Probability estimation with sparse structure

Chanwoo Lee, October 1, 2020

1 Low rank approximation

To calculate the conditional probability $\mathbb{P}(y = 1|\boldsymbol{X})$ where $\boldsymbol{X} \in \mathbb{R}^{d_1 \times d_2}$ is not necessarily a point in training dataset, we need to calculate $\boldsymbol{B}$ from the output $\boldsymbol{P}_{\text{row}}$, $\boldsymbol{P}_{\text{col}}$, and $\boldsymbol{\alpha}$. I calculate $\boldsymbol{B}$ as

$$\boldsymbol{B} = \sum_{i=1}^{n} \alpha_i y_i \left(\boldsymbol{P}_{\text{row}} \boldsymbol{P}_{\text{row}}^T \boldsymbol{X}_i + \boldsymbol{X}_i \boldsymbol{P}_{\text{col}} \boldsymbol{P}_{\text{col}}^T \right). \tag{1}$$

I have checked $\boldsymbol{B}$ calculated from the formula (1) satisfies

$$\text{fitted}_i = \langle \boldsymbol{B}, \boldsymbol{X}_i \rangle + \text{intercept},$$

where fitted_i and intercept are output of `SMMK_con`. This implies that the estimated predictor is exactly $\langle \boldsymbol{B}, \boldsymbol{X} \rangle + \text{intercept}$. Notice that this $\boldsymbol{B}$ has rank at most $2r$. The following figure shows the probability estimation based on the optimal coefficient $\boldsymbol{B}$.

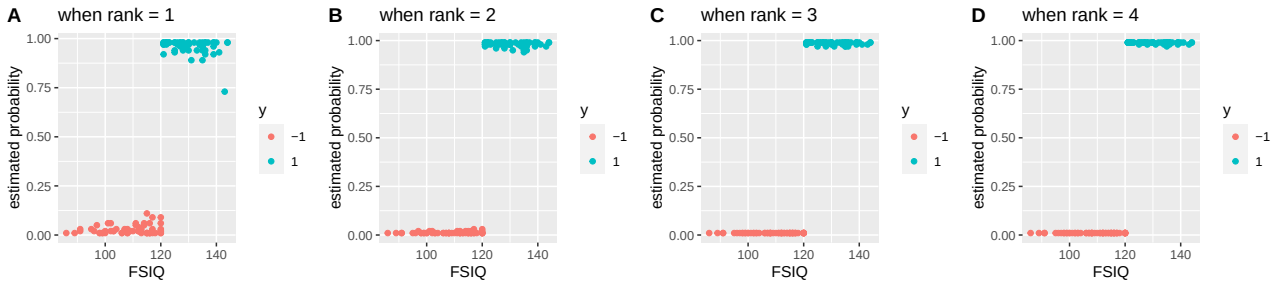


Figure 1: Probability estimation results according to different ranks based on the coefficient $\boldsymbol{B}$.

Since $\boldsymbol{X}_i$ are symmetric, we make $\boldsymbol{B}$ symmetric as $\boldsymbol{B}' = (\boldsymbol{B} + \boldsymbol{B}^T)/2$. I verified everything remains when we use $\boldsymbol{B}'$ instead of $\boldsymbol{B}$. Since we have perfect separation even when rank $= 1$, we can check that the estimated probability has clear separation far from 0.5, from which we can not have information about true IQ values.

To make the coefficient matrix $\boldsymbol{B}$ has rank r, we can use low rank approximation on $\boldsymbol{B}$ from (1) as

$$\tilde{\boldsymbol{B}} = \boldsymbol{U}_r \boldsymbol{\Sigma}_r \boldsymbol{V}_r^T, \tag{2}$$

where columns of $\boldsymbol{U}_r$ and $\boldsymbol{V}_r$ are the first r right and left singular vectors of $\boldsymbol{B}$, and $\boldsymbol{\Sigma}_r$ is a diagonal matrix whose entries are top r singular values of $\boldsymbol{B}$. Notice that there is no guarantee that $\tilde{\boldsymbol{B}}$ is the best r rank coefficient that minimizes the loss function. I have checked that the approximated $\tilde{\boldsymbol{B}}$ cannot perfectly separate the training dataset when the rank is less than 4, which implies $\tilde{\boldsymbol{B}}$ is not a global optimum among rank r matrices.

