Simulations for hyperparameters

Chanwoo Lee, September 2, 2020

1 Simulation 1: loss function comparisons

1.1 Simulation setting

1. Obtain submatrices of brain connection matrices such that $\boldsymbol{X}_i \in \mathbb{R}^{10\times 10}$ for $i = 1, \ldots, 212$.
2. Generate ground truth coefficient matrix $\boldsymbol{B} \in \mathbb{R}^{10\times 10}$ whose rank is $1, 3, 5, 7$.
3. Assign the labels for $i = 1, \ldots, 212$ such that

$$y_i = \operatorname{sign}\left(\langle \boldsymbol{B}, \boldsymbol{X}_i \rangle\right).$$

4. Begin 5 folded cross validation at given combinations of (rank, cost) $\in \{1, 2, \ldots, 10\} \times \{2, 4, \ldots, 10\}$.
5. Assess performance based on 0-1 loss, hinge loss, and Frobenius norm $\sum_{i=1}^{2} \|\boldsymbol{P}_i \boldsymbol{P}_i^T - \hat{\boldsymbol{P}}_i \hat{\boldsymbol{P}}_i^T\|^2$ where $\boldsymbol{P}_1, \boldsymbol{P}_2$ are left and right singular matrices of $\boldsymbol{B}$ and $\hat{\boldsymbol{P}}_1, \hat{\boldsymbol{P}}_2$ are estimated row-wise and column-wise projection matrices.

1.2 Simulation results

The results can be summarized by the following table. Figure 1 plots 5-folded CV result with

	Rank 1	Rank 3	Rank 5	Rank 7
0-1 loss	3 (1)	4 (6)	6 (5)	3 (8)
Hinge loss	2 (2)	5 (7)	8 (7)	6 (10)
Frobenius	1 (1)	2 (3)	4 (4)	10 (10)

Table 1: The table shows the estimated rank according to the loss functions given pre-specified true ranks. Numbers in parenthesis shows the estimated rank when feature matrices are not centerized.

different true rank when feature matrices are centered. The results from uncentered feature matrices are more or less the same. When rank is less or equals to 5, Frobenius-based rank estimation performs the best and stable. It seems that 0-1 loss based rank estimation performs better but less stable than hinge loss-based estimation.

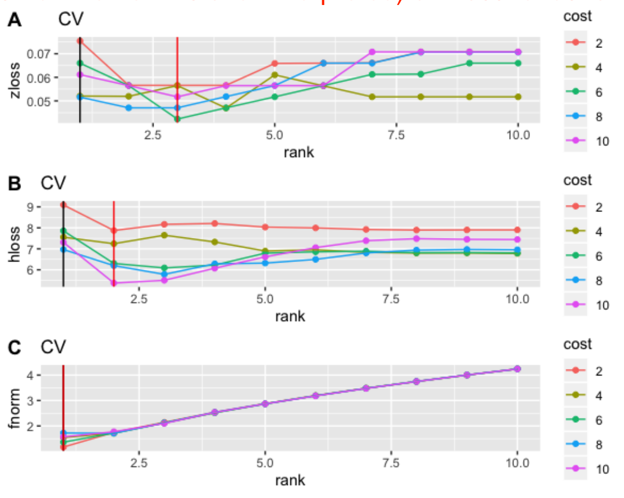
2 Simulation 2: binary vs continuous

2.1 Simulation setting

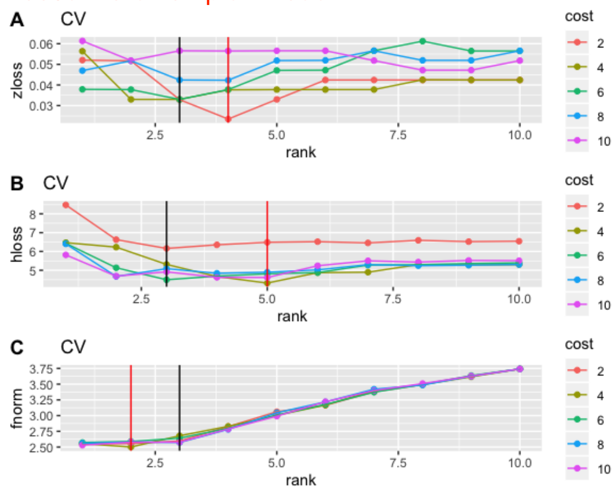
1. Obtain continuous valued matrices $\boldsymbol{X}_i^c \in \mathbb{R}^{10\times 10}$ for $i = 1, \ldots, 212$ whose entries are from i.i.d uniform distribution $U(0, 1)$.
2. Obtain binary valued matrices $\boldsymbol{X}_i^b \in \mathbb{R}^{10\times 10}$ for $i = 1, \ldots, 212$ whose entries are from i.i.d. Bernoulli distribution with probability 0.5.
3. Compare rank estimation performance following the steps 1,2,3 in Simulation 1 setting.

