MATH5472 Computer-Age Statistical Inference

Report

varbvs: Fast Variable Selection for Large-scale Regression

Romain BARRAUD
20551110
rmbarraud@connect.ust.hk

December 10, 2021

Contents

1 Summary 1

2 Paper Structure 3

3 glmnet vs. varbvs 4

4 Focus on variational inference 5

5 Results 6

6 Discussion 6

7 Conclusion 7

8 References 7

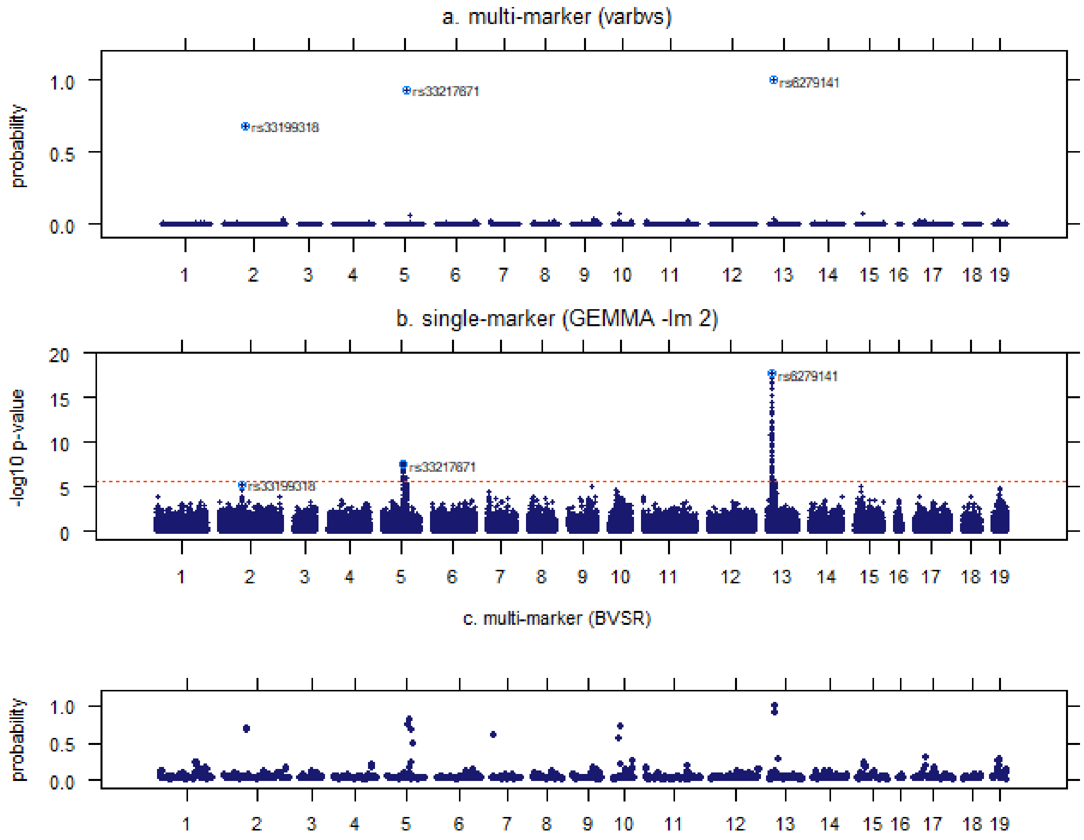
1 Summary

This report presents the research paper **varbvs: Fast Variable Selection for Large-scale Regression** published by Peter Carbonetto, Xiang Zhou and Matthew Stephens in 2017. The paper introduces a new method to perform variable selection through a Bayesian approach. Instead of computer-heavy and slow exact posterior computation, selection is achieved through optimisation via variational approximation. A new R package **varbvs** -

variational Bayesian variable selection - is created and made available on Github, in R and Matlab. Several examples are presented in the context of genomics, which is known for facing the challenge of variable selection among a very large number of features.

