VARIABLE SELECTION IN VARIATIONAL BAYESIAN STUDIES

BAICHEN TAN

Abstract. This article examines three approaches used in variational Bayesian inference: SuSiE, SparsePro, and SharePro. We explained the theoretical guarantee for the three methods and also present a new model that employs annotation data to boost causal signal detection.

Contents

1. Variational Bayesian Inference

In this section, we will introduce basic ideas about variational Bayesian inference, which is the foundation of both SuSiE and SaprsePro models.

Suppose we have $X = [x_1, \cdots, x_n]^T$ to be observations and $z = [z_1, \cdots, z_m]^T$ to be hidden variables. Let $q(z)$ be the true distribution and $p(z \mid X)$ be the posterior distribution. We use Kullback-Leibler Divergence to measure the closeness of the two distributions.

Definition 1.1. Let $q(z)$ be the true distribution density function and $p(z \mid X)$ be the posterior distribution density function. The Kullback-Leibler (KL) Divergence is defined as

$$\mathrm{KL}(q \parallel p) = \mathbb{E}_q \left[\log \frac{q(z)}{p(z \mid X)} \right] \tag{1.2}$$

Remark 1.3. The KL divergence is non-negative.

Proof. We prove that KL divergence is non-negative by proving that $-\operatorname{KL}$ is smaller or equal to 0.

$$\begin{aligned}
-\operatorname{KL}(q \parallel p) &= -\int_z q(z) \log \frac{q(z)}{p(z \mid X)} dz \\
&= \int_z q(z) \log \frac{p(z \mid X)}{q(z)} dz \\
&\leq \int_z q(z) (\frac{p(z \mid X)}{q(z)} - 1) \\
&= \int_z p(z \mid X) - \int_z q(z) \\
&= 1 - 1 \\
&= 0
\end{aligned}$$

where the third inequation holds because $\log(a) \leq a - 1$ for all $a > 0$. Then since $-\operatorname{KL} \leq 0$, we have $\operatorname{KL} \geq 0$. □

Intuitively, if the posterior $p(z \mid X)$ approximates the true distribution $q(z)$ well, then values of q and p will be close to each other so that $\log \frac{q(z)}{p(z|X)}$ will be close to 0, giving a low KL divergence value. Our goal is to pick a q from a family of distributions $\mathcal{Q}$ that minimize the KL divergence. In other words, we want to find $q^*(z)$ such that

$$q^*(z) = \arg\min_{q \in \mathcal{Q}} \operatorname{KL}(q \parallel p)$$

In reality, it is very hard to calculate the KL divergence because the KL divergence implicitly depends on the true distribution $p(X)$ of our our observations, which we do not have access to. To see why this is true, we expand the formula of KL divergence in definition 1.1.

$$\begin{aligned}
\operatorname{KL}(q(z) \parallel p(z \mid X)) &= \mathbb{E}_q \left[\log \frac{q(z)}{p(z \mid X)} \right] \\
&= \mathbb{E}_q \left[\log \frac{q(z)}{\frac{p(z,X)}{p(X)}} \right] \\
&= \mathbb{E}_q \left[\log \frac{q(z) p(X)}{p(z, X)} \right] \\
&= \mathbb{E}_q [\log q(z)] - \mathbb{E}_q [\log p(z, X)] + \mathbb{E}_q [\log p(X)]
\end{aligned}$$

As we can see, because the KL divergence depends on $p(X)$, it is not possible for us to directly calculate its value. However, we can minimize a function that is equal to the KL divergence up to a constant. Namely, we can calculate the value of $\mathbb{E}_q[q(z)] - \mathbb{E}_q[p(z, X)]$.

Definition 1.4. The evidence lower bound (ELBO) of $q(z)$ is defined as the difference between the expectation of $\log p(z, X)$ and the expectation of $\log q(z)$. Namely,

$$\operatorname{ELBO}(q) = \mathbb{E}_q[\log p(z, X)] - \mathbb{E}_q[\log q(z)]$$

We call this value evidence lower bound because it is the lower bound of $\log p(X)$. To see why this is true, we calculate the value of $\log p(X)$ by integrating z over the joint density $p(z, X)$.

$$\begin{aligned}\log p(X) &= \log \int_z p(z, X) dz \\ &= log \int_z p(x, z) \frac{q(z)}{q(z)} dz \\ &= \log \left(\mathbb{E}_q \left[\frac{p(X, z)}{q(z)} \right] \right) \\ &= \mathbb{E}_q[p(z, X)] - \mathbb{E}_q[q(z)]\end{aligned}$$

where the last equation holds because of Jensen's inequality: if f is concave, $f(\mathbb{E}[X]) \geq \mathbb{E}[f(X)]$.

We can observe that

$$\begin{aligned}\mathrm{KL}(q \parallel p) &= \mathbb{E}_q[\log q(z)] - \mathbb{E}_q[\log p(z, X)] + \mathbb{E}_q[\log p(X)] \\ &= - \left(\mathbb{E}_q[\log p(z, X)] - \mathbb{E}_q[\log q(z)] \right) + \mathbb{E}_q[\log p(X)] \\ &= - \mathrm{ELBO} + \mathbb{E}_q[\log p(X)]\end{aligned}$$

In the above equation, $\mathbb{E}_q[\log p(X)]$ is a constant that does not depend on the value of $q(z)$ so we can ignore it. Then, the larger the value of ELBO, the smaller the value of the KL divergence. We thus want to find a $q(z) \in \mathcal{Q}$ such that it maximizes the ELBO. In other words,

$$q^*(z) = \arg\min_{q \in \mathcal{Q}} \mathrm{ELBO}(q)$$

We now explain how we can calculate the value of ELBO. In variational studies, we assume that the variables $z_1, z_2, \cdots, z_m$ from the variational family are independent so that

$$q(z) = q(z_1, \cdots, z_m) = \prod_{i=1}^{m} q(z_i) \tag{1.5}$$

Notice that in general this independence does not hold in the true posterior because the hidden variables are dependent. For example, in the Gaussian mixture model, all of the cluster assignments z_i are dependent on each other.

