

Learning from multi source data

Élisa Fromont, Marie-Odile Cordier, and René Quiniou

IRISA, Campus de Beaulieu, 35000 Rennes, FRANCE,
{efromont,quiniou,cordier}@irisa.fr

Abstract. This paper proposes an efficient method to learn from multi source data with an Inductive Logic Programming method. The method is based on two steps. The first one consists in learning rules independently from each source. In the second step the learned rules are used to bias a new learning process from the aggregated data. We validate this method on cardiac data obtained from electrocardiograms or arterial blood pressure measures. Our method is compared to a single step learning on aggregated data.

1 Introduction

In many applications, correlated data are recorded from different sensors observing the same phenomenon from different points of view. We are investigating how to take advantage of this diversity and of these correlations to learn discriminating rules from such data.

Mono source learning consists in learning separately from different sources describing a common phenomenon. When dealing with clear signals, only one channel can be sufficient; when dealing with noisy signals or defective sensors, several sources can be beneficial to ensure a confident recognition. When relations between events occurring on the different sources can be provided, it is interesting to take advantage of the global knowledge. This is achieved by multi source learning. In our case, for instance, relations between events on an blood pressure (ABP) channel and on electrocardiogram (ECG) channels are exploited to help in characterizing cardiac arrhythmias.

However, in a multi source learning problem, the amount of data and the expressiveness of the language increases compared to mono source learning. These are well known problems in Inductive Logic Programming (ILP). Indeed, the computation time of ILP algorithms grows with the amount of data and so does the size of the hypothesis search space when the expressiveness of the language increases. As pointed out by Fürnkranz [1], novel techniques are still needed to reduce the dimensionality of ILP problems, in particular, the example space and the hypothesis search space. In [2], Quinlan exposes a windowing technique improved later by Fürnkranz [3], to reduce the example space. To reduce the hypothesis search space, many methods have been proposed in ILP, one of them is using a declarative bias [4]. Such a bias aims either at limiting the search space or at deciding in what order hypotheses are to be considered and which hypothesis is better than another.

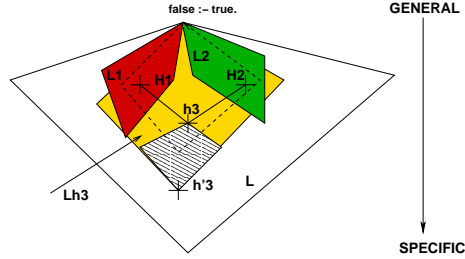


Fig. 1. Hypotheses search space for biased multi source learning

Designing an efficient global bias to cope with the multi source problems is a difficult task. We propose a divide-and-conquer strategy which consists in learning rules independently from each source and then, in using the learned rules to bias a new learning step on the whole dataset.

2 Biased multi source learning

We assume that the reader is familiar with the ILP paradigm (see [5] for details). In order to reduce the multi source learning search space we propose a method to construct a declarative language bias from a bottom clause [5] that bounds this space. In the following, we assume that a language bias can be built from a clause. The method to build such a bias is not detailed in this article.

We consider the case of data coming from two sources. This case can be straightforwardly extended to any number of sources. L_1 (resp. L_2) represents the language of source 1 (resp. 2), and L the multi source language ($L \supseteq L_1 \cup L_2$). A straightforward multi source learning method using the whole language L is complex and takes much computation time. To reduce the search space, and thus the learning complexity, we propose a two step method. Firstly, discriminating rules (H_1 and H_2) are learned on each source independently. The rules are used to select in the language L the literals that are useful to compute the multi source rules. Secondly, another learning step is performed on the whole dataset but with a reduced hypothesis language and so, a reduced search space.

Let L_{h_3} be the language created with the literals of H_1 and H_2 and all literals in L that describe relationships between elements of H_1 and elements of H_2 . To design the bias, we focus on two interesting clauses in the multi source search space (cf. Figure 2). The first one is the clause $h_3 = GSS(H_1 \cup H_2)$ (GSS : Greatest Specialization under θ -subsumption [6]). From this clause, the most restrictive bias can be created. However, this clause does not include any relations between the two sources, which can be interesting when some relationship exists. The second interesting clause is h'_3 , the most specific clause of L_{h_3} . When knowledge about the expected results is available, the bias can be restricted by choosing a clause more general than h'_3 but still more specific than h_3 (in the crosshatched area). An example of a final search space is represented

in Figure 2 by the dashed polygon. We proved that this search space is smaller than the search space associated to the whole language L and that the multi source rules learned with the bias that bounds the dashed area are at least as accurate as the mono source results for the same class. We also provide a way to design an efficient, while generative enough, multi source bias. Designing such a bias can be very complex when many sources of knowledge are involved and when complex relationships exist between elements of the different sources. In the presented method, the multi source bias is learned from partial learning steps that use biases simpler to design as they involve fewer knowledge.

3 Preliminary results

The two step method is compared to the straightforward approach. Experiments were done on five arrhythmias. We focus on two particular arrhythmias: supraventricular tachycardia (*svt*) and ventricular extra-systole (*ves*). The predictive accuracy of the learned rules is estimated by a “leave-one-out” cross-validation method because of the relatively small number of positive (about 6) and negative (about 30) examples for each arrhythmia. Rules learned for *esv* using the two methods are sound and complete. Accuracy is 0.973 for the *svt* class with both methods. This is caused by one exceptional negative example misclassified as true positive. The rules learned straightforwardly are less complex (they have about 25 % less literals). This can be explained by the fact that the bias is looser in the straightforward setting than in the biased setting: the shortest clause learned in the former setting has been excluded by the bias constructed from the mono source learned rules. However, the biased learning method reduces computation time very much. The whole computation (mono source then biased multi source) lasted 22.3 CPU seconds for the *ves* class and 49.89 CPU seconds for the *svt* class. These computation times are multiplied respectively by about 10 and 16 for the straightforward learning.

References

1. Fürnkranz, J.: Dimensionality reduction in ILP: A call to arms. In de Raedt, L., Muggleton, S., eds.: Proceedings of the IJCAI-97 Workshop on Frontiers of Inductive Logic Programming, Nagoya, Japan (1997) 81–86
2. Quinlan, J.R.: Learning efficient classification procedures and their application to chess end games. In Michalski, R.S., Carbonell, J.G., Mitchell, T.M., eds.: Machine Learning: An Artificial Intelligence Approach. Springer (1983) 463–482
3. Fürnkranz, J.: Integrative windowing. Journal of Artificial Intelligence Research **8** (1998) 129–164
4. Nedellec, C., Rouveirol, C., Ade, H., Bergadano, F., Tausend, B.: Declarative bias in ILP. In Raedt, L.D., ed.: Advances in Inductive Logic Programming. (1996) 82–103
5. Muggleton, S., De Raedt, L.: Inductive logic programming: Theory and methods. The Journal of Logic Programming **19** & **20** (1994) 629–680
6. Nienhuys-Cheng, S.H., de Wolf, R.: Least generalisations and greatest specializations of sets of clauses. Journal of Artificial Intelligence Research **4** (1996) 341–363