Interesting.

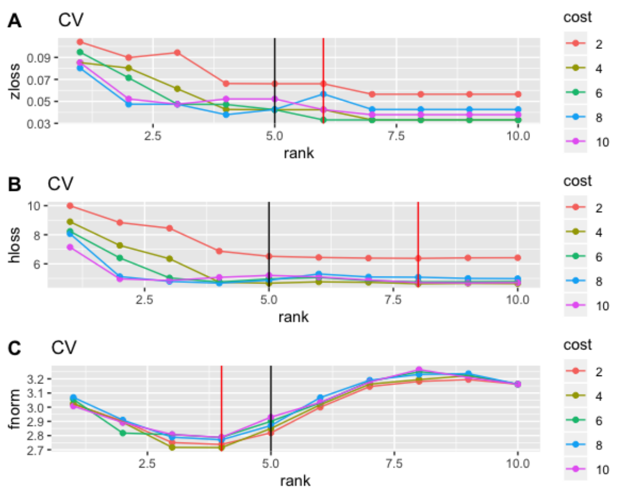
1. Both prediction losses level off in high-rank region —> performance cannot be too bad
2. The accuracy for joint search over (rank, C) is similar to single search over rank given a reasonable C
3. (not sure whether it is over-interpreted) 0-1 loss tends to choose more complex model.



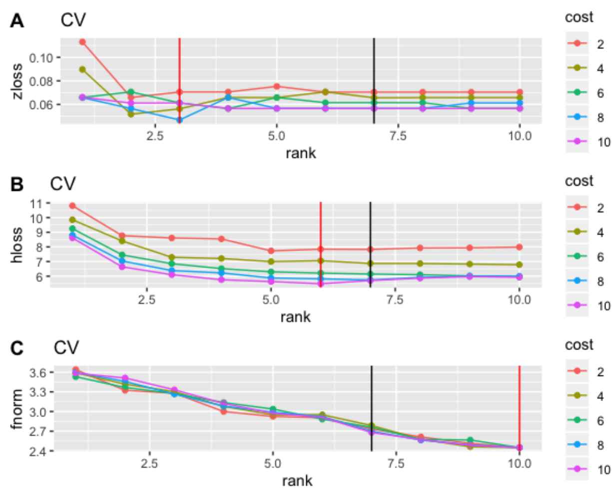
(a) When true rank = 1



(b) When true rank = 3



(c) When true rank = 5



(d) When true rank = 7

Figure 1: The figures plot the loss values according to given rank and cost in Simulation 1. Loss values are A: 0-1 loss, B: hinge loss, and C: Frobenius norm.

2.2 Simulation results

The following figures plot the loss values according to given rank and cost. Loss values are A: 0-1 loss, B: hinge loss, and C: Frobenius norm. It turns out that there is no big difference between continuous type and binary type of feature matrices. There are some tendency I want to comment about the result.

- Frobenius norm estimate the true rank the best in any ranks. The other two losses work moderately in small rank. As true rank increase, it seems to be hard to estimate the rank. But if we see the overall shape of plot, we can check that there is a local minimum around true rank.
- This simulation shows more sensitivity to cost values than Simulation 1. One possible reason is that feature matrices in Simulation 1 are sparse so that chances of having the same prediction are high.

Good explanation.

We can check that rank estimation works better in Simulation 2 than Simuation 1. One possible reason for having different performance for estimating the true rank between brain data based simulation and simulation 2 is sparsity. To be specific Simulation 1 has sparse feature matrices while Simulation 2 has quite dense one. This sparsity makes feature matrices have less information making rank estimation difficult. I verified this from a simulation. In a new simulation , I generated spare binary valued matrices whose entries are from i.i.d. Bernoulli distribution with probability 0.1. Table 2 compares the rank estimation performance between spare binary matrix case (probability = 0.1) and dense matrix case (probability = 0.5). We can check that the case of dense matrix has better estimation result that that of sparse matrix.

