MetaBayes: Bayesian Meta-Interpretative Learning using Higher-Order Stochastic Refinement

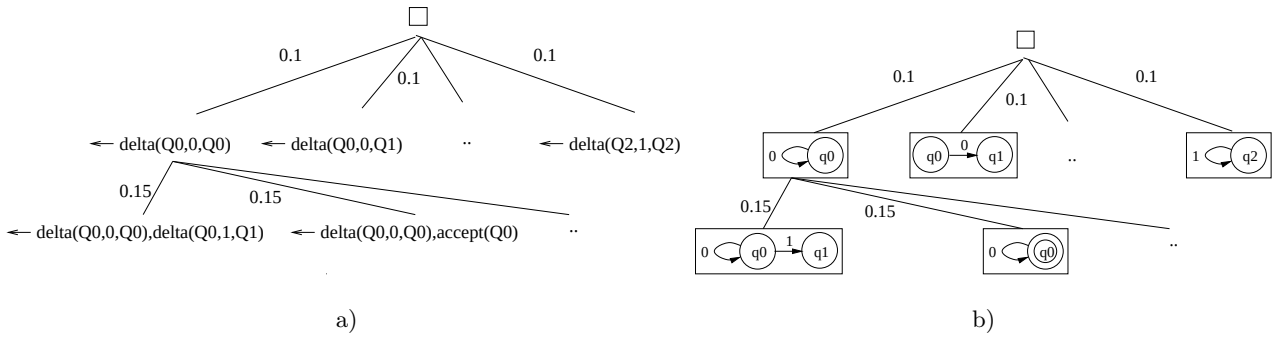
Stephen H. Muggleton, Dianhuan Lin, Jianzhong Chen and Alireza Tamaddoni-Nezhad

Department of Computing, Imperial College London

Abstract. Recent papers have demonstrated that both predicate invention and the learning of recursion can be efficiently implemented by way of abduction with respect to a meta-interpreter. In the separate cases of learning grammars, robot strategies and higher-order dyadic logic programs special purpose Prolog meta-interpreters have been shown to search the hypothesis up to 1000 times faster than state-of-the-art ILP systems. This paper shows that a generalised meta-interpreter of stochastic logic programs can be used to implement a Bayesian posterior distribution over the hypothesis space. We describe an implementation of this approach, MetaBayes, which uses higher-order stochastic refinement to randomly sample consistent hypotheses from the hypothesis space. Three variants of MetaBayes are introduced (MetaBayes$_{SR}$, MetaBayes$_{RS}$, MetaBayes$_{MAP}$) which respectively provide approximations to Bayes' prediction based on 1) Sampling consistent hypotheses with Replacement, 2) sampling consistent hypotheses without replacement (using Regular Sampling) and 3) returning a single hypothesis with Maximum A Posterior (MAP) probability. Experiments with learning grammars and family relations indicate that Bayes' prediction without replacement has significantly higher predictive accuracy than MAP, and in the case of family relations achieves 90% accuracy with on average around half the training set required by the MAP learner. However, this accuracy increase comes with up to 70 times increase in running time. We note that these results are comparable to known theoretical bounds for Bayes prediction and MAP.

1 Introduction

In [12] grammars are learned using a special-purpose Prolog meta-interpreter. Hypotheses are generated along various SLD derivation paths by abduction. The approach is generalised in this paper by 1) replacing the special-purpose meta-interpreter by a general-purpose meta-interpreter which interprets user-provided meta-rules and 2) treating the meta-rules as a Stochastic Logic Program (SLP) [10, 2]. In this setting we can view the hypotheses as being derived using Stochastic Refinement [15]. Figure 1 illustrates a Stochastic Refinement tree [15] for constructing a finite state acceptor. In this tree each path leading to a hypothesis can be interpreted either a) as a series of refinements leading to a headless Horn clause, representing the negation $\neg H$ of the ground abductive hypothesis H (see Figure 1a) or b) as the derivation of a finite state acceptor by a meta-interpreter applied to an SLP (Figure 1b).



a) b)