Figure 2 shows the probability estimation results based on $\tilde{\boldsymbol{B}}$. We can see that the estimated probability has linear relationship to the true IQ values though it looks quite noisy.

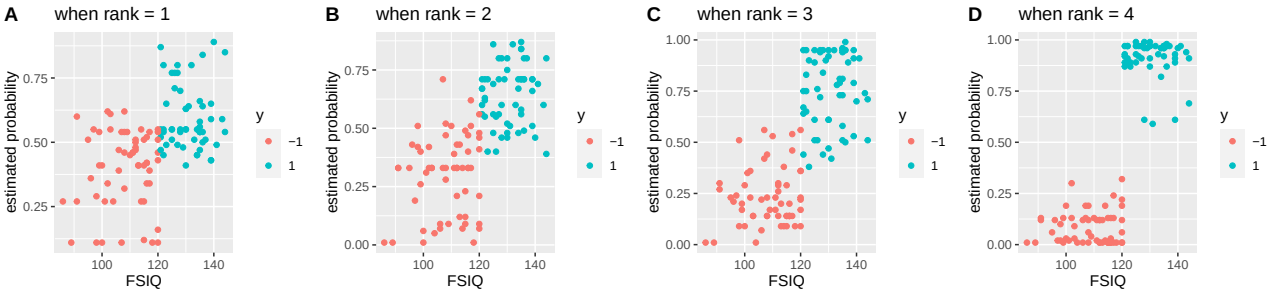


Figure 2: Probability estimation results according to different ranks based on the coefficient $\tilde{\boldsymbol{B}}$.

2 Sparse structure

I first, set the rank and sparsity as (1,66) and estimate the probability based on two updated algorithm `ADMM` and `SMMK_con`. Figure 3 compares 4 different ways to estimate the conditional probability $\mathbb{P}(y = 1|\boldsymbol{X})$. Figure A is the estimation based on $\boldsymbol{B}$ in (1), Figure B is based on low rank approximation $\tilde{\boldsymbol{B}}$ in (2), Figure C is based on the sparse structure with the algorithm `ADMM`, and Figure D is based on the sparse structure with the algorithm `SMMK_con`. We can check that methods based on the sparse structure estimate the probability values centered at 0.5. My guess for the reason is that the sparse structure does not perform well on classification. So I compare the error rates of classification on each method with the training dataset $\{(\boldsymbol{X}_1, y_1)\}_{i=1}^{114}$. The following table shows the error rate on the training set when rank = 1. Considering the response variable is

Method	SMMK	SMMK+ low rank	ADMM	SMMK+low rank+sparse
Error rate	0	0.289	0.403	0.350

Table 1: Error rates on the training dataset according to different methods when rank = 1.

binary, ADMM method has quite huge error rate.

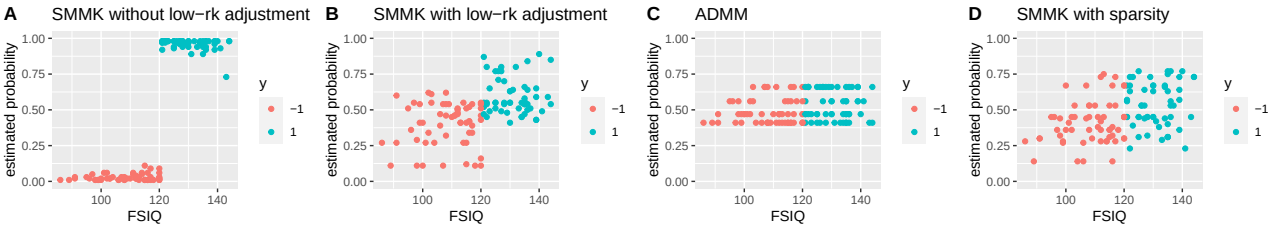


Figure 3: Comparison of the probability estimation when rank = 1.

Therefore, I find another combination of rank and sparsity that has moderate error rates. To this end, I set the rank and sparsity as (2,60) which has small cross validation error in the figure "hinge_test_corner.pdf". The following table shows the comparison of the error rates of each method when the rank = 2.

