One of the Challenges in Learning Probabilistic Relational Models: Aggregation

Abstract

Aggregation is a tool that is used in order to present the multiple instances of an individual attribute in a single value that characterizes the groups represent. However, probabilistic relational models are constructed from relational data base; these data are interrelated with different cardinalities so it is need for aggregation in some situations in order to convert the relation cardinality from “many” to “one”. This paper will shed light on the role of aggregation in learning probabilistic relational models that based on through presenting two aggregate functions one is conventional and the other is proposed and compare their effects on the produced models. This will be introduced through practical application on test data.

Keywords: Aggregation, Graphical Model, PRMs, Learning.

1. Introduction

The construction of Probabilistic Relational Models (PRMs) is done using raw data through learning process. Where the input of this process is relational data base that contains the interested data and the output is a model which encodes the dependencies between the features of input data, descriptive attributes of the entities in data base represent some features.

The learning process consists of different steps and in result the learning is divided into different stations. The initial step is to locate the features and find the suitable forms of them and in result it facilitates the next step that seeks about the dependencies between them and so on.

The descriptive attributes as features exist in relational data base are spread on different tables; these attributes can be related through multiple tables that have relations cardinalities such as one-one, one-many, and many-many. At the same time, the searching for dependencies between attributes is performed as one-to-one, so this means the “many” must be converted into single, aggregation is the tool that convert “many” to single values that as possible keep the features of “many”.

2. Probabilistic Relational Models

Probabilistic relational models (PRMs) are models that are used in mining the relational database in order to extract the regular relationships of interesting data and formulate them in suitable forms that are represented by probabilistic models. PRMs are extended traditional models with flat table. PRMs, in contrast to the traditional models devoted to the attributes of a single table, exploit the structure of relational database in addition to the attributes that they contain.

Probabilistic Relational Models (PRMs) are a recent development[1] that extends the standard attribute-based Bayesian network representation to incorporate a much richer relational structure. A probabilistic relational model specifies a template for probability distribution over a database. The template allows a generic dependence between attributes of the classes to be represented for a class of objects, which is then initiated for particular sets of entities and relations.

Construction of PRM is based on the existence of possible dependencies between different types of attributes that are belong to different entities, are stored in different tables.

It is possible to search the dependency between attributes in the same table or between attributes in different tables, whether they are connected with cardinality relations; many or one-to-one.

3. Learning Graphical Models

The graphical model that is required to be learned consists of two parts; structure and parameters. So the learning in graphical model is dedicated to two tasks according to the learned model components[2].

3.1 Learning Parameters

The learning parameters aim to infer the parameters which govern the graphical model from the data while this data are observable completely (all variables of domain are visible) or partially observed (contains hidden variable).

Basically, given data set d of n independent and identically distributed observations of the setting of all the variables in an individual graphical model d={X1,………,Xn} where Xi=$\{x_i^1,...........,x_i^m\}$, (m) represent the possible Xi states. The likelihood of the parameters(θ) is proportional to the probability of the observed data:

$$P(d \mid \theta) = \prod_{i=1}^{n} P(Xi \mid \theta) \qquad (1)$$

However, to estimate the unknown parameters from the data, the log likelihood (L) must be maximized using Maximum Aposterior Probability(MAP) for each P(Xi) value order to obtain the parameters θ that maximize the posterior probability of graphical model:

$$L(\theta) = \log(d \mid \theta) = \sum_{i=1}^{n} \log P(Xi \mid \theta) \qquad (2)$$

Using the factorization of joint distribution, we obtain:

$$= \sum_{i=1}^{n}\sum_{j=1}^{m} \log p(x_i^{(j)} \mid parents(x_i^j, \theta)) \qquad (3)$$

Assume the parameters θ_i governing the conditional probability distribution of X given with its parents distinct and functionally independent of other nodes in the graphical models, then the log likelihood of the graphical models can be expressed in a sum of terms involving subsets as individual node and its parent:

$$L(\theta) = \sum_{i=1}^{n} L_i(\theta_i) \tag{4}$$

So, $L_i(\theta_i)$ can be maximized locally and independently as a function of θ_i [3].

