Generative Models for Clustering: The Next Generation *

Adam Anthony and **Marie desJardins**
Department of Computer Science and Electrical Engineering
University of Maryland Baltimore County
1000 Hilltop Circle, Baltimore, MD 21250
(410) 455-8894
{aanthon2 , mariedj}@cs.umbc.edu

Abstract

Clustering social information is challenging when both attributes and relations are present. Many approaches commonly used today ignore one aspect of the data or the other. Relation-only algorithms are typically limited because the sparseness of relations in social information makes finding strong patterns difficult. Feature-only algorithms have the limitation that the most interesting (and useful) aspect of social information is the set of relations between objects. Recently, there has been a surge in interest in simultaneously considering both features and relations in the operation of a clustering algorithm. Several of these approaches are based on a *generative model,* which corresponds to an assumption that the data exists as part of some unobserved probability distribution. Each of the approaches discussed in this paper have good initial results. After discussing each in turn, we discuss in broad terms what has been accomplished thus far with generative models, what open problems remain, and how the development of generative models for relational data can contribute to the field of social information processing.

Introduction

Social information processing has always been a topic of interest to sociology and artificial intelligence researchers. With the emergence of social networking phenomena on the World Wide Web and the publication of social interaction data sets like the Internet Movie Data Base,[1] social information processing has become an area where exciting, innovative new research is being done to make full use of the vast new data resources that are freely available.

These new data resources contain complex interactions such that there can be multiple representations of the same data set. In this paper, we assume an *attributed graph* representation. Such a representation consists of a set of objects and a set of relations between them. Each object is represented as a vertex in a graph where the relations are edges. Furthermore, each object has an associated set of *attributes*, or *features*.

*This material is based upon work supported by the National Science Foundation under Grant No. #0545726.
Copyright © 2008, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.
[1] http://www.imdb.com

Clustering an attributed graph can be difficult. It is not always apparent how to use the two sources of information—the object attributes and the relations between objects—together in an efficient manner. Often, a first approach is to ignore the relations altogether and perform a common clustering algorithm such as K-means (MacQueen 1967) over the object features to see if any useful groupings can be found. This is not desirable because the real interest is in how the objects in a social information data set interact.

A second approach would be to use graph cutting or some novel form of graph clustering to group objects. This has the advantage of showing how different groups of individuals interact, but because object features are ignored, it is difficult to analyze *why* the groups interact in a certain way based on the features of the individual objects.

The two approaches—using features only or using relations only—have a connection in that the advantage of one compensates for the disadvantage of the other. This implies that a combined approach could find clusterings that are both socially interesting and qualitatively meaningful in terms of individuals' features. Many researchers have begun exploring methods for processing object features and relations between objects simultaneously. This paper focuses on the specific approach of using *generative models* to perform such a task. Generative models of social information give us insights into how the data is structured because they take a statistical approach to clustering. By specifying assumptions about the probability distribution that models a given data set, researchers can not only find interesting clusterings, but they can also make conclusions about the data by analyzing the values of the distribution parameters that are learned in the process of clustering. What follows is a brief survey of some generative models that can be used for relational clustering, a discussion of the advantages/disadvantages of relational clustering, and a discussion of open problems.

A Brief Survey of Generative Models for Relational Clustering

Generative models describe probabilistically how a dataset may be formed. In a sense, they are a hypothesis of the underlying distribution that created the set of data objects. Assuming that the hypothesis is true, general algorithms such as Expectation Maximization, simulated annealing or Gibbs

sampling can be used to find a clustering that best agrees with the underlying model.

Bottom-Up Approaches

The models described in this section are all common in the sense that their designers each chose specific phenomena observed in relational data and designed a model that accurately discovers the phenomena.

Probabilistic Relational Models One open-ended and flexible generative model is the Probabilistic Relational Model (PRM) (Getoor *et al.* 2002). PRMs extend the naïve Bayes model of independent data objects to include relations by allowing the class label of an object to depend on the class label of another object it is linked to. Getoor *et al.* also introduced the concept of edge existence uncertainty which allows basic reasoning about whether an edge exists between two objects. Their model has the advantage of being flexible in the tasks that it can be used for. For example, the model was originally developed for making predictions of link structure in a data set, but Taskar, Segal, & Koller (2001) use it to cluster relational data. Because the class of one cluster can depend on the class of another, they were able to generate meaningful clusters in a movie database where actors were clustered together in part by the types of movies they acted in, because an actor's class depended on the class of the movies in which he acted.

