Leveraging relational autocorrelation with latent group models

Jennifer Neville, David Jensen

Department of Computer Science, University of Massachusetts
Amherst, MA 01003
{jneville|jensen}@cs.umass.edu

Abstract. The presence of autocorrelation provides strong motivation for using relational techniques for learning and inference. Autocorrelation is a statistical dependency between the values of the same variable on related entities and is a nearly ubiquitous characteristic of relational data sets. Recent research has explored the use of *collective inference* techniques to exploit this phenomenon. These techniques achieve significant performance gains by modeling observed correlations among class labels of related instances, but the models fail to capture a frequent cause of autocorrelation—the presence of underlying groups that influence the attributes on a set of entities. We propose a latent group model (LGM) for relational data, which discovers and exploits the hidden structures responsible for the observed autocorrelation among class labels. Modeling the latent group structure improves model performance, increases inference efficiency, and enhances our understanding of the datasets. We evaluate performance on three relational classification tasks and show that LGM outperforms models that ignore latent group structure when there is little known information with which to seed inference.

Keywords. Statistical relational learning, probabilistic relational models, latent variable models, autocorrelation, collective inference

1 Introduction

Autocorrelation is a statistical dependency between the values of the same variable on related entities, which is a nearly ubiquitous characteristic of relational datasets. For example, hyperlinked web pages are more likely to share the same topic than randomly selected pages [1], and movies made by the same studio are more likely to have similar box-office returns than randomly selected movies [2]. More formally, autocorrelation is defined with respect to a set of related instance pairs $(z_i, z_j) \in Z$; it is the correlation between the values of a variable X on the instance pairs $(z_i.x, z_j.x)$.

Despite the challenges to learning, the presence of autocorrelation offers a unique opportunity to improve model performance, as autocorrelation enables inferences about one object to be used to improve inferences about related objects. Indeed, recent work in relational domains has shown that *collective inference* over an entire dataset results in more accurate predictions than conditional inference over each instance independently [3,1] and that the gains over conditional models widen as autocorrelation increases [4].

Collective inference techniques exploit autocorrelation by reasoning with *collective models* that represent the dependencies among class labels of related instances. Collective models have lower variance than models that represent autocorrelation dependencies indirectly through dependencies on other attributes of related instances [4]. Collective inference techniques use these low variance models to propagate information throughout the dataset and improve the overall set of predictions.

Modeling the correlations among class labels directly, however, fails to capture a frequent cause of autocorrelation—the presence of underlying groups, conditions, or events that influence the attributes on a set of entities. For example, in the cinematic domain, it is likely that studios *cause* the observed autocorrelation among movie returns. Movie-goers are unlikely to choose movies based on the success of other movies from the same studio. It is more likely that movie success is influenced by some unobserved properties of the studio (e.g., advertising budget). In this case, the class labels of movies are conditionally independent given studio *type* (e.g., high-budget studio). Latent-variable models that represent the correlation of unobserved group properties (e.g., studio type) with attribute values (e.g., movie returns) may be able to express density functions more accurately and compactly than approaches that directly model the observed autocorrelation.

When the groups are observable (e.g., movies made by the same studio), we can model the latent structure with a relatively simple application of the expectation-maximization (EM) algorithm. A similar approach is used in information retrieval where latent-unigram models represent each document as a *group* of word occurrences[1] that are conditionally independent given the *topic* of the document [5]. In many relational datasets, however, group membership is unobserved and must be inferred from the relations and attributes. For example, the World Wide Web contains *communities*—groups of hyperlinked pages with similar topics. Although we can not directly observe which community a web page belongs to, intra-community citations are more frequent than inter-community citations so hyperlinks are evidence of the underlying community structure. To continue the document retrieval metaphor, it is as if we have word occurrence information (attribute values) and noisy indicators of word co-occurrence within documents (link information) but we do not know the document boundaries or the topic distributions. In these situations, a joint model of attributes and links is needed to recover group membership and infer latent group properties.

