AN ACCELERATED ALGORITHM FOR MCMC BAYESIAN DECISION TREE SAMPLING

Gilles Blanchard

Fraunhofer FIRST, Berlin

MPI Tübingen Ensemble Learning Worskhop

Plan

1- Decision tree models

2- Bayesian framework

3- MCMC simulations of posterior

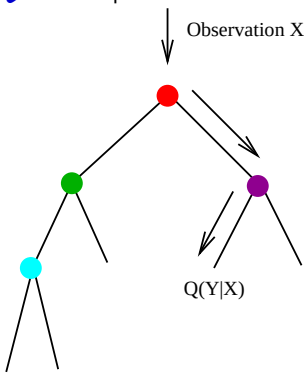
4- The accelerated algorithm

5- Examples

6- Conclusion

Decision trees for value prediction

$X \in \mathcal{X}$ observation, $Y \in \mathcal{Y}$ value to predict:



Parametrization of Decision Trees

We consider decision trees models:

$m = (T_m, K_m, \Theta_m)$

T_m : binary tree structure (we assume that there is a maximum depth d).

K_m : set of "questions" (from a pool $\mathcal{Q}$) attached to the internal nodes of T.

$\Theta_m = (\theta_s, s \in \delta T_m)$: set of parameters attached to external nodes (leaves) of T_m.

For each terminal leaf $s \in \delta T$, parameter θ_s determines a distribution $Q_s(y|x)$ over $\mathcal{Y}$.

Decision Trees Models

Examples:

Classification:

$Q_s(y)$ is a multinomial distribution on the set of classes.

Regression:

(1) $Q_s(y)$ is a Gaussian ditribution $\mathcal{N}(c, \sigma^2)$.

(2) $Q_s(y|x) = \mathcal{N}(\beta^\top x + b, \sigma^2)$ (linear model).

For $X \in \mathcal{X}$, denote $m(X)$ the node reached by X using the tree model m. Then the distribution $P(Y|X)$ modeled by m is

$$Q_m(Y|X) = Q_{m(X)}(Y)\,,$$

i.e. the distribution for Y found in the leaf reached by X.

Bayesian framework

In a Bayesian framework we assume some prior π on the set of models m. Remember each m determines a certain $Q_m(Y|X)$.

This determines a joint model for the model m and the data (X_i, Y_i):

$$Q(m, (Y_i)|(X_i)) = \pi(m) \prod_i Q_m(Y_i|X_i)\,.$$

Important quantities are then:

Posterior on the model

$$\mu_{post}(m) = Q(m|(Y_i), (X_i)) = \frac{\prod_i Q_m(Y_i|X_i)\pi(m)}{\int_{m'} \prod_i Q_m(Y_i|X_i) d\pi(m')}\,;$$

Bayesian framework

Predictive Bayesian distribution

$$\mu_{pred}(Y|X) = Q(Y|X, (Y_i), (X_i)) = \int_m Q_m(Y|X) d\mu_{post}(m)\,;$$

Expectation *a posteriori* of a real function $h(Y)$ for a given X:

$$I_h(X) = \int_Y h(Y) d\mu_{pred}(Y|X)\,;$$

Maximum *a posteriori*:

$$m_0 = Argmax_m \mu_{post}(m)\,.$$

Generalities on MCMC simulation of posterior

MCMC methods are often used to simulate the posterior (to perform approximate integration).

Metropolis-Hastings method: define a reversible Monte-Carlo chain on the set of models such that the equilibrium distribution is precisely the posterior.

Start from an (arbitrary) reversible chain with transition probability $\rho(m, m')$;

Then define the Markov chain with

$$\widetilde{\rho}(m, m') = \rho(m, m') \min\left(1, \frac{\rho(m', m)\mu_{post}(m')}{\rho(m, m')\mu_{post}(m)}\right).$$

Then obviously

$$\mu_{post}(m)\widetilde{\rho}(m, m') = \mu_{post}(m')\widetilde{\rho}(m', m)\,,$$

which implies that μ_{post} is the equilibrium distribution of the chain.