Fig. 1. Stochastic Refinement tree showing a) clause containing arcs (delta) and acceptors, b) corresponding Finite State Acceptors. Stochastic Refinement tree edge labels represent selection probabilities.

a) Generalised meta-interpreter

$prove([], Prog, Prog).$
$prove([Atom|As], Prog1, Prog2) :-$
$\quad metarule(RuleName, HO\_Sub, (Atom \text{ :- } Body), OrderTest),$
$\quad OrderTest,$
$\quad abduce(metasub(RuleName, HO\_Sub), Prog1, Prog3),$
$\quad prove(Body, Prog3, Prog4),$
$\quad prove(As, Prog4, Prog2).$

b) Meta-rules for finite state acceptors

$metarule(acceptor, [Q], ([Q, [], []] \text{ :- } []), (nonterm(Q))).$
$metarule(delta, [P, C, Q], ([P, [C|X], Y] \text{ :- } [[Q, X, Y]]),$
$\quad (nonterm(Q), nonterm(P))).$

c) Meta-rules for dyadic fragment

$metarule(instance, [P, X, Y], ([P, X, Y] \text{ :- } []), (pred(P))).$
$metarule(base, [P, Q], ([P, X, Y] \text{ :- } [[Q, X, Y]]),$
$\quad (pred\_above(P, Q), obj\_above(X, Y))).$
$metarule(tailrec, [P, Q], ([P, X, Y] \text{ :- } [[Q, X, Z], [P, Z, Y]]),$
$\quad (pred\_above(P, Q), obj\_above(X, Z), obj\_above(Z, Y))).$
$metarule(chain, [P, Q, R], ([P, X, Y] \text{ :- } [[Q, X, Z], [R, Z, Y]]),$
$\quad (pred\_above(P, R), obj\_above(X, Z), obj\_above(Z, Y))).$

Fig. 2. Prolog representation of a) Generalised meta-interpreter, b) Regular Grammar meta-rules and c) Dyadic fragment meta-rules

Furthermore, as in [1], we can view the SLP as a structural Bayes' prior over the hypothesis space. In this case, the posterior is formed by using the positive and negative examples to prune sub-trees from the prior. Following pruning, selection probabilities for each sub-tree are renormalised in the posterior.

2 MetaBayes Refinement framework

Setting The setting for Meta-Interpretive Learning (MIL) [12] assumes as input a specialised Meta-interpreter B_M together with two sets of ground atoms representing background knowledge B_A and examples E respectively. The result of learning is a revised form of the background knowledge containing the original background knowledge B_A augmented with additional ground atoms representing a hypothesis H. We assume H is derived from $B = B_A$ and E. Applying Inverse Entailment $B, H \models E$ is equivalent to $B, \neg E \models \neg H$. In this form we see that $B, \neg E$ is given to the meta-interpreter where $\neg E$ is a goal and the resulting abduced program $\neg H$ represents a headless Horn clause such as those shown in Figure 1a.

Generalised meta-interpreter A series of specific variants of special-purpose meta-interptreters are given in [12, 11] for Regular grammars ($Metagol_R$), Context-free grammars ($Metagol_{CF}$) and a fragment of Dyadic definite clause logic ($Metagol_D$). Figure 2a shows a generalised Meta-Interpreter which can emulate each special-purpose meta-interpreter using a set of domain specific meta-rules such as those shown for finite state acceptors (Figure 2b) [12] and the fragment of dyadic definite clauses (Figure 2c) investigated in [11]. As discussed in [11] a meta-rule is a higher-order wff

$$\exists \mathcal{S} \forall \mathcal{T} P(s_1, .., s_m) \leftarrow .., Q_i(t_1, .., t_n), ..$$

where $\mathcal{S}, \mathcal{T}$ are disjoint sets of variables, $P, Q_i \in \mathcal{S}$ and $s_j, t_k \in \mathcal{T}$. For instance, the second finite state acceptor meta-rule in Figure 2 indicates that with suitable higher-order ground substitition for the existentially quantified variables $\mathcal{S} = \{P, C, Q\}$ the higher-order atom $delta(P, C, Q)$ can be interpreted as the first-order clause $P([C|X], Y)$:- $Q(X, Y)$. In this way higher-order abduction of a set of atoms can be interpreted as first-order induction of a definite program.

