Propositionalization through Stochastic Discrimination

Bernhard Pfahringer and Geoffrey Holmes

Department of Computer Science
University of Waikato
Hamilton, New Zealand
{bernhard, geoff}@cs.waikato.ac.nz

Abstract. A simple algorithm based on the theory of stochastic discrimination is developed for the fast extraction of sub-graphs with potential discriminative power from a given set of pre-classified graphs. A preliminary experimental evaluation indicates the potential of the approach. Limitations are discussed as well as directions for future research.

1 Introduction

Propositionalization transforms relational learning problems into propositional ones. The transformation involves the identification of (potentially) useful relational sub-structures which are turned into boolean attributes indicating the presence or absence of the particular sub-structure in a specific example. Additional information can be extracted by counting the number of occurrences of a sub-structure, i.e. by generating a numeric attribute instead of a boolean attribute.

In principle any relational problem could be transformed this way without any loss of information, but in practice the generated propositional data-sets tend to have a huge number of attributes [1]. On the other hand there is a significant advantage of propositionalization: the state of the art of learning algorithms is much more advanced for propositional problems. Not surprisingly, there is a plethora of publications on propositionalization; [2] gives a good introduction and overview of the relevant methods and literature.

Another advantage of propositionalization is computational efficiency. Propositional systems tend to be much faster than relational ones, even though both kinds of systems have to cope with phase transitions in their search spaces [3, 4]. So in general, if both the search effort in the propositionalization phase as well as the number of generated attributes can be bounded reasonably, the combination of propositionalization and a propositional learner usually outperforms fully-fledged relational learners.

For propositionalization then the interesting question is: how can the search effort be limited but still generate a good set of structural features? Obviously one does not simply want to copy the workings of a standard relational learner, as such an approach would suffer from the same computational problems relational learners do. Basically there are two standard approaches currently in use for propositionalization, either imposing syntactic restrictions [5, 6] or using incomplete search (e.g. [7]).

In this paper we focus on the second approach by applying the theory of stochastic discrimination ([8]) to the the generation of sub-graphs of (labeled) graphs, i.e. we do not address general relational learning problems, but only a sub-class thereof.

In the following section stochastic discrimination will be explained. Section 3 will describe the application of stochastic discrimination to sub-graph generation. In Section 4 a preliminary experimental evaluation is conducted, and in the final section we describe our future road map for this project.

2 Stochastic discrimination: a one page explanation

Stochastic discrimination [8] was developed by a mathematician in the context of pattern recognition, which explains why it is almost unknown in the Machine Learning community. The basic idea is very simple:

- For each class, generate almost randomly thousands of features (or even more).
- For prediction use equal-weight voting over all these features and simply predict the class with the highest vote.

Obviously, the interesting step is the "almost random" generation of features. Given some random generator for features, a feature is accepted by the process when it is both what Kleinberg calls "enriched" and improves coverage towards "uniform coverage". A feature is called "enriched" for a class when the *percentage* of examples of this class covered by the feature is higher than the default percentage of this class in the training set. Note that for skewed class distributions enrichment can already be achieved by rules whose majority class *differs* from the class in question. For example, if in a two-class problem `class A` is represented by 10 out 100 training examples, then a feature covering 5 examples of `class A`, but also 10 examples of `class B` is "enriched" for `class A` ($\frac{1}{3} > \frac{1}{10}$), even though the majority class of the cover of that feature is still `class B`.

Enrichment on its own is not enough to guarantee good generalization and therefore good predictive accuracy on new examples. For the voting and averaging process described above the best generalization behavior is achieved when each training example of one class is covered by exactly the same number of features, i.e. when the average cover is equal to each individual cover, or to put it another way, when the variance of the coverage is zero. In practice it is not always possible to generate feature sets exhibiting such "uniform coverage", but the feature selection process is usually targeting examples that currently show below-average coverage.

Even though stochastic discrimination has been applied successfully in pattern recognition, there are not a lot of approaches applying it in a Machine Learning framework. One exception is the work by Ho on so-called "decision forests" as summarized in [9], where a lot of random and correct (and therefore obviously over-fitting) decision trees are generated and simply voted. Coverage is not a problem is this case as each tree covers each example, and enrichment is given by default, as the leaves of the overfitted tree are pure. Only conflicting instances (instances with identical attribute-values, but differing class-values) pose a problem, like they do for every learning algorithm.

Clearly there is a close relation to the "decision forest" algorithm in Machine Learning called "random forests" [10]. They differ by one detail in the tree generation process: random forest trees are generated from bootstrap samples instead of the whole training set, which should be an advantage in the case of noise in the training data. Random forests have been successfully applied to a few real-world problems recently, and some experiments show them to be competitive even with boosting, the current number one off-the-shelf ensemble learning technique.

