COMP 565 Assignment 3: Variational inference of Bayesian polygenic risk score regression with spike-and-slab prior

This assignment is worth 8% of your total grade and due at **23:59 on Feb 22, 2023**. The breakdown of the total points are indicated in the section headers below whenever applicable.

Variational inference of Bayesian linear regression

In Bayesian PRS (part 2) (Lecture 9), we covered in detail the variational inference (VI) algorithm to infer Bayesian linear regression coefficients that follows a spike-and-slab prior and re-parameterized into Bernoulli-Gaussian prior. In particular, the model is specified as below:

Prior:

$$p(\beta, s) = \prod_j \mathcal{N}(\beta_j|0, \tau_\beta^{-1})\pi^{s_j}(1-\pi)^{1-s_j}$$

Likelihood:

$$p(y|X, \beta) = \mathcal{N}(X(s \circ \beta), \tau_\epsilon^{-1} I)$$

Variational distribution:

$$q(\beta, s|y, X) = \prod_j q(\beta_j|s_j)q(s_j)$$

In this assignment, we will implement the expectation-maximization algorithm that is outlined in the Algorithm 1.

Data

First, download the `data.zip` from here:

`https://drive.google.com/file/d/1CtIaYK1z39UQFqN82fUvGosro3M-Vw9F/view?usp=sharing`

Algorithm 1 The EM algorithm of Bayesian PRS with spike-and-slab prior

E-step: For each SNP j given the estimates for all other SNPs $i \neq j$, do:

$$q^*(\beta_j|s_j = 1) = \mathcal{N}(\mu^*_{\beta_j}, 1/\tau^*_{\beta_j})$$

$$\tau^*_{\beta_j} = X_j' X_j \tau_\epsilon + \tau_\beta, \quad \mu^*_{\beta_j} = N\frac{\tau_\epsilon}{\tau^*_{\beta_j}}\left(y' X_j/N - \sum_{i\neq j}\gamma^*_i \mu^*_{\beta_i} r_{ij}\right)$$

$$q^*(s_j = 1) = \frac{1}{1+\exp(-u_j)} \equiv \gamma^*_j, \quad u_j = \ln\frac{\pi}{1-\pi} + \frac{1}{2}\ln\frac{\tau_\beta}{\tau^*_{\beta_j}} + \frac{\tau^*_{\beta_j}}{2}{\mu^*_{\beta_j}}^2$$

M-step: update hyperparameters:

$$\tau_\beta^{-1} = \sum_j \gamma^*_j(\mu^{*2}_j + {\tau^*_{\beta_j}}^{-1})/\sum_j \gamma^*_j$$

$$\pi = \sum_j \gamma^*_j/M$$

There are $M = 100$ SNPs, $N_{train} = 439$ training patients, $N_{test} = 50$ test patients. Both the genotype X_{train}, X_{test} and phenotype y_{train}, y_{test} were standardized. The marginal summary statistics $\beta_{marginal}$ saved in `beta_marginal.csv.gz` and the LD matrix R saved in `LD.csv.gz` were completely derived from the 439 training patients only:

$$\beta_{marginal} = X'_{train} y_{train}/N_{train}$$

$$R = X'_{train} X_{train}/N_{train}$$

To implement the VI algorithm, you **must only use** `LD.csv.gz` and `beta_marginal.csv.gz`. The individual data were provided to you for evaluation purpose only.

1 Expectation step (2%)

Implement the Expectation step as shown in Algorithm 1. You will need to cycle through one SNP at a time to update its posterior precision and posterior mean of the effect size and its PIP while fixing all of the rest of the SNPs.

