Summary and discussion of: "varbvs: fast variable selection for large-scale regression"

5472 Final project

ZENG Yeqin

1 Summary

1.1 Background and context

Bayesian variable selection (BVS) models, and extensions to these models, have been shown to provide attractive solutions to a number of important problems in genome-wide association studies. However, there are several limiting factors that constrain its wider application in genome-wide association studies (GWAS) and other areas. One factor is that computing posterior probabilities in BVS model is in fact a high-dimensional integration problem, which is intractable in large datasets. Existing methods using MCMC to approximate the posteriors are time-consuming and scale poorly to large datasets. Another factor is that the choice of priors has a significant effect to the results of Bayesian data analysis which needs extra work to tune it.

The paper we discuss here presented a new software toolkit **varbvs** for fitting the BVS model to large scale data sets. The software uses variational EM algorithm to approximate the posterior instead of MCMC which allows for a linear computational complexity with respect to the amounts of samples and features. It also provides default priors that are suitable for different areas and can automatically return a averaged model according to the Bayesian model averaging.

1.2 Bayesian Variable Selection Regression

Linear mixed model is a powerful tool widely used in GWAS studies as it can well control the confounding factors including population stratification, familial correlation, and relatedness among individuals. However, the basic assumption of LMM is that every genetic variant affects the phenotype with effect sizes normally distributed. In the true high dimensional cases in which only several of the million variants have effects on the phenotype we want to study, LMM may perform poorly especially when our main goal is to identify the true relevant variants. Motivated by this observation, we consider a sparse version of the LMM and expect it can well represent the sparse data.

We first begin with the basic linear model. Suppose we a have data set with n samples, p candidate predictors and q covariates, and the target phenotype is a linear combination of the candidate predictors, covariates and residual noise:

$$Y = Z\omega + X\beta + \epsilon$$

where $X \in \mathbb{R}^{n\times p}$ is the predictor matrix with effects $\beta \in \mathbb{R}^p$, $Z \in \mathbb{R}^{n\times q}$ is the covariates matrix with effects $\omega \in \mathbb{R}^q$, $y \in \mathbb{R}^n$ is the phenotype vector, and $\epsilon \sim N(0, \sigma_e^2 I_n)$ is the noise vector.

While LMM assumes that $\beta_i \sim N(0, \sigma_a^2)$ independently, we introduce a indicator vector $\gamma \in \{1, 0\}^p$ to denote whether variant i has a non-zero effect, and the prior of β_i turns to:

$$\begin{cases} \beta_i \sim N(0, \sigma_a^2) & if \gamma_i = 1 \\ \beta_i \sim \delta_0 & if \gamma_i = 0 \end{cases}$$

where $P(\gamma_i = 1) = \pi$ and $P(\gamma_i = 0) = 1 - \pi$. To be more simplified, this prior can also be wrote as:

$$\beta_i \sim (1 - \pi)\delta_0 + \pi N(0, \sigma_a^2),$$

which is called the "spike-and-slab" prior. After combining this prior with the model, we now have latent variables (β, γ) and unknown parameters$(\sigma_e^2, \sigma_a^2, \pi)$. The covariates Z are additional predictors that we want to exclude their effects from our study. To adjust them, we only need to multiply the projection matrix on the both sides of the equation and we can get a simpler model:

$$P_Z Y = P_Z X\beta + P_Z \epsilon$$

where $P_Z = I - Z(Z_T Z)^{-1} Z^T$.

1.3 Variational EM

The basic idea of variational approximation (VA) is to find a distribution to approximate the original posterior distribution which is intractable. The candidate distribution is usually a class of parameterized distribution that are easy to compute and the best one is chose by minimizing the Kullback-Leibler divergence which is well known as KL divergence.

