1 The global testing problem

Let $(X, Y) \in \mathbb{R}^{p+1}$, where X is p-dimensional. Is it of interest to study the testing problem:

$$H_0 : \max_{1 \le j \le p} |\mathrm{corr}(X_j, Y)| = 0, \quad \text{vs} \quad H_1 : \max_{1 \le j \le p} |\mathrm{corr}(X_j, Y)| > 0\,?$$

Would GWAS and/or eQTL be possible applications?

2 Lasso tuning using only summary statistics

Assume we have $Y \in \{0,1\}^n$, $X \in \mathbb{R}^{n \times (p+1)}$, where the first column of X is $\mathbf{1}_n$, which corresponds to the intercept term. We want to learn a linear model that predicts Y from X. Consider the lasso estimator

$$\hat{\beta}_\lambda = \arg\min \frac{1}{n} \|Y - X\beta\|_2^2 + \lambda \|\beta\|_1\,,$$

which is equivalent to

$$\hat{\beta}_\lambda = \arg\min 2R^T\beta + \beta^T C \beta + \lambda \|\beta\|_1\,,$$

where

$$R = \frac{1}{n} X^T Y, \quad C = \frac{1}{n} X^T X$$

are the empirical correlation between X and Y, and the empirical covariance of X. The following discussion will focus on these quantities, assuming that the original data (X, Y) is not accessible.

Remark: In the two-population problem, we may have $C = \gamma C_1 + (1 - \gamma) C_2$, $R = \gamma R_1 + (1 - \gamma R_2)$, where C_j, R_j are the sample correlations from population $j \in \{1, 2\}$. The following development is based on the two-population context. The one-population version is simpler, just dropping all the j indices.

Ideally the tuning of λ should aim at minimizing the predictive risk from one population:

$$\mathrm{Risk}_j(\hat{\beta}_\lambda) = \mathbb{E}\left[(Y^{(j)}_{\text{new}} - (X^{(j)}_{\text{new}})^T \hat{\beta}_\lambda)^2 \middle| \hat{\beta}_\lambda \right]$$

where $(X^{(j)}_{\text{new}}, Y^{(j)}_{\text{new}})$ is an independent draw from the jth population.

However this may be too much to ask. Even with fully accessible data, we need very strong assumptions to achieve this.

A more practical quantity is the "Same-X" predictive risk:

$$\text{Risk}_j(X) = \mathbb{E}\left[\|Y^{(j)}_{\text{new}}(X^{(j)}) - X^{(j)}\hat{\beta}_\lambda\|_2^2 \middle| X^{(1)}, X^{(2)}\right] \tag{1}$$

where $Y^{(j)}_{\text{new}}(X^{(j)})$ is a fresh draw of $Y^{(j)}$ from the conditional distribution of $(Y^{(j)}|X^{(j)})$ (treating the design matrix $X^{(j)}$ as fixed), and the expectation is over both $Y^{(j)}$ and $Y^{(j)}_{\text{new}}$. I.e., how accurate does $X^{(j)}\hat{\beta}_\lambda(Y, X)$ predict $Y^{(j)}_{\text{new}}(X^{(j)})$, the response in population j generated again from the same X?

Based on a result of Efron [2], the Same-X predictive risk can be related to the in sample predictive risk as follows:

$$\text{Risk}_j(X) = \|Y^{(j)} - X^{(j)}\hat{\beta}_\lambda\|_2^2 + 2\sum_{i=1}^{n_j} \text{Cov}(\hat{\mu}_i^{(j)}, Y_i^{(j)})\,, \tag{2}$$

where $\hat{\mu}_i^{(j)} = (X_i^{(j)})^T\hat{\beta}$ is the fitted mean value for $\mathbb{E}(Y_i^{(j)}|X_i^{(j)})$.

The term $\text{Cov}(\hat{\mu}_i, y_i)$ is called the "optimism" (also known as "covariance penalty"), which quantifies the difference between the actual predictive risk and the in-sample predictive risk. It is easy to check that the in-sample predictive risk $\|Y^{(j)} - X^{(j)}\hat{\beta}_\lambda\|_2^2$ can be computed using only the summary statistics:

$$\begin{aligned}\|Y^{(j)} - X^{(j)}\hat{\beta}_\lambda\|_2^2 =& (Y^{(j)})^T Y^{(j)} - 2\hat{\beta}_\lambda^T (X^{(j)})^T Y^{(j)} + \hat{\beta}_\lambda^T (X^{(j)})^T X^{(j)}\hat{\beta}_\lambda \\ =& (Y^{(j)})^T Y^{(j)} - 2n_j\hat{\beta}_\lambda^T R_j + n_j\hat{\beta}_\lambda^T C_j\hat{\beta}_\lambda\,.\end{aligned}$$

Remark: The in-sample prediction error $\|Y^{(j)} - X^{(j)}\hat{\beta}_\lambda\|_2^2$ involves the term $(Y^{(j)})^T Y^{(j)}$, which equals n_j times the average of $Y^{(j)}$ (because the entries are binary), which is also n_j times the first entry of R_j. Therefore, we only need to know n_j, the sample size of population j in the training sample, to compute this quantity.