In this work, we propose a *latent group model* (LGM) for relational data, which is a joint model of links, attributes, and groups for unipartite relational graphs. The model addresses a number of weaknesses of current collective models. First, collective models, which model autocorrelation dependencies directly, generally require computationally-complex, approximate inference techniques

[1] However, the "vocabulary" is much smaller in relational domains—we generally have less than 10 class values, whereas documents have thousands of unique words.

because inference is over a large, cyclic graphical model. If, however, the instances are conditionally independent given the underlying group structure, then exact inference is not only feasible but much more efficient. Second, collective models typically restrict their representation to the dependencies among instances that are directly linked in the data. Linkage is generally sparse[2], so this restriction constrains the space of possible dependencies to a reasonable size and prevents useful information from being drowned out by noise. However, modeling the dependencies among neighboring but unlinked instances (e.g., transitive relationships) allows information to propagate in a more elaborate manner during inference. Group models are a natural way to extend the representation to improve model performance without fully representing the $O(N^2)$ dependencies between all pairs of instances. Finally, latent-variable models recover the underlying groups and identify their associated density functions. This attempt to model the true *cause* of autocorrelation will improve domain understanding and motivate development of additional modeling techniques.

Our initial evaluation of LGMs are on out-of-sample classification tasks. More specifically, we aim to learn LGMs of datasets where the attributes and links are fully observed, and group structure is unobserved, and then apply the model to classify instances in new datasets, where the attributes are partially observed, links are fully observed and the groups are again unobserved. This approach is suited for domains with large, nearly disconnected graph structures. For example, in gene prediction tasks, models of proteins and how they interact to perform certain functions in the cell can be learned in one genome and then applied to classify the proteins in new genomes. In addition, this approach is suited for dynamic network domains, where groups emerge and and disband over time. For example, fraud detection efforts usually analyze a single dataset that is evolving over time. The data contain demographic information about individuals and transactional links (e.g., bank deposits, telephone calls) that can indicate the underlying organizations. In these domains, LGMs could be used to detect group formation and use a few hand-labeled examples to seed inference about the classifications of new group members.

In the remainder of the paper, we outline LGM, our initial algorithms for learning and inference, and related work in statistical relational learning. We present empirical evaluation on three classification tasks to demonstrate the capabilities of the model, showing that LGMs perform better than models that ignore latent groups when there is little known information with which to seed inference. Finally, we conclude with directions for future work.

2 Latent Group Models

Latent group models specify a generative probabilistic model for the attributes and link structure of a relational dataset. The model posits groups of objects in the data of various type. Membership in these groups influences the observed

[2] In the datasets we have analyzed, existing links account for 1-10% of the $O(N^2)$ possible dependencies.

attributes of objects, as well as the existence of relationships (links) among objects.

For this initial investigation of LGMs, we make several simplifying assumptions about the data. More specifically, we assume a unipartite relational data graph (single object type) with binary, undirected links, and at most one link between any pair of objects. We also assume the number of objects, groups, and group types are fixed and known. However, it is relatively straightforward to extend the model to accommodate deviations from these assumptions.

The model assumes the following generative process for a dataset with N_O objects and N_G groups:

1. For each group g, $1 \leq g \leq N_G$:
   (a) Choose a value for group type t_g from $p(T)$, a multinomial probability distribution with k values.
2. For each object o, $1 \leq o \leq N_O$:
   (a) Choose a group g_o uniformly from the range $[1, N_G]$.
   (b) For each attribute $A \in \mathbf{A_M}$:
      i. Choose a value for a_o from $p(A|G,T)$, a multinomial probability conditioned on the object's group type t_{g_o}.
3. For each object i, $1 \leq i \leq N_O$:
   (a) For each object j, $i < j \leq N_O$:
      i. Choose a value for e_{ij} from $p(E|G_i = G_j, T_i, T_j)$, a Bernoulli probability conditioned on the two objects' groups types and whether they are in the same group.