	Dense (Probability = 0.5)			Sparse (Probability= 0.1)		
TRUE rank	Rank 3	Rank 5	Rank 7	Rank 3	Rank 5	Rank 7
0-1 loss	4	4	8	2	3	5
Hinge loss	3	4	5	1	3	5
Frobenius	3	6	7	2	4	9

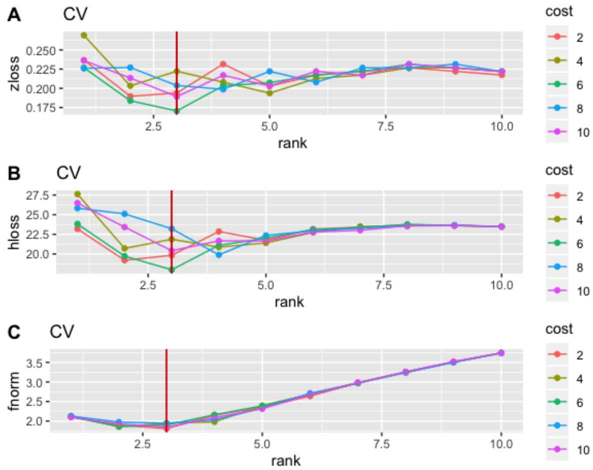
Table 2: Estimated rank based on different loss functions given true rank. The left side if the table is when feature matrices are dense while the right side one is when feature matrices are sparse

Conjecture: sparse + centered (pseudo-sparse) would have similar performance as sparse.
What matters may not be the sparsity per se, but the less similarity between features induced by sparsity

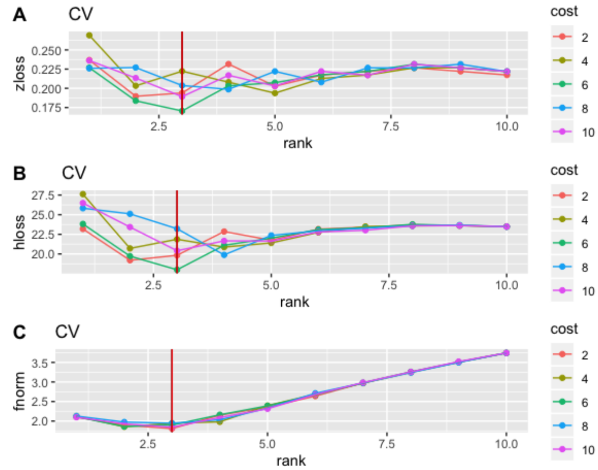
3 Real Brain dataset CV result

I perform cross validation based on our model and assess the performance in respect to 0-1 loss and hinge loss. The following figure shows the trajectory of loss values according to combinations of (rank, cost) values. 0-1 loss based estimation has rank and cost values as (8,2) while hinge loss one has (2,2).

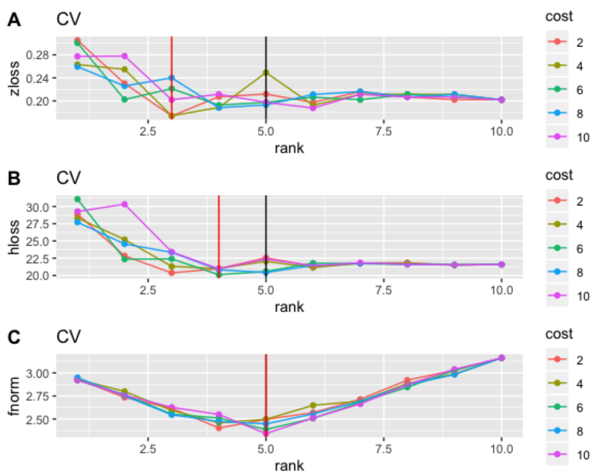
r=8 might be overestimated based on our experience