Stochastic refinement According to [15] a downward *stochastic unary refinement operator* is a function $\sigma : G \rightarrow 2^{G \times [0,1]}$ defined as follows: $\sigma(C) = \{\langle D_i, p_i \rangle | D_i \in \rho(C),\ p_i \in [0, 1]$ and $\sum p_i = 1$ for

$1 \leq i \leq |\rho(C)|\}$ and $\sigma^*(C) = \{\langle D_i, p_i\rangle | D_i \in \rho^*(C)$, $p_i \in [0,1]$ and $\sum p_i = 1$ for $1 \leq i \leq |\rho^*(C)|\}$. In [15] it is shown that the n-step stochastic refinements of a clause represent a probability distribution. In the context of the meta-interpreter of Figure 2 we can consider the refinement function ρ to consist of the selection of a consistent meta-rule followed by the related abduction of a higher-order atom[1]. Stochastic refinement with respect to a meta-interpreter involves making selections according to a probability distribution over the meta-rules.

Prior, Likelihood and Posterior The prior of H relative to background knowledge B can now be defined as $Pr(H|B) = \sum_{\langle H,p\rangle \in \sigma^*(\neg B)} p$ and $Pr(H) = Pr(H|\emptyset)$. The likelihood of examples E with respect to the background knowledge B and hypothesis H is $Pr(E|B,H) = \begin{cases} 1 & \text{if } B, H \models E \\ 0 & \text{otherwise} \end{cases}$. Using Bayes' theorem the posterior is $Pr(H|B,E) = \frac{Pr(H|B)Pr(E|B,H)}{c}$ where c is a normalisation constant. A hypothesis H is said to be MAP in the case that $H \in \text{argmax}_H Pr(H|B,E)$. A Bayes' prediction of instance x is defined by the function $\text{BayesP}(x) = \begin{cases} 1 & \text{if } \sum_H Pr(H|B,E) \geq 0.5 \\ 0 & \text{otherwise} \end{cases}$.

3 Implementation

Below we describe three implementations of Bayesian learning: MetaBayes$_{SR}$, MetaBayes$_{RS}$ and MetaBayes$_{MAP}$. Each is a variant of the generalised meta-interpreter where stochastic refinement of the meta-rules is assumed to be conducted using a uniform distribution at each internal node of the refinement tree.

MetaBayes$_{SR}$ This algorithm carries out an approximation to Bayes' prediction based on averaging over the posterior probabilities of a multiset $\mathcal{H}_n$ consisting of a sample of n consistent hypotheses. Each $H \in \mathcal{H}_n$ is derived using a random SLP derivation of the examples E with respect to the background B. The refinement function ρ is implemented within a variant of the generalised meta-interpreter using Prolog's *findall* for each metarule selection (see Figure 2a). The resulting ρ set is pruned by testing consistency with the negative examples and the next sub-tree of the refinement tree is once more selected randomly. Although simple to implement, the approach is inefficient due to sampling of large numbers of duplicate hypotheses.

MetaBayes$_{RS}$ This algorithm also carries out an approximation to Bayes' prediction based on averaging over the posterior probabilities of a set $\mathcal{H}$ consisting of a sample of consistent hypotheses. The set is generated using a method we refer to as *Regular Sampling*, in which hypotheses are generated based on a series of fractions from the sequence $0, \frac{1}{2}, \frac{1}{4}, \frac{3}{4}, \frac{1}{8}, \frac{3}{8}, \frac{5}{8}, \frac{7}{8}, ...$ This sequence has the property of being evenly distributed in the unit interval $[0,1]$ without repeating the same fractional value twice. Considering the consistent hypotheses to be ordered $H_1, H_2, ..$ left-to-right in SLD order within the derivation tree, the fraction p_i is used to find the rightmost H_j such that $\sum_{k=1}^{j} Pr(H_k|B,E) \leq p_i$. This is achieved efficiently by considering that the cumulative posterior probability associated with hypotheses found in the sub-trees under each node of the stochastic refinement tree is partitioned into equally sized intervals. Starting at the root of the refinement tree H_j will be found in the sub-tree whose cumulative posterior probability interval $[\min, \max]$ is such that $\min \leq p_i < \max$. Within this sub-tree we repeat by selecting the sub-tree containing the probability $(p_i - \min)(\max - \min)$. The iteration is terminated by the hypothesis returned by the base case of the meta-interpreter. By bounding the posterior probability sum of the sample and ignoring duplicates the approach can be made to achieve the effect of sampling without replacement. Although slightly more complex to program