In this case, we enforce a conditional independence approximation: conditioned on the hyperparameters $\theta = \{\sigma_\epsilon^2, \sigma_a^2, \pi\}$, each effect β_i is independent from the other effects. That is:

$$q(\beta, \gamma | \sigma_\epsilon^2, \sigma_a^2, \pi) = \prod_{i=1}^{p} q_i(\beta_k, \gamma_k | \sigma_\epsilon^2, \sigma_a^2, \pi)$$

And the coordinate descent approximation is as follows:

$$\beta_i | \sigma_\epsilon^2, \sigma_a^2, \pi \sim \begin{cases} N(\mu_i, s_i^2) & if \gamma_i = 1 \\ \delta_0 & if \gamma_i = 0 \end{cases}$$

$$p(\gamma_i = 1 | \sigma_\epsilon^2, \sigma_a^2, \pi) = \alpha_i = 1 - p(\gamma_i = 0 | \sigma_\epsilon^2, \sigma_a^2, \pi)$$

where

$$s_i^2 = \frac{\sigma_\epsilon^2}{(X^T X)_{ii} + 1/\sigma_a^2}$$

$$\mu_i = \frac{s_i^2}{\sigma_\epsilon^2}((X^T Y)_i - \sum_{j \neq i} (X^T X)_{ji} \alpha_j \mu_j)$$

$$\frac{\alpha_i}{1 - \alpha_i} = \frac{\pi}{1 - \pi} \times \frac{s_i}{\sigma_a \sigma_\epsilon} \times e^{\frac{\mu_i^2}{2 s_i^2}}$$

And in the M-step, we can derive the update for σ_a^2 and σ_ϵ^2 in the standard way by computing the gradient with respect to them as follows:

$$\sigma_\epsilon^2 = \frac{\|Y - Xr\|^2 + \sum_{i=1}^p (X^T X)_{ii} Var[\beta_i] + \sum_{i=1}^p \alpha_i (s_i^2 + \mu_i^2)/\sigma_a^2}{n + \sum_{i=1}^p \alpha_i}$$

$$\sigma_a^2 = \frac{\sum_{i=1}^p \alpha_i (s_i^2 + \mu_i^2)}{\sigma_\epsilon^2 \sum_{i=1}^p \alpha_i}$$

where $var[\beta_i] = \alpha_i(s_i^2 + \mu_i^2) - (\alpha_i \mu_i)^2$ and $r = \alpha_i \mu_i$.

2 Result and Discussion

In this section we will illustrate the feature of **varbvs** through several simulation and test on real data set based on our own implementation. Here we mainly focus on the linear model and the implementation of logistic model is similar, its detailed derivation can be seen in the original paper.The python source code is available for download at Github (https://github.com/yqzenggugu/5472_final_varbvs.git).

2.1 Comparison of BVS and penalized regression

This part corresponds to the section 2 in the original paper. It aims to provide some intuition for the different properties of BVS and penalized regression as implemented by **varbvs** and **LASSO**. The idea comes from that intuitively the prior π in the BVS model is similar to the penalizing parameter λ in the penalized model like LASSO, as we expect the model to be more sparse with larger λ and smaller π.

Thus we conduct simulations based on the genotype data used in the paper. The data consists of expression levels recorded for 3,571 genes in 72 patients with leukemia. The genes are the candidate variables and there is no covariates. Then we generate quantitative phenotype data according to two parameters: the number of casual effects s and the"chip heritability" h.

1. randomly sample $s = 5$ variables as casual effects.
2. generate their coefficients from standard normal distribution while others are 0.
3. generate random error e from $N(0, \sigma^2)$, where $\frac{var(X\beta)}{var(X\beta)+\sigma^2} = h$.
4. compute simulated phenotype as $y = X\beta + e$.

First we use **sklearn.linear_model.LassoCV** to fit it with 20-fold cross validation and then we use **sklearn.linear_model.Lasso** to fit it with every single *lambda* to get the solution path. This is because varbvs doesn't need cross validation to choose a appropriate level of regulation like Lasso, which we will discuss later. The result is shown in the figure 1.

