Structure Learning of Probabilistic Relational Models from Incomplete Relational Data

Xiao-Lin Li and Zhi-Hua Zhou

National Key Laboratory for Novel Software Technology
Nanjing University, Nanjing 210093, China
{lixl,zhouzh}@lamda.nju.edu.cn

Abstract. Existing relational learning approaches usually work on complete relational data, but real-world data are often incomplete. This paper proposes the MGDA approach to learn structures of probabilistic relational model (PRM) from incomplete relational data. The missing values are filled in randomly at first, and a maximum likelihood tree (MLT) is generated from the complete data sample. Then, Gibbs sampling is combined with MLT to modify the data and regulate MLT iteratively for obtaining a well-completed data set. Finally, probabilistic structure is learned through dependency analysis from the completed data set. Experiments show that the MGDA approach can learn good structures from incomplete relational data.

1 Introduction

Most machine learning algorithms work with the attribute-value setting which only allows the analysis of fairly simple objects described by a single table. To deal with complex and structured objects, one choice is to employ a relational structure which involves multiple tables. Thus, each complex object can be described by multiple records in multiple tables. To be able to analyze relational databases containing multiple relations properly, learning algorithms have to be designed for coping with the structural information in relational databases [8].

Relational learning has a precursor going back over a decade in the field of inductive logic programming (ILP) [18]. One of the most significant developments in recent years is the convergence of ILP and probabilistic reasoning and learning). ILP endows the ability of handling multiple relations; probabilistic methods endow the ability of handling uncertainty. Many approaches containing those ingredients have been proposed [19,11,1,15,16,9,21,5].

It is noteworthy that most relational learning algorithms operate on complete data, while real-world data are often incomplete, i.e., with missing attribute values. Although learning with incomplete data has been studied in attribute-value setting [6,12], few techniques can be directly applied to relational setting since the case of incomplete relational data is substantially more complex. An attribute-value learning algorithm can be seen as a relational learning algorithm which only deals with a database containing a single table. It will be computationally more expensive and the result will be much worse if such algorithms

J.N. Kok et al. (Eds.): ECML 2007, LNAI 4701, pp. 214–225, 2007.

are applied to incomplete relational data directly since there exist more poor local maxima. Actually, learning from incomplete relational data is a challenging problem in relational learning.

This paper proposes the MGDA (Maximum likelihood tree and Gibbs sampling-based Dependency Analysis) approach to learn structures of probabilistic relational models from incomplete relational data. Firstly, we fill in the incomplete relational data randomly. Then, we generate a maximum likelihood tree (MLT) [4] from the completed data sample. After that, Gibbs sampling is combined with MLT to modify the data and regulate MLT iteratively for obtaining a well-completed data set. Finally, probabilistic structure is learned through dependency analysis from the completed data set.

The rest of this paper is organized as follows. Section 2 briefly introduces the research background. Section 3 describes the proposed method. Section 4 reports on the experiments. Finally, section 5 concludes.

2 Background

Probabilistic relational model (PRM) [11] is one of the fundamental models in relational learning, which extends the standard attribute-value-based Bayesian network representation to incorporate a richer relational structure. Briefly, given a relational database of a specific schema (or a set of instances and relations between them), a PRM defines a probability distribution which specifies probabilistic dependencies between related objects (or the attributes of the instances).

Definition. [11] A PRM for a relational schema σ is defined as follows. For each entity type X and each propositional attribute $X.A$,

- A set of parents $Pa(X.A) = \{Pa_1, Pa_2, \cdots, Pa_n\}$, where each Pa_i has the form $X.B$ or $\gamma(X.\tau.B)$. τ is a chain of relations and $\gamma(\cdot)$ is an aggregation function.
- A conditional probability model for $P\left(X.A|Pa\left(X.A\right)\right)$.

The probability distribution over complete instantiation $\mathcal{L}$ of σ represented by the PRM is given by:

$$P\left(\mathcal{L}|\sigma, \mathcal{S}, \theta_{\mathcal{S}}\right) = \prod\nolimits_{X_i} \prod\nolimits_{A \in \mathcal{A}(X_i)} \prod\nolimits_{x \in \mathcal{O}^{\sigma}(X_i)} P(\mathcal{L}_{x_i.a}|\mathcal{L}_{Pa(x_i.a)})$$

As indicated by [11], the task of learning a PRM from complete data has two aspects, i.e., parameter estimation and structure learning. In parameter estimation, the input consists of the schema and training database, as well as a qualitative dependency structure. In structure learning, there is only the training database as input, while the goal is to extract an entire PRM schema from the training database automatically.