Latent Group Models The Latent Group Model (LGM) proposed by Neville & Jensen (2006) is an attempt at a general model for relational data. They first assume that objects and their features exist probabilistically in a distribution associated with each cluster. Second, they assume that in addition to objects belonging to clusters, objects also belong to a different, hidden grouping in the data, referred to as latent groups. These latent groups represent *coordinating objects* that are not part of the data set; these coordinating objects represent the reason that relations exist between objects. Using the movie data base example again, a coordinating object for the relation `directed-movie` could be a movie studio. An object's cluster also depends on the group the object belongs in, so that objects that are in the same group are more likely to be in the same cluster. Since the presence of relations determines the latent group an object belongs to, they indirectly affect the cluster membership of the object. Latent group models are an intuitive, general representation of relational data. The initial results presented use a combination of relation-only clustering and supervised inference to find a final grouping. The important finding from LGMs is that *relational autocorrelation*, the tendency for related objects to have similar feature values, can be represented as a probabilistic influence in a generative model.

Stochastic Link and Group Detection Another example generative model is presented by Kubica *et al.* (2002). In their model, links exist as the result of some event (e.g., a phone call or a meeting). They use this model to detect groups of associated people from a social network. They begin by identifying some events as coincidental and others as intentional. The interest here is in determining which events are intentional, and which people are involved in the event. As with LGMs, Kubica *et al.* assume that relations exist as a result of the objects' membership in some unobserved group, where a membership in a group means that some personal relationship (e.g., club member, coworker, co-conspirator) exists between the members. Their model contrasts with LGMs because, instead of assuming that groups occur because of linkages, Kubica *et al.* assume that groups occur because of some demographic similarity between the two objects. The final task, then, is to determine the groups that have the highest likelihood for both the observed features and the observed links. In the final output, the objects in the data are placed into groups such that it is likely that there was a reason for the observed relation (i.e., that it was not merely coincidental).

Stochastic Block Model Approaches

A posteriori block modeling is the problem of determining a partitioning of data objects into groups such that members in the same group have similar linkage patterns for a certain relation. Stochastic approaches to a posteriori block modeling were pioneered by Nowicki & Snijders (2001). The fundamental component of stochastic block modeling is to assign a specific probability of edge existence for each edge observed in a relational graph that is dependent on the groups that the connected objects belong to. A partitioning is then found that maximizes the likelihood of the observed data. Recently, several researchers have proposed interesting extensions to the basic stochastic block model approach.

Infinite Relational Models Kemp *et al.* (2006) propose the Infinite Relational Model, an extension of stochastic block models to handle the case where the number of clusters is unknown. To incorporate object features, Kemp *et al.* discretize any continuous features and then represent the features as a set of objects, where an object in the original set is related to a feature-object if it has that feature value. Because of this relationship, IRMs have the unique quality that object features can be represented in the form of a relation, so that the process of clustering data objects also results in a clustering of object features. For example, in an animals data set, the binary features `lives-in-water` and `has-gills` might be placed in a group because they help to identify a specific kind of animal, namely fish. Furthermore, additional sources of relation can exist besides object features. Kemp *et al.* add another restriction that a clustering found for objects using one relation must be the same clustering found for any other relation type. For example, if the data set was a collection of students where demographic information and friendship relations are observed, the clustering found for the students would have to simultaneously maximize the likelihood of the friendship relations, as well as the feature value relations.

Latent Space Models Hoff, Raftery, & Handcock (2002) and Handcock, Raftery, & Tantrum (2007) have recently

proposed models in which additional information about each object in a relational data set can be used to reinforce the information extracted from the relations in latent group discovery. Hoff, Raftery, & Handcock (2002) developed a *latent space* model for social networks. They assume that the objects in a network are embedded in an unknown k-dimensional euclidean space. Relation existence in their model depends on the euclidean distance between the two objects in the latent space, modeled using logistic regression. Object features can be incorporated into the model by adding them as covariate information to the logistic regression model. While Hoff, Raftery, & Handcock (2002) did not perform clustering with their model Handcock, Raftery, & Tantrum (2007) extended the model with the added observation that the objects may belong to clusters within the latent space, which he models as a mixture of gaussians. Handcock, Raftery, & Tantrum (2007) propose two methods for clustering. The first is to embed the objects in a latent space using Hoff, Raftery, & Handcock (2002)'s method, then use the EM algorithm to find a clustering in the latent space. The second is a fully Bayesian approach where the parameters are estimated using Gibbs sampling and the Metropolis-Hastings approximation for distributions that are difficult to compute. An interesting feature of their method is that it successfully captures transitivity, due to the triangle inequality in the embedded space. They showed that their Bayesian inference method is superior to the EM approach, but is intractable for large datasets.

