Extended Stochastic Logic Programs for Informative Priors over C&RTs

Nicos Angelopoulos and James Cussens

Department of Computer Science, University of York
Heslington, York YO10 5DD, UK
{nicos|jc}@cs.york.ac.uk

Abstract. A general method for defining informative priors on statistical models is presented and applied specifically to the space of classification and regression trees. Our aim is towards a Bayesian approach to learning such models from data, with the Metropolis-Hastings algorithm used to sample from the posterior. We present some preliminary results where we empirically tested the methodology.

1 Introduction

A key feature of the Bayesian approach to statistical inference is that prior knowledge (i.e. information available prior to data) can be incorporated into the learning process in a mathematically rigorous and conceptually clear manner. On the assumption that a space of possible statistical models can be defined, prior knowledge is expressed via a *prior distribution* over this space. This distribution is then updated with the data using Bayes theorem to produce a posterior.

The Bayesian framework is compellingly simple, but there are many complexities in applying it to real data analysis. Some of the complexities are being progressively alleviated by increased computing power and the subsequent use of Markov chain Monte Carlo (MCMC) techniques to sample from the posterior. In this paper our existing Bayesian method for defining and using a very flexible class of prior distributions over models is further developed and applied specifically to classification and regression tree (C&RT) models.

The method is fully Bayesian using a version of the Metropolis-Hastings MCMC algorithm which can sample from posterior distributions over the space of all possible C&RT models. Even after eliminating *a priori* uninteresting trees (for example, those containing sparsely populated leaves) this will be a very large space. The key feature of our approach is that the user can apply whatever prior knowledge s/he has available to restrict and/or bias the prior distribution over this space. Also, in contrast with other Bayesian C&RT work, MCMC sampling from the space is almost entirely driven by sampling from the prior.

2 Defining priors with stochastic logic programs

Our approach to defining priors is related to [1] and [2]. We specify a prior distribution over a space of models with a *stochastic program* rather than by some closed-form expression. A convenient language for defining such programs is *stochastic logic programming* (SLP). A stochastic logic program (SLP) can be used to define a prior over a space of statistical models by a two-step process. Firstly, a logic program is written which defines the desired model space (i.e. the *hypothesis space*). We define a unary predicate `cart/1` in a logic program such that each instantiation for `X` which makes `cart(X)` true is a term which represents a C&RT in the hypothesis space. We can exploit the expressiveness of first-order logic to represent C&RTs, choosing the most convenient logical representation. At this point any C&RT models we wish to exclude from the hypothesis space can be so excluded by adding the appropriate constraints to the logic program. For example, the program:

```
cart(leaf).
cart(Split-[L,R]) :- choose_split(Split), cart(L), cart(R).
choose_split(x1).   choose_split(x2).    choose_split(x3).
```

defines the space of binary-splitting C&RT models where there are only three binary attributes x_1, x_2 and x_3. Since the x_i are binary it suffices to supply the variable to determine the splitting rule—assume that data for which $x_i = 0$ go down the left branch and data for which $x_i = 1$ go down the right branch. To restrict the hypothesis space to have only trees where x_1 is the top node it is enough to define `cart_foo(X) :- X = x1-[L,R], cart(X).` and use `cart_foo/1` as the predicate defining the hypothesis space. Secondly, probability labels are attached to rules in the program thus converting it into an SLP. When the search for proofs has to choose between rules, a rule is chosen probabilistically according to the labels. In this paper, if a chosen rule does not lead to a proof, then backtracking will try from amongst untried rules, renormalising the probabilities as necessary. Now when the query `:- cart(X).` is asked, there is a distribution over possible instantiations for `X`.

We have extended the representational power of SLPs so that probability labels may be computed 'on the fly' rather than being fixed. For example, is an SLP based on our example above, where the probability of growing a tree by splitting a leaf depends on the depth of that leaf.

```
1 - 1/D :: [D] :: cart(leaf).
1/D :: [D] :: cart(Split-[L,R]) :- NewD is D+1, choose_split(Split),
                                   [NewD] :: cart(L), [NewD] :: cart(R).
choose_split(x1).  choose_split(x2). choose_split(x3).
```