Now we can use coordinate ascent inference to iteratively optimize the ELBO. Using the independence assumption 1.5, we can decompose $\mathbb{E}[\log q(z_{1:m})]$ by

$$\mathbb{E}[\log q(z_{1:m})] = \sum_{i=1}^{m} \mathbb{E}_{z_i}[\log q(z_i)] \tag{1.6}$$

Then, expanding the formula of ELBO in 1.4, we get

$$\mathrm{ELBO} = \mathbb{E}[\log p(z, x)] - \sum_{i=1}^{m} \mathbb{E}[\log q(z_i)] \tag{1.7}$$

At each iteration, we only have control over one $q(z_k)$ for $k = 1, 2, \cdots, m$. We can consider ELBO as a function of $q(z_k)$ and assume that other variables $z_i, i \neq k$

are fixed. We denote $z_i, i \neq k$ as z_{-k}. We can then further decompose the ELBO function in equation 1.7 by

$$\begin{aligned}
\text{ELBO} &= \mathbb{E}[\log p(z,x)] - \sum_{i=1}^{m} \mathbb{E}[\log q(z_i)] \\
&= \mathbb{E}_{z_k}\left[\mathbb{E}_{z_{-k}}[\log p(z_k, z_{-k}, x_{1:n})]\right] - \mathbb{E}_{z_k}[\log q(z_k)] - \sum_{i\neq k}^{m} \mathbb{E}_{z_i}[q(z_i)] \\
&= \mathbb{E}_{z_k}\left[\mathbb{E}_{z_{-k}}[\log p(z_k, z_{-k}, x_{1:n})]\right] - \mathbb{E}_{z_k}[\log q(z_k)] + \text{constant}
\end{aligned}$$

where we treat $-\sum_{i\neq k}^{m} \mathbb{E}_{z_i}[q(z_i)]$ as constant because we treat $z_i, i \neq k$ as fixed at the iteration of updating z_k. Remember that at each iteration, we consider ELBO as a function of $q(z_k)$, so we can write out the ELBO $= \mathbb{E}_{z_k}\left[\mathbb{E}_{z_{-k}}[\log p(z_k, z_{-k}, x_{1:n})]\right] - \mathbb{E}_{z_k}[\log q(z_k)] + \text{constant}$ in terms of integration

$$\begin{aligned}
\text{ELBO} &= \mathbb{E}_{z_k}\left[\mathbb{E}_{z_{-k}}[\log p(z_k, z_{-k}, x_{1:n})]\right] - \mathbb{E}_{z_k}[\log q(z_k)] + \text{constant} \\
&= \int q(z_k)\mathbb{E}_{z_{-k}}[\log p(z_k, z_{-k}, x_{1:n})]dz_k - \int q(z_k)\log q(z_k) dz_k + \text{constant}
\end{aligned}$$

Taking the derivative with respect to $q(z_k)$, we get

$$\frac{d\,\text{ELBO}}{dq(z_k)} = \mathbb{E}_{z_{-k}}[\log p(z_k, z_{-k}, x_{1:n})]dz_k - \log q(z_k)dz_k - 1 = 0$$

This lagrange multiplier leads us to the coordinate ascent update rule for $q(z_k)$

$$q(z_k) \propto \exp\left(\mathbb{E}_{z_{-k}}[\log p(z_k, z_{-k}, x_{1:n})]\right) \tag{1.8}$$

We can also write the update rule in terms of the conditional density function $p(z_k \mid z_{-k}, x_{1:n})$:

$$q(z_k) \propto \exp\left(\mathbb{E}_{z_{-k}}[\log p(z_k \mid z_{-k}, x_{1:n})]\right) \tag{1.9}$$

because the denominator of $p(z_k \mid z_{-k}, x_{1:n}) = \frac{p(z_k, z_{-k}, x_{1:n})}{p(z_{-k}, x_{1:n})}$ does not depend on z_k.

In the next three sections, we will explain how SuSiE and SparsePro use the update rules 1.8 and 1.9 to come up with valid algorithms that estimate the posterior distribution.

2. The Sum of Single Effects Models

We start with single effect regression model (SER model). Let y be an n dimensional vector that represents our response variable. Let $X = [x_1, x_2, \cdots, x_n]^T \in \mathbb{R}^{n\times p}$ be our observed SNP for n individuals at p locations. We use $\gamma = [\gamma_1, \gamma_2, \cdots, \gamma_p]^T \in \mathbb{R}^p$ to denote the p dimensional indicator variable that measures which SNP to include. We can write out the model as

$$\begin{aligned}
y &= X\beta\gamma + e \\
\beta &\sim N(0, \sigma_0^2) \\
e &\sim N(0, \sigma^2 I_n) \\
\gamma &\sim \text{MultiNorm}(1, \pi)
\end{aligned}$$

In the above formulas, β measures the effect size and $e \in \mathbb{R}^n$ is the randomm error. $\pi = [\pi_1, \pi_2, \cdots, \pi_p]$ is the prior probability that variable j is the effect variable for $j \in \{1, 2, \cdots, p\}$. Notice γ is a p dimensional vector with only one entry to be non-zero. Our goal is to estimate the posterior distribution of γ and β to obtain the posterior inclusion probabilities (PIPs) and the signal effect. Namely, we want to estimate

$$\gamma \mid X, y, \sigma, \sigma_0 \sim \text{MultiNorm}(1, \alpha)$$

$$\beta_j \mid X, y, \sigma, \sigma_0, \gamma_j = 1 \sim N(\mu_{1j}, \sigma_{1j}^2)$$

where $\beta_j = \beta\gamma_j$.

To estimate the posterior distribution $\alpha = [\alpha_1, \alpha_2, \cdots, \alpha_p]$ of γ, i.e., $\gamma \mid X, Y, \sigma, \sigma_0 \sim$ MultiNorm$(1, \alpha)$, we need the following theorem.

