ILP through Propositionalization and Stochastic k-term DNF Learning

Aline Paes[1⋆], Filip Železný[2⋆⋆], Gerson Zaverucha[1⋆⋆⋆], David Page[3], and Ashwin Srinivasan[4]

[1] Federal University of Rio de Janeiro
{ampaes, gerson}@cos.ufrj.br
[2] Czech Institute of Technology in Prague
zelezny@fel.cvut.cz
[3] University of Wisconsin
page@biostat.wisc.edu
[4] IBM India Research Laboratory
ashwin@cse.iitd.ernet.in

1 Motivation

ILP has been successfully applied to a variety of tasks. Nevertheless, ILP systems have huge time and storage requirements, owing to a large search space of possible clauses. Therefore, clever search strategies are needed. One promising family of search strategies is that of stochastic local search methods. These methods have been successfully applied to propositional tasks, such as satisfiability, substantially improving their efficiency. Following the success of such methods, a promising research direction is to employ stochastic local search within ILP, to accelerate the runtime of the learning process. An investigation in that direction was recently performed within ILP [Železný et al., 2004].

Stochastic local search algorithms for propositional satisfiability benefit from the ability to quickly test whether a truth assignment satisfies a formula. As a result, many possible solutions (assignments) can be tested and scored in a short time. In contrast, the analogous test within ILP—testing whether a clause covers an example—takes much longer, so that far fewer possible solutions can be tested in the same time. Therefore, motivated by both the sucess and limitations of the previous work, we also apply stochastic local search to ILP but in a different manner. Instead of directly applying stochastic local search to the space of first-order Horn clauses, we use a propositionalization approach that transforms the ILP task into an attribute-value learning task. In this alternative search space, we can take advantage of fast testing as in propositional satisfiability. Our primary aim in this paper is to reduce ILP run-time.

The standard greedy covering algorithm employed by most ILP systems is another shortcoming of typical ILP search. There is no guarantee that greedy covering will yield the globally optimal hypothesis; consequently, greedy covering often gives rise to problems such as unnecessarily long hypothesis with too

⋆ Supported by CAPES
⋆⋆ Supported by the Czech Academy of Sciences through the project 1ET101210513 Relational Machine Learning for Biomedical Data Analysis
⋆⋆⋆ Supported by CNPq

many clauses. To overcome the limitations of greedy covering, the search can be performed in the space of entire theories rather than clauses. A strong argument against this larger search is the combinatorial complexity, giving us another reason to transform the relational domains into propositional ones and to use stochastic local search in the resulting, simpler search space. Therefore, our secondary aim in this work is to verify the benefits of a non-covering approach to perform search in ILP systems.

2 Stochastic Local Search in ILP through propositionalization

In recent work, a novel stochastic local search algorithm (SLS)[1] was presented to induce k-term DNF formulae [Rückert and Kramer, 2003]. The SLS algorithm performs refinements of an entire hypothesis rather than a single rule. A detailed analysis of SLS performance compared to WalkSAT shows the advantages of using SLS to learn a hypothesis as short as possible [Rückert and Kramer, 2003].

We specifically investigate the relevance of using stochastic local search to learn k-term DNF formulae in relational domains through propositionalization. To do so we implemented a k-term DNF formulae inducer using the SLS algorithm joined to the first-order feature construction part of the RSD system [Železný and Lavrac, 2006]. We compare the run-time when performing stochastic local search through propositionalization and the run-time when doing stochastic search and enumerative heuristic search directly in the relational space.

Experiments. Due to space limitations, in this short paper we only show the result of a single basic experiment[2] on the Mutagenesis classification problem, comparing the ILP system Aleph working in its default mode with the k-DNF stochastic search performed on the propositionalized data. The experimental steps were as follows: (1) Execute Aleph, requiring that each induced clause has at most 4 negative literals and accuracy of at least 0.8. Record the run-time. (2) Propositionalize the relational data with RSD using the same language declarations as used in Aleph.[3] Record the run time. (3) Set k to the number of rules produced by Aleph and S to the score on the training data achieved (calculated as # positive examples covered - # negative examples covered). (4) Execute repeatedly (1.000 times) the stochastic k-term DNF search on the

[1] The authors would like to thank Ulrich Rückert and Stefan Kramer for giving us their SLS code.

[2] The full set of experimental results will be presented at ILP 2006.

[3] Despite the same declarations, the theory spaces explored by the two approaches are necessarily different. On one hand, only a fraction of clauses explored by Aleph form a correct feature (as defined e.g. in [Železný and Lavrac, 2006]). On the other hand, the Aleph setting of maximum number of literals in a rule here translates into the maximum number of literals in a feature; however, in the subsequent k-DNF search, the features are combined in conjunctions and the total length of a single rule thus is unbounded.

propositionalized form of the data, each time terminated upon achieving score S. Record the run-time distribution.

Results. The Aleph execution terminated after 244.93 [s] yielding a theory with $k = 19$ rules and score $S = 106$. For these k and S, the stochastic search yielded the runtime distribution shown in Fig. 1. The mean run times are summarized in the table below, also taking into account the cpu time consumed by propositionalization.

Algorithm	**Cpu time [s]**
Aleph	244.93
k-term DNF SLS (mean)	0.21
k-term DNF SLS (mean) + propositionalization	15.37

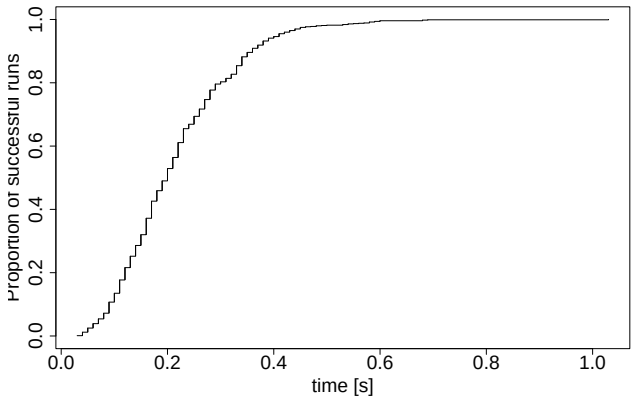

Fig. 1. The run-time distribution of the stochastic k-term DNF search on the Mutagenesis problem.

3 Conclusions

Two main observations follow: (1) a significant speed-up achieved by using stochastic k-term DNF search on the propositionalized form of the learning data, as compared to the default enumerative search conducted by Aleph, (2) the k-term SLS run-time distribution exhibits a rapid decay, unlike the heavy-tailed clause-search run-time distributions we observed in the relational domain [Železný et al., 2004].

References

[Rückert and Kramer, 2003] Rückert, U. and Kramer, S. (2003). Stochastic local search in k-term dnf learning. In *Proc. of the 20th ICML*, pages 648–655.

[Železný and Lavrac, 2006] Železný, F. and Lavrac, N. (2006). Propositionalization-based relational subgroup discovery with RSD. *Machine Learning*, 62(1–2):33–63.

[Železný et al., 2004] Železný, F., Srinivasan, A., and Page, D. (2004). A monte carlo study of randomised restarted search in ILP. In *Proc. of the 14th Int. Conf. on ILP*, volume 3194 of *LNCS*, pages 341–358. Springer. Extended version to appear in Machine Learning 2006.