3 Priors for Bayesian C&RT

A C&RT model T, maps an example described by a set of p predictor variables $X = X_1, X_2, \ldots X_p$ to a distribution over a response variable Y. In the case of classification trees Y is discrete and finite, and in the case of regression trees Y is continuous. Each T defines a conditional probability distribution $P(Y|T, X)$. For the purposes of determining a posterior distribution over trees, this conditional is a likelihood function derived by Bayes theorem: $P(T|Y,X) = \frac{P(T|X)P(Y|T,X)}{P(Y|X)}$ The key point is that the prior $P(T|X)$ is conditional on the predictor variables, thus we can use the observed values of X to help define sensible priors. This allows us to rule out *a priori* trees with leaves which contain few examples. Ruling out such trees is normal in this research area and greatly reduces the size of the model space. In our experiments only trees whose leaves have at least 5 examples are permitted. We use two types of priors over trees. The first, the GROW prior, is essentially that used by [1]. The stochastic process defined by this prior grows a C&RT tree by starting with a single leaf node and then repeatedly splits each leaf node η with a probability $\alpha(1+d_\eta)^{-\beta}$, where d_η is the depth of node η and α and β are prior parameters set by the user to control the size of trees. If a node is split, the splitting rule for that split is chosen uniformly. An abbreviated fragment of the SLP which expresses this prior is :

```
split_cart([H|T],Alpha/Beta) :- ...
            Split is Alpha*exp((1+D),-Beta), %split prob
            [Split] :: split_leaf(D,Ids,P,LV,RmLvPrs,NxLvPrs),
            split_cart(NxLvPrs,Alpha/Beta).
split_cart([],_ABeta ).
Split :: [Split] :: split_leaf(D,Ids,[Hp|Tps],LV,LvPrs,NwLvPrs) :-
            D1 is D + 1, ... %increase depth
            branch(Rest,ChL,PL),  branch(Rest,ChR,PR), ...
(1 - Split)::[Split]::split_leaf(D,Ids,P,[leaf(D,Ids,P)],Lvs,Lvs).
```

In the EDIT prior an 'initial' C&RT model is supplied. This can be any tree, for example the tree produced by the standard greedy algorithm. This tree is then probabilistically edited. Our approach is closely related to the proposal mechanisms of [1–3]. Hopefully, the SLP fragment in Table 1 conveys the general idea. With probability 4/5 the tree is edited with one of three equiprobable editions: growing, pruning and changing a splitting rule.

```
4/5 :: edit(CartIn,CartOut) :- one_edit(CartIn,CartMid), edit(CartMid,CartOut).
1/5 :: edit(CartIn,CartOut) :- one_edit(CartIn,CartEdt),
                               edcart_to_concrete_cart(CartEdt,CartOut).
1/3 :: one_edit( CartIn, CartOut ) :- ... % grow by splitting a leaf
1/3 :: one_edit( CartIn, CartOut ) :- ... % prune by merging neighbours
1/3 :: one_edit( CartIn, CartOut ) :- ... % change a splitting rule
```

Table 1. EDIT prior defined with an SLP

4 Experiments

We have used the Metropolis-Hastings (MH) algorithm for running MCMC simulations. Our main algorithm is effectively that presented in [4]. In brief, leaves in the stochastic SLD tree instantiate query variables to models. The MCMC makes stochastic proposals of next model in the chain. The algorithm accepts

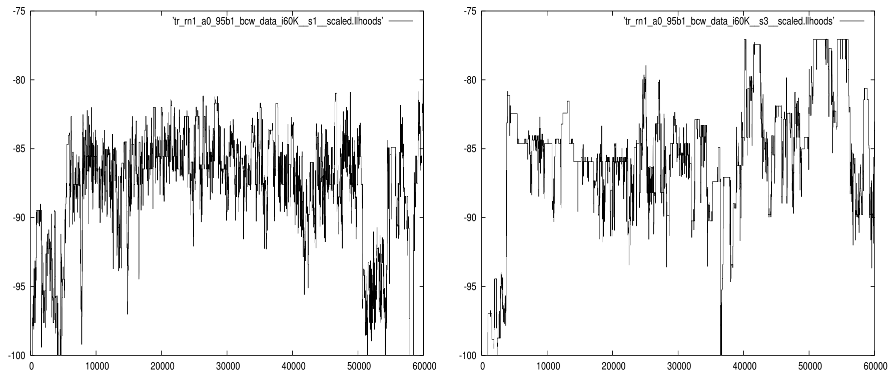

Fig. 1. Log-likelihood trajectories using two different seeds (low values truncated). Prior=GROW ($\beta = 1$), Data=BCW, MCMC=RN

this model with a stochastic choice which is influenced by the prior and the likelihood of the current and proposed models (i.e. the relative fitness of the two models for explaining the data). In our experiments we have used two datasets: the Wisconsin breast cancer (BCW) and the Kyphosis (K). BCW was used by [1] and it contains 16 missing data values the records of which were deleted. In both cases the machine learning task is binary classification using integer predictors. All splits are binary. There are 683 datapoints for BCW and 81 for K.