Keywords: variable selection, Bayesian, regression, Bayesian, variational inference, genetic, R.

Example from section 4 ”trait mapping in outbred mice”



Important note on code reuse and simulations

- The code and simulations presented in this report are originated from the official repository of the **varbvs** package hosted at `https://github.com/pcarbo/varbvs/tree/master/varbvs-R`.

- We have focused on the R implementation of the package. Matlab implementation is available in the official repository.

- For this report, we have particularly leveraged the code used in document **varbvs.Rmd**. The resulting output is hosted in our repository `https://github.com/RomainBarraud/HKUST-MATH5472-report-varbvs` created along with this report.

- The code to run section 5 ”Example: mapping Crohn’s disease risk loci” and section 6 ”Example: gene set enrichment analysis in Crohn’s disease” could not be reproduced due to privacy protection on the underlying data set. As per the paper, the simulation took 39 hours. Running successfully the code might have in any case been impossible in the light of the computing requirements. The source code, coming from the official repository, is also hosted to our own repository for the reader’s interest.

2 Paper Structure

In this section we outline the organisation of the paper and the keys ideas for each section.

1. Introduction.

2. The first part puts **glmnet**, a widely used package to do variable selection and using penalty regressions, and **varbs** side by side to explicit their high-level principle and differences to perform variable selection. **glmnet** relies on penalty-based regression while **varbs** takes a Bayesian approach. The 2 methods are then compared on a real example based on the analysis of genes in patients suffering from leukemia.

3. The next part details how **varbvs** performs variable selection. 4 subsections breaks down the procedure and quick guide of the package is given at the end. The simulation has been reproduced in ”MATH5472 report varbvs Romain Barraud.Rmd” in our repository.

   3.1. This subsection introduces the equations of linear regression and logistic regression used by **varbvs**.

   3.2. This section details how the priors of the selected regression model are chosen.

   3.3. The third section explains how **varbvs** manages to optimise a posterior distribution for a given set of hyperparameters, using KL-divergence as the target metrics and coordinate descent to optimize this metrics.

   3.4. The fourth section focuses on the computation of the hyperparameters of the candidates distributions chosen to approximate the exact one.

   3.5. The fifth section introduces the interface of **varbvs** and the main parameters in use.

4. The fourth part takes the example of a the genome of mice to apply **varbvs** on a trait (i.e. a specific combination of genes) defining the weight of a testis. The simulation has also been reproduced in ”MATH5472 report varbvs Romain Barraud.Rmd” in our repository.

5. The firth part is also an application. **varbvs** is used to detect the position of the genes leading to Crohn’s diseases (provoking inflammation of intestines).

6. The sixth part is a variant of the previous section aiming at comparing models. Incorporating into the model the presence of a specific protein (the cytokine), **varbvs** features allow to detect strong correlation between Crohn's disease and that protein.

7. Summary and associated discussion.

3 glmnet vs. varbvs

The paper puts side by side the use of constraint-based regression models via **glmnet**, and the newly created **varbvs** based on Bayesian variable selection. This comparison is accompanied by examples, including computation time which brings tangible evidence of the speed of the methodology used. The comparison is made in the case of a logistic regression.

glmnet problem formulation takes the form of Elastic Net:

$$\min_{\beta_0 \in R, \beta \in R^p} -\frac{1}{n}\sum_{i=1}^{n} Pr(y_i|x_i, \beta_0, \beta) + \frac{\lambda}{2}(1-\alpha)\|\beta\|_2^2 + \lambda\alpha\|\beta\|_1 \qquad (1)$$

$\lambda = 0$ (resp. $\lambda = 1$) then allows to apply a Ridge (resp. Lasso) model.