[1] abduce/3 only adds a higher-order atom a to a program P to give P' when $a \notin P$.

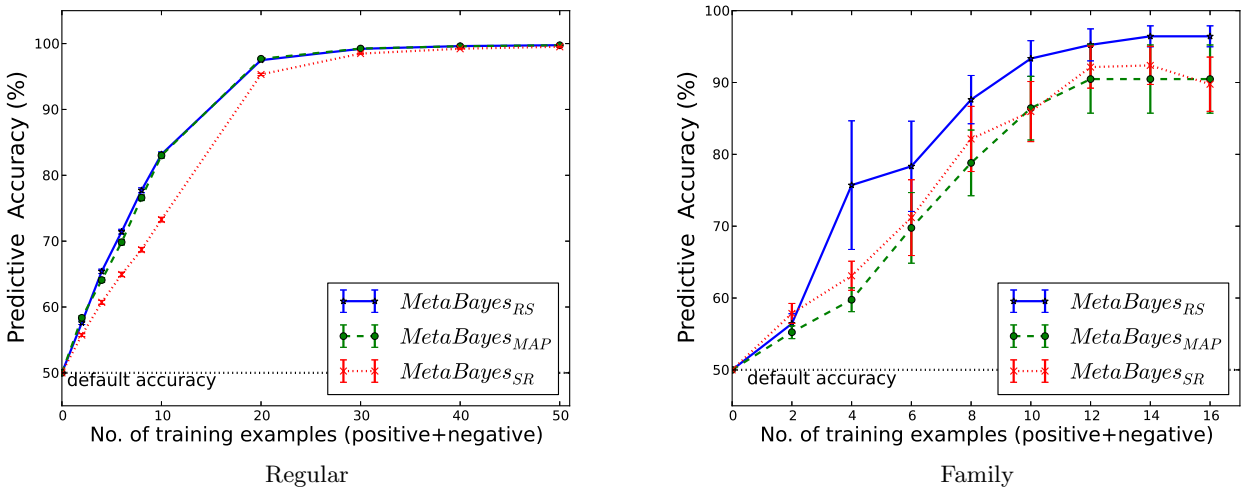


Fig. 3. Average predictive accuracies for learning Regular Grammars and Family Relationships

than MetaBayes$_{SR}$ the approach achieves higher efficiency by minimising duplicate sampling due to the spread of hypotheses chosen by the sequence of fractions.

MetaBayes$_{MAP}$ This algorithm carries out predictions based on the leftmost consistent hypothesis at minimal depth in the stochastic refinement tree. This hypothesis can be found efficiently using iterative deepening of derivations from the generalised meta-interpreter.

4 Experiments

In this section we describe experiments on learning regular grammars (RG) and family relationships.

Null hypothesis The predictive accuracy of MetaBayes$_{RS}$ is never significantly higher than MetaBayes$_{MAP}$.

Materials and Methods [2] In the RG experiment, 200 randomly chosen deterministic regular grammars were generated. Specifically, a set of sequences were randomly chosen from Σ^* for $\Sigma = \{a, b\}$, and used as input for Metagol$_R$, which ensures minimality of the generated grammars [12]. For each target grammar, 1500 different examples with half positive and half negative were randomly chosen from Σ^* for $\Sigma = \{a, b\}$. Among them, 1000 were reserved as a test data, while the other 500 were divided into 10 training sets of size 50.