Generalities on MCMC, continued

The Metropolis-Hastings algorithm is very practical because we only have to compute ratios

$$\frac{\mu_{post}(m)}{\mu_{post}(m')} = \frac{\pi(m)}{\pi(m')} \frac{\prod_i Q_m(Y_i|X_i)}{\prod_j Q_{m'}(Y_i|X_i)};$$

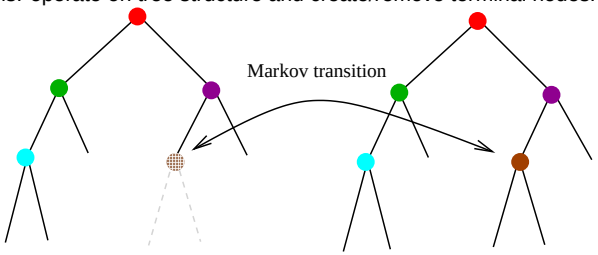
in other words the normalizing constant does not have to be computed.

Furthermore, once this quantity is computed the chain can be realized by a simple accept/reject transition mecanism.

MCMC for trees, fi rst version

We present a first version for MCMC inspired by (Denison, Mallick and Smith 1998) and (Chipman, George, McCulloch 2002).

Transitions: operate on tree structure and create/remove terminal nodes.



Back to trees. . .

To ease things we will consider a "simplified" Bayesian setting for trees in which θ_m is fixed for each decision tree structure (T_m, K_m).

For example, θ_m is estimated for each model by another data sample, or by some other method.

In this setting m is discrete and the only parameters are (T_m, K_m).

(A fully Bayesian approach is possible! This simplification is only in hope to have a clearer presentation of the idea)

MCMC for trees, fi rst version, continued

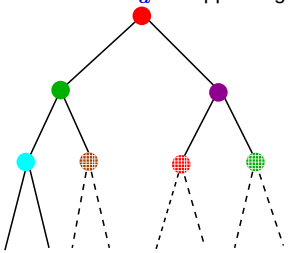
Advantage: the posterior ratio is "local" and only involves the examples and distributions for the terminal nodes concerned. The rest cancels out.

Inconvenient: there are many deep local minima in the "landscape" of the Markov Chain. Requires many restarts for suitable sampling.

The factorization idea, 1

General idea: do not perform Markov Chain on tree structure, only on questions.
Integrating the posterior on tree structure given the questions can be done in an exact way!

For this, consider a slight change of parametrization of the model: K_m is now the set of questions on the internal nodes of the complete binary tree $\mathcal{T}_d$ of depth d. T_m is a subset of $\mathcal{T}_d$ (questions at the nodes of $\mathcal{T}_d$ not appearing in T_m are just ignored!)



The factorization idea, 2

Assumptions on the prior:

$$\pi(T_m, K_m) = \pi_1(T_m)\pi_2(K_m)\,.$$

and π_1 is a binary branching process of parameter ρ restricted to trees of depth $\leq d$:

$$\pi_1(T_m) = C\rho^{|T\setminus\delta T|}(1-\rho)^{|\delta T|}.$$

We must be able to compute two things for the factorization idea:

- the marginal posterior on the questions K_m (up to normalization constant)
- the integration of a target function over the posterior on tree structure T_m given fixed questions K_m (this includes normalization constant)

Marginal posterior on questions

$$\begin{aligned}\mu_{post}(K) &= \sum_{T\in\mathcal{T}_d} \mu_{post}(T,K)\\ &\propto \pi_2(K)\sum_{T\in\mathcal{T}_d}\pi_1(T)\prod_{i=1}^{N} Q_{T,K}(Y_i|X_i)\\ &\propto \pi_2(K)\sum_{T\in\mathcal{T}_d}\prod_{s\in\delta T}\left((1-\rho)\prod_{i:s\in K(X_i)} Q_s(Y_i)\right)\prod_{s\in T\setminus\delta T}\rho\,.\end{aligned}$$

We can now use the following fact: any sum of the form

$$S(f,g) = \sum_{T\in\mathcal{T}_d}\left(\prod_{s\in\delta T} f(s)\prod_{s\in T\setminus\delta T} g(s)\right),$$

can be calculated efficiently by a single pass through the nodes of $\mathcal{T}_d$ (context tree weighting algorithm, Willems et al. 1995)

Factorized computation

$$S(f,g) = \sum_{T\in\mathcal{T}_d}\left(\prod_{s\in\delta T} f(s)\prod_{s\in T\setminus\delta T} g(s)\right),$$

Define a function σ on the nodes of $\mathcal{T}_d$ with

$$\sigma(s) = f(s)$$

for s terminal node (leaf) of $\mathcal{T}_d$; and

$$\sigma(s) = f(s) + g(s).\sigma(\text{left child}(s))\sigma(\text{right child}(s))\,;$$

for s internal node of $\mathcal{T}_d$.