Remark: If the goal is to select tuning parameters, then we do not even need to evaluate the term $(Y^{(j)})^T Y^{(j)}$, because it does not involve λ, and hence is common for all λ's.

2.1 Approximating the covariance penalty

Let $(Y_1^{(1)*}, ..., Y_{n_1}^{(1)*})$, $(Y_1^{(2)*}, ..., Y_{n_2}^{(2)*})$ be a bootstrap sample, then we can approximate the optimsim term by

$$\widehat{\text{Cov}}(\hat{\mu}_i^{(j)}, Y_i^{(j)}) = \text{Cov}_*(\hat{\mu}_i^{(j)*}, y_i^{(j)*})$$

where $\mu_i^{(j)*} = (X_i^{(j)})^T\hat{\beta}^*$, and $\hat{\beta}^*$ is the bootstrap version of $\hat{\beta}$ using the bootstrap sample.

The bootstrap sample is generated by

$$y_i^{(j)*} = \hat{\mu}_{i,0}^{(j)} + \epsilon_i^{(j)*}$$

where $\epsilon_i^{(j)*}$ is a centered Bernoulli noise such that $\mathbb{E}_* \epsilon_i^{(j)*} = 0$ and $\mathbb{E}_* (\epsilon_i^{(j)*})^2 = \hat{\mu}_{i,0}^{(j)}(1 - \hat{\mu}_{i,0}^{(j)})$.

Remark: Here $\hat{\mu}_0$ is a "preliminary estimate", which is expected to be fairly accurate although not optimal. Such an estimate can usually be optained by a small-ish λ.

Now

$$\begin{aligned}\sum_{i=1}^{n} \mathrm{Cov}_*(\hat{\mu}_i^{(j)*}, Y_i^{(j)*}) =& \sum_{i=1}^{n_j} \Big[\mathbb{E}_* \hat{\mu}_i^{(j)*} (\hat{\mu}_{i,0}^{(j)} + \epsilon_i^{(j)*}) - \mathbb{E}_* \hat{\mu}_i^{(j)*} \mathbb{E}_* (\hat{\mu}_{i,0}^{(j)} + \epsilon_i^{(j)*}) \Big] \\ =& \sum_{i=1}^{n_j} \mathbb{E}_* \hat{\mu}_i^{(j)*} \epsilon_i^{(j)*} \\ =& \mathbb{E}_* (\hat{\beta}^*)^T (X^{(j)})^T \epsilon^{(j)*} \,.\end{aligned}$$

In order to obtain $\hat{\beta}^*$, we will need the bootstrap versions of R_1 and R_2, where

$$\begin{aligned}R_j^* = (X^{(j)})^T Y^{(j)*} =& (X^{(j)})^T (X^{(j)} \hat{\mu}_0 + \epsilon^{(j)*}) \\ =& n_j C_j \hat{\mu}_0 + (X^{(j)})^T \epsilon^{(j)*} \,. \end{aligned} \tag{3}$$

Therefore, in order to obtain the optimism estimate, we only need to generate

$$(X^{(j)})^T \epsilon^{(j)*}$$

for $j = 1, 2$.

Let $X^{(j)} = U_j D_j V_j^T$ be the singular value decomposition of $X^{(j)}$. If we only have the summary statistic C_j, then U_j is not accessible but (V_j, D_j) are. Here we keep in mind the setting that $X^{(j)} \in \mathbb{R}^{n_j \times (p+1)}$ where $p > n_j$. So U_j is full rank orthonormal.