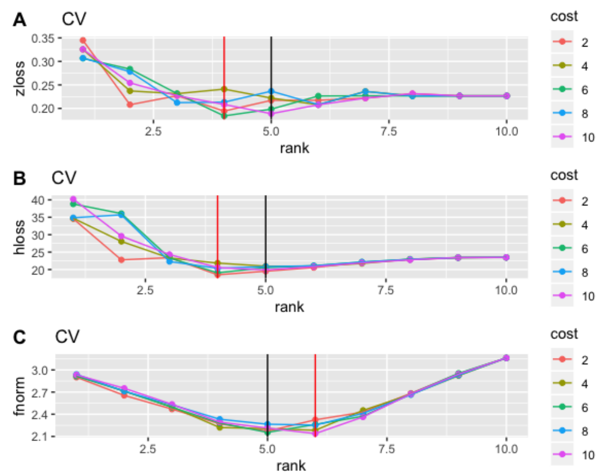
(a) Continuous feature when rank = 3



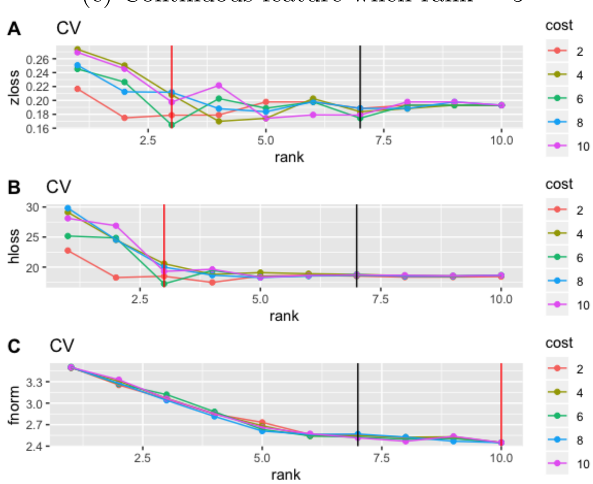
(b) Binary feature when rank = 3



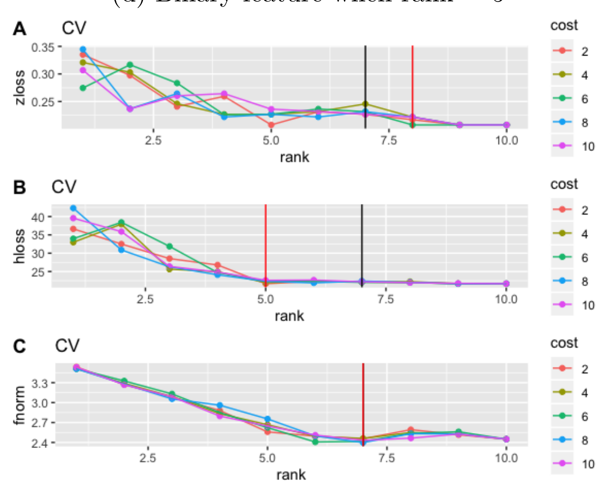
(c) Continuous feature when rank = 5



(d) Binary feature when rank = 5



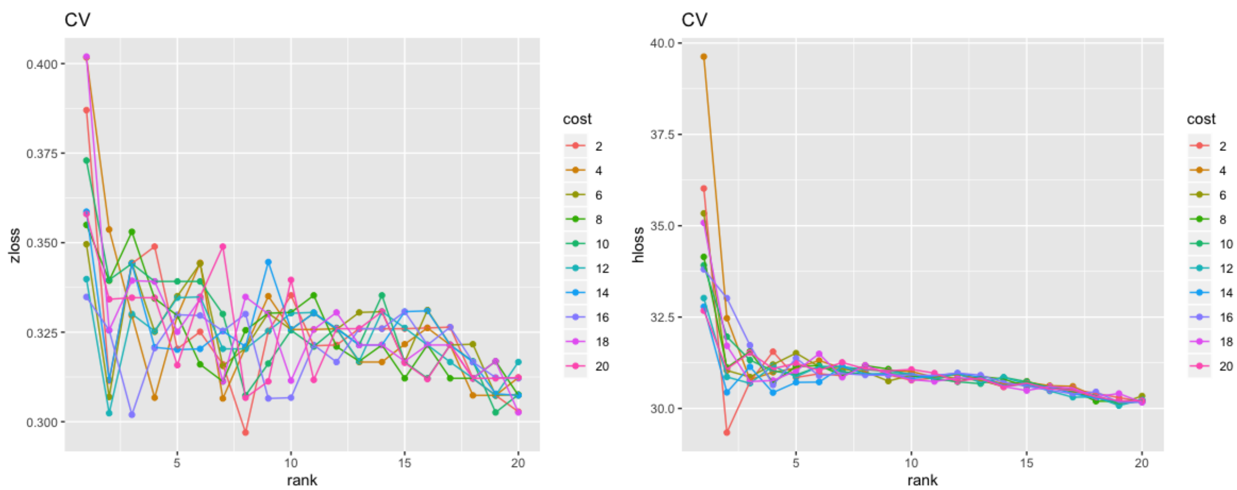
(e) Continuous feature when rank = 7



(f) Binary feature when rank = 7

Figure 2: The figures plot the loss values according to given rank and cost in Simulation 2. Loss values are A: 0-1 loss, B: hinge loss, and C: Frobenius norm.

1. Full data, or sub-data?
2. Simulated y or real y?



(a) 0-1 loss CV

(b) hinge loss CV

Figure 3: Figure (a) shows the averaged 0-1 loss from CV and has the least prediction error at (rank, cost) = (8,2). Figure (B) plots the averaged hinge loss from CV and has the least prediction error at (rank, cost) = (2,2).