For the experiment on learning family relationships, we used the dataset from [11], which is a small family tree consisting of 17 persons. The background knowledge contains only facts of $father/2$ and $mother/2$. The target hypothesis involves both recursion and predicate invention. The examples were a mixture of *parent*, *greatgrandparent* and *ancestor*, while examples of *grandparent* were not provided and left for predicate invention. A set of 80 positive and 80 negative examples were systematically generated. Half of them were reserved as a test data, while the other half was randomly divided into 5 training sets. Results from different training sets were averaged in both experiments.

The learning systems being compared were MetaBayes$_{RS}$, MetaBayes$_{SR}$ and MetaBayes$_{MAP}$. The sample size of MetaBayes$_{RS}$ was limited by its posterior probability sum ($SumPostProb$). In the RG experiment, we set $SumPostProb = 0.8$ while in the family relationships experiment we set $SumPostProb = 0.6$ due to the excessive running time for larger sample sizes. The sample rate in

[2] All datasets and learning systems in these experiments will be made available in any long version of the paper.

MetaBayes$_{SR}$ is decided by the number of samples n. In both experiments, we set $n = 10$ due to the constraint of running time.

Results and Discussion The results (Figure 3) from the family experiment show that MetaBayes$_{RS}$ outperforms MetaBayes$_{MAP}$ in terms of predictive accuracy with significant level 0.0032 (p-value in two-tailed t tests). Therefore the Null hypothesis is refuted. In the experiment of learning RG, MetaBayes$_{RS}$ has slightly higher accuracy than MetaBayes$_{MAP}$, in particular when the training sizes were 4 and 6, but their differences were not significant according to the two-tailed t test. This is due to the fact that a complex automata can be approximated by a smaller one. Thus the automata generated by MetaBayes$_{MAP}$ have comparable accuracy to those hypothesised by MetaBayes$_{RS}$. MetaBayes$_{RS}$ also significantly outperforms MetaBayes$_{SR}$. This is due to the relatively higher sample rate in MetaBayes$_{RS}$ due to a lack of duplicates.

Running times It should be noted that the running times for MetaBayes$_{RS}$ were up to 70 times higher than for MetaBayes$_{MAP}$. This can be explained by the fact that MetaBayes$_{RS}$ predictions are based on sampling more than half the probability volume of all consistent hypotheses in the space, while MetaBayes$_{MAP}$ simply uses the simplest consistent hypothesis for prediction.

5 Related work

According to Bayesian learning theory [5, 3], maximal predictive accuracy in learning is achieved by using a diversity of models, with predictions weighted according to the posterior probability of the corresponding hypothesis. In [8] error bounds for various Bayesian algorithms were analysed. The paper notes that while MAP maximises probability of exact identification of the target, it may have relatively high expected error. The paper goes on to show that the Gibbs algorithm, which randomly chooses a consistent hypothesis from the posterior distribution, has an error bound which is at most twice that of a Bayes's predictor (which is known to be optimal). These theoretical results are consistent with the experiments described in Section 4, and are also pertinent to a number of more *ad hoc* approaches to "model averaging" which have demonstrated significant predictive accuracy increases. These approaches are usually grouped under the title of *ensemble methods* and include *boosting* [6, 9] and *bagging* [4, 17]. Unlike the approach described in the present paper ensemble approaches use a more *ad hoc* approach to model-averaging, not based on an explicit Bayesian prior over the hypothesis space. However, the use of such a prior is directly comparable to the use of SLPs for sampling Bayes' nets investigated in [1]. The present paper extends this general approach to an ILP context, and demonstrates its predictive accuracy advantages in a context which supports the invention of relations and recursive programs. Within the ILP literature randomised search [16, 13] has been widely investigated. However, unlike the approaches described in this paper, these searches involve heuristic step-wise optimisation, rather than sampling and averaging predictions over a posterior distribution. The related areas of Probabilistic ILP (PILP) [14] and Statistical Relation Learning (SRL) [7] involve combining Bayesian inference and ILP, though this is in the context of Probabilistic Logic representations. The treatment of a set of meta-rules as an SLP is akin to this, though the logical reasoning is necessarily in terms of higher-order clauses rather than the probabilistic first-order representations used in PILP and SRL.