Theorem 2.1. *For each jth SNP, its posterior distribution α_j can be calculated from*

$$\alpha_j = \Pr(\gamma_j \mid X, y, \sigma, \sigma_0) = \frac{\pi_j \operatorname{BF}(x_j, y, \gamma_j, \sigma, \sigma_0)}{\sum_{i=1}^p \pi_i \operatorname{BF}(x_i, y, \gamma_i, \sigma, \sigma_0)} \tag{2.2}$$

where BF *is the Bayes factor*

$$\operatorname{BF}(x_j, y, \gamma_j, \sigma, \sigma_0) = \frac{\Pr(y \mid \gamma_j, x_j, \sigma, \sigma_0)}{\Pr(y \mid \gamma_j = 0, x_j, \sigma, \sigma_0)} \tag{2.3}$$

Proof. We first notice that

$$\Pr(\gamma_j \mid X, y, \sigma, \sigma_0) = \frac{\Pr(\gamma_j, X, y, \sigma, \sigma_0)}{\Pr(X, y, \sigma, \sigma_0)} = \frac{\Pr(X, y, \sigma, \sigma_0 \mid \gamma_j) \Pr(\gamma_j)}{\Pr(X, y, \sigma, \sigma_0)}$$

where $\Pr(\gamma_j) = \pi_j$ is the prior distribution of γ_j. Then, expanding the equation 2.2 by substituting the Bayes factor BF with equation 2.3, we have

$$\begin{aligned}
\Pr(\gamma_j \mid X, y, \sigma, \sigma_0) &= \frac{\pi_j \operatorname{BF}(x_j, y, \gamma_j, \sigma, \sigma_0)}{\sum_{i=1}^p \pi_i \operatorname{BF}(x_i, y, \gamma_i, \sigma, \sigma_0)} \\
&= \frac{\pi_j \frac{\Pr(y|\gamma_j, x_j, \sigma, \sigma_0)}{\Pr(y|\gamma_j=0, x_j, \sigma, \sigma_0)}}{\sum_{i=1}^p \pi_i \frac{\Pr(y|\gamma_i, x_i, \sigma, \sigma_0)}{\Pr(y|\gamma_i=0, x_i, \sigma, \sigma_0)}} \\
&= \frac{\Pr(\gamma_j) \Pr(y \mid \gamma_j, x_j, \sigma, \sigma_0)}{\sum_{i=1}^p \Pr(\gamma_i) \Pr(y \mid \gamma_i, x_i, \sigma, \sigma_0)} \\
&= \frac{\Pr(y, \gamma_j, x_j, \sigma, \sigma_0)}{\Pr(X, y, \sigma, \sigma_0)}
\end{aligned}$$

where the last equation holds because $\Pr(\gamma_i) \Pr(y \mid \gamma_i, x_i, \sigma, \sigma_0) = \Pr(y, \gamma_i, x_i, \sigma, \sigma_0)$ and summing over all γ_is gives $\Pr(y, X, \sigma, \sigma_0)$ □

Then we can estimate β_j by the posterior mean

$$\beta_j = \mathbb{E}[\beta_j \mid X, y, \sigma, \sigma_0] = \alpha_j \mu_{1j}$$

Now suppose instead of having a single effect variable, we have L variables. Intuitively, this means that we are sampling the indicator variable γ L many times and summing them over, which leads to the SuSiE model.

$$
\begin{aligned}
&y = X\beta\gamma + e\\
&\beta = \sum_{l=1}^{L} \beta_l \gamma_l\\
&\beta_l \sim N(0, \sigma_{0l}^2)\\
&e \sim N(0, \sigma^2 I_n)\\
&\gamma_l \sim \text{MultiNorm}(1, \pi)
\end{aligned}
$$

In the model, we assume that the signals are sampled L many times from the multinomial distribution MultiNorm$(1, \pi)$. The indicator variables $\gamma_l \in \mathbb{R}^p, l = 1, 2, \cdots, L$ represents each of the signals. We add up all the indicators to represent the overall effects $\beta = \sum_{l=1}^{L} \beta_l \gamma_l$. Therefore, the SuSiE model can be viewed as the sum of L many single effect model. We assume that the signals are far less than the number of variables, namely, $L << p$. Because we are sampling with replacement, we allow the sampling β_ls to be overlap, and there are at most L non-zero coefficients.

After constructing the SuSiE model, we now turn to estimate the signal effects $\beta_1, \cdots, \beta_L$. Intuitively, given $\beta_1, \cdots, \beta_{L-1}$, estimating β_L involves fitting a single effect model. This suggest an iterative approach to fitting the SuSiE model: at each iteration, we estimate β_l conditioned on $\beta_{l'}, l' \neq l$. We then write out the algorithm for this iterative method.

Algorithm 1: SuSiE Algorithm

```
Input: Data $X, y$
Output: $\alpha = [\alpha_1, \cdots, \alpha_L], \beta = [\beta_1, \cdots, \beta_L]$
Set $\beta_1, \cdots, \beta_L = 0$
repeat
    for $l \in 1, 2, \cdots L$ do
        $\bar{r} \leftarrow y - X \sum_{l=1}^{L} \beta_l$
        $(\alpha_l, \mu_{1l}, \sigma_{1l}) \leftarrow \text{SER}(X, \bar{r}, \sigma, \sigma_{0l})$
        $\beta_l \leftarrow \alpha_l \odot \mu_{1l}$
until Convergence;
return $\alpha = [\alpha_1, \cdots, \alpha_L], \beta = [\beta_1, \cdots, \beta_L]$
```

Under SuSiE, the effect of variable j is $b^j = \sum_{l=1}^{L} b_{lj}$ which is equal to 0 if an only if $b_{lj} = 0$ for all $l = 1, 2, \cdots, L$. Since the b_{lj} are independent across l, we define

Definition 2.4. The posterior inclusion probability of jth variable is

$$PIP_j = \Pr(b^j \neq 0 \mid X, y) = 1 - \prod_{l=1}^{L} (1 - \alpha_{lj})$$

To see why algorithm 1 connects to ELBO maximization, we need to prove the following proposition.

