weights of those edges that are "broken" in the partition. A bi-partition is shown by the dotted line in part B of Figure 1. Only one edge whose weight is 0.3 is "broken" in the partition. Hence, the cut of the partition is 0.3. The partition cuts the original graph into two bipartite graphs. Vertex sets of each new sub-graph form a cluster pair. Thus, a bi-partition co-clusters vertices into two cluster pairs. Clusters of the same pair preserve all features of the original graph except by losing the connections with other cluster pairs. One way to measure the similarity between two concept clusters is the sum of weights for all edges connecting the two clusters. Ideally, we want clusters from the same pair to be as similar as possible, which means that clusters from different pairs are less similar. Since weighs of edges that are "broken" in a partition reflect similarity of clusters from different pairs, this leads to the idea of finding the partition that minimizes the cut. The partition showed in part B of Figure 1 is already the partition with the minimal cut.

Given the partition that minimizes the cut, we can establish the mapping relationship for each cluster pair. In the bi-partition example shown in Figure 1, we map vertices a, b, and c to vertices f and g, and vertices d and e to vertices h, i, and j. This way we have established mappings that allow for many-to-many concept correspondences.

Similarly, we can co-cluster a bipartite graph into k cluster pairs with a k-partition, where the value of k is of

our choice. An ontology mapping problem then becomes seeking the k-partition that minimizes the cut of the partition. It has been studied that simply minimizing the cut of the partition usually results in clusters of very unbalanced size. A better way is to minimize a normalized variant of the cut, which constrains the size of clusters. Details of this approach can be found in (Dhillon 2001; and Zha et al. 2001).

Conclusions and Future Work

We proposed modeling two ontologies as a weighted bipartite graph. Concepts of one ontology form one vertex set, while concepts of the second ontology form the other vertex set. Weights of graph edges are calculated through the use of current similarity measure techniques. Graph partition techniques are applied to co-cluster the vertex sets of the bipartite graph. We established mappings between concepts in the two ontologies based on the resulting cluster pairs. Our approach combined mapping methods that rely solely on similarity measures with an unsupervised learning technique called bipartite graph co-clustering. The advantage of the combination was that it allowed the mappings to have many-to-many concept correspondences.

Resuming this work, we will apply this method on mediating the sharing of scientific documents between different information communities across environmental sciences. There are two questions we are especially interested in investigating when applying the method. First, which is the best similarity measure to be used when assigning weights to the edges? A challenge to the proposed method is that modeling two ontologies as a bipartite graph implicitly ignores the internal structure within ontologies. This may negatively affect the mapping result because we do not fully take advantage of all available information. To face this challenge, we hope the chosen similarity measure can implicitly reflect part of the structural information as a remedy. The answer to this question, we believe, will be problem dependent. Our second question is if we can deal with situations when semantic contents of concepts within an ontology are overlapping. Our proposed method considered so far only "hard" clustering (Dhillon 2001), i.e., the situation in which a concept vertex belongs to one and only one cluster. In many situations, semantic contents of concepts within an ontology are overlapping to some degree. It would be useful to allow concept vertices to belong to several clusters. We intend to explore the potential of bipartite graph co-clustering on dealing with this problem.

References

Berlin, J.; and Motro, A. 2002. Database Schema Matching Using Machine Learning with Feature Selection. In *Proceedings of the 14th International Conference on Advanced Information Systems Engineering (CAiSE02)*. Toronto, Ontario, Canada.

Berners-Lee, T.; Hendler, J.; and Lassila, O. 2001. The Semantic Web: A New Form of Web Content That Is Meaningful to Computers Will Unleash a Revolution of New Possibilities. *The Scientific American* 284(5): 34-43.

Dhillon, I.S. 2001. Co-Clustering Documents and Words Using Bipartite Spectral Graph Partitioning, Technical Report, 2001-5, CS Dept, Univ. of Texas, Austin.

Doan, A.; Domingos, P.; and Halevy, A.Y. 2001. Reconciling Schemas of Disparate Data Sources: A Machine Learning Approach. In *Proceedings of ACM SIGMOD Conference on Management of Data*, 509-520. Santa Barbara, California.

Doan, A.; Domingos, P.; and Halevy, A.Y. 2002. Learning to Map between Ontologies on the Semantic Web. In *Proceedings of the 11th International Conference on World Wide Web*, 662-673. Honolulu, Hawaii.

Lacher, M.S.; and Groh, G. 2001. Facilitating the Exchange of Explicit Knowledge through Ontology Mappings. In *Proceedings of the 14th International FLAIRS conference*. Key West, Florida.

Noy, N.F.; and Musen, M.A. 2000. PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. In *Proceedings of the 17th National Conference on Artificial Intelligentce (AAAI'00)*, 450-455. Austin, Texas: AAAI Press.

Noy, N.F.; and Musen, M.A. 2001. Anchor-PROMPT: Using Non-Local Context for Semantic Matching. In *Proceedings of the Workshop on Ontologies and Information Sharing at the International Joint Conference on Artificial Intelligence (IJCAI)*. Seattle, Washington.

Rahm, E.; and Bernstein, P.A. 2001. A Survey on Approaches to Automatic Schema Matching. *VLDB Journal* 10(4): 334-350.

Rishe, N.D., 1992. Database Design: The Semantic Modeling Approach. McGraw- Hill.

Rodríguez, M.A.; and Egenhofer, M.J. 2003. Determining Semantic Similarity Among Entity Classes from Different Ontologies. *IEEE Transactions on Knowledge and Data Engineering* 15(2): 442-456.

Wache, H.; Vogele, T.; Visser, U.; Stuckenschmidt, H.; Schuster, G.; Neumann, H.; and Hubnet, S. 2001. Ontology-Based Integration of Information - A Survey of Existing Approaches. In *Proceedings of IJCAI-01 Workshop on Ontologies and Information Sharing*, 108-117. Seattle, Washington.

Zha, H.; He, X.; Ding, C.; Simon, H.; and Gu, M. 2001, Bipartite Graph Partitioning and Data Clustering. In *Proceeding of 10th International Conference of Information and Knowledge Management (CIKM 2001)*. Atlanta, Georgia.