CV varification and influential edge detection

Chanwoo Lee, October 8, 2020

1 What happens when rank = 1 and sparse =68?

Let $\boldsymbol{B}_i$ be the coefficient matrix and b_i the intercept when the weight $\pi = 0.01 * i$, rank $r = 1$, and sparsity $s = 68$ are used. I found that $\boldsymbol{B}_i$ has only one non zero entry which is located in diagonal location. I think the sparsity constrain impose all non diagonal entries to be 0 because $\boldsymbol{B}_i$ is symmetric. One another interesting thing I noticed is that the intercept has value

$$b_i = \begin{cases} 1 & \text{if } i = 1, \dots 49, \\ -1 & \text{if } i = 50, \dots 99. \end{cases}$$

part 1 is constant zero over all sample points.

From those observations, we can easily check that only part 2 in (1) dominates the predicted $y_k^{(i)}$ for all $k \in \{1, \dots, 114\}$ and $i \in \{1, \dots, 99\}$ resulting in the following prediction.

Does this phenomenon have connection to Figures 3 and 4 in 100120.pdf?

$$y_k^{(i)} = \text{sign}\left(\underbrace{\langle \boldsymbol{B}_i, \boldsymbol{X}_k \rangle}_{\text{part 1}} + \underbrace{b_i}_{\text{part 2}} \right) = \begin{cases} 1 & \text{if } i = 1, \dots 49, \\ -1 & \text{if } i = 50, \dots 99. \end{cases}. \tag{1}$$

Therefore, the estimated probability has $\mathbb{P}(y = 1|\boldsymbol{X}) = 0.49$ for any given $\boldsymbol{X}$. I have checked the loglikelihood values we obtained from cross validation are numerical same as

$$loglikelihood = \sum_{y_i = 1} log(0.49) + \sum_{y_i = -1} log(0.51).$$

This result implies that random guess with probability 0.5 can be the best cross validation performance on test datasets having the smallest log-likelihood.

maximal

What does it imply?
1. signal in data is too small?
2. ADMM is numerically unstable for super sparse + super low rank case?

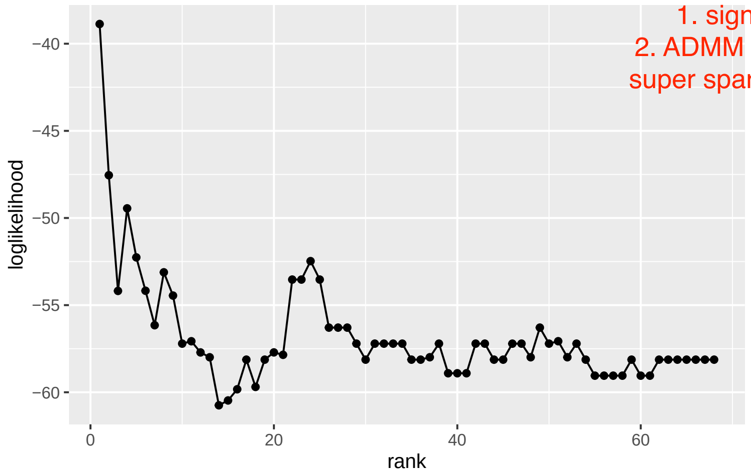

Figure 1: Cross validation result when sparsity $s = 0$

Figure 1 shows that rank 1 performs the best even when there is no sparse structure. Since this result is somewhat different from cross validation result where we consider 0-1 loss based on equal weight classifier, I requested jobs to check whether cross validation result based on log-likelihood from `SMMK` algorithm (with sparsity = 0) has the same trend.

My guess is yes.

2 Influential edges detection based on a set of coefficients

In previous note, I calculate $\mathbb{P}(y = 1|\boldsymbol{A}_{pq})$ and pick the edge (p, q) that has greatest conditional probability. One shortcoming of this approach was that most of edges have conditional probability 0.93. The main reason for this phenomenon is that the intercept dominates the predictions (the coefficient matrix is too small compared to intercept). When we ignore part 1 in (1) setting 0 as,

$$y_k^{(i)} = \text{sign}\,(b_i)\,.$$

We can get exactly conditional probability $\mathbb{P}(y = 1|\boldsymbol{X}) = 0.93$. This explains why many edges have the estimated probability 0.93.

