A propositionalisation that Preserves More Continuous Attribute Domains

Julien Lesbegueries, Nicolas Lachiche and Agnès Braud

LSIIT - University of Strasbourg

Abstract. The aim of this paper is to extend local transformation functions for propositionalisation. We follow the works of [1] by adding a transformation function that reverses the thresholding problem. Indeed, we propose an innovative method that manages numeric attributes without discretisation nor aggregation.

1 Introduction

Aggregation-based features and discretisation cause well-known biases in multi-relational learning. Indeed, there is a loss of information for both techniques since multiple values of an attribute are summarized in order to produce one (or few) features. However these techniques are often necessary. Aggregation-based attributes in multi-relational problems allows to deport information into the target table in order to be used in propositional and efficient learning processes. Discretisation occurs when numeric attributes are not managed by learners or implicitly when the learners choose a threshold.

The motivation of our proposition is that in some cases, it is fundamental to preserve the whole information and it cannot be aggregated. Let us take an example with a spatial dataset, composed of districts and buildings of a city (see figure 1). This city dataset is a relational database composed of 2 tables. The target table represents districts of a city and the other table represents buildings contained in the districts. Their attributes give geometric and topologic information about surface, elongation, convexity, etc. The learning problem consists in categorizing a district as *housing surface*, *specific urban surface*, etc.

There can be several buildings per district and we want to classify these districts depending on all their characteristics. It is obvious that the ratio of buildings with a housing shape can determine wether a given district is a housing surface or not. Similarly, one or few buildings with a large area can determine the district to be a specific urban surface (with industries, etc.). However, if the district is also composed of little buildings, aggregate functions (like the average of the buildings areas) can hide the relevant information carried by the relations cardinalities.

Thus we propose a technique where cardinalities of such relations are involved in a propositionalisation process in order to avoid aggregation and discretisation on numeric attributes. Based on database-oriented methods like RELAGGS [1],

Presented at the 19th International Conference on Inductive Logic Programming (ILP'09).

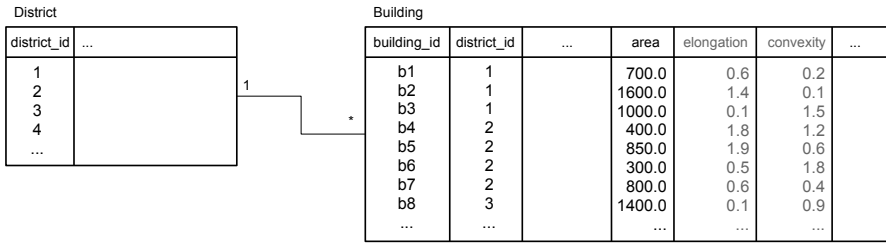

Fig. 1. Schema of the district example.

our method implies a local transformation function that inverts classical problems of propositionalisation by pointing up cardinalities instead of thresholds.

Section 2 presents related works on propositionalisation (database-oriented and logic-oriented methods), Section 3 presents our approach. Then experiments allow to compare the different approaches and their combination (on benchmarks and in particular on a geographic domain that motivated firstly our proposition). The last section discusses these experiments and perspectives.

2 Related Work

Propositionalisation aims at converting a relational problem into an attribute-value one. As explained in [2], approaches for constructing the new set of attributes can be divided in two trends. The first one follows the Inductive Logic Programming tradition and is logic-oriented. Systems of that kind are based on existential features, namely conjunctions of litterals, which constrain the language of clauses learnt. This trend includes the first representative LINUS [3] and its descendants, the latest being SINUS, RSD [2] and HiFi [4]. The second approach is database-inspired and appeared later, in 2001, with two systems Polka [5] and RELAGGS [1] in two different research groups. Those systems build attributes which summarize information stored in non-target tables by applying usual database aggregate functions such as count, min, max, etc. While logic-based approaches may perform well on purely symbolic problems, they may encounter lots of difficulties when numeric data are involved. In this case, database-oriented approaches have proven to be powerful [2].

Propositionalisation algorithms can be defined as transformation functions, that take as input a set of clauses (i.e. a query) and an example (i.e. a tuple in the target table). The result is a tuple of desired features values for the example according to the query [1]. Let us formally define the transformation function φ as

$$\varphi : C, e \mapsto (v_1, ..., v_{n_{\varphi,C}}), \tag{1}$$

where $C \in \mathcal{C}$ is the set of clauses (included in the set of all possible clauses), e is the example.