Set the initial values for the posterior estimates for all SNPs to the following values:

$$\forall j \in \{1, \ldots, M\} \quad \mu^*_{\beta_j} = 0, \quad \tau^*_{\beta_j} = 1, \quad \gamma^*_j = 0.01$$

Set the initial values for the hyperparameters to the following values:

$$\tau_\epsilon = 1, \quad \tau_\beta = 1/(0.5/M) = 200, \quad \pi = 0.01$$

For numerical stability, after a full cycle of E-step, cap the resulting PIP γ_j of each SNP within [0.01, 0.99]. That is, if $\gamma_j < 0.01$ set it to 0.01, and if $\gamma_j > 0.99$ set it to 0.99.

2 Maximization step (2%)

Implement the Maximization step as shown in Algorithm 1. To make things simpler, we don't need to update τ_ϵ in the M-step and keep its value always as 1, and only implement the update for τ_β and π.

3 Evidence lower bound (2%)

The evidence lower bound (ELBO) of the model is:

$$\mathcal{L}_{ELBO} = \mathrm{E}_{q(\beta,s)}[\ln p(y|\beta,s)] + \mathrm{E}_{q(\beta)}[\ln p(\beta|s)] + \mathrm{E}_{q(s)}[\ln p(s|\pi)] - \mathrm{E}_{q(\beta|s)}[\ln q(\beta|s)] - \mathrm{E}_{q(s)}[\ln q(s)]$$

To evaluate the total ELBO, first implement the above five individual ELBO terms, and then bring them together in the final ELBO:

$$\begin{aligned}
\mathrm{E}_q[\ln p(y|\beta,s)] &= \frac{N}{2}\ln\tau_\epsilon - \frac{\tau_\epsilon}{2}y'y + \tau_\epsilon(\gamma^* \circ \mu^*)'X'y \\
&\quad - \sum_j \frac{\tau_\epsilon}{2}(\gamma_j^*({\mu_{\beta_j}^*}^2 + 1/\tau_{\beta_j}^*)x_j'x_j - \tau_\epsilon \sum_{j=1}^{M}\sum_{k=j+1}^{M} \gamma_j^*\mu_{\beta_j}^*\gamma_k^*\mu_{\beta_k}^* x_k'x_j \\
\mathrm{E}_{q(\beta,s)}[\ln p(\beta|s)] &= \sum_j -\frac{1}{2}\ln 2\pi\tau_\beta^{-1} - \sum_j \frac{\tau_\beta}{2}\gamma_j^*({\mu_j^*}^2 + {\tau_{\beta_j}^*}^{-1}) \\
\mathrm{E}_{q(\beta,s)}[\ln q(\beta|s)] &= \sum_j -\frac{1}{2}\ln 2\pi\tau_\beta^{-1} - \frac{1}{2}\sum_j \gamma_j^*\ln\tau_\beta \\
\mathrm{E}_{q(s)}[\ln p(s|\pi)] &= \sum_j \gamma_j^*\ln\pi + (1-\gamma_j^*)\ln(1-\pi) \\
\mathrm{E}_{q(s)}[\ln q(s)] &= \sum_j \gamma_j^*\ln\gamma_j^* + (1-\gamma_j^*)\ln(1-\gamma_j^*)
\end{aligned}$$