To overcome this situation, we focus on the coefficient matrices $\boldsymbol{B}_i$. Notice that great values of $\langle \boldsymbol{B}_i, \boldsymbol{X}\rangle$ across i implies high impact on prediction and the greater chance of high probability $\mathbb{P}(y = 1\boldsymbol{X})$. Therefore, we can pick the edge (p, q) such that $\langle \boldsymbol{B}_i, \boldsymbol{A}_{pq}\rangle = [\boldsymbol{B}_i]_{pq}$ has significant values away from 0 across $i \in \{1, \dots, 99\}$.

Let $\boldsymbol{B}'_i$ be the truncated $\boldsymbol{B}_i$ such that

$$[\boldsymbol{B}'_i]_{pq} = \begin{cases} [\boldsymbol{B}_i]_{pq} & \text{if } |[\boldsymbol{B}_i]_{pq}| > 0.1 \\ 0 & \text{otherwise.} \end{cases}$$

There are two ways to pick the edges.

1. Pick (p, q) such that $|\{i \in [99] : [\boldsymbol{B}'_i]_{pq} > 0\}| - |\{i \in [99] : [\boldsymbol{B}'_i]_{pq} < 0\}|$ is big, where $|\cdot|$ is a cardinality of a set.
2. Pick (p, q) such that mean $\left(\{[\boldsymbol{B}_i]_{pq}\}_{i=1}^{99}\right)$ is big.

Figure 2 plots $[\boldsymbol{B}_i]_{pq}$ values of the top 5 edges (p,q) across weight $i = 1, \dots 99$ according to two different ways to pick the edges. Unlike other edges whose $[\boldsymbol{B}_i]_{pq}$ values are around 0 across all weights, we can see big values of entries.

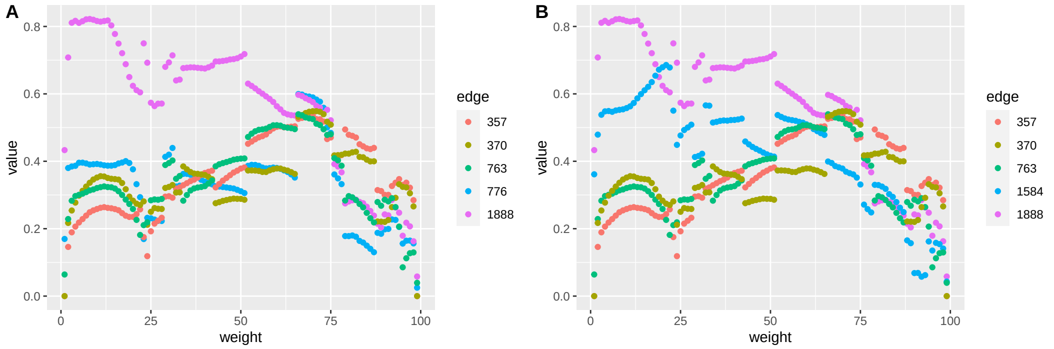

Figure 2: $[\boldsymbol{B}_i]_{pq}$ values of the top 5 edges (p,q) across weight $i = 1, \dots 99$. Figure A is based on the number of positive non zero elements while Figure B on the mean value of elements.

I do not understand Figure 2. Why would we have a value for ``every'' weight?    do you mean ``cumulative mean''?

The following table is comparison of top 5 influential edges based on conditional probability , the number of positive elements, and mean of corresponding entries of the coefficient matrices for each edge. We can check that some edges are contained regardless of methods. Those edges have so strong signal that each method can pick them among all edges.

	conditional probability	the number of positive values	mean of corresponding entries
1st	(13,51)	(13,51)	(40,56)
2nd	(40,60)	(40,56)	(31,40)
3rd	(6,38)	(6,38)	(6,38)
4th	(6,51)	(6,51)	(13,38)
5th	(31,40)	(13,38)	(6,51)

Table 1: top 5 edges according to different criteria. 1st through 4th place are tied on method based on conditional probability.