Proposition 2.5. *For variable β_l given β_{-l}, maximizing* $\mathrm{ELBO}(q(\beta_l))$ *is equivalent to solving the single effect regression* $\mathrm{SER}(X, \bar{r}, \sigma, \sigma_{0l})$. *Namely,*

$$\arg\max_{q(\beta_l)} \mathrm{ELBO}(q(\beta_l), q(\beta_{-l}), \sigma, \sigma_0) = \mathrm{SER}(X, \bar{r}, \sigma, \sigma_{0l})$$

where σ_0 denotes $\sigma_{01}, \cdots, \sigma_{0L}$ and $\bar{r} = y - X \sum_{l'} \bar{\beta_{l'}}, l' \neq l$ for $\bar{\beta_{l'}} = \mathbb{E}[\beta_l' \mid X, y, sigma_{0l'}, \gamma_{l'}]$

The formal proof of this proposition can be found at supplement section B of [1]

3. SparsePro Model

In this section, we will explain the SparsePro model, which introduce the concept of annotation in variational inference. In the SuSiE model, we do not specify the prior distribution π for $\gamma_l \sim \mathrm{MultiNorm}(1, \pi)$. Certainly, we can assume that for p SNP, we give equal weights for all the SNPs, namely, $\pi_i = \frac{1}{p}$ for π_i from $\pi = [\pi_1, \cdots, \pi_p]$. However, in many cases, we can obtain side information that helps us specify other forms of prior distribution. For instance, we can give more weights to some of the SNPs if we have some knowledge in advance. We call those information annotations. The SparsePro model provides methods in estimating signal effects with the aid of annotations. The SparsePro Model, however, offers a way for us to construct other forms of prior distribution of π.

Suppose we have p gene SNPs in total and for each $g \in \{1, 2, \cdots, p\}$, we use $A_g \mathbb{R}^{1 \times M}$ to denote its functional annotations. Let $w \in \mathbb{R}^{M \times 1}$ denotes the vector of logrithm of relative enrichment. Then gth gene SNP's prior distribution π_g, we define its value as

Definition 3.1. For the gth gene SNP, let $A_g \mathbb{R}^{1 \times J}$ denote its functional annotations. Let $w \in \mathbb{R}^{J \times 1}$ denotes the vector of logrithm of relative enrichment. Then the prior distribution of gth gene SNP π_g is defined as

$$\pi_g = \mathrm{softmax}(A_g w) = \frac{\exp(A_g w)}{\sum_{i=1}^{p} \exp(A_i w)} \tag{3.2}$$

We use the softmax function to ensure that the prior probability density function π_g is a normalized and valid probability function.

After constructing the prior distribution π by definition 3.1, we can know construct the SparsePro Model. Let $A = [A_1, A_2, \cdots, A_p]^T \in \mathbb{R}^{p \times J}$ be the functional annotation information for the p gene SNPs.

$$\begin{aligned}
&y = X\beta + e \\
&\beta = \sum_{l=1}^{L} \beta_l \gamma_l \\
&\beta_l \sim N(0, \sigma_{0l}^2) \\
&e \sim N(0, \sigma^2 I_n) \\
&\pi = \mathrm{softmax}(Aw) \\
&\gamma_l \sim \mathrm{MultiNorm}(1, \pi)
\end{aligned}$$

We can observe that the SparsePro model shares the same form as the SuSiE model. However, the SparsePro miodel, instead of using an iterative Bayesian algorithm to solve for SER regression models like the SuSiE model does, adopts a variational inference algorithm for estimating the posterior distribution of βs.

Recall that in variational Bayesian inference, our goal is to estimate the posterior distribution

$$p(\gamma_{1:L}, \beta_{1:L} \mid y, X) = \frac{p(y, \gamma_{1:L}, \beta_{1:L} \mid X)}{p(y \mid X)}$$

.

We will use a paired mean field factorized variational family $q(\gamma_{1:L}, \beta_{1:L})$ to approximate the posterior distribution.

$$q(\gamma_{1:L}, \beta_{1:L}) = \prod_{i=1}^{L} q(\gamma_i, \beta_i) = \prod_{i=1}^{L} q(\gamma_i) q(\beta_i \mid \gamma_i) \tag{3.3}$$

where the second equation holds because of the independence assumption in the equation 1.5.

To find the best approximation of the posterior distribution $p(\gamma_{1:L}, \beta_{1:L} \mid y, X)$ by $q(\gamma_{1:L}, \beta_{1:L})$, we need to maximize the ELBO, which is

$$\text{ELBO} = \mathbb{E}_q[\log p(y, \gamma_{1:L}, \beta_{1:L} \mid X)] - \mathbb{E}_q[\log q(\gamma_{1:L}, \beta_{1:L})] \tag{3.4}$$

By the update rule 1.8, the following requirement should be satisfied for each $l \in \{1, 2, \cdots, L\}$.

$$\log q(\gamma_l, \beta_l) = \mathbb{E}_{q(\gamma_{-l}, \beta_{-l})}[\log p(y, \gamma_{1:L}, \beta_{1:L} \mid X)]$$

where γ_{-l} and β_{-l} means taking expectation excluding the lth component. Then we can calculate

$$\begin{aligned} \log q(\gamma_{l,g} = 1, \gamma_{l,-g} = 0, \beta_l) &= \mathbb{E}_{q(\gamma_{-l}, \beta_{-l})}[\log p(y, \gamma_{1:L}, \beta_{1:L} \mid X)] \\ &= \mathbb{E}_{q(\gamma_{-l}, \beta_{-l})}[\log p(y \mid X, \gamma_{1:L}, \beta_{1:L}) \\ &+ \sum_{i}^{L} \log p(\beta_i \mid \sigma_{0i}) + \sum_{i}^{L} \log p(\gamma_i \mid \pi)] \\ &= \text{constant} - \frac{1}{2\sigma_{0l}^2} \beta_l^2 - \frac{1}{2\sigma^2} X_g^T X_g \beta_l^2 \\ &+ \frac{1}{\sigma^2} \beta_l X_g^T (y - X\tilde{\beta}_{-l}) + \log \pi_g \end{aligned}$$

where $\gamma_{l,g}$ means that the gth entry of the lth indicator variable γ_l is 1 and $\gamma_{l,-g}$ means that the entries other than the gth one of the lth indicator variable γ_l is 0. $X_g \in \mathbb{R}^{n \times 1}$ is the vector that only contains the gth SNP column of the dataset X. $\tilde{\beta}_{-l} = \mathbb{E}_{q(\gamma_{-l}, \beta_{-l},)}[\sum_{l' \neq l} \gamma_{l'} \beta_{l'}]$. The formal proof of the calculation process can be found at [2], which involves expanding the probability density functions of multinormal distribution.