3.2 Learning Structure

The goal of learning structure is to find the best structure that represents the joint distribution and can predict correctly the related data. The structure represents the skeleton of graphical model, so learning structure sometimes is defined as learning graphical, however it faces the real challenges. The structure learning is interested in discovering possible required structure from its constructions; nodes and arcs. So the task is concentrated on specifying which candidate nodes and arcs can realize this goal.

Nowadays, the problem of learning or estimating a network from data is receiving increasing attention with the community of researchers into Uncertainty in Artificial Intelligence. In the literature, two common approaches to the learning problem are used: -

- Methods based on conditional independence tests
- Methods based on a scoring metric

The algorithms based on independence tests carry out a qualitative study on the dependence and independence properties among the variable in the domain, and then try to find a network representing these properties. So, they take as the input a list of conditional independence relationships (obtained, for example, from a database by means of conditional independence tests), and the output is a graph displaying these relationships as far as possible[4].

The algorithms based on a scoring metric try to find a graph which has the minimum number of links that ‘properly’ represents the data. They all use a function(the scoring metric) that measures the quality of each candidate structure, and an heuristic search method to explore the space of possible solutions, trying to select the best one.

Mathematically, the learning of structure in given a graph (G) with MAP:

$$P(G \mid D) \alpha P(G) P(D \mid G) \qquad (5)$$

And this can be measured by maximizing the likelihood of data (D) given a graph P(D|G).

3.3. Evaluation Measures

An evaluation measures serve to assess the quality of a given candidate graphical model a given database of samples cases, so that it can be determined which of a set of candidate graphical models best fits the given data[5]. A desirable property of an evaluation measure is decomposability i.e. the total network quality should be computable as sum or product of local scores, while local score is concerned with pairs (child | parents). This can enable to maximize the quality of each pair independently then in result the total network.

The evaluation measures are split into two classes, first the measures that are concerned with determining the structures of model by allocating the nodes and arcs, and the other are concerned with the evaluation the best parameters of models.

Most evaluation measures of structure model are based on measures of dependence, it is necessary to measure the strength of dependence of two or more variables, either in order to test of conditional independence or in order to find the strongest dependences. One of the dependences measures is Mutual Information (MI); it can be used to measure the strength of interaction between two or more variables, so it has a threshold (0.001) [6]. If MI for random variable is over this threshold, these variables have some dependence, else the variables are independent. The random variable with high score of dependence can be introduced in the candidates’ structures.

$$MI(X;Y) = \sum_{x,y} P(x,y) \log \frac{P(x,y)}{P(x)P(y)} \qquad (6)$$

Where X,Y random variables; MI(X;Y) is the mutual information between X and Y; x is the individual state of X and y is individual state of Y, P(x,y) is the joint probability of x,y; P(x) refers to probability of state (x); P(y) refers to probability of state (y).

4. Aggregate based Model Learning

The seeking for random variables in order to formulate the hypothesis space of candidate models in order to create PRMs is considered the main task of learning process, different possible values can be attached to them; description attributes, existence of possible objects, links between different relations, and aggregate descriptive attributes.

The learning of PRMs aggregate based models require to prepare and care the hypothesis space that contain the random variables, have aggregate attributes so the early step in such learning process is the extraction the suitable aggregate values of specified attributes and present as a virtual attribute and then complete the learning process will be done in similar way to other probabilistic models.