3 Discussion on λ in consistency proof

Theorem 3.1. *Assume that*

A.1 For some positive sequence such that $s_n \to 0$ as $n \to \infty$, there exists $f_\pi^* \in \mathcal{F}_r(\cancel{M})$ such that $e_V(f_\pi^*, \bar{f}_\pi) \le s_n$.

A.2 There exist constant $0 \le \alpha < \infty$, $a_1 > 0$ such that, for any sufficiently small $\delta > 0$.

$$\sup_{\{f \in \mathcal{F}: e_{V^T}(f, \bar{f}_\pi) \le \delta\}} \|sign(f) - sign(\bar{f}_\pi)\|_1 \le a_1 \delta^\alpha.$$

A.3 Considered feature space is uniformly bounded such that there exists $0 < G < \infty$ satisfying $\sqrt{\mathbb{E}\|\boldsymbol{X}\|_F^2} \le G$

Then, for the estimator $\hat{p}$ obtained from our algorithm with function class $\mathcal{F}_r(M)$, there exists a constant a_2 such that

$$\mathbb{P}\left\{\|\hat{p} - p\|_1 \ge \frac{1}{2m} + \frac{a_1}{2}(m+1)\delta_n^{2\alpha}\right\} \le 15\exp\{-a_2 n(\lambda J_\pi^*)^{2-\alpha\wedge 1}\},$$

provided that $\lambda^{-1} \ge \max\left(\frac{MJ_\pi^*}{2\delta_n^2}, \frac{4J_\pi^*}{\delta_n^2}\right)$ where $J_\pi^* = \max(J(f_\pi^*), 1)$,$\delta_n = \max\left(\mathcal{O}\left(\frac{\log(n/r(d_1+d_2))^{1/(2-\beta)} + 2\log(GM)}{(n/r(d_1+d_2))^{1/(2-\beta)}}\right), s_n\right)$ and $\beta = \alpha \wedge 1$.

In the above theorem, We have the condition $\lambda^{-1} \ge \max\left(\frac{MJ_\pi^*}{2\delta_n^2}, \frac{4J_\pi^*}{\delta_n^2}\right)$. First term $\frac{MJ_\pi^*}{2\delta_n^2}$ comes when I simplify the Assumption 1 confing $\mathcal{F}^V(k)$ as ~~bounded~~ linear function class. To be specific, I replaced $L = L(\epsilon, \lambda, k) = \min\{\epsilon^2 + \lambda(k/2-1)J_\pi^*, 1\}$ by $\mathcal{O}(\epsilon^2)$ assuming $\epsilon^2 > \lambda M J_\pi^*/2 > \lambda(k/2-1)J_\pi^*$. The second term $\frac{4J_\pi^*}{\delta_n^2}$ is from a condition in Theorem 3 in the reference paper. In the proof of Theorem 3, this λ constraint is used to find lower bound the first moment

no need. max_k phi(k) is obtained when k=2

$$\mathbb{E}\{\tilde{V}^T(f, Z) - \tilde{V}^T(f_\pi^*, Z)\},$$

where $\tilde{V}^T(f, z) = (1 - f(Z))_+ \wedge T + \lambda J(f)$.

There is no lower bound of λ but only upper bound of λ is imposed in the proof. In the reference paper, the authors shows consistency in Corollary 1 with extra condition that $n(\lambda J_\pi^*)^{2-\beta}$ is bounded away from 0 to make upper bound of the probability converge to 0.

Assumption 1. *For some constant $a_3, a_4, a_5 > 0$, and $\epsilon_n > 0$,*

$$\sup_{k \ge 2} \int_{a_4 L}^{\sqrt{a_3 L^\beta}} \sqrt{H_2(\omega, \mathcal{F}^V(k))}\, d\omega / L \le a_5\sqrt{n}, \quad \textit{where } L = L(\epsilon, \lambda, k) = \min\{\epsilon^2 + \lambda(k/2-1)J_\pi^*, 1\}.$$

sup is obtained at k=2 as long as entropy H(k) grows slower than k^0.5.
In our case, H(k) ~ log k, slower than any polynomial function of k.

Your calculation in earlier note (0716.pdf) can be simplified by replacing H(x) ~ log x by x^alpha.
Compute the integration of the polynomial function, then let alpha —> 0.