Now we have

$$\begin{aligned} \log q(\gamma_{l,g} = 1, \gamma_{l,-g} = 0, \beta_l) = \text{constant} &- \frac{1}{2\sigma_{0l}^2} \beta_l^2 - \frac{1}{2\sigma^2} X_g^T X_g \beta_l^2 \\ &+ \frac{1}{\sigma^2} \beta_l X_g^T (y - X\tilde{\beta}_{-l}) + \log \pi_g \end{aligned}$$

because we know that the posterior

$$q(\beta_l \mid \gamma_{l,g} = 1, \gamma_{l,-g} = 0) \sim N(\mu_{lg}, \sigma_{lg}^2) \tag{3.5}$$

Matching the sufficient statistics of $q(\gamma_{l,g} = 1, \gamma_{l,-g} = 0, \beta_l)$ for this normal distribution 3.5, we have

$$\sigma_{lg}^2 = \frac{1}{\sigma^2 X_g^T Xg + \sigma_{0l}^2} \tag{3.6}$$

$$\mu_{lg} = \frac{\sigma^2}{\sigma_{lg}^2} X_g^T (y - X\tilde{\beta}_{-l}) \tag{3.7}$$

By integrating β_k in $\log q(\gamma_{l,g} = 1, \gamma_{l,-g} = 0, \beta_l)$, we obtain

$$\log q(\gamma_{l,g} = 1, \gamma_{l,-g} = 0) = \log \pi_g - \frac{1}{2} \log \frac{1}{2\pi\sigma_{lg}^2} + \frac{1}{2\sigma_{lg}^2}\mu_{lg}^2 + \text{constant}$$

Therefore, the posterior probability of gth variant being causal in the lth effect group can be estimated as

$$\gamma_{l,g}^* = q(\gamma_{l,g} = 1, \gamma_{l,-g} = 0) = \text{softmax}(\log \pi_g - \frac{1}{2} \log \frac{1}{2\pi\sigma_{lg}^2} + \frac{1}{2\sigma_{lg}^2}\mu_{lg}^2) \tag{3.8}$$

Then, we can take $\gamma_{l,g}^*$ back to the ELBO formula and obtain

$$\begin{aligned} \text{ELBO} &= \text{constant} + \sum_{l,g} \gamma_{l,g}^* \log \pi_g \\ &= \text{constant} + \sum_{l,g} \gamma_{l,g}^* \log \frac{\exp(A_g w)}{\sum_g \exp(A_g w)} \\ &= \text{constant} + \sum_{l,g} \gamma_{l,g}^* \left(A_g w - \log(\sum_g \exp(A_g w)) \right) \end{aligned}$$

We use w_j to denote the kth entry of the enrichment w. To maximize

$$\text{ELBO} = \text{constant} + \sum_{l,g} \gamma_{l,g}^* \log \pi_g = \text{constant} + \sum_{l,g} \gamma_{l,g}^* \left(A_g w - \log(\sum_g \exp(A_g w)) \right) \tag{3.9}$$

We can take partial derivative of ELBO with respect to w_j.

Theorem 3.10. *The partial derivative of ELBO with respect to w_j is*

$$\frac{\partial\, \text{ELBO}}{\partial w_j} = r_1 - (r_1 + r_0) \frac{k_1 e^{w_j}}{k_1 e^{w_j} + k_0}$$

where

$$k_1 = \sum_g [A_{gj} = 1]\, \text{softmax}(\sum_{m'=m} A_{gm'})$$

$$k_0 = \sum_g [A_{gj} = 0]\, \text{softmax}(\sum_{m'=m} A_{gm'})$$

$$r_1 = \sum_{k,g} [A_{gj} = 1] \gamma_{kg}^*$$

$$r_0 = \sum_{k,g} [A_{gj} = 0] \gamma_{kg}^*$$

The proof of this theorem can be found at the supplemntary of [2].

Setting the derivative to 0, we have

$$\frac{k_1 e^{w_j}}{k_1 e^{w_j} + k_0} = \frac{r_1}{r_1 + r_0}$$

and solve for w_j

$$w_j = \log\left(\frac{r_1/r_0}{k_1/k_0}\right)$$

We can notice that while SparsePro and SuSiE both apply ELBO to estimate the posterior distribution, they adopt ELBO in different ways. SuSiE uses an iterative Bayesian approach to solve a single effect model for each signal $l = 1, 2, \cdots, L$, while SparsePro directly take partial derivatives on ELBO to solve the optimization problem.

Now we can present the algorithm of SparsePro.

Algorithm 2: SparsePro Algorithm

```
Input: Data $X, y$
Output: $\alpha = [\alpha_1, \cdots, \alpha_L], \beta = [\beta_1, \cdots, \beta_L]$
Calculate $X^T X, X_g^T y$
repeat
    for $l \in 1, 2, \cdots L$ do
        update $\alpha_l = q(\gamma_l)$
        update $q(\beta_l \mid \gamma_l)$
    estimate $w$
    update $\pi$
until Convergence;
return $\alpha = [\alpha_1, \cdots, \alpha_L]$
```

In Sparsepro, the posterior inclusion probability is defined as

Definition 3.11. The posterior inclusion probability is defined as

$$PIP_j = \max(\alpha_{1j}, \alpha_{2j}, \cdots, \alpha_{Lj})$$

4. Models with multiple response types

So far we have been working with a single response type. In this section, we will introduce a model that uses expectation-maximization algorithm to estimate posterior distribution when multiple response types are present. Suppose we have m different response types, the key idea of this algorithm is to assume that for each response type, they share the same π defined in 3.1. Then, at each iteration, we first run SuSiE on each of the response type and aggregate their posterior inclusion probabilities $\alpha = [\alpha_1, \cdots, \alpha_p]$ to estimate the prior π. Intuitively, if the response types are similar (for example, they are all heart diseases), we hope that, assuming that they share the same prior π, they can aid in each other in detecting more signals.

We now present the EM model.

let $m = 1, ..., M$ be responses type. We have the base model, for each response type $y^{(M)}$,

$$y^{(M)} = X\beta^m + \epsilon$$
$$\epsilon \sim N_n(0, \sigma^2 \boldsymbol{I}_n)$$

For individual causal SNP $\beta^{(m)} = \sum_{l=1}^{L} \beta_l^{(m)} \gamma_l^{(m)}$, where

$$\gamma_l^{(m)} \overset{\text{i.i.d}}{\sim} MultiNorm(1, \pi)$$

Notice that we assume that for different response type $\gamma^{(m)}$, they share the same distribution probability π.

$$\beta_l^{(m)} \overset{\text{i.i.d}}{\sim} N_1(0, \sigma_{0l}^2)$$

We introduce EM algorithm for estimating the parameter π.