Basically, Aggregation is the approach used to deal with the attributes across different relations with this type of relationships and aggregate functions are suggested to conclude information and convert into single value according to some features that functions perform. PRMs learning procedure formulates the resulted aggregate values in virtual attribute, which would be joined to the opposite relation. The new virtual attributes would be tested in order to reveal any dependencies with the attributes that are located in the opposite relation. The pairs of attributes (virtual and opposite) are input into dependency tests. Then the pair that overcomes threshold value of specified measure will be candidates to be into other tests that construct the model through training step. The virtual attributes holds aggregate values are dealt as same as the original attributes so the learning steps of model are the same in more of them for both types of attributes. Algorithm(1) contains the required steps to aggregate the interested attributes in one relation and formulate them into virtual attributes that holds aggregate values and join them to opposite relation. The aggregation is accomplished using the corresponding between the foreign key in opposite relation in allocating the bags in an interested relation and extracted the aggregate values for each bag.

Algorithm(1) Construction Virtual Attribute by Aggregation

```
Input : Two Relations X,Y are connected with cardinality(One_to_Many)
Output : Virtual attribute(V) would be joined with X as X.V
Processing:
    1- Let Y.A be discrete attribute in Y to be aggregeated
    2- Grouping Data(D) in Y relation into bags of record (Di), where the
       number of bags is determined by the tuples in X. and size of (Di) is the
       number of tuples  related with (i) tuple in X
    3- For i=1 to n:
           a- Execute Aggregate Function on (Di)
           b- $v_i$ is resulting single value
    4- Formulate values($v_n$) in X.V and join it to X as X.V
```

5. Practical Work

The practical work consists of multiple tasks in order that we present in this section.

5.1 Description of Used Aggregate Functions

The aggregate functions we are interested in are the functions related with discrete values. Two functions are used in this work; conventional function and proposed function.

5.1.1 Conventional Function

The conventional aggregate discrete function is concerned with "Mode", where it seeks the state that has higher frequency in bags and presents it as a single aggregated value. It requires computing frequencies of states in each bag separately and there is no need for further information about whole data. The Algorithm (2) performs this function. The required data to this algorithm are the attribute to be aggregate and the instances of this attribute and the number of bags that group the attribute instances and this is specified by related opposite relation.

Algorithm(2) Conventional Aggregate Function

```
Input: Attribute A in Relation(X) with attribute instances Data(D) and number of bags(m)

Output: aggregated values for each bag Di

Processing:

    1- For i=1 to m (m number of sets) execute step(2-4)

    2- Compute freq(j), $j \in$ Val(X.A) in Di as:

    3-Search for the largetst frequency from set of frequencies

    4- Let freq(k) be the largets frequency

       So k is the state and its value represents the aggregated
    value to represent Di for attribute A
```

The state of aggregated attribute that catches higher frequencies in each bag will have the chance to be represented and the rest will be hidden even if they have trivial difference from the largest frequency.

5.1.2 Proposed Function

The previous function penalizes the states with low frequencies without taking into account the possible significant of them. From this point the proposed function was suggested in this work in order to present two different functions that extract different features from the same data.

The idea of proposed function is to take into consideration the frequencies of states in whole data then grant specific weights by employing these weights in computing the frequencies in each bag. As a result the significant frequencies will be emerged according to the local frequencies in bag and to degree of scarce of the state value in whole data. The required algorithm to perform this function on the interested data is Algorithm(3).