Relational Push-Pull Models Finally, Anthony & desJardins (2007) present the Relational Push-Pull Model (RPPM), which assumes that edge existence depends both on the cluster assignments of the connected objects as well as the feature values of the related objects. In a similar manner as in Handcock, Raftery, & Tantrum (2007), this dual dependence allows the model to take advantage of features that tend to be correlated when relations are present. Anthony & desJardins presented some initial results on the IMDB data set in which a relation existed between two actors if they starred in a movie together. Their model was able to discover the phenomenon that experienced actors tend to act in movies with less experienced actors by discovering that two clusters—one with experienced actors and one with inexperienced actors—had a higher probability of linkage between each other, rather than within.

Discussion

The models discussed above all have a common result: they are able to extract additional information from the presence of relations that would not be present otherwise. Taskar, Segal, & Koller were able to use PRMs to discover a clustering where actors that frequently acted in the same genre were clustered together. Because Kemp *et al.* enforce the requirement that the clustering found using object relations must be the same clustering that groups objects in the feature relation set, the information found from the object relations contributes to the grouping of object features into categories. Neville & Jensen were able to identify underlying groups that represent the reason a relation is present. Kubica *et al.* succeeded in detecting when relations implied a personal relationship, as opposed to a coincidental meeting. Handcock, Raftery, & Tantrum (2007) found clusterings that successfully modeled the transitive relations between objects in a data set. Anthony & desJardins succeeded in detecting not only logical clusters, but also in identifying the reason that edges exist in the data based on the cluster parameter values and the common features between objects in the clusters.

The ability to harness the latent association between object features and object relations enables the use of these various models to discover interesting patterns in the data. Generative model approaches to clustering allow for the targeting of specific phenomena that a researcher feels is important. They are a good first approach because their design is simple and there are various standard learning techniques such as simulated annealing and Gibbs sampling that can be used to fit the model to a specific data set.

Generative models have certain drawbacks, however. First and foremost, they are limited in terms of speed and memory usage. The algorithms used for learning generative models can be very slow, depending on the number of parameters that must be learned. Even if the learning algorithm is fast, there is generally a restriction imposed by memory usage. Depending on the representation used for the data and the density of the relation set, the amount of memory used can grow quadratically with respect to the number of objects. This problem is compounded if there are different types of relations.

Open Problems

Despite the early successes in designing generative models for clustering, several problems remain. The first is the diversity of relational data. Traditional clustering algorithms make the assumption that data are independent and identically distributed (IID). This means that all of the objects are of the same type and no relations are observed between the objects. In a relational data set, this independence assumption can no longer be made for objects in the set. Furthermore, many relational data sets contain multiple types of objects, where an intuitive comparison is not readily available. Examples of hard-to-compare objects are actor–movie, person–city, and website–server. The first approach is to only cluster objects of the same type together, as Taskar, Segal, & Koller and Kemp *et al.* did. In their work, the user must perform a post-clustering analysis to observe the connection between actors and movies. Ideally, for example, it would be useful to have actors and movies clustered together. Such a method might also make it easier to find a place to live, given a person's features, or to decide which web server company to use, given a website's content.

A second open problem is how to handle relation dependence. As with any probabilistic approach to a problem, independence assumptions are often made to simplify the problem and speed up computation. However, it is frequently the case that relations are not independent. Consider three people A, B and C. If A is friends with B and C, it is much more likely that B and C are also friends. Such a phenomenon has already been captured by Handcock, Raftery,

& Tantrum (2007). However, there are many other kinds of relation dependence properties a data set could have. Guha *et al.* (2004) listed several cases in which a `trust` relation can be propagated in a data set, such as *co-citation* where if i trusts persons j and k, and m separately trusts j, the relation has a co-citation property if m also trusts k. In their research, they added additional links to a data set, to enrich a sparse trust matrix. However, the properties they list could be used as clustering criteria as well, where instead of computing the closure, the detection of such a situation as co-citation would lead to a better clustering.