6 Conclusions and further work

This paper extends previous work on Meta-Interpretive Learning [12, 11] by demonstrating that a Bayesian prior can be implemented as a meta-interpreter over a stochastic logic program consisting

of higher-order meta-rules. The approach supports sampling of hypotheses consistent with a given set of examples and background knowledge, and has been used to implement various Bayesian algorithms involving approximations of Bayes' prediction and MAP. Our experiments indicate that approximated Bayes' prediction can significantly outperform MAP in terms of predictive accuracy, in line with theoretical predictions. Further work will address efficiency improvements in the algorithms as well as extensions to handle classification noise in the data and the use of Bayesian inference in active learning.

Acknowledgements

The authors would like to acknowledge the support of Syngenta in its funding of the University Innovations Centre at Imperial College. The first author would like to thank the Royal Academy of Engineering and Syngenta for funding his present 5 year Research Chair.

References

1. N. Angelopoulos and J. Cussens. Markov chain Monte Carlo using tree-based priors on model structure. In *UAI-2001*, Los Altos, CA, 2001. Kaufmann.
2. A. Arvanitis, S.H. Muggleton, J. Chen, and H. Watanabe. Abduction with stochastic logic programs based on a possible worlds semantics. In *Short Paper Proceedings of the 16th International Conference on Inductive Logic Programming*. University of Corunna, 2006.
3. J.M. Bernardo and A.F.M Smith. *Bayesian theory*. Wiley, New York, 1994.
4. L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.
5. W. Buntine. *A Theory of Learning Classification Rules*. PhD thesis, School of Computing Science, University of Technology, Sydney, 1990.
6. Y. Freund and R. Shapire. A decision theoretic generalisation of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55:119–139, 1997.
7. L. Getoor. Tutorial on statistical relational learning. In Stefan Kramer and Bernhard Pfahringer, editors, *Proceedings of the 15th International Conference on Inductive Logic Programming*, volume 3625, page 415, 2005.
8. D. Haussler, M Kearns, and R. Shapire. Bounds on the sample complexity of Bayesian learning using information theory and the VC dimension. *Machine Learning Journal*, 14(1):83–113, 1994.
9. H. Lodhi and S.H. Muggleton. Modelling metabolic pathways using stochastic logic programs-based ensemble methods. In *Proceedings of the 2nd International Conference on Computational Methods in System Biology*. Springer-Verlag, 2004.
10. S.H. Muggleton. Stochastic logic programs. In L. de Raedt, editor, *Advances in Inductive Logic Programming*, pages 254–264. IOS Press, 1996.
11. S.H. Muggleton and D. Lin. Meta-interpretive learning of higher-order dyadic datalog: Predicate invention revisited. In *Proceedings of the 23rd International Joint Conference Artificial Intelligence (IJCAI 2013)*, 2013. In Press.
12. S.H. Muggleton, D. Lin, N. Pahlavi, and A. Tamaddoni-Nezhad. Meta-interpretive learning: application to grammatical inference. *Machine Learning*, 2013. Published online.
13. S.H. Muggleton and A. Tamaddoni-Nezhad. QG/GA: A stochastic search for Progol. *Machine Learning*, 70(2–3):123–133, 2007. DOI: 10.1007/s10994-007-5029-3.
14. L. De Raedt, P. Frasconi, K. Kersting, and S.H. Muggleton, editors. *Probabilistic Inductive Logic Programming*. Springer-Verlag, Berlin, 2008. LNAI 4911.
15. A. Tamaddoni-Nezhad and S.H. Muggleton. Stochastic refinement. In *Proceedings of the 20th International Conference on Inductive Logic Programming*, pages 222–237, 2011.
16. F. Zelezny, A. Srinivasan, and D. Page. Lattice-search runtime distributions may be heavy-tailed. In S. Matwin and C. Sammut, editors, *Proceedings of the 12th International Conference on Inductive Logic Programming*, volume 2583, pages 333–345, 2003.
17. J. Zhu, H. Zou, S. Rosset, and T. Hastie. Multi-class adaboost. *Statistics and its Interface*, 2:349–360, 2009.