Comparatively, **varbvs** relies on the following equation:

$$log\left\{\frac{Pr(Y=1)}{Pr(Y=0)}\right\} = \sum_{i=1}^{m} Z_i u_i + \sum_{i=1}^{p} X_i \beta_i \qquad (2)$$

Let us also mention the case of a linear regression (which is the second option of the package):

$$Y = \sum_{i=1}^{m} Z_i u_i + \sum_{i=1}^{p} X_i \beta_i + \epsilon \qquad with\ \epsilon \sim \mathcal{N}(0, \sigma^2) \qquad (3)$$

We can note here the inclusion of the covariate terms which are not present in the simplest definition of linear and logistic regressions. Covariates allow to capture more variability and therefore to decrease random effects (i.e. unpredictable variability). They become particularly relevant in the context of variable selection where we are precisely seeking the most relevant features, hence their inclusion in the equation used by **varbvs**.

We will summarise here the comparison to highlight the pros and cons of each approach.

Advantages:

- **Uncertainty estimation**. Bayesian methods inherently bears the capability to estimate uncertainty around the parameters. **varbvs** intrinsically bears this advantage.
- **Comparison**. Using Bayesian factors (e.g. Bayesian Information Criterion), one can compare the performance of different models.

Limitations and how varbvs address them

- **Interpretation**. Using approximated distributions must be taken with caution when trying to interpret results.
- **hyperparameters priors setup**. **varbvs** uses by default model averaging, which requires the user to specify the priors for the hyperparameters. This part may be challenging.
- **Initialization**. Variational approximation suffers from the initialization choice.

4 Focus on variational inference

In this section we describe the stages adopted by **varbvs** to perform variable selection. Let us recall that the main advances of the package is to transform posterior computation into optimisation via variational inference in order to tackle the difficulty, if not impossibility, to compute exact posteriors in a reasonable time.

At a high level, the process is a pipeline of 3 stages: **Model selection** → **Prior selection** → **Variational inference**. The variational inference can also be broken into 2 loops, the second one being nested into the first one. The first obtains the candidates hyperparameters, and the second fits the candidate distributions for the given hyperparameters.

Let's detail each stage.

1. **Model selection**. This step will choose the right model, logistic or linear regression, based on the target problem to solve. The form of the equation is expressed by **(2)** (resp. **(3)**) in section 3 **glmnet vs. varbvs**.
2. **Prior selection**. The selection of the prior relies on "spike-and-slab" priors:

   $$Pr(\beta_i|\pi, \sigma_a^2) = (1-\pi)\delta_0 + \pi\mathcal{N}(0, \sigma^2) \quad for \quad i = 1, ..., p \quad and \quad 0 \leq \pi \leq 1$$

   where π is the probability that β_i is part of the selected variables and σ_a the variance (of the prior) of the all the coefficients being not null.

   We can understand in this formulation that we will obtain a sparse model when π is high (resp. a dense model when π is low).

   Grid search would be used to select the hyperparameters $(\sigma^2, \sigma_a^2, \pi)$. They can also be estimated by Expectation Maximization if they are unknown.

3. **Variational inference**

   3.1. **Outer loop**. There are 3 cases. When $(\sigma^2, \sigma_a^2, \pi)$ are known, we can fit the posteriors to β_i considering that $(\sigma^2, \sigma_a^2, \pi)$ represent a lower bound that we want to maximize. Another possibility is to estimate $(\sigma^2, \sigma_a^2, \pi)$. **varbvs** gives the possibility to allocate priors to the hyperparameters, and then use model averaging (Hoeting et al. 1999). This method makes the following approximation:
   for $j = 1, ..., n$ priors, noting $= (\sigma^2, \sigma_a^2, \pi$ and $LB()$ the lower bound of the distribution formed by :

$$w^{(j)} = LB(\theta) / \sum_{k=1}^{n} LB(\theta^{(k)})$$

   and

$$Pr(\beta | X, Z, y) \approx \sum_{j=1}^{n} w^{(j)} Pr(\beta | X, Z, \theta^{(j)})$$

   The third case is an hybrid version of the 2 previous ones. Thus, the theory remains the same.