Another interesting open problem that must be addressed is the sparseness of relations in real-world sets. Many data sources exhibit less than 10% edge density. With any approach, this can be limiting, but in the generative model approach, the consequences can be significant. Because the approach is to estimate distribution parameters given a sample, the goodness of fit for any model decreases with the size of the sample. Prior research in learning with unbalanced training sets, such as in the work by Kubat, Holte, & Matwin (1997) may be applicable to this problem.

One open problem that is particularly important to the social information processing community is how to handle the temporal nature of social information. Many data sources, such as friendship graphs or business networking graphs evolve over time. Both people and links between people appear and disappear over time. Most research has focused on taking a sample of the data from a particular point in time for analysis. It clearly would be useful to observe how relations change over time to help evaluate the relative significance of a relation between two people. Very little work has been done in this area so far, though Sharan & Neville (2007) have presented preliminary work on representing a temporal graph using a single weighted summary graph in which the weights imply the persistence of the edges. They used this summary graph as input to a supervised learning algorithm to predict the topic of papers in a citation data set. The performance increase was significant, indicating that it may be worthwhile to investigate other uses of the summary graph technique.

Conclusion

In this paper, we discussed the use of generative models for clustering social information. Such an approach allows for careful control of clustering criteria and analysis. The results from some recent generative models are interesting and encourage further research in the area. There are many open problems, the solution of which would result in good progress in social information processing. Clustering heterogeneous objects together would reduce the amount of post-processing that would be required if differently typed objects had to be clustered separately. Proper handling of relation dependence must be determined to advantageously harness the information found in latent dependencies in a data set. Finally, the algorithms need to be improved in terms of speed and memory, or discrete methods must be developed in order to process large social information data sets. Overall, clustering social information is a common task that is useful in many ways. Further progress in developing generative models will result in the discovery of useful clusterings in a variety of social information data sets.

References

Anthony, A., and desJardins, M. 2007. Data clustering with a relational push-pull model. In *Proceedings of the ICDM Workshop on Optimization-based Data Mining Techniques with Applications*.

Getoor, L.; Friedman, N.; Koller, D.; and Taskar, B. 2002. Learning probabilistic models of link structure. *Journal of Machine Learning Research* 3:679–707.

Guha, R.; Kumar, R.; Raghavan, P.; and Tomkins, A. 2004. Propagation of trust and distrust. In *International World Wide Web Conference (WWW2004)*.

Handcock, M. S.; Raftery, A. E.; and Tantrum, J. M. 2007. Model-based clustering for social networks. *Journal of the Royal Statistical Society* 170(2):1 – 22.

Hoff, P. D.; Raftery, A. E.; and Handcock, M. S. 2002. Latent space approaches to social network analysis. *Journal of the American Statistical Association* 97(460).

Kemp, C.; Tenenbaum, J. B.; Griffiths, T. L.; Yamada, T.; and Ueda, N. 2006. Learning systems of concepts with an infinite relational model. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence*, 381 – 388. Menlo Park, California: American Association for Artificial Intelligence.

Kubat, M.; Holte, R.; and Matwin, S. 1997. Learning when negative examples abound. In *European Conference on Machine Learning*, 146–153.

Kubica, J.; Moore, A.; Schneider, J.; and Yang, Y. 2002. Stochastic link and group detection. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence*, 798–804. AAAI Press/MIT Press.

MacQueen, J. B. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Symposium on Math, Statistics, and Probability*, 281 – 297.

Neville, J., and Jensen, D. 2006. Leveraging relational autocorrelation with latent group models. In *Proceedings of the Fifth IEEE International Conference on Data Mining*.

Nowicki, K., and Snijders, T. A. B. 2001. Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical Association* 96(455):1077 – 1087.

Sharan, U., and Neville, J. 2007. Exploiting time-varying relationships in statistical relational models. In *Proceedings of the 1st SNA-KDD Workshop, 13th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Taskar, B.; Segal, E.; and Koller, D. 2001. Probabilistic classification and clustering in relational data. In Nebel, B., ed., *Proceeding of IJCAI-01, 17th International Joint Conference on Artificial Intelligence*, 870–878.