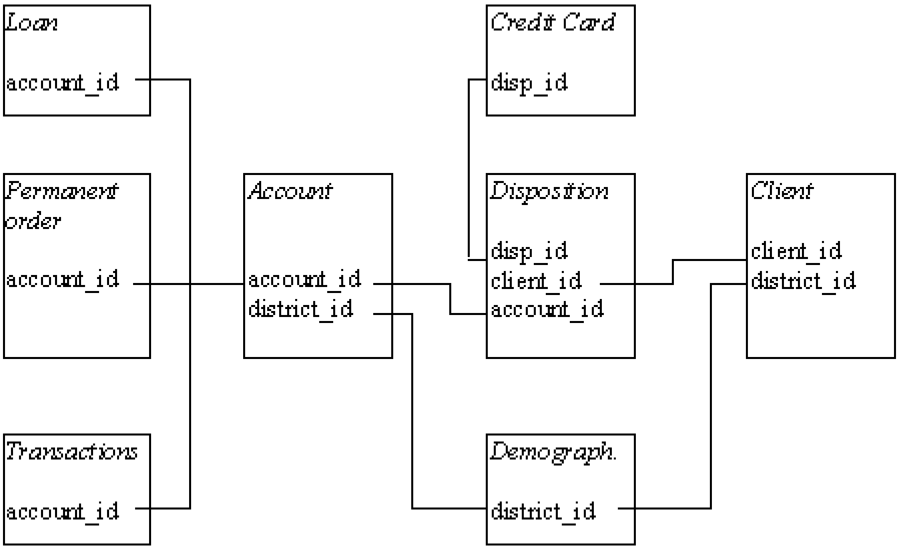
Algorithm(3) Proposed Aggregate Function

```
Input: Attribute A in Relation(X) with attribute instances Data(D) and number of bags(m)

Output: aggregated values for each bag Di

Processing:

1- Compute freq(j) of each state $j \in$ Val(X.A) in D
2- For i=1 to m (m number of bags) execute step(3-5)
3- For each state $j \in$ Val(X.A) Compute Distribution Dis as:
        a- Freq(j) in Dom(Di)
        b- $Dis_{ij}$=freq(j) in (D)/freq(j) in ($D_i$)
4- Sort ($Dis_{ij}$) ascending
5- Pick the state that has the first Distribution in the sorted list. To represent the aggregated value of X.A in $D_i$.
```

Each of both functions has some characteristics that can be suitable for kind of data and to specific cases. This is what we will see in the practical section.

5.2. Expirement Data Description

The experiment data we use are from PKDD'99 Discovery Challenge "Guide to the Financial Data Set" and are given free for research purposes[7] as shown in Figure(1). Data are organized in relational data base that consists of (8) relations. Each relation is referred in Figure(1) by "box" and the line between two boxes refers to association between them; these association are addressed with different cardinality. Our work in this paper is focused on the association with "many-one" cardinality and that we found between "Account" relation and "Transaction" relation.



Figure(1) Relational Structure Schema Of Experimented data

5.3. Scope of Experiment Data

The relations in experimented data that are used, which satisfy our work objects, are "Transactions" and "Account". These relations are linked with relationship “Many-to-One”, where each instance in “Account” is connected with multiple instances in “Transactions”. Also these relations have additional property that is sufficient instances are available. As a result it can consolidate the results. On the other hand, other relations in the experimented database not satisfy these characteristics therefore we exclude them from our work.

The practical work is specified with discrete attributes; therefore these attributes in both relations are marked to be in the practical procedures. Relation “Account” contains two discrete attributes; “Date” and “Frequency”. Relation “Transactions” contains discrete attributes like “Date”, “Type”, “Operation”, “Symbol_k”, and “Bank_to”; while “Amount” is continuous attribute that is converted into discrete attribute.

Also before we proceed in practical work, we need to know average size of each set of objects in D_i that refers to number of instances connected to one instance in other relation in order to evaluate the results. Average size of D_i=no. of instances in relation X/no. of instance in relation Y. where X-Y link has cardinality (n-1)

For relation “Transaction:

size (D_j)=no. of instances in transaction/no. of instances in account

$$= \frac{1056320}{4500} = 234.7377$$

That means the instance in relation “Account” may be linked with average instances around (234) from “Transactions”.

5.4. Computing Mutual Information(MI)

As related, here we determine the legal structure, which specifies the dependency structures. The practical step is to find the elements of structure. The basic elements of the structure are nodes and arcs; nodes refer to random variables, arcs refer to dependency that exists between them. How they can be located and extracted from raw data is the problem.

The task entails focusing on the decision which nodes it will contain; the nodes that are candidate to connect and interact through the structure. What the factor that must be taken in determining the candidate nodes? , the strength of the relation between different nodes is the indicator for this decision that is considered with the amount of information which moves between the attributes in order to determine if there is dependency or independency. According to this measure, the nodes are added or not to the networks and the result decide the connections between them.