This generative model specifies that attribute values and link existence are conditionally independent given group membership and type information. More specifically, the class labels of objects are conditionally independent.

The joint distribution of the dataset D, with groups N_G, objects N_O, and links L, is thus given by:

$$p(D) = \prod_{g \in N_G} p(t_g) \prod_{A \in \mathbf{A}} \prod_{o \in N_O} p(a_o|g_o, t_{g_o}) \cdot$$
$$\prod_{l_{ij} \in L} p(e_{ij}{=}1|g_i{=}g_j, t_{g_i}, t_{g_j}) \prod_{l_{ij} \notin L} p(e_{ij}{=}0|g_i{=}g_j, t_{g_i}, t_{g_j})$$

See figure 1 for a graphical representation of LGM. The model is similar to hierarchical Bayesian models but extended to a relational domain where the generative process is responsible for generating both attributes and links. More specifically the model is a form of probabilistic relational model (PRM) [6] that combines a directed relational Bayesian network, link existence uncertainty, and hierarchical latent variables.

2.1 Learning

Learning an LGM consists of learning the parameters of the distribution and inferring the latent variables on both objects (group membership) and groups

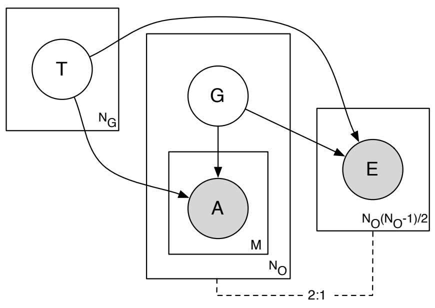

Fig. 1. LGM graphical model representation. The plates represent replicates: N_G groups; N_O objects, each with M attributes; and $N_O(N_O-1)/2$ possible binary links. The shaded nodes are observed variables: A are object attributes, E is a binary variable indicating link existence. The unshaded nodes are unobserved variables: T is the group's type, G is the object's group membership. The dashed line indicates the underlying link relationships in the data, which each involve two objects; consequently each E will be influenced by two G and two T variables in the unrolled Bayes net.

(group type). Ideally, we could learn the model using a straightforward application of the EM algorithm—iterating between inferring the latent variables (E-step) and estimating the parameters (M-step). Unfortunately, there are difficulties with this approach. First, there are N_O latent group variables with N_G possible values and N_G latent type variables with k possible values. When the average group size is small ($N_G = O(N_O)$), we expect that EM will be very sensitive to the start state. Furthermore, the E-step requires that we run inference over a large, complex, rolled-out Bayes net with $2N_O|A_M| + 4N_O(N_O-1)/2$ edges, where objects' group memberships are all interdependent (given the link observations). Exact inference in this situation is impractical, although approximate inference techniques such as loopy belief propagation or variational methods may allow accurate inference. However, given the number of latent variables and their dependency on sparse link information ($L \ll N_O(N_O-1)/2$), the likelihood function will have many local (suboptimal) maxima and we expect that EM will not converge to a reasonable solution.

Because collective models propagate information only on existing links, we expect the autocorrelated groups will have more intra-group links than inter-groups links. We exploit this characteristic to decouple the group discovery from the remainder of the estimation process and propose the following approximate learning algorithm:

1. Hypothesize group membership for objects based on the observed link structure alone.
2. Use EM to infer group types and estimate the remaining parameters of the model.

A hard clustering approach, which assigns each object to a single group, greatly simplifies the estimation problem—we only need to estimate the latent group type variables and parameters of $p(T)$ and $p(A|G,T)$. To this end, we employ a recursive spectral decompostion algorithm with a norm-cut objective function [7] to cluster the objects into groups with high inter-group and low intra-group linkage. Our approach appears to work well in practice, but refinements that iterate the clustering and EM steps, or incorporate soft clusterings, may improve results even further.