Thus we only need to generate $\tilde{\epsilon}^*$:

$$\tilde{\epsilon}^* = U^T \epsilon^* \,.$$

The frist and second moments of $\tilde{\epsilon}^{(j)*}$ are

$$\mathbb{E}(\epsilon^{(j)*}) = 0$$

and

$$\mathbb{E} \epsilon^{(j)*} (\epsilon^{(j)*})^T = \mathrm{diag}(\hat{\mu}_{1,0}^{(j)} - (\hat{\mu}_{1,0}^{(j)})^2, ..., \hat{\mu}_{n,0}^{(j)} - (\hat{\mu}_{n,0}^{(j)})^2)$$

We can use the following a "partially-second-order" Gaussian approximation

$$\tilde{\epsilon}^{(j)*} \sim N(0, \tau_j^2 I_{n_j}) \tag{4}$$

where

$$\begin{aligned}\tau_j^2 =& \left((\hat{\mu}_0^{(j)})^T \mathbf{1}_{n_j} - (\hat{\mu}_0^{(j)})^T \hat{\mu}_0^{(j)}\right)/n_j \\ =& \left(\hat{\beta}_0^T (X^{(j)})^T \mathbf{1}_{n_j} + \hat{\beta}_0^T (X^{(j)})^T X^{(j)} \hat{\beta}_0\right)/n_j \\ =& \hat{\beta}_{1,0} + \hat{\beta}_0^T C_j \hat{\beta}_0 \,. \end{aligned} \tag{5}$$

Here $\hat{\beta}_{1,0}$ is the first coordinate (i.e., intercept) of the preliminary estimate $\hat{\beta}_0$, and we used the fact that the other columns in $X^{(j)}$ are centered and hence sum to 0.

Procedure: Input: Summary statistics R_j, C_j for $j = 1, 2$ and corresponding sample sizes n_j; Preliminary estimate $\hat{\beta}_0$; bootstrap sample size B

For $b = 1, ..., B$, $j = 1, 2$

1. Generate $\tilde{\epsilon}^{j(*)}$ according to (4) with τ_j^2 given in the formula (5).
2. Generate (emulate) $(X^{(j)})^T \epsilon^{(j)*}$ by $V_j D_j \tilde{\epsilon}^{(j)*}$ where $V_j D_j^2 V_j^T$ is the SVD of nC_j.
3. Compute $\hat{\beta}^*$ using (R_1^*, C_1, R_2^*, C_2), with R_j^* determined by (3).

The covariance panelty in (2) is approximated by the average of $(\hat{\beta}^*)^T (X^{(j)})^T \epsilon^{(j)*}$ over the bootstrap repetitions.

3 Variable selection and fine mapping

Consider regression model

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

where $\beta \in \mathbb{R}^p$ is sparse, with $\|\beta\|_0$ likely being only 1, 2, or 3. The problem of testing whether $\beta = 0$ is the global effect test (which is related to Section 1), and can also be solved using the method presented here.

A main challenge in the fine mapping problem is that the columns of X can be highly correlated. Making it impossible to consistently recover the support of β: $S = \{1 \le j \le p : \beta_j \neq 0\}$. Instead, we aim at finding promising candidates (e.g., confidence sets) of S.

A Bayesian method SuSiE considers a similar problem, but outputs a posterior credible set that covers at least one element of S.

3.1 The proposed procedure

Our idea combines the cross-validation based model confidence set developed by [4] (CVC) and the model path selection idea of [3] (MPS). At a high level, CVC uses the randomness of validating data to construct a confidence set among a given set of candidate models, such that any "nearly optimal" ones are included with pre-specified frequentist coverage probability. The MPS method explores the candidate models in a tree structure using the forward stepwise selection. The proposed method keeps the forward stepwise tree search, using the CVC as selection and stopping criteria.

The CVC-MPS algorithm

Input: Y, X, type I error level α

1. Initialize the depth $k = 0$, and the model confidence set $\hat{\mathcal{T}}_0 = \{\emptyset\}$ having the empty set as the only element.
2. Use CVC to find all models of the form $S \cup \{j\}$ with $S \in \hat{\mathcal{T}}_k$ that have the most significant improvement from all $S \in \hat{\mathcal{T}}_k$.
    (a) If such models exist then use them to form $\hat{\mathcal{T}}_{k+1}$, update $k \leftarrow k+1$, and repeat Step 2.
    (b) If such models do not exist, then the algorithm stops and outputs $\hat{\mathcal{T}}_k$

Remark: one can of course set a pre-specified tree depth (for example, 2).