   3.2. **Inner loop. Distribution optimisation**. For the given hyperparameters $(\sigma^2, \sigma_a^2, \pi)$ from the outer loop, we will minimise a chosen metrics, in this case the KL-divergence, between the distribution selected for the approximation and the exact distribution. This part will be realised through co-ordinate descent and will continue until a maximum number of iterations is reached or the difference between the distributions is below a chosen threshold.

5 Results

The results are available at the github repository https://github.com/RomainBarraud/HKUST-MATH5472-report-varbvs in ”MATH5472 report varbvs Romain Barraud.Rmd”. The sources and resulting knitted files are all accessible to the reader. The README.md also included in the repository provides help and additional details to reproduce the simulations.

6 Discussion

Beyond the actual use of **varbvs** to perform variable selection, the key advances revolve around the capacity to compute posteriors in a reasonable time. The approach, consisting in moving away from an exact computation to an optimisation, is the fundamental enabler. This principle paves the way to apply Bayesian methods more widely, adding the advantage of evaluating uncertainty around a given estimate. Applications are everywhere, ranging from high-precision mechanics, passing by aeronautics and going to finance. In this respect, **varbvs** and Bayesian Variable Selection more generally methods have a tangible edge.

It is worth noting that the **varbvs** package was also purposely designed to offer an interface very close to **glmnet**. This choice is assuredly relevant to ease and accelerate its adoption. Furthermore, a significant effort was put to supporting features, default values (such as for the choice of priors), and metrics to allow model comparison like with the Bayesian Information Criterion. These extra features can make the difference between a successful and unused package.

Finally, let us reflect on the use of R to solve variable selection in genetics. This domain inherently requires extremely intensive computations. R is known for not going at scale. In this respect, other technical approaches, and languages, look more relevant to tackle these problems. Being able to leverage GPU should be seen as a natural way to run larger and loads at a faster speed. Bayesian computations that are GPU-enabled seem best suited. One may consider deep-learning frameworks in this setup. TensorFlow has **TensorFlow Probability** and Pytorch has **Pyro** to perform (deep) probabilistic programming. Porting **varbvs** onto one of these 2 frameworks, and using Python, could expand its performance either by reducing computation time or increasing the size of data sets while keeping a nearly similar interface.

7 Conclusion

We have analyzed and reproduced an example of **varbvs: Fast Variable Selection for Large-scale Regression**, the research paper introducing a new package to perform Bayesian Variable Selection. In addition to the natural advantage of Bayesian methods to provide uncertainty around computed values, **varbvs** also counters the usual computational wall inherent to Bayesian methods by transforming posterior computation into an optimisation problem. In addition to the theoretical contribution, **varbvs** also brings a familiar interface purposely close to the popular **glmnet** package. As a result, the methodology and packages make the use of Bayesian Variable Selection more accessible and a complementary approach to penalty-based regressions.

8 References

Peter Carbonetto, Xiang Zhou, Matthew Stephens (2017), **varbvs: Fast Variable Selection for Large-scale Regression**

Peter Carbonetto, Matthew Stephens† (2021), **Scalable Variational Inference for Bayesian Variable Selection in Regression, and Its Accuracy in Genetic Association Studies**

Pr. Yang Can (2021), **HKUST MATH 5472. Computer-Age Statistical Inference**

Hoeting et al. 1999, **Bayesian Model Averaging: A Tutorial**

Clyde MA, Ghosh J, Littman ML (2011). **Bayesian Adaptive Sampling for Variable Selection and Model Averaging." Journal of Computational and Graphical Statistics**

varbvs: large-scale Bayesian variable selection in R. https://github.com/cran/varbvs

Lasso and Elastic-Net Regularized Generalized Linear Models. https://github.com/cran/glmnet

TensorFlow Probability https://www.tensorflow.org/probability/

Pyro https://pyro.ai/