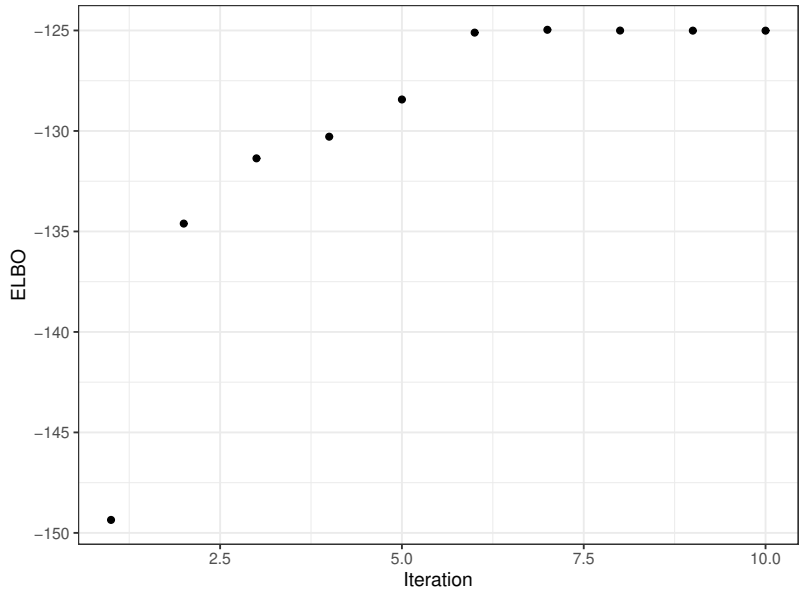

Figure 1: Evidence lower bound as a function of EM iteration.

where γ_j^*, μ_j^*, and τ_j are the inferred PIP, mean and precision of the effect size for SNP j at the E-step, respectively; $\circ$ is the elementwise product of two vectors.

After that, run your implemented EM-update code for 10 iterations and show the ELBO as a function of iterations. If you see the same nice convergence curve as in Figure 1, your implementation should be generally correct. Otherwise, you have a bug in your implementation of the E-step, the M-step, and/or ELBO evaluation.

4 Evaluating PRS prediction (1%)

Once you confirm that the ELBO of our code behaves appropriately, you may take on the PRS prediction task. First, import X_{train}, y_{train}, X_{test}, y_{test} from the respective CSV files:

`X_train.csv.gz`, `y_train.csv.gz`, `X_test.csv.gz`, `y_test.csv.gz`

Then, predict PRS for both the 439 training patients and the 50 testing patients using their genotypes:

$$\hat{y}_{train} = X_{train}(\gamma^* \circ \mu^*)$$
$$\hat{y}_{test} = X_{test}(\gamma^* \circ \mu^*)$$

where γ^* and μ^* are the inferred PIP and expected effect size at the E-step, respectively. $\circ$ is the elementwise product of two vectors.

