Induction of Abstraction Operators using Unsupervised Discretization of Continuous Attributes

M. Biba, S. Ferilli, T.M.A. Basile, N. Di Mauro, and F. Esposito

Department of Computer Science, University of Bari, Italy
{biba, ferilli, basile, ndm, esposito}@di.uniba.it

Abstract. Most complex real-world domains are described by both relations among objects and numeric features. While ILP allows to handle relations, it is less suited to deal with numeric values. Abstraction operators can help in these cases by replacing specific values with intervals. Since it is difficult for humans to provide useful discretizations, we propose a method for automatically learning the abstraction operator from available observations. The resulting abstraction theories are then tested on a well-known dataset.

1 Introduction

Most complex real-world domains are described by numeric information, but attribute-value representation languages could be not enough powerful to capture their complexity, raising effectiveness problems for machine learning systems. In such domains, it is often necessary to adopt a more powerful representation language, like first-order logic that is able to describe relations between objects. However, while being very suitable for handling symbolic descriptions, it is less tailored to cope with numerical (and specifically real-valued) information. A possible solution to overcome such a problem, and enhance the learning process, is obtained by replacing specific numerical values with symbolic descriptors that represent the interval they belong to. Conceptually, this corresponds to the extension of pure induction with other inference strategies, and specifically with abstraction, defined as a mapping between representations that are related to the same world but contain less detail (typically, only the information that is relevant to the achievement of the goal is maintained).

Abstraction deals with cases when learning can be more effective if it can take place at multiple (different) levels of complexity, which can be intended as a language bias shift. Indeed, it can be used by machine learning systems to perform a shift to a higher level language when the current one is not enough powerful to capture the target predicate definition (a well-known problem in the traditional approach to the inductive concept-learning). An Abstraction Theory (an operational representation of such a mapping) is used to perform such a shift to a higher level representation: It is a collection of alternative definitions for intermediate concepts.

A useful perspective for the integration of abstraction inferences in an inductive learning framework has been given by Zucker [1]. According to such a perspective, abstraction takes place by means of a set of operators, that generally includes operators for grouping indistinguishable objects into equivalence classes; grouping ground objects to form a compound object (that replaces them in the abstract world); ignoring terms (that disappear in the abstract world); merging values that are considered indistinguishable; reducing the arity of a function or relation (even up to elimination of all arguments). Definitions of such operators make up the Abstraction Theory. Providing proper abstraction theories, and specifically sensible discretizations of real values, is very difficult for humans, even domain experts. Under this motivation, after facing in previous work the automatic learning of operators for grouping and ignoring objects, here we focus on the automatic inference of value-merging (i.e., discretization of continuous values into intervals) operators.

2 Unsupervised discretization

Supervised discretization methods use class information while generating the sub-intervals in which the values of a continuous attribute should be split. Since observations descriptions often do not carry class information about many of the involved objects, such methods are not applicable to the continuous-valued attributes of those objects. On the other hand, few works exist on unsupervised discretization methods [2]. We developed an unsupervised discretization method that uses non-parametric density estimators to automatically adapt sub-intervals size to the data. Among the currently produced intervals, the proposed algorithm searches for the next interval to be split by evaluating the best cut point on the basis of the density it induces by splitting into the sub-intervals and the outcome of a kernel density estimator for each of them. We introduced a principle for splitting the intervals based on the idea that the next sub-intervals to be produced for successive splitting are those that reflect the worst fit to the data density distribution given by a kernel density estimator. The cut point that produces the worst two sub-intervals in terms of fitting the density distribution of the data, is chosen as the best. The algorithm uses the cross-validated log-likelihood to select the number of intervals. It tries to group together in intervals continuous values based on their density and to capture the changes in density in different separate intervals in order to identify only the most relevant regions that can represent the set of continuous values. The new proposed method has been compared to classical unsupervised discretization methods through experiments on 200 continuous attributes of 15 different datasets from UCI repository that are usually used as a test-bed for discretization algorithms. Table 1 shows the number of cases in which our technique was significantly better, equal or worse compared to the other methods according to a t-test based on 10-fold cross-validated log-likelihood, resulting in better performance in many cases.

Table 1. Cross-validated log-likelihood t-test comparison with other discretization methods

	Our method significantly more accurate	Not significantly different	Our method significantly less accurate
Equal-Width (10 bins)	59	132	9
Equal-Frequency (10 bins)	56	136	8
Equal-Width Cross-validated	44	149	6

3 Experiments

Given the good performance of the proposed discretization procedure, we decided to exploit it as a tool for the automatic induction of a value-merging abstraction operator to be used in abstraction theories for supporting ILP systems in dealing with numerical information. Specifically, we embedded it in the symbolic version of the ILP system INTHELEX [3] and ran experiments on the classical mutagenesis dataset from which we considered the atom bond descriptions, the indicator variables, *logp* and *lumo*. The aim was studying the system performance when provided with the automatically learned abstraction operator concerning the three continuous attributes that describe the examples, and specifically: *logp*, *lumo* and *charge*. As measures of performance, we use predictive accuracy, runtime and theory complexity intended as number of clauses. Initially we tested the abstraction theory automatically generated by the discretization algorithm by letting it choose the best number of intervals, which was 19, 18 and 16 for logp, lumo and charge, respectively. A 10-fold cross-validation produced these results: 6985 seconds as processor time, 11 clauses and 83.5% predictive accuracy. In order to understand how the learning task is influenced by the number of intervals, we generated other abstraction theories by forcing the number of intervals in the discretization method. Among different numbers of intervals we found that the best results in terms of predictive accuracy (86.9%) were reached by generating 7 intervals for each of the three continuous attributes, at the expense of a slightly more complex theory: 4258 seconds of processor time and 12 clauses. This is quite interesting because it shows that by searching in the space of abstraction theories for a theory that can focus on more relevant information, it is possible to increase the performance of the learning system. Although the automatically found triple 19-18-16 produced good results compared to the maximal performance obtained with the triple 7-7-7, it is useful to investigate how the best number of intervals can be achieved. As a future work we plan to extend the algorithm that generates the abstraction theories in order to not only provide a valid discretization algorithm but also to search for the number of intervals that can maximize the performance of the learning system.

References

[1] J.-D. Zucker. Semantic abstraction for concept representation and learning. In R. S. Michalski and L. Saitta, editors, *Proceedings of the 4th International Workshop on Multistrategy Learning*, pages 157–164, 1998.

[2] J. Dougherty, R. Kohavi, and M. Salami. Supervised and unsupervised discretization of continuous features. In A. Prieditis and S. Russell, editors, *Proc.Twelfth International Conference on Machine Learning,* Morgan Kaufmann, 1995.

[3] F. Esposito, S. Ferilli, N. Fanizzi, T. M.A. Basile, and N. Di Mauro. Incremental Multistrategy Learning for Document Processing. *Applied Artificial Intelligence: An Internationa Journal*, 17(8/9):859–883, Taylor \& Francis, 2003.