2.2 Inference

There are two ways to apply LGMs to relational data. First, given a dataset with observed attributes and links, we can use the model to cluster objects into groups with similar attribute values and patterns of linkage. Second, we can apply the model to a new unseen dataset with partially observed attributes and/or links to infer both the group memberships and the unobserved attributes/links. This approach exploits the underlying group structure to improve predictions by jointly inferring the latent groups, their types, and the unknown attributes/links. Our initial investigation of LGMs has focused primarily on this latter task to enable an objective comparison of LGMs with current, alternative techniques. In the experiments reported below, we learn the model on a dataset with fully observed attributes and links, then we apply the model to a new dataset with partially observed attributes and fully observed links to jointly infer the group memberships and unknown attributes. We designed our learning algorithm with this out-of-sample classification task in mind. When classifying a new dataset with partially observed attributes, we can cluster the objects into groups using the observed links and then use the learned model to jointly infer the group types and the unobserved attributes.

3 Related Work

There are two types of statistical relational models related to LGMs: models that represent joint distributions of attributes conditioned on link structure, and models that cluster objects into groups based on link and attribute structure.

3.1 Joint models

The first category consists of models that represent a joint distribution of the attributes of set of instances. Relational Markov networks (RMNs) [1] and relational dependency networks (RDNs) [8] model autocorrelation in a procedural fashion. RMNs use clique templates to model the pairwise correlations among class labels of related instances, whereas RDNs use aggregated features. These techniques model the autocorrelation dependencies at a global level—the autocorrelation dependencies are assumed to be uniform across each link in the data and parameters of features are tied across the entire dataset. As such, these models will not be able to distinguish among regions with varying levels of autocorrelation.

Probabilistic relational models (PRMs) [6] are also able to model autocorrelation relationships, but only if the autocorrelation can be structured to be acyclic (e.g., with temporal constraints). The use of latent variables in PRMs has been explored in limited settings where the groups are known and represented as objects in the data [9]. For example, in the cinematic domain we could use a PRM with a latent variable on studios to model the autocorrelation of movie returns. However, latent groups are not posited by the model and group variables are not conditioned on the observed link structure.

3.2 Cluster models

The second category consists of models that cluster relational data into groups. Popescul and Ungar [10] cluster relational database tables and use the cluster IDs as features in *individual* classification models that reason about each instance independently. This approach has been shown to improve classification performance, but it can only be employed in situations where the test set instances link into the clusters used during training because the features use the identity of the clusters rather than generalizing over the properties of the groups.

Kubica, Moore, Schneider and Yang [11] use a latent variable model that is a special case our proposed model to cluster objects into groups based on their attribute values and link structure. Their approach is geared toward clustering data with multiple transactional links (e.g., phone calls, email and meetings) where the links patterns are homogeneous with respect to the groups. In other words, it is assumed that all groups have the same distribution of intra- and inter-group linkage. A situation where the patterns of linkage differ among groups is, however, easy to imagine. For example, consider machine learning papers: Reinforcement learning papers tend to cite papers in optimization, operations research, and theory, but genetic algorithm papers cite primarily other genetic algorithm papers. Allowing the link probabilities to vary among groups will be important for modeling group structures in large heterogeneous domains.

3.3 Discussion

Consider the case where there are k group types, C class values, and each object has a latent variable. There is a spectrum of group models ranging from $k=C$ to $k=N_G$. RMNs can reason at one end of the spectrum ($k=C$) by tying the parameters across all links and creating a feature for each class. Techniques that cluster the data for features to use in conditional models (e.g., [10]), can reason at the other end of the spectrum ($k=N_G$) by using the identity of each cluster. The approach of [11] uses $k=1$ in the sense that it ties the parameters of intra- and inter-group link probabilities across all groups.

When group size is large there may be enough data to reason about each group independently, setting $k = N_G$. For example, in the cinematic domain, once a studio has made a sufficient number of movies, we can reason about the likely returns of its next movie independent of the rest of the data. However, when group size is small, assuming that all groups are drawn from the same distribution, and setting $k = C$, will offset the limited data available for each group. A model that can smoothly vary k can be thought of as a backoff model, where we smooth to the background signal when we don't have enough data to estimate about a group's type in isolation. LGMs offer a principled framework within which to explore this spectrum.