For example, we can illustrate this definition by the existential function expressed as:

$$\varphi_{\exists}(C, e) := \{ (1)\ if \mid T \mid > 0, (0) \text{ otherwise.} \tag{2}$$

where T is the set of tuples directly linked with e and satisfying C, and $\mid T \mid$ its cardinality. In [1], the proposed algorithm RELAGGS is based on a specific class of transformation function, called local transformation function. The equation 3 shows how it is used:

$$\varphi(C, e) = \bigoplus_{i=1..width(T)} \varphi'(A_i) \tag{3}$$

where A_i denotes the i^{th} numeric attribute of T, $\oplus$ denotes the tuple concatenation and the $width()$ function gives the number of attributes (or columns).

The specific φ' function proposed in (original) RELAGGS corresponds to:

- for numeric attributes: $\varphi'(A_i) := (avg(A_i), max(A_i), min(A_i), sum(A_i))$
- for nominal attributes: $\varphi'(A_i) := \bigoplus_{v \in domain(A_i)}(count(v, A_i))$

where $domain(A_i)$ is the ordered set of possible values for A_i and $count(v, A_i)$ is a function that provides the number of occurrences of value v for A_i in T.

The RELAGGS method gives encouraging results for relational data thanks to the representativeness of classic statistic operators used. Nevertheless it causes aggregation biases, e.g. concerning information carried by the cardinality of relationships between the examples and the non-target table. Its main principle actually consists in taking some relevant values from a given bag (min, max, avg, ...). Our idea is that taking each value of the bag into account can extract specificities at the cardinal level (there is n objects with a numeric attribute A smaller than X). This hypothesis is argued in the next section.

3 Propositionalisation by "Cardinalisation"

Our hypothesis is that the cardinality specificity of non-target relations can be relevant in some cases, when some discriminant attributes are numeric. In particular, information preserved by cardinalisation is different from other propositionalisation processes outputs. Some classes of our geographic learning problem can be separated knowing the number of "small" buildings present in the area. Existing aggregation methods loose this kind of information (the number of ...). With our method, the classifier will choose both the best attribute (i.e. the appropriate cardinality to separate classes) and the best threshold on it.

Thus, we define the function $\mathcal{TH}(A_i, k, T)$. Given A_i the i^{th} attribute of T, k a cardinality (between 1 and $\mid T \mid$) and T, the set of tuples directly related to e (for now, there is no constraint given by C, as in the RELAGGS proposition):

$$\mathcal{TH}(A_i, k, T) := min(th \in \mathbb{R} \text{ s.t. } \mid \{t \in T \text{ s.t. } v_{A_i}(t) \leq th\} \mid \geq k) \tag{4}$$

District

district_id	...	TH(area, 1, buildings(district_id))	TH(area, 2, buildings(district_id))	...
1		700.0	1000.0	
2		300.0	400.0	
3		...	...	
4				
...				

Fig. 2. Cardinalisation process for one related table *Buildings* and one attribute *area*.

where th is the minimal threshold, t is a tuple in T, and $v_{A_i}(t)$ returns the value of the attribute A_i for t. The global transformation function, entitled "cardinalisation", for an example e and an attribute A_i consists in using a concatenation of φ', defined as:

$$\varphi'(A_i) := \bigoplus_{k=1..|T|} \mathcal{TH}(A_i, k, T) \tag{5}$$

A set of SQL queries is computed for each attribute of each related table. We can find an analogy between our method and the "cumulative binary representation" defined in [6]. However that method was used to represent the ordering between discretized values of a numeric attribute in multiple columns in a binary way.

Figure 2 illustrates the process with a sample of the urban dataset. For the attribute *area*, a list of N columns is computed,

$$N = max_{e \in \text{Target Table}}(\mid T_e \mid) \tag{6}$$

where T_e is the set of tuples related to e. N is the maximum cardinality of *Building* tuples for one *District* tuple. These columns are put in the target table *District*. Let us emphasize that the number of features (i.e. of columns) is bound by the cardinality of the one-to-many relation and not by the number of distinct values of the attribute *area*. The generated clauses, containing actually 2 imbricated inequalities, are expressed in natural language as the *minimal value of attribute A (area) such that there are at least c related objects (buildings).*

In this way, numeric attributes thresholds can be chosen in a later step by a propositional attribute-value learner. It will choose the best new feature for which it will find the best threshold. Indeed, we reverse the thresholding problem by selecting values according to the cardinality of the relation. Thus we do not lose information with threshold searches.

It extracts further information from data, by associating a count to each threshold (the number of non-target tuples corresponding to this threshold) and that can be useful when classic aggregation operators like *average()*, *count()*, *sum()*, etc. do not provide enough information.

