Variational Bayes for high-dimensional linear regression with sparse priors

Théo Gnassounou, Nicolas Nguyen
Bayesian Machine Learning, Master MVA
theo.gnassounou@gmail.com, nicolas.nguyen@etu.emse.fr

Abstract

Variational Bayes (VB) has gained popularity in approximate inference problems in recent years. This paper [1] proposes a mean-filed spike and slab VB approximation to Bayesian model selection priors in high-dimensional linear regression. The paper proposes a new model based on Laplace slab prior and study its theoretical and numerical properties. We test this model on real data and concrete problem and show that the method successes to find sparse representation of regression parameter in a high-dimensional setting.

1 Introduction and problem setup

Consider a sparse linear regression problem for n samples and p parameters with typically $p \gg n$ (sparse setting). The linear regression problem is given by

$$Y = X\theta + Z \ , \tag{1}$$

where

$$Y \in \mathbb{R}^p, X \in \mathbb{M}_{n,p}(\mathbb{R}), \theta \in \mathbb{R}^p, Z \sim \mathcal{N}(0, I_n)$$

.

The particularity of this setup is that θ is sparse, meaning that it contains many zeros values. The Bayesian model selection appears in random variables which model probabilistic weights for each entries of θ_i.

One crucial choice consists in choosing the model selection prior. A natural way to construct this prior is to consider the following hierarchical prior :

1. Select a dimension s from a prior π_p on $\{0, .., p\}$: $s \sim \pi_p(s)$
2. Select a subset S of $\{1, ...p\}$ of cardinal s with uniform probability : $S||S| = s \sim \mathcal{U}(S)$
3. Select a set of $(\theta)_{i,i\in S}$ of non-zeros values from a chosen prior (called slab prior).

One way to summarise this approach is to consider a spike-and-slab prior for the parameters $(\theta_i)_i$ where

$$\theta_i | z_i \sim z_i p_{slab}(\lambda) + (1 - z_i)\delta_0 \ , \tag{2}$$

and where $(z_i)_{i\in[1,p]}$ are the latent variables described above and λ a hyperparameter characterising the slab prior p_{slab}.

As we will show in next sections, the choice of p_{slab} directly influences the performances of the algorithm. Previous work deal with Gaussian prior slab, meanwhile the involved paper uses Laplace prior slab. Using Gaussian slabs induces closed-form formulas for the variational parameters update, but it can yield poor performances due to excessive shrinkage of the estimated coefficient. We will compare the 2 approaches in section 3 .

Using a spike-and-slab prior is a classical way to handle Bayesian model selection when facing sparsity. However the main goal of this approach is to makes the computation scalable, since searching over all possible models involves searching over 2^p models which is unfeasible for large values of p. However approximate inference techniques which permit large-dimensional computations. 2 main approaches have recently been studied for Bayesian approximations : MCMC and Variatonal Bayes. While MCMC are known to give nice theoretical properties on convergence, it is usually too computationally expensive. Another approach is Variational Bayes, aiming at approximate the posterior distribution with simpler distributions.

2 Variational approximation

2.1 Principle

VB permits to approximate the posterior distribution $\Pi(.|Y)$ associated to the prior defined by (2). One main assumption is to take a VB approximation using the mean field family, which is a common hypothesis in VB literature.

A natural way to approximate the posterior is to suppose a spike-and-slab distribution with Gaussian slabs (but Laplace slab true prior) :

$$\mathcal{P}_{MF} = \{P_{\mu,\sigma,\gamma} = \otimes_{i=1}^{p}(\gamma_i \mathcal{N}(\mu_i, \sigma_i^2) + (1-\gamma_i)\delta_0)\} \tag{3}$$

Where $\mu \in \mathbb{R}^p, \sigma \in \mathbb{R}^p+, \gamma_i \in [0,1]^d$ are the variational parameters.

VB aims at minimising the KL-distance between an element of the family $\mathcal{P}_{MF}$ and the true posterior, meaning finding an approximate $\tilde{P}i$ such that :

$$\tilde{\Pi} = \underset{P_{\mu,\sigma,\gamma} \in \mathcal{P}_{MF}}{\text{argmin}} \; KL(P_{\mu,\sigma,\gamma} || \Pi(.|Y)) \tag{4}$$

2.2 Variational parameters update

From this one can compute the minimizer of this KL distance by minimising the following functions :

$$f_i(\mu_i|\sigma, \mu_{-i}, \gamma, z_i = 1) = \mu_i \sum_{k \neq i} (X^T X)_{ik} \gamma_k \mu_k + \frac{1}{2}(X^T X)_{ii} \mu_i^2 - (Y^T X)_i \mu_i + \lambda \sigma_i \sqrt{\frac{2}{\pi}} e^{-\frac{\mu_i^2}{2\sigma_i^2}} + \lambda \mu_i (1 - 2\phi(-\frac{\mu_i}{\sigma_i})) \tag{5}$$