Obviously, structure learning is the key of learning a PRM. There are two general approaches to graphical probabilistic model learning, i.e., the *search & scoring* approaches and the *dependency analysis* approaches. The main difficulty in the first kind of approaches is how to find out a good dependency structure from the many possible ones, which are potentially infinite. For Bayesian networks, the task of finding the highest scoring network is NP-hard [3]. PRM learning is at least as hard as Bayesian network learning. The second kind of approaches, i.e. dependency analysis approaches, try to discover the dependency relationship from the data, and then use these dependencies to infer the structure. The approach proposed in this paper belongs to this category.

Many applications and extensions of PRM have been described. Getoor and Sahami [13] applied PRM to collaborative filtering. Getoor *et al.* [14] applied PRM to hypertext classification. Taskers *et al.* [23] proposed a general class of models for classification and clustering based on PRM. They have considered incomplete data for parameter estimation, but have not touched structure learning. Sanghai *et al.* [22] extended dynamic Bayesian networks where each time slice is represented by a PRM. To the best of our knowledge, structure learning of PRM from incomplete relational data has only been studied recently [17], where an evolutionary algorithm is used and the PRM structure is learned by filling in the missing data with the best evolved structure in each generation.

In traditional attribute-value setting, learning from incomplete data has been studied by many researchers. In particular, approaches for learning Bayesian networks from incomplete data require an approximation for incomplete data. One kind of approaches is based on Monte-Carlo or sampling [12]. These approximations can be very accurate, but these approaches are often intractable when the sample size is very large. Another kind of approaches is based on the expectation-maximization (EM) algorithm [6]. EM algorithm can be powerful for parameter estimation in relational learning with missing data, but for structure learning, since the number of possible structures to be explored is too huge, in the E step of EM it would be difficult to efficiently determine how to modify the current structure. It has been noted [10] that such algorithms often get stuck in local maxima when the search landscape is large and multi-modal.

3 The MGDA Approach

The MGDA approach is summarized in Table 1, which will be introduced step-by-step in this section.

3.1 Initialization

Incomplete data make the dependency relationship between attributes more disordered and it is difficult to learn an creditable PRM structure directly. If the incomplete data can be filled in accurately, then the fittest structure can be learned. Standard Gibbs sampling conducts sampling from full conditional distribution. As the conditional set is with high dimensionality, the complexity is exponential in the number of attributes. It will be computationally expensive if standard Gibbs sampling is extended to relational learning directly. So, an improved approach needs to be proposed.

Table 1. The MGDA approach

1. Fill in the incomplete relational data randomly and generate an MLT from the obtained complete data set (details in Section 3.1);
2. Repeat until the stop criterion is satisfied:
   a) Modify the incomplete relational data (details in Section 3.2.1);
   b) Modify the corresponding parameter according to the latest modified data set (details in Section 3.2.2);
   c) Regulate the MLT structure according to the completed relational data and test the stop criterion (details in Section 3.2.3);
3. Regulate the PRM structure learned from the well-completed data set by using the proposed dependency analysis approach (details in Section 3.3).

Here we combine Gibbs sampling with MLT to modify the incomplete data and regulate MLT iteratively for obtaining a well-completed data set. The incomplete relational data are filled in randomly at first. Then, an MLT is generated from the completed data set. MLT is the fittest tree-like structure of Bayesian network, which has a simple structure. Chow and Liu [4] proposed a well-known method for learning tree-like Bayesian networks, which reduces the problem of constructing an MLT to the finding of a maximal weighted spanning tree. We extend this procedure on relational conditions as follows:

1. Compute $I(X_i.A; X_j.B)$ between each pair of attributes ($A \neq B$), where
$$I(X_i.A; X_j.B) = \sum\nolimits_{\substack{X_i,\\ X_j}} \sum\nolimits_{\substack{A\in A(X_i),\\ B\in B(X_j)}} \sum\nolimits_{\substack{x_i.a,\\ x_j.b}} P(x_i.a, x_j.b) \log \frac{P(x_i.a, x_j.b)}{P(x_i.a)P(x_j.b)};$$
2. Build a complete undirected graph where the weight of the edge connecting $X_i.A$ to $X_j.B$ is $I(X_i.A; X_j.B)$;
3. Choose a root attribute for the tree and set the direction of all edges to be outward from it.