We have used the GROW prior with $\alpha = 0.95$ and $\beta \in \{0.5, 1, 1.5\}$ to give a bias towards big, medium and small trees. Generally speaking, the results are as expected: the method works but more work needs to be done in ensuring rapid convergence of the chain to the posterior. We used Sicstus Prolog 3.11 running under Linux. Running a chain for 60,000 iterations typically took 1-2 hours.

Comparison to greedy algorithm. The `rpart` (recursive partitioning) R package implements a version of the standard greedy algorithm. By default, 10-fold cross-validation is used to prune trees to avoid overfitting. The trees found by `rpart` for BCW and K using default settings have log-likelihoods -97 and -39, and sizes 7 and 5, respectively. By looking through the log of visited trees it is easy to find 'better' trees for many of the combinations of prior and MH variant that we use. For BCW we can find a tree with only 5 leaves and log-likelihood of -86 and for K a tree with only 3 leaves and log-likelihood of -36.

Analysing convergence Fig 1 records the trajectories of the log-likelihood as the MH algorithm moves through the space of trees. In both cases exactly the same experimental parameters have been used, just the random seed has been altered. We can see that although the distribution of log-likelihoods is similar in both cases, it appears that the two chains have wandered into different areas of tree space: the righthand trajectory has more plateaux and also reaches higher log-likelihoods than the lefthand one.

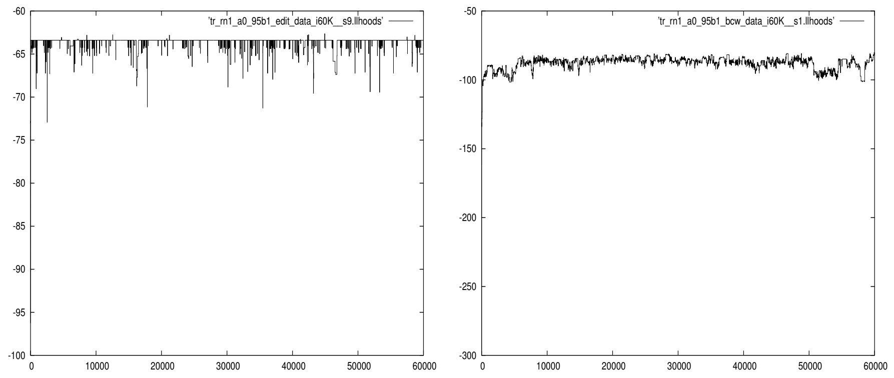

Fig. 2. Log-likelihood trajectories. In both cases: Data=BCW, MCMC=RN. For LHS: Prior=EDIT. For RHS: Prior=GROW ($\beta = 1$)

Influence of a strong prior Fig 2 compares log-likelihood trajectories using the EDIT prior and the GROW prior respectively with all other parameters being equal. The distinct horizontal line in the EDIT trajectory is clear evidence that the EDIT prior is pulling the Markov chain back to the initial tree.

5 Conclusions and future work

We have a working system (`http://www.cs.york.ac.uk/~nicos/sware/slps`) which permits the user to express prior information about model structure by writing an SLP. EDIT results show that prior information has the desired effect of significantly altering the posterior (at least for the moderately sized datasets used here). Our results, not just those presented, indicate that local jumps are essential to adequately explore the posterior. However, it appears that proposals which are heavily biased towards local jumps (e.g. RN used for Fig 1) lead to the chain getting stuck in local areas of tree space. Concerning actual implementation of the chain, our current program does not use Prolog's built-in backtracking: if this were possible this would permit a significant speed-up. A more radical approach is to adapt an existing Prolog system to have our probabilistic mechanisms built-in. Tabling could also be used advantageously.

References

1. Chipman, H.A., George, E.I., McCulloch, R.E.: Bayesian CART model search. Journal of the American Statistical Association **39** (1998) 935–960
2. Denison, D.G.T., Mallick, B.K., Smith, A.F.M.: A Bayesian CART algorithm. Biometrika **85** (1998) 363–377
3. Denison, D.G.T., Holmes, C.C., Mallick, B.K., Smith, A.F.M.: Bayesian Methods for Nonlinear Classification and Regression. Wiley (2002)
4. Angelopoulos, N., Cussens, J.: Markov chain Monte Carlo using tree-based priors on model structure. In Breese, J., Koller, D., eds.: Proc. UAI-01, Seattle (2001)