One of the primary advantages of LGMs is that influence can propagate between pairs of objects that are not directly linked but are close in graph space (e.g., in the same group). In RMNs and RDNs, the features of an object specify its Markov blanket. This limits influence propagation because features are generally constructed over the attributes of objects one or at most two links away in the data. Influence can only propagate farther by influencing the probability estimates of attribute values on each object in a path sequentially. An obvious way to address this issue is to model the $O(N_O^2)$ dependencies among all pairs of objects in the data, but dataset size and sparse link information makes this approach infeasible for most datasets. PRMs with existence uncertainty [12] are the only current models that consider the full range of dependencies and their influence on observed attributes. LGMs are a natural way to expand current representations while limiting the number of dependencies to model. LGMs can aggregate influence over a local neighborhood, instead of only passing on autocorrelation information through changes to the probability distributions of each object in a path sequentially.

4 Experimental Results

The experiments in this section demonstrate the utility of latent group models in relational domains. Using three classification tasks, we evaluate whether the models can leverage autocorrelation to improve model accuracy and illustrate the conditions under which the models will perform well.

We present results for two variations of LGM. The first variation, LGM-k, sets the number of group types to the number of class label values, $k = C$ (e.g., for binary tasks, $k = 2$); the second variation, LGM-2k, sets $k = 2C$. We compare the LGM to four alternative models. The first two are individual classification models that reason about each instance independently and do not use the class labels of related instances: the relational probability tree (RPT) model [13] is a decision tree model and the relational Bayesian classifier (RBC) model [14] is a naive Bayes model. The third model is a relational dependency network (RDN) [8] that reasons about networks of instances collectively. The fourth model (RDN-ceil) is a probabilistic ceiling for the RDN model, where we allow the true labels of related instances to be used during inference. This model shows the level of performance possible if the RDN model could infer the true labels of related instances with perfect accuracy.

To limit the confounding effects of feature construction and model selection, we consider the restricted task of predicting class labels using only the class labels of related instances and/or the group membership. For the RPT and RBC models, we clustered the training and test sets together and used cluster ID as the sole attribute in the model. The performance of these baseline models illustrates the baseline utility of clustering without typing the groups and serves as a comparison to previous work [10], which clusters the data to generate additional features for classification. For the LGM, RDN and RDN-ceil, we used the class label of related instances as the sole attribute available for modeling. When possible, we used exact inference but RDNs require approximate inference techniques. In the RDN experiments, we used Gibbs with chains of length 500 and burn-in of 100. (At this length, acccuracy and AUC had converged.) During inference we varied the number of known class labels available to seed the inference process. We expect performance to be similar when other information serves to seed the inference process—either when some labels can be inferred from intrinsic attributes, or when weak predictions about many related instances serve to constrain the system.

4.1 Data and Tasks

The first data set is drawn from the Internet Movie Database (www.imdb.com). We used a sample of 1,382 movies released in the U.S. between 1996 and 2001. The binary classification task was to predict movie opening weekend returns ($>\$2million$). Based on past work that showed movie receipts to be autocorrelated through studios [2], we considered a unipartite graph of movies, where links indicate movies that are made by a common studio.

The second data set is drawn from Cora, a database of computer science research papers extracted automatically from the Web using machine learning techniques [15]. We considered the unipartite co-citation graph of 4,330 machine-learning papers. The classification task was to predict paper topic. There are seven topics including *Neural Networks* and *Reinforcement Learning.*

The third data set was collected by the WebKB Project [16]. The data consist of a set of 3,877 web pages from four computer science departments, manually labeled with the categories: course, faculty, staff, student, research project, or other. We considered the unipartite co-citation web graph. The classification task was to predict page category. As in previous work on this dataset, we do not try to predict the category *Other*; we remove them from the data after creating the co-citation graph.