Then, we use the learned MLT structure to decompose the joint probability. This process can convert the sampling from n-order full conditional probability to second-order conditional probability since each attribute of an MLT has only one parent at most. So, it can not only meet the requirement of full conditional distribution in standard Gibbs sampling but also reduce the computational cost. After modifying the incomplete data, a new MLT can be generated.

3.2 Modification and Regulation

There are three tasks in each iteration, i.e., modifying the incomplete relational data, modifying the parameters, and regulating the MLT structure. The order of sampling attributes with incomplete data is based on the order of nodes (attributes) of the learned MLT structure and the order of records of the data set. Assume that $X_i.A$ has a missing entry in the mth record, which is denoted by $x_{im}.a$ and the modified value is denoted by $\hat{x}_{im}.a$. The possible values of $X_i.A$ are $x_i^1.a, \cdots, x_i^r.a$. MGDA uses the latest modified data set to modify the next missing attribute value.

Modifying the Incomplete Relational Data. If $P(L|\sigma, S, \theta_s)$ contains non-zero probabilities, MGDA modifies the missing data by Gibbs sampling. For a random λ, the value of $X_i.A$ is:

$$\hat{x}_{im}.a = \begin{cases} x_i^1.a,\ 0 < \lambda \leq \hat{p}\left(L_{x_{i.a}^1} | L_{P_a(x_{im}.a)}\right) \\ \cdots \quad \cdots \\ x_i^h.a,\ \sum_{j=1}^{h-1} \hat{p}\left(L_{x_{i.a}^j} | L_{P_a(x_{im}.a)}\right) < \lambda \leq \sum_{j=1}^{h} \hat{p}\left(L_{x_{i.a}^j} | L_{P_a(x_{im}.a)}\right) \\ \cdots \quad \cdots \\ x_i^r.a,\ \lambda > \sum_{j=1}^{r-1} \hat{p}\left(L_{x_{i.a}^j} | L_{P_a(x_{im}.a)}\right) \end{cases}$$

Modifying the Parameters. If $P(L|\sigma, S, \theta_s)$ contains zero probabilities, i.e. $\hat{p}\left(L_{x_{i.a}^u} | L_{P_a(x_{im}.a)}\right) = 0$, MGDA modifies the probabilities by using Laplacian-correction [7]:

$$\hat{p}\left(L_{x_{i.a}^u} | L_{P_a(x_{im}.a)}\right) = \frac{1}{N\|P_a(x_{im}.a)\| + \|x_i^u.a\|}$$

where N is the number of records, $\|x_i^u.a\|$ is the number of the values of attribute $X_i.A$, and $\|P_a(x_{im}.a)\|$ is the number of the parent combination of $Pa(X_i.A)$.

If $x_{im}.a \neq \hat{x}_{im}.a$, the corresponding parameters are modified as follows:

$$\hat{p}\left(L_{\hat{x}_{im}.a} | L_{P_a(x_{im}.a)}, \hat{D}_m\right) = \hat{p}\left(L_{\hat{x}_{im}.a} | L_{P_a(x_{im}.a)}, D_m\right) + 1/N$$

$$\hat{p}\left(L_{x_{im}.a} | L_{P_a(x_{im}.a)}, \hat{D}_m\right) = \hat{p}\left(L_{x_{im}.a} | L_{P_a(x_{im}.a)}, D_m\right) - 1/N$$

where D_m and $\hat{D}_m$ are respectively the database before and after modifying $x_{im}.a$.

Regulating the MLT Structure. After modifying the incomplete relational data, MGDA generates a new MLT structure from the completed data set. The MGDA modifies the incomplete relational data and regulates MLT iteratively for obtaining a well-completed data set. The iteration will stop when the stop criterion is satisfied. For the modification on the incomplete relational data and parameters, we test the coherence of two consecutive iterations. $x_1^t, x_2^t, \cdots, x_k^t, \ldots, x_n^t$ and $x_1^{t+1}, x_2^{t+1}, \cdots, x_k^{t+1}, \ldots, x_n^{t+1}$ $(k \in \{1, \cdots, n\})$ are two sequences of the incomplete relational data in two consecutive iterations, respectively, then

$$sig(x_k^t, x_k^{t+1}) = \begin{cases} 0,\ x_k^t = x_k^{t+1} \\ 1,\ x_k^t \neq x_k^{t+1} \end{cases}$$

For a given threshold $\eta > 0$, if $\frac{1}{n} \sum_{k=1}^{n} sig(x_k^t, x_k^{t+1}) < \eta$ then stop the modification and generate an MLT structure from the latest modified data set. Thus, when the above process terminates, a well-completed data set and a well-regulated MLT are obtained.