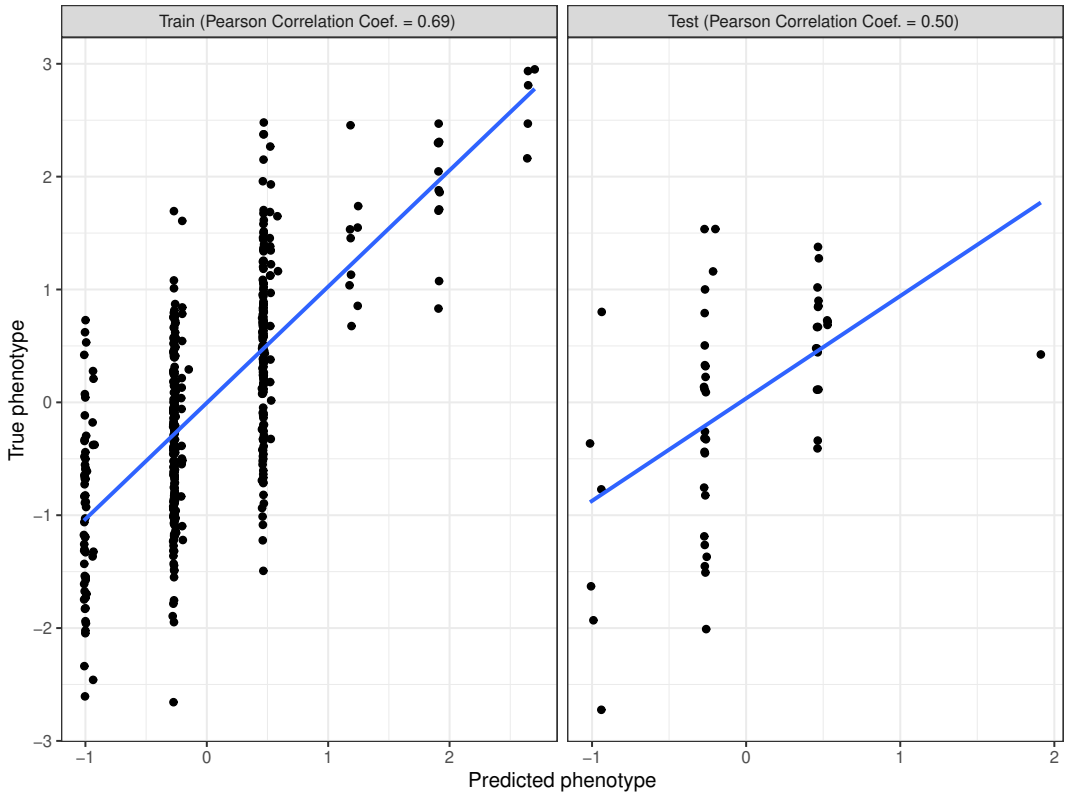

Figure 2: PRS prediction on training and testing set.

Calculate the Pearson correlation coefficient (PCC) between your predicted phenotypes and the true phenotypes. Generate scatter plots as displayed in Figure 2. The training PCC is 69% and the testing is 50% - not bad considering the true heritability is 50%!

5 Evaluating fine-mapping (1%)

Plot your inferred PIP γ as in Figure 3. The causal SNPs are rs9482449, rs7771989, and rs2169092. Color them in red on your plot. Although we did quite well in this fine-mapping application, caveat should be taken in fine-mapping loci with tighter and more complex LD as we discussed in class.

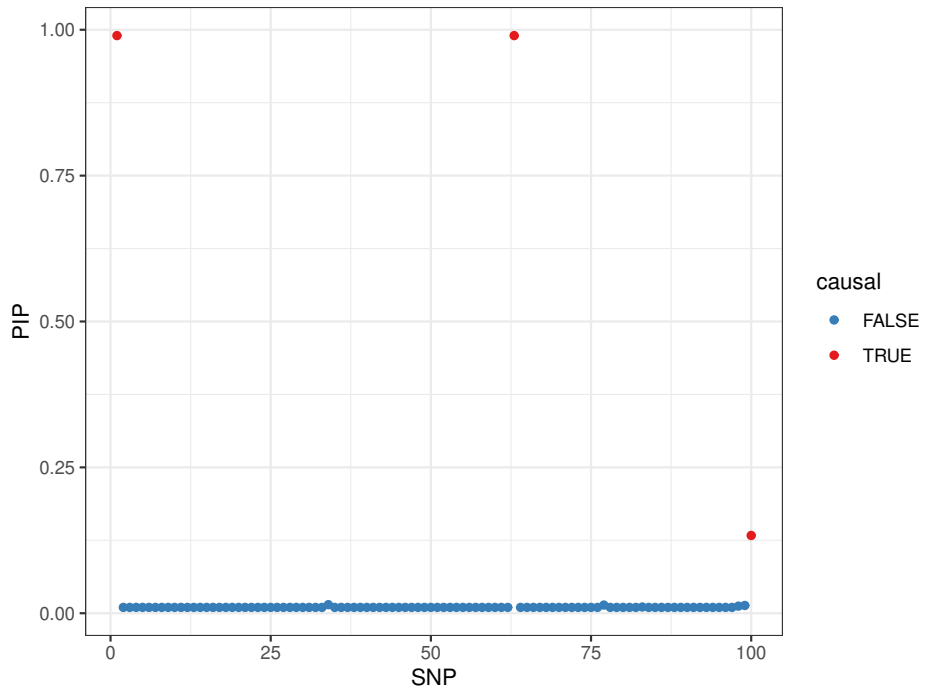

Figure 3: Inferred PIP. Causal SNPs are colorred in red.

6 File to submit

Submit your code with name `COMP565_A3_prs.py` or `COMP565_A3_prs.R` that implements *all of the above five tasks* in ONE Python or R script. We should be able to run your code on the same data and produce the above plots to evaluate the correctness of your implementation.