4.2 Results

Figure 2a-c shows area under the ROC curve (AUC) results for each of the models on the three classification tasks[3]. The graph shows AUC for the most prevalent class, averaged over 4-5 training/test splits. For IMDb and Cora, we used temporal samples where we learned the model on one year and applied the model to the subsequent year. For WebKB, we used cross-validation by

[3] Accuracy results, over all classes, show similar behavior.

department, learning on three departments and testing on the fourth. For each training/test split we ran 10 trials at each level of labeling, except the RDNs where we ran 5 trials due to relative inefficiency of RDN inference. The error bars indicate the standard error of the AUC estimates for a single training/test split, averaged across the training/test splits. This illustrates the variability of performance within a particular sample, given the random initial labeling.

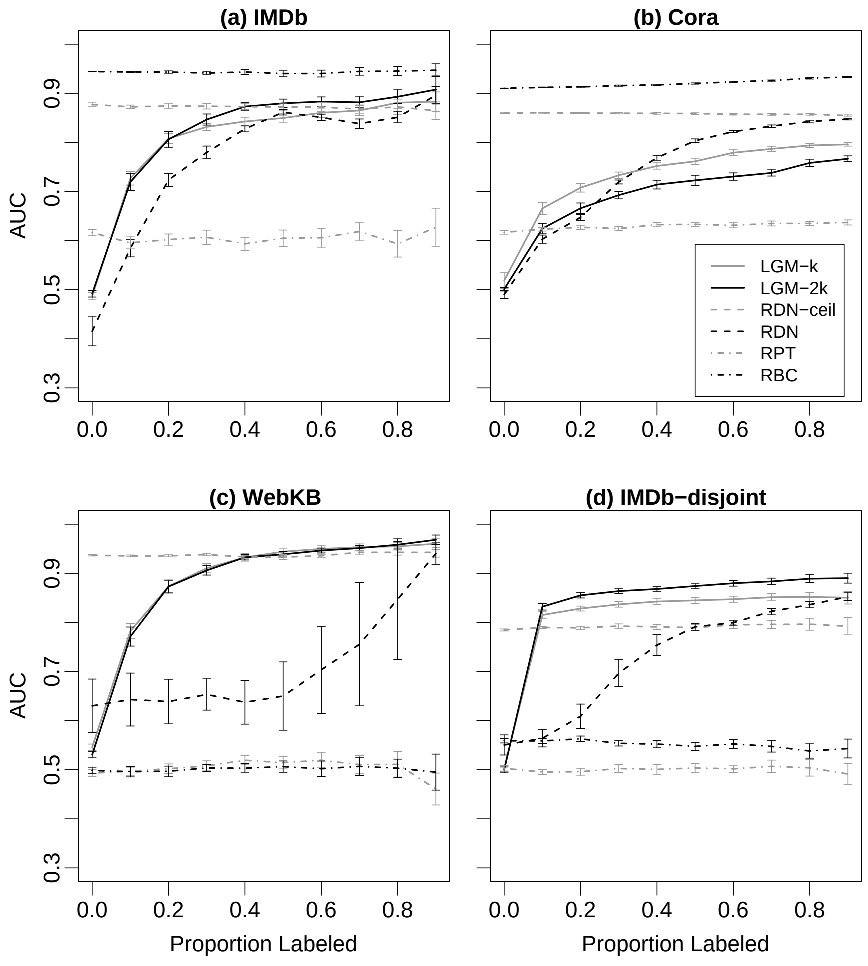

Fig. 2. Model performance as the proportion of labeled instances during inference is varied.