3.3 Structure Learning of PRM

MGDA uses class dependency graph [11] structures as the candidate PRMs. The MLTs here are also class dependency graphs. In those graphs, an attribute can depend on any attributes of all the classes except itself. If parents of the attribute are attributes of other class, they relate with the attribute by chains of relations. In this way, we can get the class dependency graph which contains latent relations and also get PRM structure with relations.

There are three basic dependencies [20] between attributes, i.e., transitive dependencies, non-transitive dependencies and induced dependencies, which can be described by the Bayesian network framework of information channels and pipelines [2]: (1) Transitive dependencies indicate that information flow can directly pass two nodes in Bayesian networks and not be blocked by any other nodes. In other words, the two nodes are conditional dependent. (2) Non-transitive dependencies indicate that information flow can not directly pass two nodes in Bayesian networks, but can pass through the open path which connects the two nodes and be blocked by the nodes in cut-set. Namely, the two nodes are conditional independent given the nodes in cut-set. (3) Induced dependencies are induced by V-structure. Information flow can not directly pass two nodes and be induced by the collider in V-structure [20]. [1] Learning Bayesain network is to keep transitive dependencies and get rid of other dependencies. As an extension of Bayesian network, PRM can be learned through the new dependency analysis approach from the well-completed data set.

To measure the conditional independence, we use mutual information and conditional mutual information. Conditional mutual information is defined as:

$$I(X_i.A; X_j.B|C) = \sum\nolimits_{\substack{X_i,\\ X_j}} \sum\nolimits_{\substack{A,\\ B}} \sum\nolimits_{\substack{x_i.a,\\ x_j.b}} P(x_i.a, x_j.b|c) \log \frac{P(x_i.a, x_j.b|c)}{P(x_i.a|c)P(x_j.b|c)}$$

where C is a set of nodes. When $I(X_i.A, X_j.B|C)$ is smaller than a certain small value ε, we say that $X_i.A$ and $X_j.B$ are conditionally independent given C. Then, we use dependency analysis to regulate the PRM structure from the well-completed data set.

Transitive Dependencies Regulation. We use the latest regulated MLT as the initial PRM structure. For each pair of nodes $(X_i.A, X_j.B)$ ($X_i.A$ is in front of $X_j.B$ in node ordering) without edge connection, compute conditional mutual information $I(X_i.A, X_j.B|Pa)$, where Pa are the nodes in all the parents of $X_j.B$ that are on the path linking $X_i.A$ and $X_j.B$. If $I(X_i.A, X_j.B|Pa) > \varepsilon$, add an edge $X_i.A \rightarrow X_j.B$. This process requires at most $\frac{(n-1)(n-2)}{2}$ number of conditional independence (CI) tests. The regulated structure is denoted by G_1.

Non-transitive Dependencies Regulation. For G_1, compute $I(X_i.A, X_j.B|Pa)$ for each pair of nodes $(X_i.A, X_j.B)$ with edge connection, where Pa are the

[1] For three nodes X, Y and Z, there are only three possible types of V-structures, i.e. (1) $X \rightarrow Y \rightarrow Z$, (2) $X \leftarrow Y \rightarrow Z$, and (3) $X \rightarrow Y \leftarrow Z$. Among them only the third type makes X and Z depend conditionally on $\{Y\}$.

nodes in all the parents of $X_j.B$ that are on the path linking $X_i.A$ and $X_j.B$. If $I(X_i.A, X_j.B|Pa) < \varepsilon$, delete the edge between them. This process requires at most $n(n-1)$ number of CI tests. The regulated structure is denoted by G_2.

Inductive Dependencies Regulation and Edges Orienting. For each pair of nodes, compute $I(X_i.A, X_j.B)$ according to the node ordering. If $I(X_i.A, X_j.B) < \varepsilon$, test the pairs of nodes by collider identification. Any nodes which can form a V-structure with $X_i.A$ and $X_j.B$ are denoted as $X_{m1}, \cdots, X_{mt}$ ($X_{mh} \neq X_i.A, X_j.B$, $h \in \{1, \cdots, t\}$). For a given threshold $\delta > 0$, if $\frac{I(X_i.A, X_j.B|X_{mh})}{I(X_i.A, X_j.B)} > 1+\delta$, then $X_i.A$, $X_j.B$ and X_{mh} form a V-structure and orient edges $X_i.A \rightarrow X_{mh}$ and $X_j.B \rightarrow X_{mh}$. If there is an edge between $X_i.A$ and $X_j.B$, then delete the edge. This process requires at most $\frac{n(n-1)(n-2)}{2}$ number of CI tests.