4.1. **E step.** We denote $\boldsymbol{\gamma} = [\gamma_1^{(m)}, ..., \gamma_L^{(m)}]$. Then our full log-likelihood is equal to

$$\sum_m \mathbb{E}_{\boldsymbol{\gamma}|y^m\pi^t}[\mathrm{LH}(y^s, \boldsymbol{\gamma})] = \sum_m \mathbb{E}_{\boldsymbol{\gamma}|y^m\pi^t}[\mathrm{LH}(y^m \mid \boldsymbol{\gamma})] + \mathbb{E}_{\boldsymbol{\gamma}|y^m\pi^t}[\mathrm{LH}(\boldsymbol{\gamma} \mid \pi)]$$

where LH denotes the likelihood function.

Notice that the first term doesn't depend on π, so our EM algorithm aims for a π that maximizes the second term. So our problem becomes

$$\pi^{t+1} = \arg\max_{\pi} \mathbb{E}_{\boldsymbol{\gamma}|y^m\pi^t}[\mathrm{LH}(\boldsymbol{\gamma} \mid \pi)]$$

4.2. **M step.**

$$\pi^{t+1} = \arg\max_{\pi} \mathbb{E}_{\boldsymbol{\gamma}|y^m\pi^t}[\mathrm{LH}(\boldsymbol{\gamma} \mid \pi)]$$

To solve this maximization problem, notice that $\gamma_l^{(m)} \overset{\text{i.i.d}}{\sim} MultiNorm(1, \pi)$, therefore

$$\mathrm{LH}(\boldsymbol{\gamma} \mid \pi) = \sum_{l=1}^{L} LH(\gamma_l^m \mid \pi) = \sum_{l=1}^{L} \log(\prod_{j=1}^{p} \pi_j^{\gamma_{l,j}^m}) = \sum_{l=1}^{L} \sum_{j=1}^{p} \gamma_{l,j}^m \log(\pi_j)$$

Then, substitute this into our formula, we have

$$\sum_m \mathbb{E}_{\boldsymbol{\gamma}|y^m\pi^t}[LH(\boldsymbol{\gamma} \mid \pi)] = \sum_m \sum_{l=1}^{L} \sum_{j=1}^{p} \mathbb{E}_{\boldsymbol{\gamma}|y^m,\pi^t}[\gamma_{l,j}^m] \log(\pi_j) = \sum_m \sum_{j=1}^{p} \alpha_{j,m} \log(\pi_j)$$

where $\alpha_{j,m}$ denotes the posterior inclusion probability of jth SNP in the mth response type.

We denote $\log(\pi_j)$ as λ_j and run poisson regression

$$\sum_m \alpha_{j,m} \sim Poisson(\lambda_j)$$

Remark 4.1. Here we claim that the MultiNomial GLM and the Poisson GLM are equivalent because:

We denote each $\sum_m PIP_{j,m}$ as y_j. First, we want to maximize the value of the function $\sum_{j=1}^{p} y_j \log \pi_j$.

Notice that this is in the same form of the likelihood function for a multinomial response model, choose π_J to be our baseline category so we can construct the multinomial model for a single observation

$$\log \frac{\pi_j}{\pi_J} = \sum_k x_{jk} \beta_k$$

where x_{jk} is the annotated information for the j^{th} SNP, and we can write the formula in a matrix form

$$\log \frac{\pi_j}{\pi_J} = X_j \boldsymbol{\beta_j}$$

where X_j and $\boldsymbol{\beta_j}$ denotes the annotated variables and their respective parameter values for the j^{th} SNP.

In this way, we can notice that $\pi_j = \frac{\exp X_j \boldsymbol{\beta_j}}{\sum_i \exp X_i \boldsymbol{\beta_i}}$

This is the same as running a poisson loglinear model: let $\lambda_j = \mathbb{E}[y_j]$. Then we can construct the model

$$\lambda_j = \exp(\sum_k x_{jk}\beta_k)$$

where x_{jk}'s are the annotated information for the j^{th} SNP. Namely, we construct $y_j \sim Poisson(\lambda_j)$ and then $\sum_j y_j \sim Multi(n, \pi)$ where $\pi = (\pi_1, ..., \pi_J)$ and

$$\pi_j = \frac{\lambda_j}{\sum_i \lambda_i}$$

.

The likelihood factors into two independent functions, one for $\sum_j \lambda_j$ and the other for π. The total n carries no information about π and vice-versa. Therefore, likelihood-based inferences about are the same whether we regard $y_1, ..., y_j$ as sampled from j independent Poissons or from a single multinomial, and any estimates, tests, etc. for π or functions of π will be the same, whether we regard n as random or fixed.

Algorithm 3: Sussie with same pi

```
Data: SNP statistics, response type 1...m
Result: $\pi$
initialize posterior $\pi_j = 1$
repeat
    for each trait 1, ..., m
        run Sussie to get $PIP_{j,m}$
    estimate $\sum_m PIP_{j,m} \sim Poisson(\lambda_j)$
until convergence return $\pi_j$
```

Comparing the EM algorithm with the SparsePro model, we can observe that the M step is essentially the same as maximizing the ELBO in the SparsePro model. To take a closer look, in the EM algorithm, we want to maximize the term $\sum_m \sum_{j=1} \alpha_{j,m} \log(\pi_j)$, which shares mathematically the same form as the equation in 3.9.

4.3. **SharePro Model.** In this section, we introduce the SharePro model, which is adapted from the SparsePro model. Specifically,

$$\begin{aligned}
& y_1 = X_1 \sum_{l=1}^{L} \beta_{l1} \gamma_l c_{l1} + e \\
& \beta_{l1} \sim N(0, \sigma_{l1}^2) \\
& y_2 = X_2 \sum_{l=1}^{L} \beta_{l2} \gamma_l c_{l2} + e \\
& \beta_{l2} \sim N(0, \sigma_{l2}^2) \\
& e \sim N(0, \sigma^2 I_n) \\
& \gamma_l \sim \text{MultiNorm}(1, \pi) \\
& c_{l1} c_{l2} \sim \text{Bernoulli}(p)
\end{aligned}$$

where y_1, y_2 represent two different response types, X_1, X_2 represents two different variable values, and $\beta_{l1}, \beta_{l2}, \gamma$ follows the same definition as in SparsePro model. We also include two indicator variables c_{k1}, c_{k2} from a Bernoulli distribution with probability p to indicate whether each lth effect group is causal for y_1 and y_2.