So what are the limitations of random forests, or stochastic discrimination in general? Predictive performance is good to excellent, and that is all that is needed in pattern recognition. But for some Machine Learning applications predictive performance is needed as well as readability or intelligibility of the induced model. Certainly, no domain expert is going to inspect tens of thousands of features to determine whether they contain interesting new knowledge, or why they led to certain conclusions, etc.

This is the main reason we have taken a two-step approach for incorporating stochastic discrimination in a Machine Learning framework. Stochastic discrimination is used as a first and transformational step, generating lots of potentially useful features for a second step, which applies some arbitrary standard Machine Learning algorithm to the transformed representation.

3 SD applied to labeled graphs

In this paper we focus on the application of stochastic discrimination on one specific and interesting sub-problem in relational learning: the generation of propositional features from structural background knowledge. Specifically we will be looking at a situation where each training example is represented by a (labeled) graph, and the extracted features will be (labeled) sub-graphs. Typical application problems represented in such a way are biochemical applications trying to predict properties like mutagenicity, carcinogenicity, or biodegrability of chemical compounds. The 2D-structure of such compounds is naturally represented by labeled graphs.

Applying standard relational learning algorithms to such problems has been rather successful, but has always been plagued by runtime complexity [11]. Such algorithms usually perform some kind of heuristic hill-climbing always searching for the next best step. As the branching factor is huge in typical relational learning problems, this core step is destined to be slow. The process described below for randomly generating connected sub-graphs is extremely fast in comparison. Still, it also suffers one huge potential efficiency problem: in the worst case determining subgraph-isomorphism is unfortunately NP-complete. When randomly generated sub-graphs are checked for enrichment and coverage, it must be determined which training examples contain this sub-graph. Fortunately in practice, at least for labeled graphs like chemical compounds, the worst case is rarely encountered. Furthermore smart matching algorithms can utilize methods developed in the area of constraint programming like forward checking or the first-fail principle to speed up the matching process.

For the generation of random connected sub-graphs we have designed the following simple algorithm, which ensures that the generated sub-graphs will be enriched and improve coverage. It borrows an important idea from the Progol algorithm [12]: search can be guided efficiently by using focus examples. Only literals that are true for the particular focus example maybe used to extend partial rules.

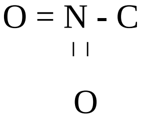
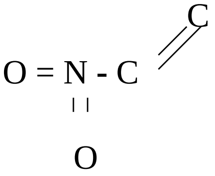
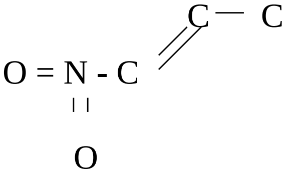
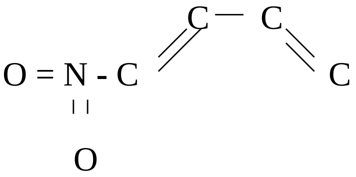
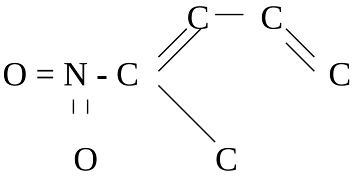
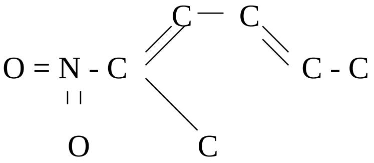
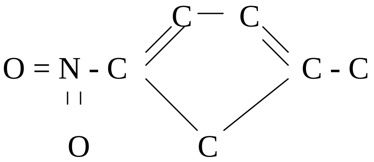
- For each class do N times (where N is a user defined parameter):
  1. Randomly choose an example whose coverage is currently below average.
  2. Randomize the list of all edges of the example graph, but ensure that every edge (except for the first one) in the list is connected to at least one other edge appearing prior to it in the list. Thus each prefix of the list describes a random connected sub-graph of the chosen example. Different randomizations lead to different sub-graphs.
  3. For each prefix ordered by size, determine its enrichment, and return the first (= smallest) prefix that is enriched for the class in question.

In the worst case this algorithm terminates with the full example itself, which will probably not be a very good feature, but at least it should be enriched (in general, but see below), and every rule returned will improve the coverage of the example in question, as it was an example suffering from below-average coverage. In theory, pathological cases are possible, e.g. a pattern covering every example of the class, thereby not changing the average coverage. First of all, such a feature is probably valuable, unless it prevents the generation of other useful features. Second, the random nature of the generation process seems to usually guard against such pathologies. The other problem with the above algorithm is a more general version of the conflicting examples problem mentioned in the previous section: if a graph A in one class is a proper sub-graph of some graph B in another class, it is impossible to construct sub-graphs of A which are not also sub-graphs of B. Therefore the algorithm can fail to find an enriched rule for some graphs. If the algorithm fails multiple times for the same example, we simply discard it.