Using collider identification, we can identify all the V-structures of the third type in a probabilistic model and orient the edges in such structures using tests on conditional independence. The number of edges which can be oriented by collider identification is constrained by the network structure. In an extreme case, when the network does not contain any V-structure of the third type, these methods could not orient any edges at all. However, this method is popular in Bayesian network learning owing to its efficiency and reliability.

For edges that could not be oriented by collider identification, we orient them by computing the joint cross-entropy. For two discrete attributes $X_i.A = \{x_{i1}.a, x_{i2}.a, \cdots, x_{iM}.a\}$ and $X_j.B = \{x_{j1}.b, x_{j2}.b, \cdots, x_{jN}.b\}$, suppose the joint probabilistic distribution of $X_i.A$ and $X_j.B$ is $p_1(x_{im}.a, x_{jn}.b)$ under assumption H_1; the joint probabilistic distribution of $X_i.A$ and $X_j.B$ is $p_2(x_{im}.a, x_{jn}.b)$ under assumption H_2, where $m = 1, 2, \cdots, M$, $n = 1, 2, \cdots, N$. Then the joint cross-entropy of $X_i.A$ and $X_j.B$ can be defined as:

$$I(p_2, p_1; X_i.A, X_j.B) = \sum\nolimits_{m=1}^{M} \sum\nolimits_{n=1}^{N} p_2(x_{im}.a, x_{jn}.b) \cdot \log \frac{p_2(x_{im}.a, x_{jn}.b)}{p_1(x_{im}.a, x_{jn}.b)}$$

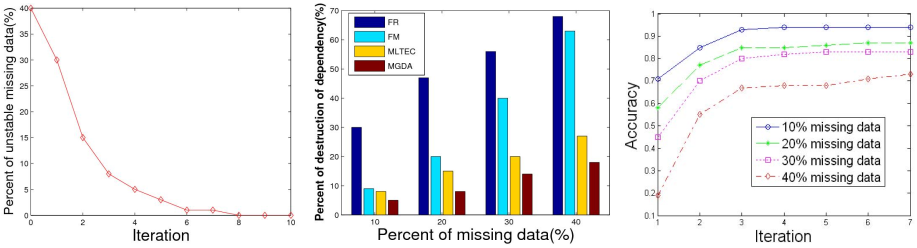
Let the assumptions H_1 and H_2 be $X_i.A \rightarrow X_j.B$ and $X_i.A \leftarrow X_j.B$, respectively. Compute $I(p_1, p_2; X_i.A, X_j.B)$ and $I(p_2, p_1; X_i.A, X_j.B)$:

if $I(p_1, p_2; X_i.A, X_j.B) > I(p_2, p_1; X_i.A, X_j.B)$,
then orient edges $X_i.A \rightarrow X_j.B$; **otherwise**, orient edges $X_i.A \leftarrow X_j.B$.

4 Experiments

We begin by evaluating the proposed MGDA approach on a synthetic data set generated by a school domain whose structure is shown in Fig. 2(a). The learning approach takes only the data set as input. We generate 4 data sets with the same size 5,000. Here the size of a data set corresponds to the number of students involved. These data sets are with 10%, 20%, 30%, and 40% missing data, respectively. These missing data are generated by randomly removing 10%, 20%, 30%, and 40% attribute-values from the original data sets, respectively.

We compare MGDA with FR, FM and MLTEC. FR and FM are two straightforward approaches. FR fills in the incomplete relational data randomly and then



(a) (b) (c)

Fig. 1. Results on the school domain. (a) Percentage of unstable missing data on the database with 40% missing data when MGDA is used; (b) Comparison of the destruction of dependency relations in structures learned from data sets; (c) Average accuracy of the structures learned by MGDA.

learns the structures of PRMs from the obtained complete data. FM fills in the incomplete relational data with the mean values of attributes and then learns the structures of PRMs from the obtained complete data. To the best of our knowledge, MLTEC [17] is the only approach for learning PRM structure from incomplete relational data, where an evolutionary algorithm is used and the PRM structure is learned by filling in the missing data with the best evolved structure in each generation. We run MLTEC for ten times on each data set and regard the median PRM structure of the ten runs as the result. This is because that MLTEC is an approach based on evolutionary computation, whose result could be very different in different runs. We wish that the median PRM structure of the ten runs could reflect the median performance of MLTEC.