Figure 4 shows the probability estimation results on each method. The performance of methods based on the sparse structure seems to improve compared to when the rank is one. If you make

Method	SMMK	SMMK+ low rank	ADMM	SMMK+low rank+sparse
Error rate	0	0.157	0.157	0.245

Table 2: Error rates on the training dataset according to different methods when rank = 2.

the classification rule as

$$y = \begin{cases} 1 & \text{if } \mathbb{P}(y = 1|\boldsymbol{X}) > 0.5, \\ -1 & \text{othewise.} \end{cases}$$

We can check that there is less chance to assign y = 1 when IQ value is small or y = -1 when IQ value is high when we use SMMK with the low rank approximation.

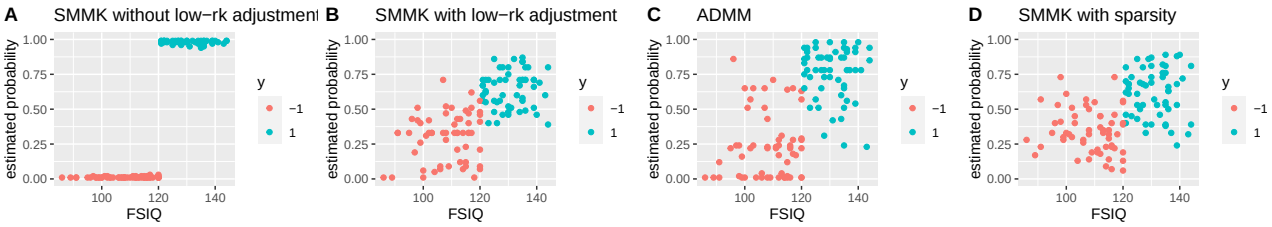


Figure 4: Comparison of the probability estimation when rank = 2.

3 Discussion on setting hyper parameters

We set hyper parameters $(rank, cost, sparsity)$ based on cross validation result when the weight $p = 0.5$, and use the combination of parameters to estimate the conditional probability $\mathbb{P}(y = 1|\boldsymbol{X})$. There are two questions:

1. why do we fix $(rank, cost, sparsity)$ to estimate the probability estimation.
2. why are we based on the best combinations $(rank, cost, sparsity)$ of equal weight $\pi = 0.5$ case.

First, we pre-specify the considered function class

$$\mathcal{F} = \{f(\cdot) = \langle \boldsymbol{B}, \cdot \rangle + b \colon \text{rank}(\boldsymbol{B}) \leq r, \text{sparsity}(\boldsymbol{B}) \leq s\},$$

which means the rank and the sparsity are the global parameters that we have to stick to regardless of π. In addition, our consistency theorems are based on the fixed rank, sparsity, and the cost values. One example is that we have the condition $\lambda^{-1} \geq 4\delta_n^{-2} J_\pi^*$, in the theorem for the consistency probability estimation. We also used the same cost λ in the proof as well as the rank. I have checked that the reference papers [1, 2] keep using the same cost value across different weights.

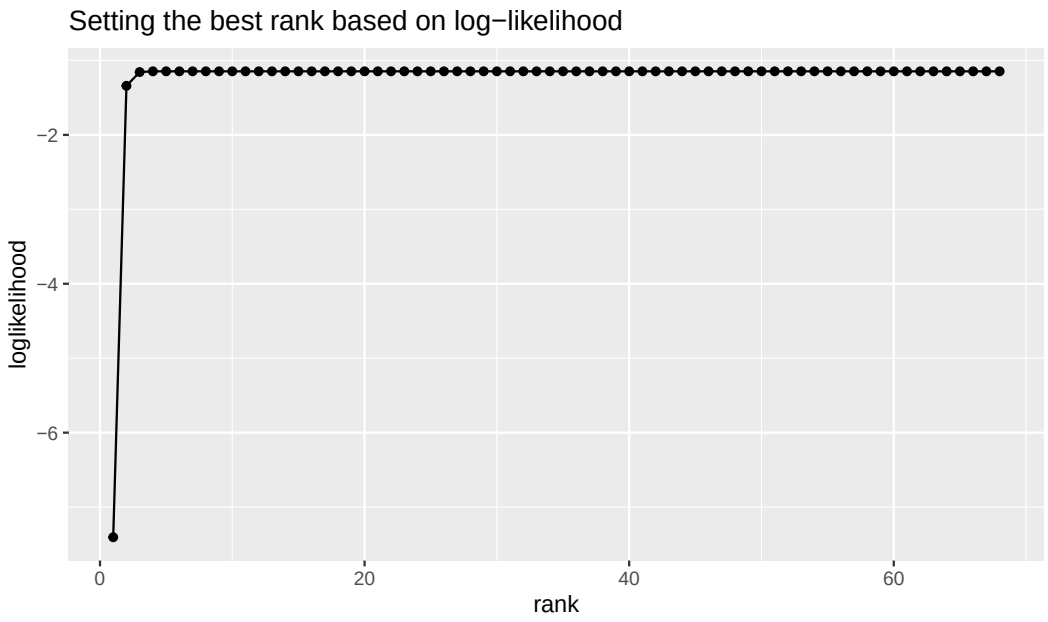
Second, the second question is a fair question. There is no specific reason set the rank based on the cross validation results at equal weight. One possible answer to this is since there is no information about which misclassification is more important (misclassify as y = 1 when y = -1 vs as misclassify y = -1 when y = 1) we assume the two types of misclassification are equally important setting the