The final propositionalization step for a set of graphs simply determines either the presence of each generated sub-graph in each graph or also counts the number of unique instances of the sub-graph in each graph. In the preliminary experiments reported in the next section we found that the latter usually yields better results.

We illustrate the mechanics of the algorithm with a simple example. Figure 1 depicts the full 2D structure of the mutagenic compound D63. Table 1 lists a sequence of randomly growing connected sub-graphs of compound D63 together

Table 1. Algorithm steps 1-9: a random connected subgraphs of compound D63.

Step	Subgraph	Active covered	Inactive covered
1	N - C	125	63
2	O = N - C	125	63
3		125	63
4		125	62
5		125	61
6		125	61
7		125	61
8		57	14
9		1	0

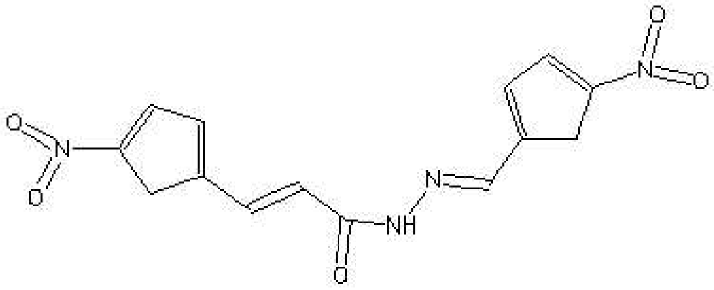


Fig. 1. Full structure of the mutagenic compound D63.

with their coverage information. In this example sub-graphs 4 to 9 would be enriched for class Active, as the percentage of examples of that class being covered by the respective sub-graph is strictly larger than the default percentage of class Active.

The current implementation actually chooses sub-graph 4, which is the smallest one being enriched. But judging from this example where sub-graph 8 looks rather more interesting from a chemical point of view, it would be better if any enriched sub-graph could be chosen in general. In the future we will modify the algorithm to allow this, maybe by selecting all enriched sub-graphs, or by just choosing a single one somehow. To make such a choice either a heuristic could be employed for trading off coverage versus accuracy (e.g. weighted relative accuracy [13] which is popular for subgroup discovery), or maybe one of the possible sub-graphs is just chosen at random (in the spirit of stochastic discrimination).

4 Preliminary experimental evaluation

For this preliminary experimental evaluation we have only employed a small and well-known chemical applications dataset: prediction of mutagenicity [14].

Only the 2D-structure formulas are used without any additional background knowledge, neither global (like molecular weights, luminosity, etc.) nor local (like Quanta types, charges, etc.) knowledge. Thus we only compare the results to the best result reported in [15] for a similar setting. Their Table 3 gives in row one (which is labeled BG1) an overview of results achieved with a variety of algorithms working solely from the 2D structure. The best result there was an accuracy of 79% achieved by the ICL system. This is rather similar to the results reported below for a binary representation with enough attributes, whereas counts usually perform better (see below for details). One should keep in mind that the results published here are not supposed to show the superiority of the new approach for realistic learning problems, but should just act as a kind of first sanity check ensuring that the new approach makes sense at all. This experiment has used only the 188 so-called "regression-friendly" compounds and has performed one ten-fold cross-validation using the standard predefined folds of this problem.

Table 2. Predictive accuracies for mutagenicity using various types and numbers of random structural attributes, and various classifiers. The size of a boosted stump ensemble is always equal to the number of attributes given. For comparison purposes: the ICL result using background knowledge $BG1$ was 79%.

AttrType	Number of Attrs	Ripper	linear SVM	boosted Stumps
boolean	50	72.34	77.13	72.87
	100	70.74	77.66	76.06
	200	71.28	79.26	79.79
counts	50	86.70	85.11	85.63
	100	84.04	84.04	86.17
	200	85.64	84.04	87.23

In Table 2 we give some sample results of what can be achieved by combining stochastic discrimination with a few different propositional learning algorithms. One batch of results is computed using boolean attributes, a second one is computed using counts. As is evident from the table, counts outperform a boolean representation.

Different algorithms were employed for their different properties:

- Ripper: usually generates a small, but accurate set of rules, therefore it should work well generating good as well as interesting rule sets for domain experts to examine.
- Linear SVM: does not perform any attribute selection, but does not seem to suffer from an abundance of even sometimes strongly correlated attributes, and generates a linear model, which can also be analyzed by domain experts. Interestingly, non-linear or RBF kernels mostly do not perform as well in this setting.
- Boosted Decision stumps: this is usually the slowest method (if enough iterations are performed for producing competitive accuracy), but can usually outperform the previous two methods in terms of predictive accuracy. Furthermore, it is still possible to inspect the generated ensembles, as they are basically just a sum of step-wise constant functions over the selected attributes. Additionally, the selected attributes have implicitly been ranked by the boosting algorithm, which can help interpretation as well.