Details of implementing Step 2: Given $\hat{\mathcal{T}}_k$, let $\mathcal{M}_k = \{S \cup \{j\} :\ S \in \hat{\mathcal{T}}_k,\ 1 \leq j \leq p\} \cup \hat{\mathcal{T}}_k$. In other words, $\mathcal{M}_k$ contains all subsets in $\hat{\mathcal{T}}_k$ and those obtained by augmenting a subset in $\hat{\mathcal{T}}_k$ by one element. Let $\mathcal{A}_k = \mathrm{CVC}(X, Y, \mathcal{M}_k, \alpha)$, where α is a pre-specified type I error (not too conservative, e.g., $\alpha = 0.1,\ 0.2$ etc). If $\mathcal{A}_k \cap \hat{\mathcal{T}}_k \neq \emptyset$, then there exists a model in $\hat{\mathcal{T}}_k$ that performs nearly as well as the best performing augmented model, thus the augmented models do not outperform the current models $\hat{\mathcal{T}}_k$. This is the case (b) in Step 2 above. Otherwise, the models in $\mathcal{M}_k$ all significantly outperform those in $\hat{\mathcal{T}}_k$, so we set $\hat{\mathcal{T}}_{k+1} = \mathcal{M}_k$, and move on to the next step (case (a) of Step 2 above).

properties the estimated model path set

(1) Let S^* be the true support of β. When the algorithm stops, the probability that $S^* \in \hat{\mathcal{T}}_k$ is lower bounded by a pre-specified coverage level.

(2) All models in $\hat{\mathcal{T}}_k$ have similar predictive performance.

The second property comes intuitively from the construction, while the coverage guarantee requires careful analysis and tools in adaptive analysis, as the hypothesis testing in each step is conducted after the previous forward selection. A key technical assumption is that the inclusion of a model in the forward selection confidence set is stable against entry-wise perturbation in the input data.

3.2 Application to fine mapping

Suppose we apply the CVC-MPS algorithm to a fine mapping data set, resulting $\hat{\mathcal{T}}_2$ (suppose the algorithm stops after selecting two variables, either by CVC stopping rule or by pre-specified subset size). The output $\hat{\mathcal{T}}_2$ have the following properties:

(1) For each model $S \in \hat{\mathcal{T}}_2$, the two variables in S have an order of inclusion (one entered in the first step, the other entered in the second step). The one entered in the first step likely has a larger effect size.

(2) For all variables that appear in at least one $S \in \hat{\mathcal{T}}_2$, we can construct a graph, whose nodes are these variables, and an edge between two nodes means that these two variables form an $S \in \hat{\mathcal{T}}_2$. It is natural to expect that such a graph will be close to bi-partite, with one group being the ones highly correlated to the first non-zero coordinate in β, and other one being those highly correlated to the other non-zero coordinate in β.

In short, we should be confident that those entered in the first step cover the first non-zero coordiante (with larger effect size), and those who entered in the second step cover the second.

3.3 Adaptation to more than one population

Suppose we have data from two populations: $\{(X_i, Y_i) : i \in \mathcal{I}_1\}$ and $\{(X_i, Y_i) : i \in \mathcal{I}_2\}$, where $n_2 = |\mathcal{I}_2| \ll |\mathcal{I}_1| = n_1$.

We work under the assumption that the regression coefficients are the same: $Y_i = X_i^T \beta + \epsilon_i$ with the same β for all $i \in \mathcal{I}_1 \cap \mathcal{I}_2$.

Remark: The method described here can be modified to work under the relaxed condition, such that $Y_i = X_i^T \beta_j + \epsilon_i$ for $i \in \mathcal{I}_j$ $(j = 1, 2)$, but β_1, β_2 share the same sparse pattern, and also maybe ϵ_i's have different variance depending on which population the individual i is in.

The multi-population CVC works the same way as the regular cross-validation, exepct the validating sample points are weighted differently. For example, let V be a positive integer that evenly divides both n_1 and n_2. Let $\mathcal{I}_{jv}$ be the vth fold of $\mathcal{I}_j$.

Under the "commmon β" assumption, the weighted cross-validation criterion is

$$\mathrm{CV} = \sum_{j=1}^{2} \sum_{v=1}^{V} \sum_{i \in \mathcal{I}_{1v}} \frac{1}{n_j} \ell(Y_i, X_i; \hat{\beta}_{-v})$$

where $\hat{\beta}_{-v}$ is the regression coefficient obtained using all data except the vth fold.

The CVC modification is straightforward given the weighted cross-validation criterion.

Under the "different β but common support" assumption, the CV criterion is then

$$\mathrm{CV} = \sum_{j=1}^{2}\sum_{v=1}^{V}\sum_{i\in\mathcal{I}_{1v}} \frac{1}{n_j}\ell(Y_i, X_i; \hat{\beta}_{-jv})$$

where $\hat{\beta}_{jv}$ is the regression coefficient using data from $\bigcup_{u\neq v}\mathcal{I}_{ju}$. Here β_1 and β_2 are estimated separately from the two populations, but subject to the requirement of common support. This is applicable if the models to be compared correspond to different support patterns.