Then

$$\sigma(\text{root}) = S(f,g)\,.$$

Exact integration over T given K

Integration of a certain function $H(m(X))$ over T given K has the form

$$S(X) = \sum_{T \in \mathcal{T}_d} \mu_{post}(T|K) H(m(X))$$
$$= \frac{\sum_{T \in \mathcal{T}_d} H(m(X)) \pi_1(T) \prod_i Q_{T,K}(Y_i|X_i)}{\sum_{T' \in \mathcal{T}_d} \pi_1(T') \prod_i Q_{T',K}(Y_i|X_i)}$$

Again, numerator and denominator can be put under the form allowing fast computation.

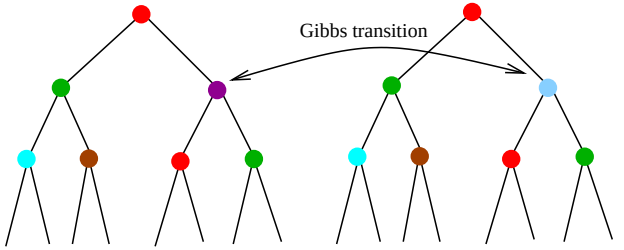
Comparison with the previous method

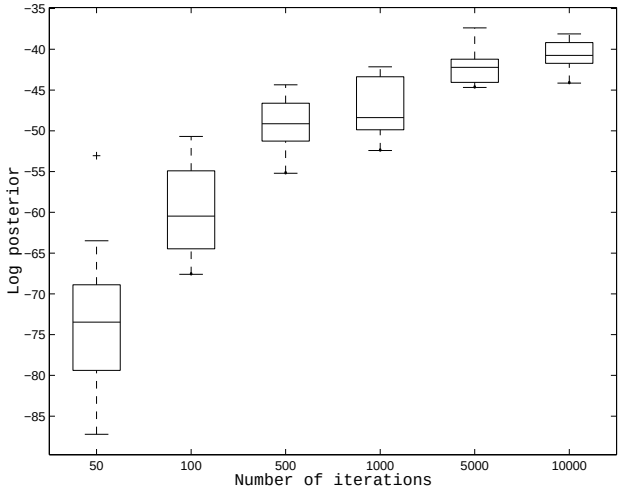
- each step is more expensive in computation (up to 2^d nodes to update is the root question is changed)
- on the other hand, the number of models averaged at each step is of order 2^{2^d}. This entails a massive gain in the number of models effectively sampled.
- in practice the landscape of energies seems less “canyoned”.

The algorithm: a Gibbs sampler for trees

To sample from the marginal posterior on questions, we adopt the following “Gibbs sampler” strategy:

- At each step, choose an internal node and a new question at random for this node.
- Accept or reject the transition by calculating the marginal posterior ratio.





For true Bayesians...

One would like to also adopt a Bayesian point of view for the parameters Θ_m. In particular because it entails the "Occam's razor" effect which Bayesians like (lead to penalization of models with more parameters). Then we also need to integrate out the Θ_m. Assume the prior on $\Theta_m = (\theta_s, s \in \delta T)$ is of the form

$$\pi_3(\Theta_m|T,K) = \prod_{s\in\delta T} \pi_3(\theta_s)\,.$$

Then the marginal posterior on (T,K) is of the product form

$$\begin{aligned}\mu_{post}((T,K)) &= \pi_1(T)\pi_2(K)\int_\Theta \prod_i Q_{T,K,\Theta}(Y_i|X_i)d\pi_3(\Theta)\\ &= \pi_1(T)\pi_2(K)\prod_{s\in\delta T}\int_{\theta_s}\prod_{i:m(X_i)=s} Q_{T,K,\Theta}(Y_i|X_i)d\pi_3(\theta_s)\,.\end{aligned}$$

If this integral can be computed (using standard parametric Bayesian tools), the product structure of the posterior marginal entails that we can apply the same principles as before.