Second, we use **varbvs** to fit it with default setting. As we have said above, **varbvs** has a set of default setting for the prior just like **LassoCV**, however it selects model according to the Bayesian inference approach which automatically weighs the accuracy of the model

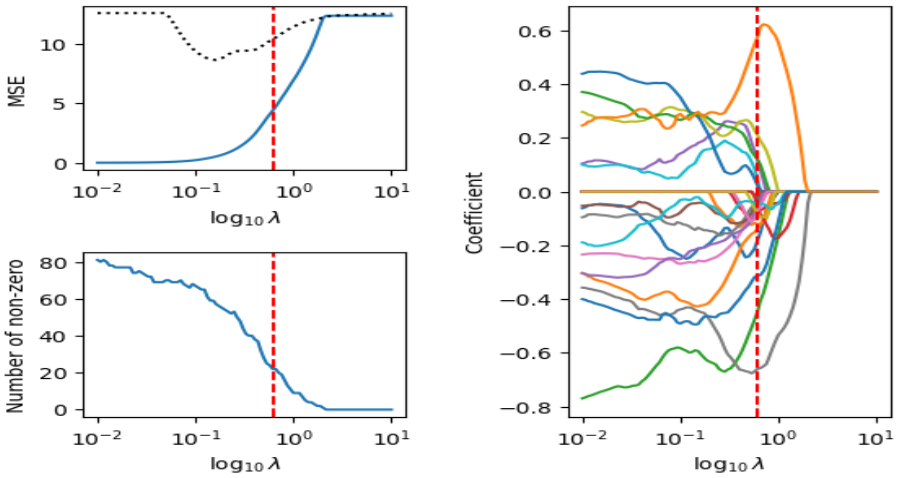

Figure 1: Lasso analysis of simulated data. *top-left penal*: l_1-penalty strength parameter(λ) against mean squared err for the model fitted to the entire dataset, which is conducted for comparison to the varbvs result. The black line corresponds to the CV MSE. *bottom-left panel*: Number of variables with non-zero effects at each λ setting. *right-hand panel*: Regression coefficients path of coefficients selected by cross-validation. The red vertical line corresponds to the selected λ.

predictions against the model complexity. Then we fit it with every single π to get the solution path. The result is shown in the figure 2.

Consider the top-left and bottom-left penal of two figures, we can see that the prior inclusion probability π has a regulation effect on the model similar to λ: As π decreases, the variable is less likely to be included in the model, and the model becomes more sparse with fewer expected number of non-zero effects and larger MSE. This is similar to the pattern in Lasso result.

Though in the performance of prediction error there are no indistinguishable difference, their shrinkage behavior are quite different. It seems that the number of non-zero effects both converges to zero when the model becomes more sparser, but in fact the expected number of non-zero effects in varbvs converges to 1.49 in the bottom-left penal of figure 2. And this expectation mainly focus on one of the true casual effects. We can see that from the figure 3.

This shows that, varbvs can achieve strong shrinkage of effects near zero without correspondingly strong shrinkage of the important predictors which is a highly desirable feature that convex penalization methods such as the Lasso and Elastic Net struggle to achieve. The figure also show another advantage of BVS model: it provides a simple and efficient approach to account for the uncertainty for selecting variables.