Fig. 1(a) shows the percentage of unstable missing data, i.e., the portion of missing data being modified in each iteration of MGDA, on the data set with 40% missing data. It can be found that the missing data to be modified become fewer and fewer as the iteration proceeds. Moreover, by comparing the filled values and real values, we found that among the values filled in by MGDA, 93% are correct; for MLTEC, 90% are correct; while for FR and FM the correctness is only 61% and 66%, respectively.

Fig. 1(b) presents the comparison between FR, FM, MLTEC and MGDA on the destruction of dependency relations in the learned structures. It can be found that the performance of MGDA is apparently better than that of FR. This is not difficult to understand. When the missing data are filled in randomly, noises are introduced and thereby the dependency relations between attributes are corrupted to a great extent. By taking advantage of the information in the observed data, the noises will be smoothed through refining the missing data iteratively. Thus the corrupted dependency relations are recovered. The performance of MGDA is also apparently better than FM, especially when the level of missing data is high. This is not difficult to understand either. When the level of missing data is low, the mean values of attributes filled in the missing data can represent the distribution of the real data to some degree. With the increasing of the level of missing data, the mean values of attributes could not represent

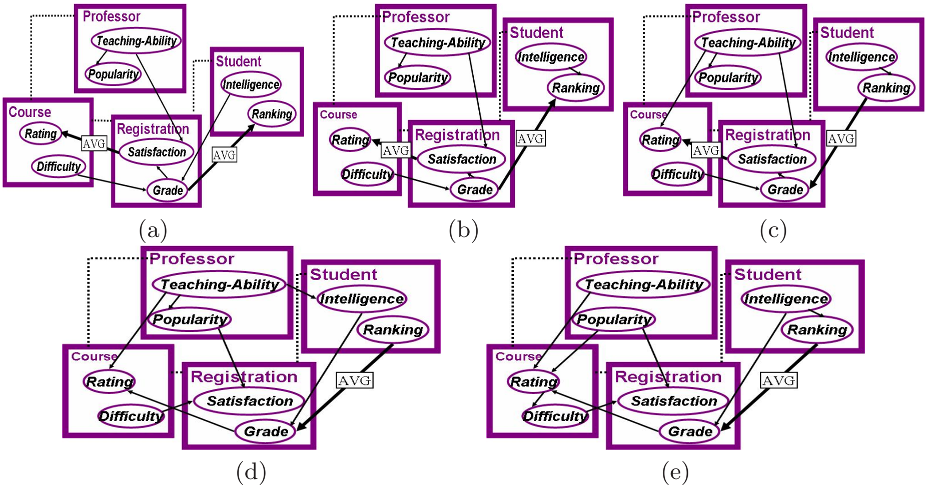


Fig. 2. The PRM structures on the school domain. (a) True structure. Dotted lines indicate relations between classes while solid arrows indicate probabilistic dependencies; (b) Result of MGDA; (c) Result of MLTEC; (d) Result of FM; (e) Result of FR.

the distribution of the real data well any more. Therefore, the performance of FM degenerates seriously. It can also be found that although the performance of MLTEC is better than FR and FM, it is worse than MGDA.

Fig. 1(c) compares the accuracies of the structures learned by MGDA on different data sets. Here we define *accuracy* in the following way. Suppose the ground-truth structure has a edges, the learned structure added b redundant edges but missed c edges, then the accuracy of the learned structure is $1-(b+c)/a$. It is obvious that the accuracy of a perfect model is 1, while the accuracy of some very poor models could be negative. Since very poor models are useless, it is not meaningful to distinguish them. Thus, we assign zero accuracy to them.

It can be found from Fig. 1(c) that as the iteration proceeds, the accuracy of MGDA increases. The accuracies of FR, FM and MLTEC are respectively 57%, 80% and 86% on 10% missing data, 14%, 57% and 80% on 20% missing data, 0, 14% and 71% on 30% missing data, and 0, 0 and 57% on 40% missing data. It is evident that the accuracy of MGDA is better than them. By inspecting the structures, we find that MGDA did not produce many redundant edges even in a high level of missing data. This might owe to the combination of Gibbs sampling with MLT, which has simple structure to modify the incomplete data and thus suffers less from overfitting.