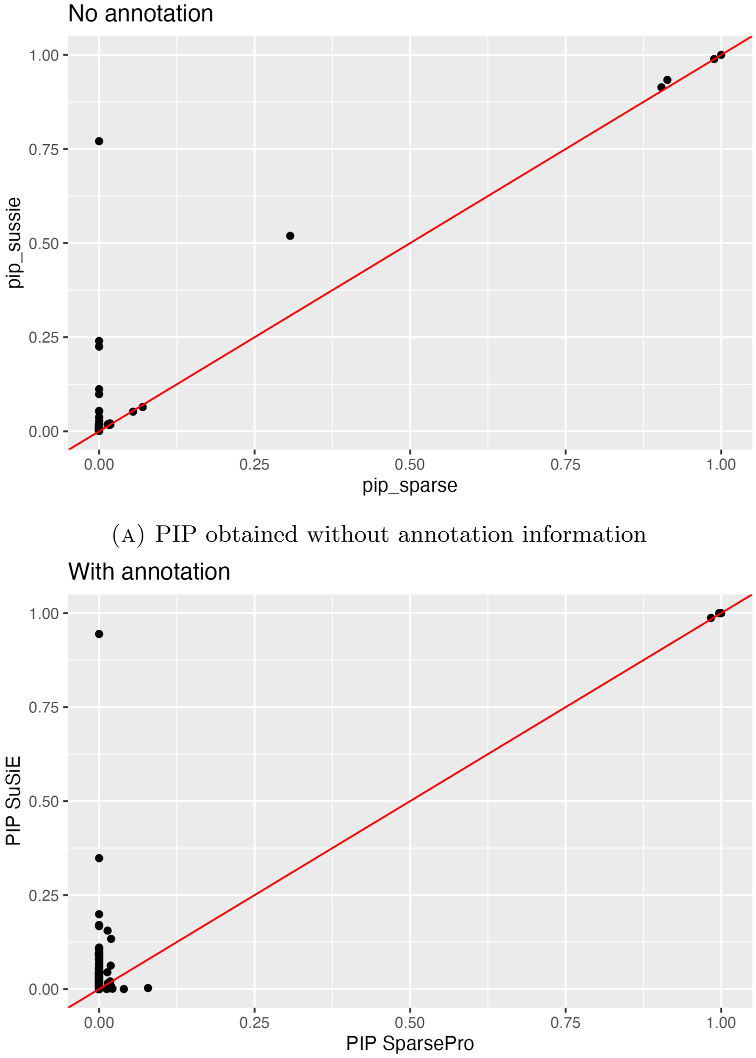
5. Comparison of Models

In this section, we first compare the performances of SuSiE, SparsePro when there is a single response type. For different response types, we will compare the performance of the EM algorithm when sharing prior distribution and not sharing prior distribution.

5.1. **Comparison of SuSiE and SparsePro.** To evaluate SuSiE and SparsePro, we apply the simulations used in the SparsePro paper [2]. Specifically, we use the UK Biobank genotype 82 data from multiple genomic loci (Methods), considering different numbers of causal variants and functional enrichment settings. We have set the number of causal variants to 10 and investigated the performance of SuSiE and SparsePro.

For 95% confidence level causal variants without annotation information, SparsePro and SuSiE return the same number of signals: rs557364786, rs117728004. For 95% confidence level causal variants with annotation information, SparsePro and SuSiE returns more but still the same number of signals: rs557364786, rs117728004, rs80055673, rs182440662. Figure 1 shows the value of all PIPs after running two methods without and with annotation information. As we can see, in general, SuSiE returns with higher PIP values. This is because SuSiE returns PIP that combines the α information across L selections while SparsePro only selects the maximum value. Consider for an example that for some jth variable such that $max(\alpha_{1j}, \cdots, \alpha_{Lj}) = \alpha_l j > 0$ for some l and we denote $\alpha_{l'j} > 0, l' \neq l$. Then for SparsePro, we will return $PIP_j = \alpha_{lj}$. For SuSiE, however, we will return

$$1 - \prod_{l=1}^{L} (1 - \alpha_{lj})$$



(A) PIP obtained without annotation information

(B) PIP obtained with annotation information

Figure 1. PIPs Comparison

Because each of $\alpha_{l'j} > 0$ and $1 - \alpha_{l'j} < 1$, the product $\prod_{l=1}^{L}(1-\alpha_{lj}) \leq 1 - \alpha_{lj}$ and so

$$1 - \prod_{l=1}^{L}(1-\alpha_{lj}) >= 1 - (1 - \alpha_{lj}) = \alpha_{lj}$$

Therefore, in general, SuSiE returns higher values of PIPs than SparsePro does, which is indeed reflected in Figure 1.

5.2. **Comparison of the model that share the same prior and the model that doesn't share the same prior when multiple response types exists.** In this section, we want to investigate the performance of the model that share the same prior with one that doesn't share. We continue to use the data from the UK Biobank with 9 different response types. For the model that shares the same prior, we use the model described in section 4; for the model that doesn't share the same prior, we run SuSiE 9 times separetely to obtain the statistics. Theoretically, we should predict that after sharing the same prior π, the PIPs across the 9 different types should have a higher correlation matrix since they are using each other's data information as enrichment while estimating the posterior α.

Figure 2 shows the correlation heatmap of using model that shares the same prior and the one that doesn't share the same prior. To take a closer look, we subtract the two correlation matrix and wish to observe which entries have positive values. The result is plotted in Figure 3. As we predicted, the correlation of PIPs accross 9 responses in general are stronger after assuming that they share the same prior. Figure 4 shows the variance of the estimated PIPs the two models. As we can see, for SNPs whose estimated PIPs are already relatively big, assuming that the models share the same prior will lower their variance.