Where we denote the notation u_{-i} as the vector u without the i^{th} component, and ϕ the cumulative distribution function of $\mathcal{N}(0,1)$.

$$g_i(\sigma_i|\mu, z_i = 1) = \frac{1}{2}(X^T X)_{ii} \sigma_i^2 + \lambda \mu_i \sigma_i \sqrt{\frac{2}{\pi}} e^{-\frac{\mu_i^2}{2\sigma_i^2}} + \lambda \mu_i (1 - \phi(\frac{\mu_i}{\sigma_i})) - log(\sigma_i) \tag{6}$$

The parameters μ_i and σ_i are respectively the arguments which minimise the 2 functions described above. The last parameter γ_i is given by the closed-formed formula $\gamma_i = logit^{-1}(\Gamma_i(\mu, \sigma, \gamma_i))$ where Γ is defined as :

$$\Gamma_i(\mu, \sigma, \gamma_i) = \log(\frac{a_0}{b_0}) + \log \sqrt{\frac{\pi}{2}} \sigma_i \lambda + (Y^T X)_i \mu_i - \mu_i \sum_{k \neq i} (X^T X)_{ik} \gamma_k \mu_k - \frac{1}{2}(X^T X)_{ii}(\sigma^2 + \mu_i^2) \tag{7}$$

$$- \lambda \sigma_i \sqrt{\frac{2}{\pi}} e^{-\frac{\mu_i^2}{2\sigma_i^2}} - \lambda \mu_i (1 - 2\phi(-\frac{\mu_i}{\sigma_i}) + \frac{1}{2} \tag{8}$$

Since minimising f_i and g_i is a non-convex and non-trivial problem, there is no closed-form formula and we used the SCIPY package to handle the optimisation procedure. Section 2.3 gives more details about the numerical challenges.

2.3 Prioritised updating order

CAVI algorithm is sensitive to initialisation and updating order because the VB objective function is often non-convex. Updating parameters in lexicographic order leads to poor performances. One way is to randomise the choice of the updating order. Nevertheless experiment (see 3) shows that, because of the large number of local minima, random order performs also poorly. The authors propose a new approach to prioritise the order of the update. First, the estimator $\hat{\mu}^{(0)}$ which is the mean of the vector μ is compute. The order is chosen according to the decreasing order of the absolute value of $\hat{\mu}^{(0)}$ i.e. describe by $\mathbf{a} = (a_1, \ldots, a_p)$ the permutation indices where $|\hat{\mu}^{(0)}_{a_i}| \geq |\hat{\mu}^{(0)}_{a_j}|$ with $i < j$

This method allows to update bigger coefficient first and avoid to assign signal strength to small coefficient.

2.4 Conditions on the design matrix X

This section states some assumptions which are necessary for the model in this sparse setup.

Previous work suggest a exponentially decreasing prior, meaning that there are constant $A_1, A_2, A_3, A_4 > 0$ such that :

$$A_1 p^{-A_3} \pi_p(s-1) \leq \pi_p(s) \leq A_2 p^{-A_4} \pi_p(s-1) \quad \forall s = 1, ..., p. \tag{9}$$

Moreover, the regularisation parameter has to vary with p in the following range :

$$\frac{\|X\|}{p} \leq \lambda \leq 2\bar{\lambda} \qquad \bar{\lambda} = 2\|X\|\sqrt{\log p} \tag{10}$$

By defining for a model $S \subseteq \{1, .., p\}$ the *compatibility number* $\phi(S)$ and the *smallest scaled sparse singular value* $\tilde{\phi}(S)$ such that :

$$\phi(S) = \inf\left\{ \frac{\|X\theta\|_2 \sqrt{|S|}}{\|X\|\|\theta\|_1} : \|\theta_{S^c}\|_1 \leq 7\|\theta_S\|_1, \theta_S \neq 0 \right\}$$

$$\tilde{\phi}(s) = \inf\left\{ \frac{\|X\theta\|_2}{\|X\|\|\theta\|_2} : 0 \neq |S_\theta| \leq s \right\}$$

Then define the set Θ_{ρ_n, s_n} which is crucial in the theoretical analysis in the paper, such that

$$\Theta_{\rho_n, s_n} = \left\{ \theta \in \mathbb{R}^p : \phi(S_0) \geq c_0, \quad |S_0| \leq s_n, \quad \tilde{\psi}_{\rho_n}(S_0) \geq c_0 \right\}$$

for any $\rho_n \longrightarrow +\infty$, such that for $M > 0$, for any model S,

$$\tilde{\psi}_M(S) = \tilde{\phi}\left(\left(2 + \frac{4M}{A_4}\left(1 + \frac{16}{\phi(S)} \frac{\lambda}{\bar{\lambda}} \right) \right) |S| \right)$$

All theoretical guarantees in the paper are proved for $\theta \in \Theta_{\rho_n, s_n}$, which impose some restrictions on design matrix X. Among design matrices that satisfy these conditions, one can cite orthogonal matrices or *iid* random matrices. We will discuss about these conditions in section 4.