The PRM structure learned by MGDA on the data set with 40% missing data is shown in Fig. 2(b). Comparing it with Fig. 2(a) we can find that it missed a dependency between the *Intelligence* of a student and the *Grade* of the registration and added a dependency between the *Intelligence* of a student and its *Ranking*. Fig. 2(c) shows the structure learned by MLTEC, which also missed the dependency between the *Intelligence* of a student and the *Grade* of

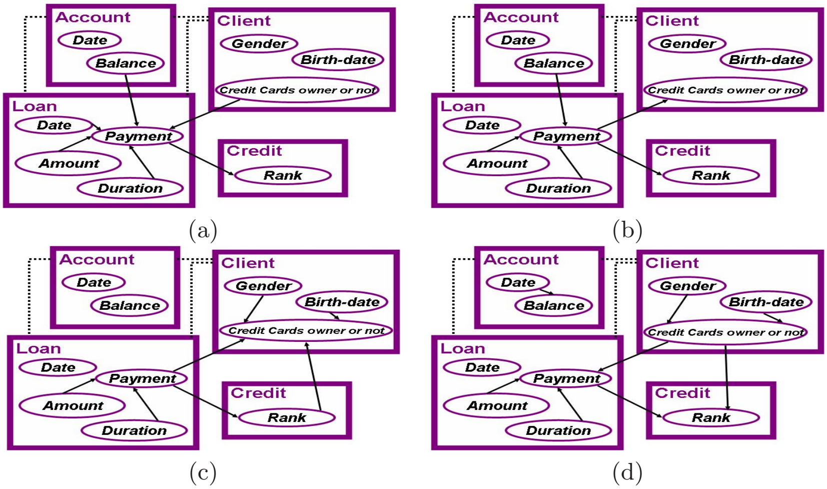


Fig. 3. The PRM structures learned on the financial domain. (a) By MGDA; (b) By MLTEC; (c) By FM; (d) By FR.

the registration. Moreover, it reversed the dependency between the *Grade* of the registration and the *Ranking* of a student, and added two redundant dependencies. Figs. 2(d) and (e) show the structures learned by FM and FR, respectively. Comparing them with Fig. 2(a) we can find that they have many redundant dependencies and missed many dependencies. In short, the structure learned by MGDA is more credible than those learned by the compared approaches.

We also evaluate the proposed MGDA approach on a real-world domain. This domain is a financial database taken from the PKDD2000 Discovery Challenge. The database contains data from a Czech bank, which describes the operation of 5,369 clients holding 4,500 accounts. The bank wants to improve their services by finding interesting groups of clients. The eight tables in the database are: *account*, *client*, *disposition*, *permanent*, *order*, *transaction*, *loan*, *credit card*, and *demographic data*. We focus on the question of clients' credit and choose a subset from the database, which consists of 4 relations, i.e. *account*, *client*, *loan* and *credit*. The extraction results in an incomplete data set. Since the data are from real-world and the ground-truth model is not known, it is not feasible to compare the proposed approach with other approaches quantitatively. Thus, we adopt the experimental methodology used by previous research [11,22], i.e., qualitatively comparing the learned structures.

The PRM structures learned by MGDA, MLTEC, FM and FR are shown in Figs. 3(a) to (d), respectively. MGDA learned that the *Payment* of a loan depends on its *Date*, *Amount* and *Duration*, the *Balance* of the account, and *the Credit cards owner or not* of the client. It also learned a dependency which can relate the tables: the *Rank* of Credit depends on the *Payment* of the loan. By comparing the structure learned by MLTEC with that learned by MGDA, we can find that MLTEC missed a dependency relation between the *Date* and the *Payment* of the loan and reversed the dependency between the *Payment* of a loan and the *Credit cards owner or not* of the client. However, the missed dependency seems not very important and the reversed dependency looks still reasonable. From Figs. 3(c) and (d) we can find that FM and FR missed an important dependency between the *Balance* of the account and the *Payment* of the loan and both generated more redundant dependency relationships.

5 Conclusion

Relational learning algorithms are capable of dealing with multiple tables or relations which could not be tackled by attribute-value-based methods. However, although real-world data are often incomplete, learning with incomplete relational data is largely understudied. In this paper, we propose the MGDA approach and experiments show that it can learn reasonable structures from incomplete relational data.