Generally, the performance seems to be quite good in comparison to standard approaches, but there seems to be an optimal number of generated attributes, as some of the approaches apparently suffer from overfitting when presented with too many attributes.

5 Conclusion

In this paper we have introduced a new method for propositionalization based on the theory of stochastic discrimination. The example implementation described here can only construct connected sub-graphs of labeled graphs. A very preliminary experimental evaluation indicates the potential of the method.

As this is work in progress, there are quite a few obvious directions for future work, and a few less obvious ones as well:

- The most important first direction will be better integration into Weka. Currently a mixture of Prolog and Lisp programs is used to generate Arff files for Weka, whereas ideally one would like to be able to use a new data type (e.g. "SmilesString") that would allow the direct inclusion of a textual representation of a graph in an Arff file plus a number of Weka filters that would generate useful attributes from these graphs similar to the way string attributes for text mining are handled in Weka.
- At lot more experiments must be performed, incorporating larger datasets, and exploring the sensitivity of the algorithm to parameter settings. These experiments will also have to include accurate timing and comparison to fully-fledged ILP systems. Additionally, the utility of explicit feature subset selection and of different classifiers should be investigated.
- Alternatives to the current one-shot two-step process should be explored, especially closer integration of feature generation with the subsequent learning algorithm. The idea developed in [16] will be a good and interesting starting point for such work.
- Stochastic discrimination-based ideas will have to be generalized from graphs to arbitrary clauses to be of use for general ILP.

In summary we claim that stochastic discrimination can be used as theory for building effective and efficient relational learning systems. Future work will try to substantiate this claim.

References

1. De Raedt, L.: Attribute value learning versus inductive logic programming: The missing links (extended abstract). In Page, D., ed.: International Workshop on Inductive Logic Programming (ILP98). Volume 1446 of LNAI., Springer (1998) 1–8
2. Kramer, S., Lavrac, N., Flach, P.: Propositionalization approaches to relational data mining. In S., D., N., L., eds.: Relational Data Mining. Springer (2001) 262–291
3. Rückert, U., Kramer, S., De Raedt, L.: Phase transitions and stochastic local search in k-term dnf learning. In T. Elomaa, H. Mannila, H.T., ed.: Machine Learning: ECML 2002, Springer (2002) 405–417
4. Giordana, A., Saitta, L.: Phase transitions in relational learning. Machine Learning **41** (2000) 217–251

5. Lavrač, N., Džeroski, S.: Inductive Logic Programming: Techniques and Applications. Ellis Horwood (1994)
6. Kramer, S., Frank, E., Helma, C.: Fragment generation and support vector machines for inducing sars. SAR and QSAR in Environmental Research **13(5)** (2002) 509–523
7. Kramer, S., Pfahringer, B., Helma, C.: Stochastic propositionalization of nondeterminate background knowledge. In Page, D., ed.: International Workshop on Inductive Logic Programming (ILP98). Volume 1446 of LNAI., Springer (1998) 80–94
8. Kleinberg, E.M.: Stochastic discrimination. Annals of Mathematics and Artificial Intelligence **1** (1990) 207–239
9. Ho, T.K.: The random subspace method for constructing decision forests. IEEE Transactions on Pattern Analysis and Machine Intelligence **20** (1998) 832–844
10. Breiman, L.: Random forests. Machine Learning **45** (2001) 5–32
11. Železný, F., Srinivasan, A., Page, D.: Lattice-search runtime distributions may be heavy-tailed. In Matwin, S., Sammut, C., eds.: Twelfth International Conference on Inductive Logic Programming (ILP02). Volume 2583 of LNAI., Springer (2003) 333–345
12. Muggleton, S.: Inverse entailment and Progol. New Generation Computing, Special issue on Inductive Logic Programming **13** (1995) 245–286
13. Todorovski, L., Flach, P., Lavrač, N.: Predictive performance of weighted relative accuracy. In Zighed, D.A., Komorowski, J., Zytkow, J., eds.: 4th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD2000), Springer-Verlag (2000) 255–264
14. Srinivasan, A., Muggleton, S., Sternberg, M.J.E., King, R.D.: Theories for mutagenicity: A study in first-order and feature-based induction. Artificial Intelligence **85** (1996) 277–299
15. Laer, W.V., Raedt, L.D.: How to upgrade propositional learners to first order logic: A case study. In S., D., N., L., eds.: Relational Data Mining. Springer (2001) 235–261
16. Kramer, S.: Demand-driven construction of structural features in ilp. In Rouveirol, C., Sebag, M., eds.: Eleventh International Conference on Inductive Logic Programming (ILP01). Volume 2157 of LNAI., Springer (2001) 132–141