Acknowledgments

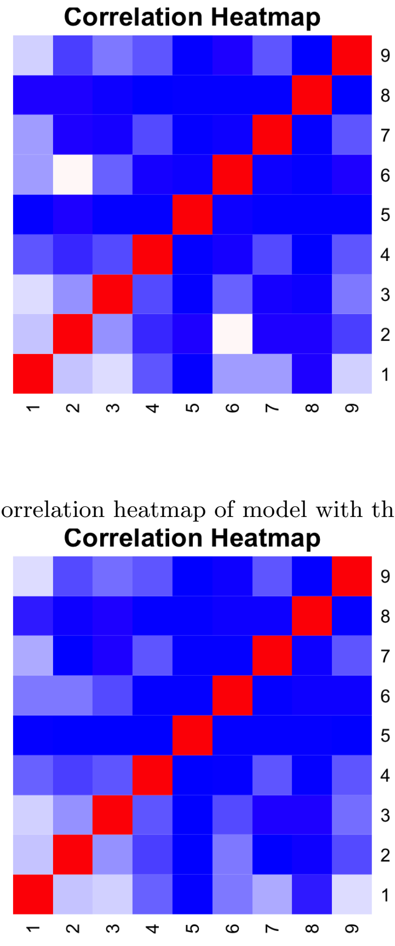
I sincerely thank Prof.Jingshu Wang for giving all the support during the exploration of this research topic.

6. bibliography

References

[1] Wang, G., Sarkar, A., Carbonetto, P. and Stephens, M. (2020). A simple new approach to variable selection in regression, with application to genetic fine mapping. Journal of the Royal Statistical Society, Series B 82, 1273–1300. https://doi.org/10.1111/rssb.12388

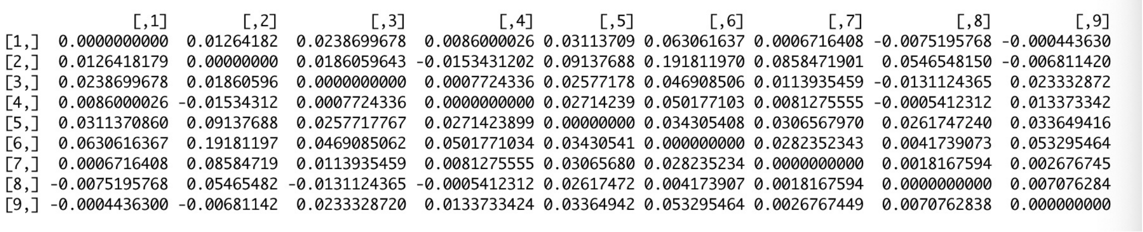
[2] Wenmin Zhang, Hamed Najafabadi, Yue Li. SparsePro: an efficient genome-wide fine-mapping method integrating summary statistics and functional annotations bioRxiv 2021.10.04.463133; doi: https://doi.org/10.1101/2021.10.04.463133



(A) Correlation heatmap of model with the same prior

(B) Correlation heatmap of model with different prior

Figure 2. Correlation Heatmap



```
               [,1]        [,2]          [,3]          [,4]       [,5]        [,6]         [,7]          [,8]         [,9]
[1,]  0.0000000000  0.01264182  0.0238699678  0.0086000026 0.03113709 0.063061637 0.0006716408 -0.0075195768 -0.000443630
[2,]  0.0126418179  0.00000000  0.0186059643 -0.0153431202 0.09137688 0.191811970 0.0858471901  0.0546548150 -0.006811420
[3,]  0.0238699678  0.01860596  0.0000000000  0.0007724336 0.02577178 0.046908506 0.0113935459 -0.0131124365  0.023332872
[4,]  0.0086000026 -0.01534312  0.0007724336  0.0000000000 0.02714239 0.050177103 0.0081275555 -0.0005412312  0.013373342
[5,]  0.0311370860  0.09137688  0.0257717767  0.0271423899 0.00000000 0.034305408 0.0306567970  0.0261747240  0.033649416
[6,]  0.0630616367  0.19181197  0.0469085062  0.0501771034 0.03430541 0.000000000 0.0282352343  0.0041739073  0.053295464
[7,]  0.0006716408  0.08584719  0.0113935459  0.0081275555 0.03065680 0.028235234 0.0000000000  0.0018167594  0.002676745
[8,] -0.0075195768  0.05465482 -0.0131124365 -0.0005412312 0.02617472 0.004173907 0.0018167594  0.0000000000  0.007076284
[9,] -0.0004436300 -0.00681142  0.0233328720  0.0133733424 0.03364942 0.053295464 0.0026767449  0.0070762838  0.000000000
```

Figure 3. Correlation difference of the two models

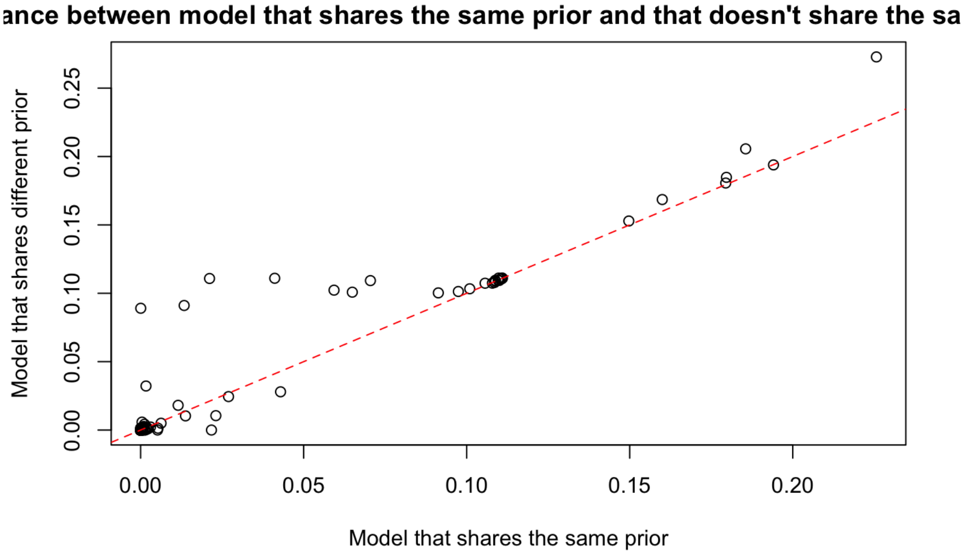


Figure 4. Variance comparison of models with and without the same prior