The probabilities that are used in these calculations are estimated from data set by using the frequencies of the states of attributes.

MI is executed on the experimented data according to the specified attributes whether original and virtual attributes. The results from MI test are listed in Table(1) as related with aggregated attributes from conventional approach, while the results, which are related with aggregated attributes with proposed approach are listed in Table(2). From comparison between the results in Tables(1), (2), it is clear the good results come from the good distribution that comes from the proposed approach. Attribute “Date” that is well distributed from using proposed approach continues in this step and this is consolidated by giving reasonable results. “Date” is combined with “Frequency” and “Date” in “Account” relation individually. As with pair variables “Date-Frequency”, the MI indicates to independence between these variables and when the values of MI for both cases (aggregated attribute with conventional and proposed) are compared, we see the MI value for the variable (aggregated attribute with proposed approach) is less than the MI value for variable (aggregated attribute with conventional approach). As MI value comes down, it refers to strong independence between variables and vice versa. So we can evaluate aggregated value with proposed approach by the high score in this case of independence.

On the other hand, pair “Date-Date”, the computed MI value refers to good dependence between these two variables. Also, we see the pair that comprises variable (aggregated attribute with proposed approach) has higher score than the pair related with (aggregated attribute with conventional approach). As a conclusion from this case, it is obvious the proposed approach makes the sharp results whether dependence or independence and this is what is required of constructing good models.

6. Conclusion

According to the results, we can distinguish the difference between the effects of conventional aggregate function from proposed aggregate function. No preference one on the other, but there is some different features that can be exploited in discovering multiple dependencies between the virtual variable in an individual relation and the other variable in the opposite relation. This means some effort will be required in the choice of the aggregate function in order to extracting the best PRM with good dependencies.

Table (1) mutual information of association relations account and transaction with conventional approach

Account Attribute	Transaction Attribute "Aggregated"	Mutual Information
Frequency	Date	9.990431e-004
Frequency	Type	4.448369e-004
Frequency	Operation	6.695549e-003
Frequency	K-symbol	1.999152e-005
Frequency	Bank_to	-1.486657e-001
Date	Date	4.887551e-001
Date	Type	1.329568e-002
Date	Operation	1.091149e-003
Date	K_symbol	5.825361e-004
Date	Bank_to	-1.561241e-001

Table (2) mutual information of association relations account and transaction with aggregated attributes by proposed approach

Account Attribute	Transaction Attribute "Aggregated"	Mutual Information
Frequency	Date	1.818759e-004
Frequency	Type	8.077454e-002
Frequency	Operation	1.093737e-002
Frequency	K-symbol	2.885031e-003
Frequency	Bank_to	-5.087133e-002
Date	Date	1.128466e+000
Date	Type	2.224417e-002
Date	Operation	2.786915e-003
Date	K_symbol	6.064359e-003
Date	Bank_to	-8.013645e-002

References

[1] Koller D. and Pfeffer A., "Probabilistic frame-based systems”, 1998, Proc. AAAI.

[2] Murphy K., “A Brief Introduction to Graphical Models and Bayesian Networks”, 1998, www.cs.ubc.ca/~murphyk/Bayes/bnintro.html.

[3] Ghahramani Z., "Graphical models: parameter learning”, www.gatsby.ucl.ac.uk/~zoubine/

[4] Geiger D.and Pearl, "On the logic of casusal models", 1988, Proc. 4th Workshop on Uncertainty in AI, St Paul, Minn, p.p.136-147.

[5] Borgelt C. and Kruse R., “Learning from Imprecise Data: Possibilistic Graphical Models”, http://citseer.ist.psu.edu/462991.html.

[6] Cheng J., Bell D.A., and Liu W., “An algorithm for Bayesian belief network construction from data”, Proceedings of AI &STAT97, p.p.83-90, 1997.

[7] Berka P., “Guide to the financial data set. The ECML/PKDD 2000 Discovery Challenge.