We observed that MGDA did not produce many redundant edges even when the missing rate was quite high. So, its performance may be improved by incorporating some mechanism for dealing with missing edges. This will be studied in the future. When several values for an attribute are almost equally likely, the current stopping criterion might encounter some problem. A possible solution may be to compute the Euclidean distance between the parameters for two consecutive steps and stop when this distance goes below a threshold. This is another future issue. Moreover, combining MGDA with collective classification is also worth studying in the future.

Acknowledgments

We wish to thank the anonymous reviewers for their helpful comments and suggestions. This work was supported by NSFC (60635030), the China Postdoctoral Science Foundation (20060390921) and the Jiangsu Planned Projects for Postdoctoral Research Funds (0601017B).

References

1. Anderson, C., Domingos, P., Weld, D.: Relational Markov models and their application to adaptive web navigation. In: KDD'02, Edmonton, Canada, pp. 143–152 (2002)
2. Cheng, J., Greiner, R., Kelly, J.: Learning Bayesian networks from data: An efficient algorithm based on information theory. Artificial Intelligence 137, 43–90 (2002)
3. Chickering, D.M.: Learning Bayesian networks is NP-complete. In: Fisher, D., Lenz, H.J. (eds.) Learning from Data: Artificial Intelligence and Statistics V, pp. 121–130. Springer, Berlin (1996)
4. Chow, C.K., Liu, C.N.: Approximating discrete probability distributions with dependence trees. IEEE Trans. Information Theory 14, 462–467 (1968)

5. De Raedt, L., Kimmig, A., Toivonen, H.: ProbLog: A probabilistic prolog and its application in link discovery. In: IJCAI'07, Hyderabad, India, pp. 2462–2467 (2007)
6. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society - B 39, 1–39 (1977)
7. Domingos, P., Pazzani, M.: On the optimality of the simple Bayesian classifier under zero-one loss. Machine Learning 29, 103–130 (1997)
8. Dzeroski, S., Lavrac, N. (eds.): Relational Data Mining. Springer, Berlin (2001)
9. Flach, P., Lachiche, N.: Naive Bayesian classification of structured data. Machine Learning 57, 233–269 (2004)
10. Friedman, N.: The Bayesian structural EM algorithm. In: UAI'98, Madison, WI (1998)
11. Friedman, N., Getoor, L., Koller, D., Pfeffer, A.: Learning probabilistic relational models. In: IJCAI'99, Stockholm, Sweden, pp. 1300–1307 (1999)
12. Geman, S., Geman, D.: Stochastic relaxation: Gibbs distributions and the Bayesian restoration of images. IEEE Trans. Pattern Analysis and Machine Intelligence 6, 721–742 (1984)
13. Getoor, L., Sahami, M.: Using probabilistic relational models for collaborative filtering. In: Masand, B., Spiliopoulou, M. (eds.) WebKDD'99. LNCS (LNAI), vol. 1836, Springer, Heidelberg (2000)
14. Getoor, L., Segal, E., Taskar, B., Koller, D.: Probabilistic models of text and link structure for hypertext classification. In: IJCAI'01 Workshop on Text Learning, Seattle, WA, pp. 24–29 (2001)
15. Getoor, L., Friedman, N., Koller, D., Taskar, B.: Learning probabilistic models of link structure. Journal of Machine Learning Research 3, 679–707 (2002)
16. Kersting, K., De Raedt, L.: Basic principles of learning Bayesian logic programs. Technical report, Institute for Computer Science, University of Freiburg, Freiburg, Germany (2002)
17. Li, X.L., Zhou, Z.H.: An approach to learning of PRM from incomplete relational data (in chinese). Chinese Journal of Software (2007) (in press)
18. Muggleton, S. (ed.): Inductive Logic Programming. Academic Press, London (1992)
19. Muggleton, S.: Stochastic logic programs. In: De Raedt, L. (ed.) Advances in Inductive Logic Programming, pp. 254–264. IOS, Amsterdam, The Netherland (1996)
20. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, San Mateo, CA (1988)
21. Richardson, M., Domingos, P.: Markov logic networks. Technical report, Department of Computer Science and Engineering, University of Washington, Seattle, WA (2005)
22. Sanghai, S., Domingos, P., Weld, D.: Relational dynamic Bayesian networks. Journal of Artificial Intelligence Research 24, 1–39 (2005)
23. Taskar, B., Segal, E., Koller, D.: Probabilistic classification and clustering in relational data. In: IJCAI'01, Seattle, WA, pp. 870–876 (2001)