Example: for classification and a uniform prior π_3 on the Bernoulli parameter $\theta_s = Q_s(Y=1)$, then we have

$$\int_{\theta_s}\prod_{i:m(X_i)=s} Q_{T,K,\Theta}(Y_i|X_i)d\pi_3(\theta_s) = \int_\theta \theta^{n_{1,s}}(1-\theta)^{n_{0,s}}d\theta = \binom{n_s+1}{n_{1,s}}\,.$$

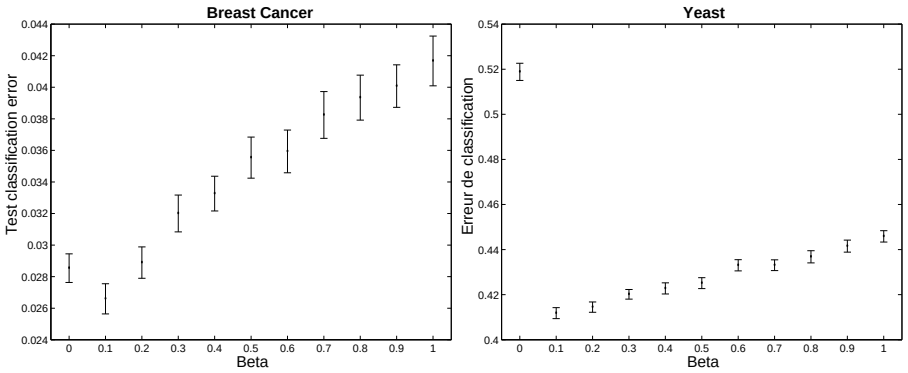
Example in regression: choosing conjugate priors for the Gaussian linear model leads to exactly computable integrals.

Extensions

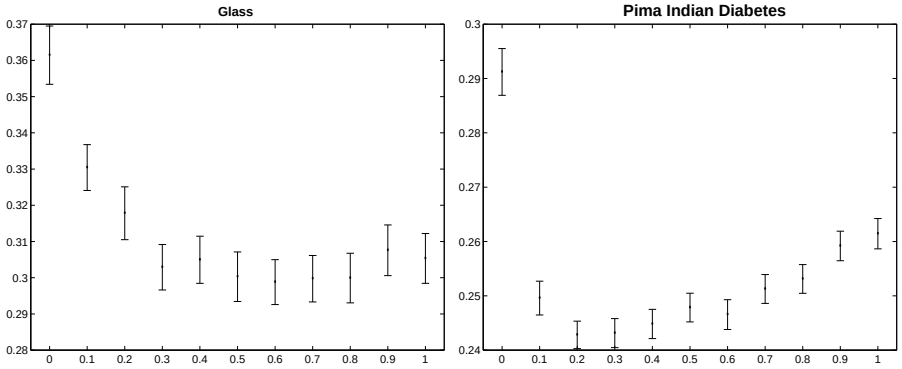
- More general prior: branching parameter can depend on the node (allows control over prior shape of trees).
- Can be used for finding the model with maximum *a posteriori* by some straightforward modifications (need to be able to find the MAP structure T for fixed K).
- we can also apply the algorithm to "annealed" posteriors (Gibbs-estimators style) of the form

$$P_\beta(m, Y_i|X_i) \propto \pi_(m)\left(\prod_i Q_m(Y_i|X_i)\right)^\beta$$

Some proof of concept experiments (classifi cation)



Experiments



Conclusion

►Bayesian trees allows to combine trees using the posterior distribution on tree space.

►The principle of MCMC tree sampling has been proposed in previous works showing its interest. Can be used for prediction, importance of variables estimation, etc. . .

►We developped an improved MCMC approach allowing to sample models in a more efficient way, based on a “factoring” idea.

►Allows to combine more models and/or to do a more efficient model search.