2.5 Variational algorithm

All this leads to the following algorithm for Laplace prior slabs.

The stopping criterion relies on the binary entropy H, which measures the change of values between two iterations.

Algorithm 1 VB for Laplace prior slabs

```
  Initialize:
      $(_H, \sigma, \gamma)$ arbitrary
      $\mu \leftarrow \hat{\mu}^{(0)}$
      $a \leftarrow order(\mu)$
  while $\Delta_H \geq \epsilon$ do
      for $i = 1, ...p$ do
          $i \leftarrow a_j$
          $\mu_i \leftarrow argmin_{\mu_i} f(\mu_i|\mu_{-i}, \sigma, \gamma, z_i = 1)$
          $\sigma_i \leftarrow argmin_{\sigma_i} g_i(\sigma_i|\mu, z_i = 1)$
          $\gamma_{old,i} \leftarrow \gamma_i$
          $\gamma_i \leftarrow logit^{-1}(\Gamma_i(\mu, \sigma, \gamma_{-i}))$
      $\Delta_H \leftarrow max_i\{|H(\gamma_i) - H(\gamma_{old,i})|\}$
```

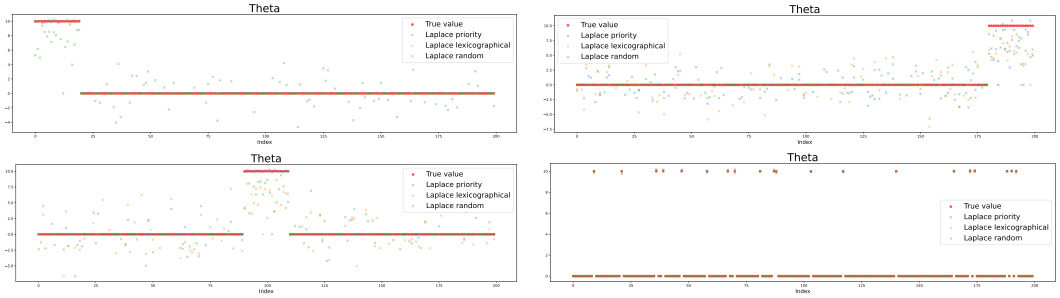

Figure 1: Results for different regression problems, when the zeros are placed in the beginning of θ (top-left), at the end (top-right), at the middle (bottom-left) and randomly (bottom-right). We study 4 different parameter update scheme for each cases.

3 Experiments

In this section we will investigate the benefits of the variational algorithm with Laplace prior. First experiment shows the importance of choosing a prioritise order for the update of the parameters to avoid to be stuck in a local minima. The second experiment shows the benefits to take Laplace prior over Gaussian prior. The last experiment shows results on real data.

3.1 Experiment 1: Prioritised updates

One of the contribution of the article is to propose a new approach to update the parameters in a prioritise order. They propose an numerical experiment to show the benefits of their method over other update. In this section we propose to implement the same experiment to verify their results. The experiment is done with the hyperparameters $a_0 = 1$, $b_0 = p$, $\lambda = 1$ and the stopping threshold for the entropy $\Delta_H = 10^{-5}$. To initialise the mean of the Gaussian μ the authors propose a ridge regression estimator $\hat{\mu}^{(0)} = (X^T X + I)^{-1} X^T Y$ instead of LASSO initialisation because LASSO will set some coefficients to zero. Taking $n = 100$, $p = 200$, $s = 20$, $\theta_i = 10$ for non zero coefficient and $X_{ij} \sim \mathcal{N}(0, 1)$, four scenarios are considered for the placement of the non-zero coefficient, at the beginning, at the end, on the middle and randomly. To prove the efficacy of the prioritising, the experiments are done for a lexicography order, an random order and a prioritise order.

Our experiment is done only one time because of the computation time. Our results show the efficacy of the prioritising order. For each position of the non-zero coefficient, prioritising order allows most of the time to make a good recovery of the parameters θ. Nevertheless, sometimes, despite the prioritising, the algorithm does not succeed see figure 1 (top, right). On contrary, it is possible that random or lexicographical order give good results see figure 1(bottom, right).