3.1 Cardinality discretisation

Our approach is dependent on the numbers of tuples involved in a one-to-many relation with the target table. Then, a restricted version of the method can be used in order to manage high cardinality relations. It is a discretisation of the cardinality but not at all a discretisation of the numeric attribute itself. Indeed, for technical reasons (maximum number of columns reached in a DBMS) or overfitting (if the number of attributes is too high according to the dataset), we can select a subset of values by spanning the cardinality from 1 to $\mid T \mid$ with a given *step* higher than 1. This way, we do not discretize on attributes domain but on the number of related objects. The equation 7 corresponds to this discretized version:

$$\varphi'(A_i) := \bigoplus_{k=1..|T|}^{k:=k+step} \mathcal{TH}(A_i, k, T) \tag{7}$$

3.2 Experiments

Experiments are made on our specific city dataset and on the PKDD financial data. The latter is taken from the Discovery Challenge organised at PKDD 1999 and PKDD 2000. It is based on data from a Czech bank and describes the operations of 5369 clients holding 4500 accounts. The data is stored in the tables *loan*, *transactions* and *permanent order* for client activities, *account*, *demograph*, *disposition*, *credit card* and *client* for clients information. The most relevant tables are *loan* that contains the status to be learned (problem or not in the loan) and the table *transactions* with which there is one-to-many relation.

The table 1 compares, for each dataset, the accuracy of the decision tree J48 with attributes of the target table only, the RELAGGS attributes (using the Proper Toolbox [7]), the Cardinalisation attributes and the RELAGGS and Cardinalisation attributes together. Accuracy is estimated using a 10 fold cross-validation. Our approach gets results close to RELAGGS' whereas cardinalisation does not get categorical attributes into account. Moreover, the combination of the two techniques improve the results. Cardinalisation provides additional information complementary to Relaggs method.

We can explain this improvement by the fact some classes are close to each other and some very fine characteristics can only be detected by non-aggregation-based attributes. For example, *building_min_2_area* attribute (giving the minimal surface such that there are at least 2 buildings in the area) can discriminate mixed areas and non mixed ones. Another example is *building_min_2_convexity* (giving the minimal convexity such that there are at least 2 buildings in the area) that discriminates high density housing surface areas and industrial / commercial areas (specific urban surfaces).

Concerning the financial data set, let us notice that transactions realized after the corresponding loan is granted have been dropped. Moreover, only loan and transaction relations have been used for our method.

Table 1. Experiments with C4.5 algorithm on attributes generated by RELAGGS, Cardinalisation and both.

Dataset	Learner	TargetTable	RELAGGS (R)	Cardinalisation (C)	R+C
City	J48	63.2%	71.97%	70.21%	**74.04%**
PKDD (AC-BD)	J48	88.86%	92.23%	91.64%	**92.52%**

4 Conclusion and Future works

These experiments show that our proposition is suitable for some relational learning problems, in particular when the number of non-target objects is low or when a threshold on some continuous attribute of related objects is important and can be relevant for classification. When there are more sub-objects, either cardinalisation can be discretized or statistic functions proposed by RELAGGS can be used. Moreover, cardinalisation can be joined to the discretisation of continuous attributes, and to aggregation, to provide an additional perspective on a relational domain for a propositional learner.

Future works concern an extension of our method by its recursive use, when there are several consecutive one-to-many relations in dataset schemas.

5 Acknowledgments

Our work is led in the framework of the national ANR project Geopensim. We want to thank its members and in particular geographers who helped us building the urban dataset. We are also grateful to Jérôme Lavoie who helped on the implementation and experiments.

References

1. Krogel, M., Wrobel, S.: Transformation-based learning using multirelational aggregation. In Springer, ed.: Inductive Logic Programming. (2001) 142–155
2. Krogel, M.A., Rawles, S., Zelezný, F., Flach, P., Lavrač, N., Wrobel, S.: Comparative evaluation of approaches to propositionalization. In Horvath, T., Yamamoto, A., eds.: Proceedings of the 13th International Conference on Inductive Logic Programming (ILP'2003), Springer-Verlag Heidelberg (October 2003) 194–217
3. Lavrac, N., Dzeroski, S.: Inductive Logic Programming: Techniques and Applications. Routledge, New York, NY, 10001 (1993)
4. Kuželka, O., Železný, F.: Hifi: Tractable propositionalization through hierarchical feature construction. In Železný, F., Lavrač, N., eds.: Late Breaking Papers, the 18th International Conference on Inductive Logic Programming. (2008)
5. Knobbe, A.J., Haas, M.D., Siebes, A.: Propositionalisation and aggregates. In: In Proceeding of the 5th PKDD, Springer-Verlag (2001) 277–288
6. Knobbe, A., Ho, E.: Numbers in multi-relational data mining. In: Knowledge Discovery in Databases: PKDD 2005. Volume 3721/2005 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg (2005) 544–551
7. Reutemann, P., Pfahringer, B., Frank, E.: A toolbox for learning from relational data with propositional and multi-instance learners. In: Proc 17th Australian Joint Conference on Artificial Intelligence. Cairns, Australia, Springer (2004) 1017–1023