2.2 Example: mapping a complex trait in outbred mice

This part corresponds to the section 4 in the original paper. It aims to illustrate the features of varbvs for genome-wide mapping of a complex trait. Here we repeat this experiment on

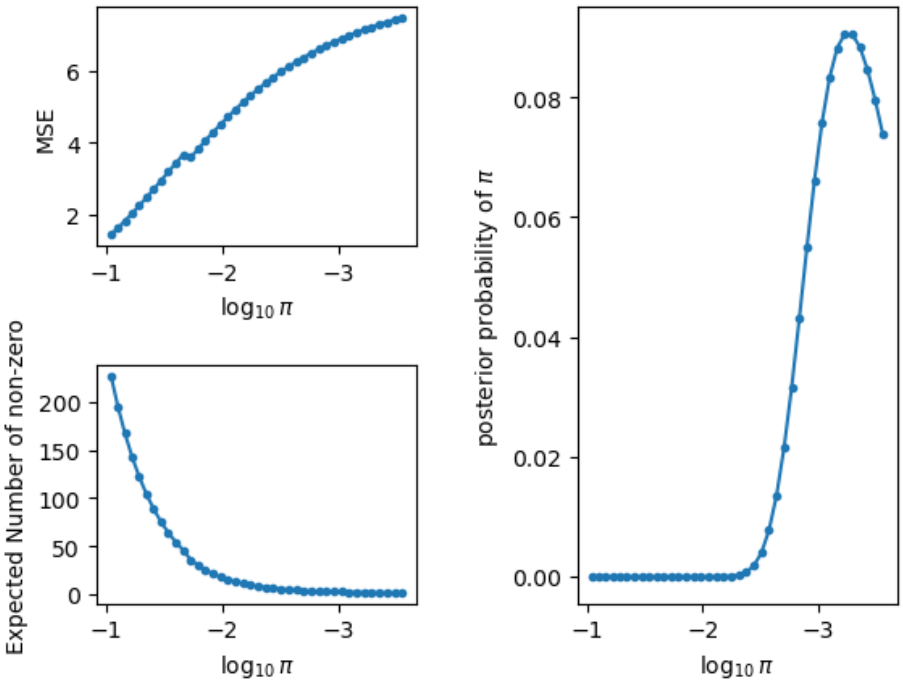

Figure 2: Varbvs analysis of simulated data. *top-left penal*: Prior inclusion probability (π) against mean squared err for the model fitted to the entire dataset. *bottom-left panel*: Expected number of variables with non-zero effects at each π setting. *right-hand panel*: Estimated posterior distribution of π.

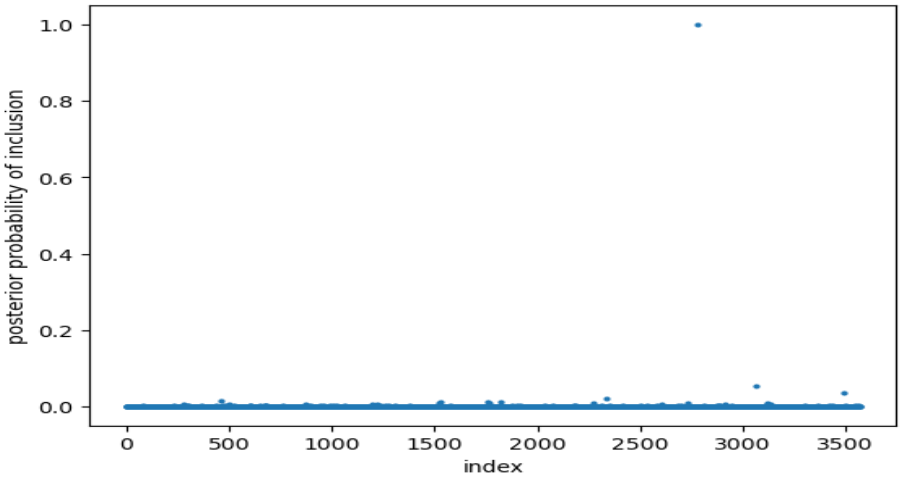

Figure 3: Posterior inclusion probability from varbvs analysis

the same data. The data are body and testis weight measurements recorded for 993 outbred mice, and genotypes at 79,748 single nucleotide polymorphisms (SNPs) for the same mice. The main goal is to identify genetic variants contributing to variation in testis weight.