The results show that LGM performance quickly reaches performance levels comparable to RDN-ceil. (Note that RDNs and LGMs cannot be expected to do better than random at 0% labeled.) On IMDb and WebKB, LGM performance asymptotes at less than 40% known labels, indicating that the model is able to exploit group structure when there is enough information to accurately infer the group type. The RDN doesn't converge as quickly to the ceiling level of performance. There are two explanations for this effect. First, when there are few constraints on the labeling space (e.g., fewer known labels), RDN inference may not be able to fully explore the space of labelings. Although we saw performance plateau for Gibbs chains of length 500-2000, it is possible that longer chains, or alternative inference techniques, could further improve RDN performance. The second explanation is that joint models are disadvantaged by the data's sparse linkage. When there are few labeled instances, influence may not be able to propagate to distant objects over the existing links in the data. A group model that allow influences to propagate in more elaborate ways may be able to exploit the seed information more successfully. Future work will attempt to quantify the amount of error due to each of these sources.

LGM performance does not reach that of RDN-ceil in Cora. Although the LGM outperforms the RDN when there is little know information, eventually the RDN takes over as it converges to RDN-ceil performance. We conjecture that this effect is due to the quality of the clusters recovered in Cora. The distribution of cluster sizes has high variance—ranging from large clusters with more than 100 papers to small clusters with 2-3 papers. LGM is not able to model papers in the large clusters as accurately as the RDN because the distribution of class labels within these groups is too uniform. The clustering technique has likely conflated several groups and returned them as one cluster when they should be partitioned. A technique the iterates over clustering and model estimation may be more robust in this situation.

RPT performance is near random on all three datasets. This is because the RPT algorithm uses feature selection and there is significant correlation between only a few cluster IDs and the class label. This indicates that there is little evidence to support generalization about cluster identities themselves. The RBC, on the other hand, does not do any feature selection, it simply uses cluster IDs without regard to their support in data. On IMDb and Cora, the RBC significantly outperforms all other models. However, these are the two samples where the test set instances link into the training set. In the third dataset, where the training and test sets are nearly disjoint, the RBC does no better than random. To further explore this effect, we partitioned the IMDb data into five disjoint training/test splits using snowball sampling of movies through studios. Figure 2d graphs performance on this view of the IMDb. RBC performance drops to random when the training and test sets are nearly disjoint.

For a subjective evaluation of the clustering abilities of LGM, table 1 lists the studios associated with the IMDb clusters. We group the clusters by their (inferred) type values and present a sample of the associated studios and the estimated probability distribution of movie returns.

Table 1. Studios associated with IMDb groups

Group type 1 (6 clusters): $P(returns > \$2mil \mid G) = 0.90$ Paramount Pictures, Universal Pictures, Buena Vista Pictures, New Line Cinema, DreamWorks, MGM
Group type 2 (24 clusters): $P(returns > \$2mil \mid G) = 0.59$ Columbia Pictures, Warner Bros., 20th Century Fox, Destination Films, Lot 47 Films, Margin Films
Group type 3 (7 clusters): $P(returns > \$2mil \mid G) = 0.30$ Artisan Entertainment, Miramax, Gramercy Pictures, Sony Pictures, RCV Film Distribution, United Artists, Trimark Pictures
Group type 4 (32 clusters): $P(returns > \$2mil \mid G) = 1.5e{-4}$ Seventh Art Releasing, Strand Releasing, Zeitgeist Films, October Films, The Shooting Gallery, Curb Entertainment

5 Discussion and Conclusions

This paper presents a latent group model that reasons jointly about attribute information and link structure to improve reasoning in relational domains. To date, work on statistical relational models has focused on models of attributes conditioned on the link structure (e.g., [1]), or on models of link structure conditioned on the attributes (e.g., [12]). These restrictions to the model space make learning and inference more tractable but limit the manner in which influence can propagate in the data. However, as our initial investigation has shown, modeling the interaction among links and attributes promises to improve model generalization and interpretability.

Latent group models are a natural means to model the attribute and link structure simultaneously. The groups decouple the link and attribute structure, thereby offering a way to learn joint models tractably. Preliminary investigations of latent variable models in the social networks community for link prediction [17,18], and in the relational learning community for clustering [11] and augmentation of classification models [9] show promise. However, the power and range of applicability of these models is yet to be fully explored. Our analysis has shown that group models outperform collective models when there is little information to seed inference. This is likely because a smaller amount of information in needed to infer group type than is needed to propagate information throughout sparse relational graphs. This suggests *active inference* as an interesting new research direction—where techniques choose which instances to label based on estimated improvement to the collective predictions.