4 Co-localization via distribution comparison

The co-localization problem is concerned with two regression models

$$Y = X\beta + \epsilon, \qquad Z = X\gamma + \zeta$$

and aims at testing whether the two sparse regression coefficients β, γ have the same support.

Again, the setting is that X have highly correlated columns and support recovery consistency is impossible, and the level of uncertainty in the estimates is moderate to high.

Bayesian methods such as SuSiE output posterior inclusion probabilities $\mathrm{PIP}_j = \mathbb{P}(\beta_j \neq 0|X, Y)$. The quesiton is how to test whether the two sets of PIP's (one for β and one for γ) suggest co-localization.

In the single-effect version of SuSiE, only one β_j (and γ_j) is non-zero and the PIP_j's can be viewed as a probability distribution over the SNPs. Let $\pi = (\pi_j : 1 \leq j \leq p)$ with $\pi_j = \mathrm{PIP}_j$ for β, and define $\kappa = (\kappa_j : 1 \leq j \leq p)$ similarly for the PIP of γ. The problem reduces to compare the two probability distributions π, κ on $\{1, ..., p\}$.

Which distance matric should we use to measure the agreement between π and κ? One feature of the co-localization problem is that the coordinates of X represent SNPs, which have spatial location. So traditional L_p distances are not suitable, because if π puts all the probablity mass at some j, and κ puts all probability mass at a neighboring $j + 1$, then these two distributions should be considered to be close. But these two distributions will have the maximal L_p distance. To this end, we define the distance between two SNPs j, k as $d(j, k) = |j - k|/p$ (imagine lining up the SNPs on the unit interval with equal distance). Then a sensible choice is the Wasserstein distance [6, 1, 5]:

$$W_q(\pi, \kappa) = \min_c \left[\mathbb{E}_{(Z_1,Z_2)\sim c} d^q(Z_1, Z_2)\right]^{1/q}$$

where the minimization is taken over all joint distributions c on $\{1, ..., p\}^2$ such that its marginal distributions are π and κ respectively. Each $q \geq 1$ defines a Wasserstein distance. For simplicity we can start from the case $q = 1$.

The Wasserstein-1 distance has a clear geometric interpretation (the shortest distance that one needs to move the probability mass to change π to κ), and has a natural range: $W_1(\pi, \kappa) = 0$ if and only if $\pi = \kappa$ and $W_1(\pi, \kappa) = 1$ if and only if π and κ are most different.

Challenges: (1) How does this work if there are multiple effects? The SuSiE PIP will no longer be a probability distribution over the SNPs. (2) How does this work if we use CVC-MPS? In the single effect case, the CVC outputs a collection of confidence sets indexed by the confidence level, but these are not probability distributions. One possibiliy is to aggregate the differences between the confidence set at different levels.

Using CVC for co-localization. The output of CVC is a collection of confidence sets at different confidence levels. We can convert such a collection of confidence sets to a probability distribution and then use the Wasserstein distance to compare the proximity of the two distributions for GWAS and eQTL, using the same idea described above for the PIP distributions. Here we describe how to convert the confidence sets to a probability distribution.

Let A be the set of SNPs. Since there are finitely many SNPs, conceptually we can find a sequence of confidence levels $0 = a_0 < a_1 < a_2 ... < a_K = 1$ such that for each $1 \leq k \leq K - 1$ the corresponding level a_k confidence set A_k records the sequence of distinct confidence sets when we increase the confidence level from 0 to 1. The probability distribution $\pi = [\pi(j) : j \in A]$ corresponding to this collection of confidence sets is

$$\pi(j) = \sum_{k=1}^{K} (a_k - a_{k-1}) \frac{\mathbb{1}(j \in A_k \backslash A_{k-1})}{|A_k \backslash A_{k-1}|}$$

with the default $A_0 = \emptyset$, $A_K = A$.

References

[1] Computational optimal transport. *Foundations and Trends in Machine Learning*, 11(5-6):355–607, 2019.

[2] Bradley Efron. The estimation of prediction error: covariance penalties and cross-validation. *Journal of the American Statistical Association*, 99(467):619–632, 2004.

[3] Nicholas Kissel and Lucas Mentch. Forward stability and model path selection. *arXiv preprint arXiv:2103.03462*, 2021.

[4] Jing Lei. Cross-validation with confidence. *Journal of the American Statistical Association*, 115(532):1978–1997, 2020.

[5] Victor M Panaretos and Yoav Zemel. *An invitation to statistics in Wasserstein space*. Springer Nature, 2020.

[6] Cédric Villani. *Optimal transport: old and new*, volume 338. Springer, 2009.