In the varbvs analysis, the quantitative trait (testis weight) is modeled as a linear combination of the covariate (body weight) and the candidate variables (the 79,748 SNPs). And we set $sa = 0.05$ and $logodds = np.linspace(-5, -3, 9)$ with others as default which is the same setting to the paper. The mapping result is as follows.

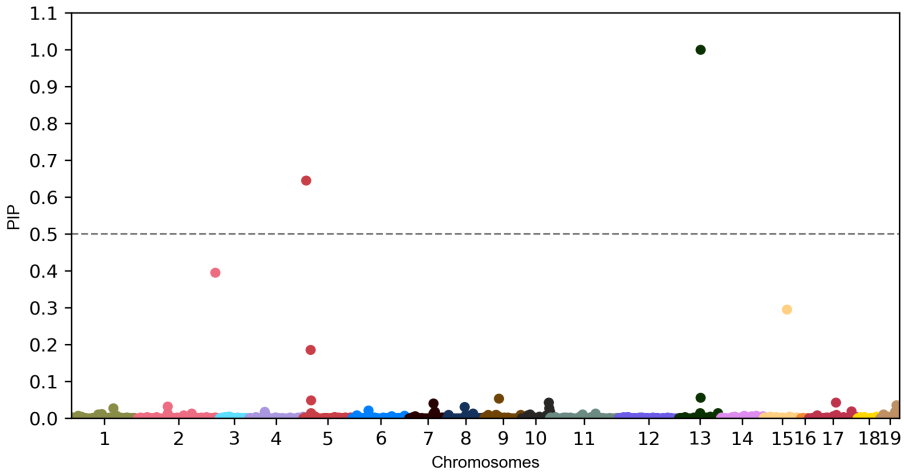

Figure 4: QTL mapping of a complex trait in outbred mice. Posterior inclusion probabilities for all 79,748 candidate SNPs on chromosomes 1–19 computed using varbvs.

The analysis takes about 4 minutes in my computer. And the result is similar to the result I get from running the "cfw" vignette provided by original paper. We can see that only 5 out of the 79,748 SNPS are included with PIP more than 0.1 and only 2 are included with PIP more than 0.5.

The analysis results data of p-value and BVSR method are directly from the original paper. And the BVSR implemented by GEMMA using MCMC approach takes about 9 hours which is much more than the implementation by varbvs.

Their QTL mapping results generally share a similar pattern. Compared with result from BVSR implemented by GEMMA, the model yielded by varbvs is much more simple with less included effective snps. Compared with result of p-value, we can see that, varbvs tend to concentrate the posterior mass on a single variable due to its fully-factorized variational approximation while in GWAS study there are many highly correlated neighbour SNPS. And the result may vary with different variable order input into the model.

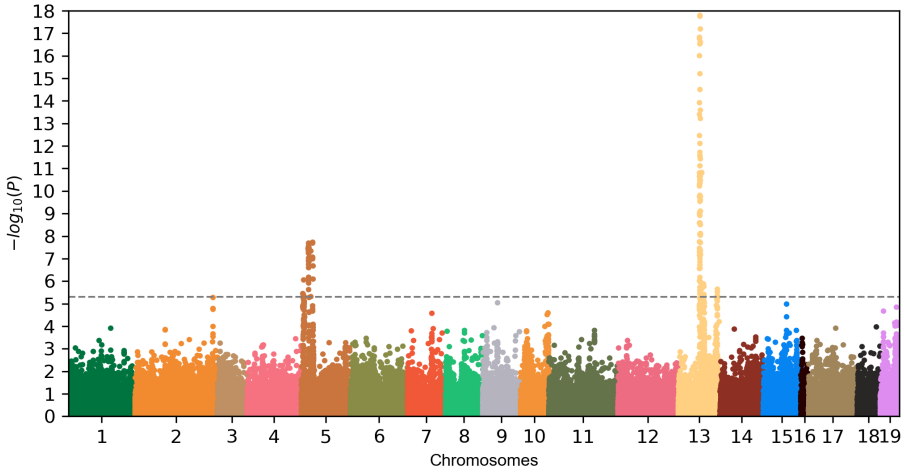

Figure 5: QTL mapping of a complex trait in outbred mice. p values for the same candidate SNPs computed using GEMMA. threshold determined via permutation analysis, at p_value = 2 ×10-6.