Latent group models extend the manner in which collective models exploit autocorrelation to improve model performance. One of the reasons collective inference approaches work is that the class labels are at the "right" level of abstraction—they *summarize* the attribute information that is relevant to related objects. Group models also summarize the information but at higher level of abstraction (e.g., group membership and type). Positing the existence of groups decouples the search space into a set of biased abstractions and could be considered a form of predicate invention [19]. This allows the model to consider a wider range of dependencies to reduce bias while limiting potential increases in variance and promises to unleash the full power of statistical relational models. Indeed, the results we report for LGMs use only the class labels and the link information but they achieve nearly the same level of performance reported by relational models in the recent literature.

References

1. Taskar, B., Abbeel, P., Koller, D.: Discriminative probabilistic models for relational data. In: Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence. (2002) 485–492
2. Jensen, D., Neville, J.: Linkage and autocorrelation cause feature selection bias in relational learning. In: Proceedings of the 19th International Conference on Machine Learning. (2002) 259–266
3. Chakrabarti, S., Dom, B., Indyk, P.: Enhanced hypertext categorization using hyperlinks. In: Proceedings of the ACM SIGMOD International Conference on Management of Data. (1998) 307–318
4. Jensen, D., Neville, J., Gallagher, B.: Why collective inference improves relational classification. In: Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. (2004) 593–598
5. Blei, D., Ng, A., Jordan, M.: Latent dirichlet allocation. Journal of Machine Learning Research **3** (2003) 993–1022
6. Getoor, L., Friedman, N., Koller, D., Pfeffer, A.: Learning probabilistic relational models. In: Relational Data Mining. Springer-Verlag (2001)
7. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **22(8)** (2000) 888–905
8. Neville, J., Jensen, D.: Dependency networks for relational data. In: Proceedings of the 4th IEEE International Conference on Data Mining. (2004) 170–177
9. Taskar, B., Segal, E., Koller, D.: Probabilistic classification and clustering in relational data. In: Proceedings of the 17th International Joint Conference on Artificial Intelligence. (2001) 870–878
10. Popescul, A., Ungar, L.: Cluster-based concept invention for statistical relational learning. In: Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. (2004) 665–670
11. Kubica, J., Moore, A., Schneider, J., Yang, Y.: Stochastic link and group detection. In: Proceedings of AAAI/IAAI 2002. (2002) 798–806
12. Getoor, L., Friedman, N., Koller, D., Taskar, B.: Learning probabilistic models of link structure. Journal of Machine Learning Research **3** (2002) 679–707
13. Neville, J., Jensen, D., Friedland, L., Hay, M.: Learning relational probability trees. In: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. (2003) 625–630
14. Neville, J., Jensen, D., Gallagher, B.: Simple estimators for relational bayesian classifers. In: Proceedings of the 3rd IEEE International Conference on Data Mining. (2003) 609–612

15. McCallum, A., Nigam, K., Rennie, J., Seymore, K.: A machine learning approach to building domain-specific search engines. In: Proceedings of the 16th International Joint Conference on Artificial Intelligence. (1999) 662–667
16. Craven, M., DiPasquo, D., Freitag, D., McCallum, A., Mitchell, T., Nigam, K., Slattery, S.: Learning to extract symbolic knowledge from the world wide web. In: Proceedings of the 15th National Conference on Artificial Intelligence. (1998) 509–516
17. Nowicki, K., Snijders, T.: Estimation and prediction for stochastic blockstructures. Journal of the American Statistical Association **96** (2001) 1077–1087
18. Hoff, P., Raftery, A., Handcock, M.: Latent space approaches to social network analysis. Journal of the American Statistical Association **97** (2002) 1090–1098
19. Stahl, I.: Predicate invention in inductive logic programming. In: Advances in Inductive Logic Programming. (1996) 34–47