case $\pi = 0.5$ as criterion. To avoid this issue, we can use log-likelihood

$$L(cost, rank) = \sum_{i=1}^{n} y_i \log\left(\hat{\mathbb{P}}(y_i = 1|\boldsymbol{X}_i, cost, rank)\right) + (1 - y_i) \log\left(1 - \hat{\mathbb{P}}(y_i = 1|\boldsymbol{X}_i, cost, rank)\right).$$

and choose the combination that maximizes the $L(cost, rank)$ [2]. Another possible approach is to use estimated generalized Kullback-Leibler loss [1]. One issue for alternatives way is that it is time consuming procedure. For example, IQ data set needs at least 68×99 computations to compare the log likelihood for each setting. If we add sparsity, we might have to calculate $68 \times 99 \times sparsity$ combinations to pick the best one.

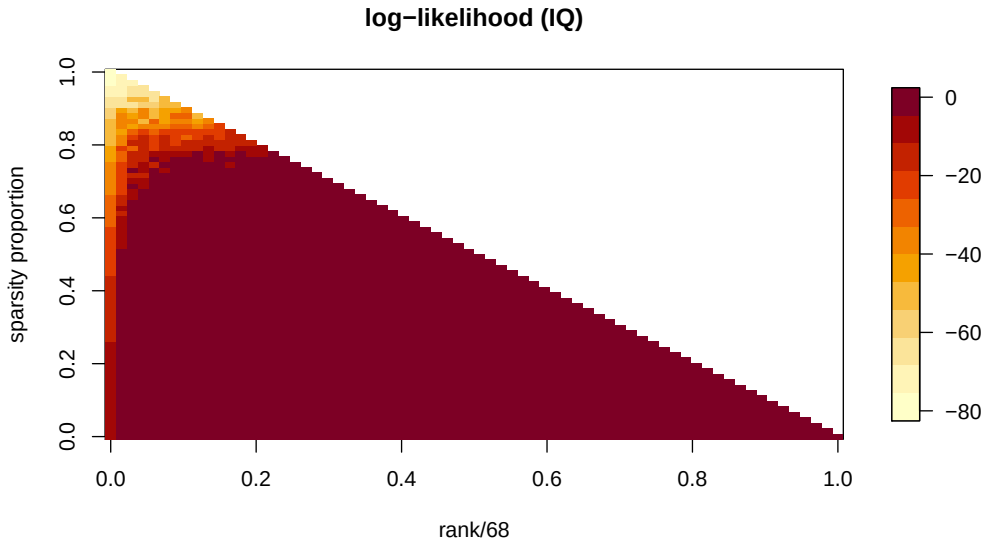
The following figure is the loglikelihood result



References

[1] Junhui Wang, Xiaotong Shen, and Yufeng Liu. Probability estimation for large-margin classifiers. *Biometrika*, 95(1):149–167, March 2008.

[2] Yichao Wu, Hao Helen Zhang, and Yufeng Liu. Robust model-free multiclass probability estimation. *Journal of the American Statistical Association*, 105(489):424–436, 2010.


log−likelihood (IQ)
sparsity proportion
0.0
0.2
0.4
0.6
0.8
1.0
rank/68
0.0
0.2
0.4
0.6
0.8
1.0
0
−20
−40
−60
−80