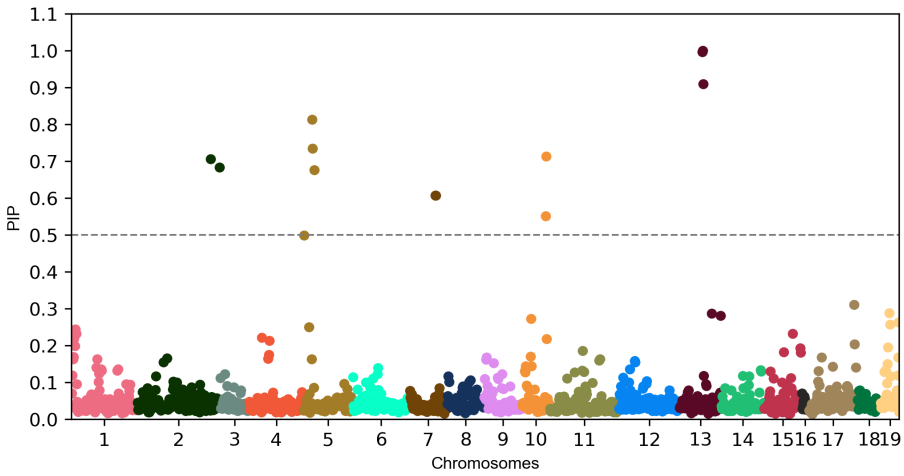

Figure 6: QTL mapping of a complex trait in outbred mice. Posterior probabilities computed using the BVSR method in GEMMA version 0.96.

3 Conclusion

In my perspective, there are two main advantage of **varbvs** over methods before. First, fast-computation of posterior distribution in BVS model is made possible in the large scale data set by the formulation of a variational approximation derived from a simple conditional independence assumption. The computational complexity of the co-ordinate ascent algorithm for fitting the variational approximation is linear in the number of samples and in the number of variables.

Second, it provides a default approach to do the prior selecting and parameter initialization according to Bayesian inference approach. This could decrease the influence of EM algorithm converges to a local maximum and save some extra work for tuning parameters due to the sensitivity of results

Besides in the paper it also illustrates some advantages of BVS model over penalized model as it introduces a prior into the model to accounts for the uncertainty of selecting a variable. And the interface of varbvs is designed to be similar to **glmnet** in R, this is aimed to make researchers already familiar with these methods can easily explore the benefits of the BVS approach.

However it doesn't accounts for the highly correlated data structure in GWAS study. This could make the algorithm take much more iterations to converge. The concentrated posterior inclusion probability may cause higher false discovery rate.

References

[1] Peter Carbonetto, Xiang Zhou, and Matthew Stephens. *varbvs: Fast Variable Selection for Large-scale Regression*. 2017. arXiv: 1709.06597 [stat.CO].

[2] Yongtao Guan and Matthew Stephens. "Bayesian variable selection regression for genome-wide association studies and other large-scale problems". In: *The Annals of Applied Statistics* 5.3 (2011), pp. 1780–1815. DOI: 10.1214/11-AOAS455. URL: https://doi.org/10.1214/11-AOAS455.

[3] Xiang Zhou, Peter Carbonetto, and Matthew Stephens. "Polygenic Modeling with Bayesian Sparse Linear Mixed Models". In: *PLOS Genetics* 9.2 (Feb. 2013), pp. 1–14. DOI: 10.1371/journal.pgen.1003264. URL: https://doi.org/10.1371/journal.pgen.1003264.