PRS Parameter Tuning

Tianyu Zhang

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

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

which is equivalent to

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

where

$$R = \frac{1}{n}X^TY, \quad C = \frac{1}{n}X^TX \tag{3}$$

Ideally we want to use summary statistics calculated from the same original data (X, Y). However, in practice, we obtain R from a disease-specific study and the correlation matrix C is calculated from another public source (such as 1000 Genome Project). So we should actually write $C = n^{-1}X_r^\top X_r$ to distinguish the difference between two source genotype matrices X, X_r. Note that we have no access to X or Y, but we do have access to X_r. One important feature that we can leverage is: the rows of X_r are independent from each other and independent from the rows of X. But presumably the rows of X and X_r are drawn from the same distribution.

For each tuning parameter λ, we have an estimate $\hat{\beta}_\lambda$. Suppose the goal of our parameter tuning is identifying the one that minimizes the covariance-penalized risk:

$$\text{Risk}_j(X) = \left\|Y^{(j)} - X^{(j)}\hat{\beta}_\lambda\right\|_2^2 + 2\sum_{i=1}^{n_j}\text{Cov}\left(\hat{\mu}_i^{(j)}, Y_i^{(j)}\right) \tag{4}$$

where $\hat{\mu}_i^{(j)} = X_i^{(j)}\hat{\beta}_\lambda$ is the fitted mean value for $\mathbb{E}\left(Y_i^{(j)} \mid X_i^{(j)}\right)$. Here $j \in \{1,2\}$ is the subpopulation index (European ancestor v.s. African ancestor), n_j is the sample size (in the disease-specific study) of the corresponding population. The Cov penalty is a "population-level" and we need to estimate via bootstrap.

It is easy to check that the in-sample predictive risk $\left\|Y^{(j)} - X^{(j)}\hat{\beta}_\lambda\right\|_2^2$ can be computed using only the summary statistics:

$$
\begin{aligned}
\left\|Y^{(j)} - X^{(j)}\hat{\beta}_\lambda\right\|_2^2 &= \left(Y^{(j)}\right)^T Y^{(j)} - 2\hat{\beta}_\lambda^T \left(X^{(j)}\right)^T Y^{(j)} + \hat{\beta}_\lambda^T \left(X^{(j)}\right)^T X^{(j)}\hat{\beta}_\lambda \\
&\sim \left(Y^{(j)}\right)^T Y^{(j)} - 2\hat{\beta}_\lambda^T \left(X^{(j)}\right)^T Y^{(j)} + \hat{\beta}_\lambda^T \left(X_r^{(j)}\right)^T X_r^{(j)}\hat{\beta}_\lambda \\
&= \left(Y^{(j)}\right)^T Y^{(j)} - 2n_j\hat{\beta}_\lambda^T R_j + n_j\hat{\beta}_\lambda^T C_j\hat{\beta}_\lambda
\end{aligned}
\tag{5}
$$

0.1 Approximating the covariance penalty

Let $\left(Y_1^{(1)*}, \ldots, Y_{n_1}^{(1)*}\right), \left(Y_1^{(2)*}, \ldots, Y_{n_2}^{(2)*}\right)$ be a bootstrap sample, then we can approximate the covariance term by

$$
\widehat{\text{Cov}}\left(\hat{\mu}_i^{(j)}, Y_i^{(j)}\right) = \text{Cov}_*\left(\hat{\mu}_i^{(j)*}, Y_i^{(j)*}\right) \tag{6}
$$

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

The bootstrap sample is generated by

$$
y_i^{(j)*} = \hat{\mu}_{i,0}^{(j)} + \epsilon_i^{(j)*} \tag{7}
$$

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

Remark: Here $\hat{\mu}_0^{(j)} = X^{(j)}\hat{\beta}_0$ 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.

Although the above bootstrap scheme is more rigorous in the sense that the disease-specific genotype matrices $X^{(j)}$ show up in the mathematical formulas, but it has several drawbacks:

1. The matrices $X^{(j)}$ are not available.

2. We eventually still use $X_r^{(j)}$ to calculate the SVD of $X^{(j)}$.

3. The SVD of such a large (block diagonal) matrix is still computationally expensive.

4. There is one step that we need to approximate $\left(X^{(j)}\right)^T \epsilon^{(j)*}$ by $V_j D_j \tilde{\epsilon}^{(j)*}$ where $V_j D_j^2 V_j^T$ is the SVD of nC_j. This approximation omitted the row relationship and is expected to work "on average". I propose approximating $\left(X^{(j)}\right)^T \epsilon^{(j)*}$ by $\left(X_r^{(j)}\right)^T \epsilon^{(j)*}$. These two quantities should be rather similar, since the rows of $X^{(j)}$ and $X_r^{(j)}$ are drawn from the same distribution).

The covariance term is penalizing those hyperparameters λ that lead to overfitting estimators. We just need to evaluate a similar quantity that can achieve similar functions. I propose the following more direct bootstrap scheme:

Let $\left(Y_1^{(1)*}, \ldots, Y_{n_1}^{(1)*}\right), \left(Y_1^{(2)*}, \ldots, Y_{n_2}^{(2)*}\right)$ be a bootstrap sample, then we can approximate the covariance term by

$$\widehat{\mathrm{Cov}}\left(\hat{\mu}_i^{(j)}, Y_i^{(j)}\right) = \mathrm{Cov}_*\left(\hat{\mu}_{i,r}^{(j)*}, Y_{i,r}^{(j)*}\right) \tag{8}$$

where $\hat{\mu}_i^{(j)*} = X_{i,r}^{(j)}\hat{\beta}_\lambda^*$, and $\hat{\beta}_\lambda^*$ is the bootstrap version of $\hat{\beta}_\lambda$ using the bootstrap sample.

The bootstrap sample is generated by

$$Y_{i,r}^{(j)*} = \hat{\mu}_{i,r,0}^{(j)} + \epsilon_{i,r}^{(j)*} \tag{9}$$

Here $\hat{\mu}_{r,0}^{(j)} = X_r^{(j)}\hat{\beta}_0$ can be evaluated. So is the noise: $\epsilon_{i,r}^{(j)*}$ is a centered Bernoulli noise such that $\mathbb{E}_*\epsilon_{i,r}^{(j)*} = 0$ and $\mathbb{E}_*\left(\epsilon_{i,r}^{(j)*}\right)^2 = \hat{\mu}_{i,r,0}^{(j)}(1- \hat{\mu}_{i,r,0}^{(j)}\Big)$. The bootstrap outcome vector $Y_r^{(j)}$ is also available.

Now

$$\sum_{i=1}^{n_j}\mathrm{Cov}_*\left(\hat{\mu}_{i,r}^{(j)*}, Y_{i,r}^{(j)*}\right) = \mathbb{E}_*\left(\hat{\beta}_\lambda^*\right)^T\left(X_r^{(j)}\right)^T\epsilon_r^{(j)*} \tag{10}$$

All the quantities above can be directly evaluated.

References