The local minima prevent the algorithm from always falling on the optimal solution. The prioritise order allow to limit this effect.

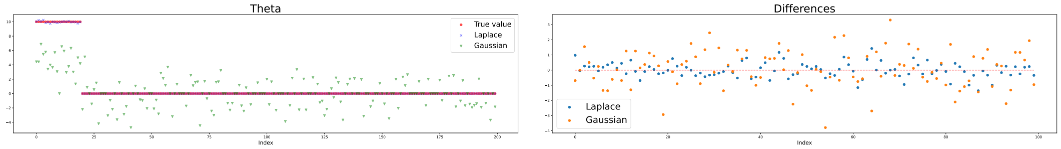

Figure 2: Parameters θ for Laplace prior and Gaussian prior (left) and difference of the true value of Y and the reconstruction of Y with Laplace prior result or Gaussian prior result (right).

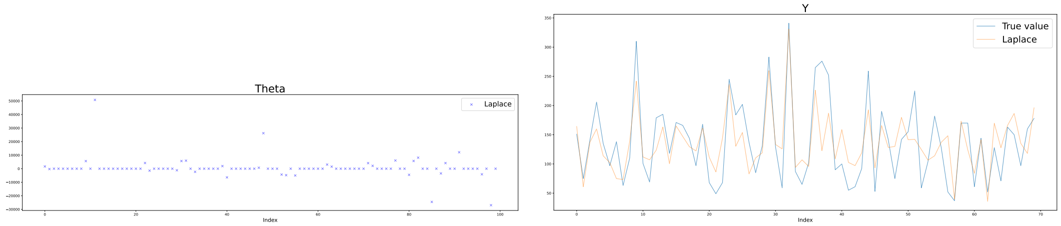

Figure 3: Predicted parameter θ (left) and predicted response vector $\hat{Y}$compared with the true response vector Y (right) for real data.

3.2 Experiment 2: Laplace prior vs. Gaussian prior

To compare the type of prior, the experiment is done with the same parameters as experiment 1 and considering only when the non-zero coefficient are at the beginning and the update order is prioritise. For the variational algorithm with Gaussian prior, the threshold to stop the training for the entropy is 10^{-5} like Laplace prior.

The results figure 2 (left) show that with Laplace prior, the predicted parameters θ are sparse. Only twenty coefficient equals 10 as expected. On contrary, with Gaussian prior almost zero coefficient are equals to zero. Nevertheless, the coefficient with most weight are the one which have to be equal to ten. That is why the reconstruction is not so bad for both choice of prior. In the figure 2 (right), the difference between the signal reconstructed and the true signal are plot. It is clear that choosing a Laplace prior increase the quality of the reconstruction, in fact the l2 error of the reconstruction is 0.055 for Laplace prior and 0.352 for Gaussian prior.

3.3 Experiment 3: Variational Bayes for real data

For the last experiment we choose to test the algorithm on a real dataset. We choose the diabetes dataset available on scikit-learn [2]. This dataset contains 442 samples and 10 parameters. To study the possible non linearity we consider all the interaction terms between the parameters, including with themselves. This leads to 100 parameters. To have $p > n$ we choose to study only 70 samples.

VB success to found sparse parameters with 70 zero coefficients (see Figure 3). The reconstruction of the signal Y has a l2 error of 102, which is not negligeable. It is possible that the hyperparameters of training are not perfectly tune cause of time computation.

4 Conclusion and Criticism

Authors propose a new approach for the VB algorithm using Laplace prior and results show the efficacy of this method. One main contribution of the authors is the prioritisation scheme for the variational updates ; although it appears to be an empirical result. Our results show that sometimes VB algorithm does not succeed to have good prediction. One solution could be to change order at each iteration instead of choosing a constant order for all the training.

The results on real data show poorer prediction than on synthetic data. It could be explain by the fact we do not tune perfectly our hyperparameters. An other possibility it is that the design matrix X does not respect the condition. For real data, this condition is hard to prove and there is no proof of convergence without it. Moreover, the conditions in Section 2.4 assumptions are tedious to interpret and are quite restrictive.

Some related problems have been investigated with the same method, such as classification [3]. This would be interesting to apply it on problems with real data since the setup where the number of parameters is bigger than the number of samples finds many applications (health, high-dimensional bayesian times series, recommendations,..).

References

[1] Kolyan Ray and Botond Szabó. "Variational Bayes for high-dimensional linear regression with sparse priors". In: *Journal of the American Statistical Association* (2021), pp. 1–12.

[2] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[3] Kolyan Ray, Botond Szabo, and Gabriel Clara. "Spike and slab variational Bayes for high dimensional logistic regression". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 14423–14434.