Cross validation results on datasets

Chanwoo Lee, September 21, 2020

1 Cross validation on bbnet68 spatial orientation

I perform cross validation with cost values in $\{0.01, 0.1, 1\}$ based on preliminary result that cost values over 1 do not affect the output. I set the 10 multiple initializations to estimate the classifiers. The following is the cross validation result evaluated by 0-1 loss and hinge loss.

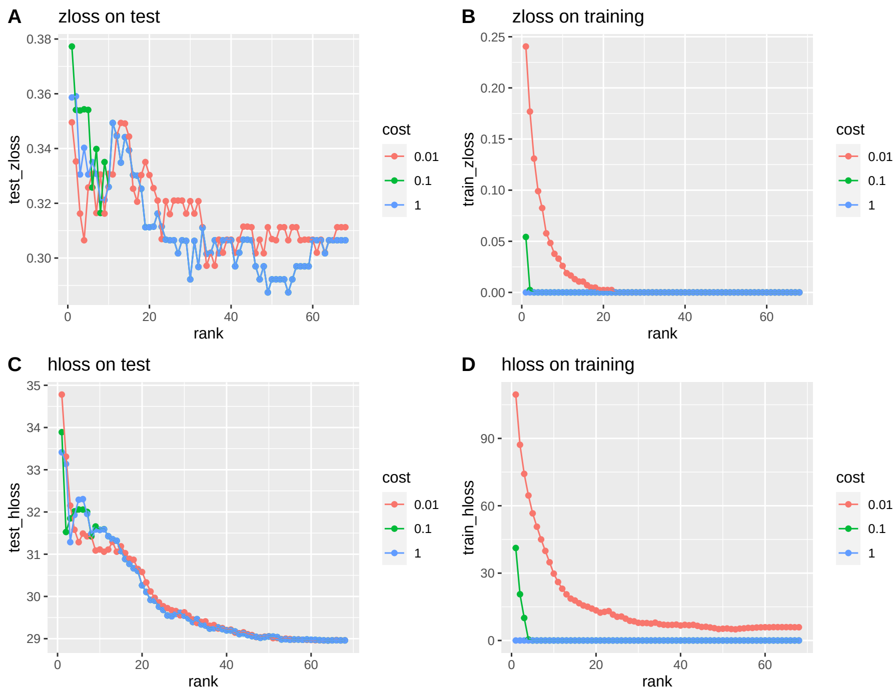

Figure 1: Cross validation results on the first data set. **A**: 0-1 loss on test datasets, **B**: 0-1 loss on training datasets, **C**: hinge loss on test datasets, and **D**: hingeloss on training datasets.

Figure 1 shows that loss values on training datsets converge to 0 as function complexities increase. When we use 0-1 loss, loss values keep decreasing until rank is around 40 and increase again as the rank increases. However, loss values are monotonically decreasing according to the rank size when hinge loss is used.

I checked the standard error of cross validations at each combination of (rank, cost). Considering the magnitude of mean of loss values, I think each cross validation result is quite stable.

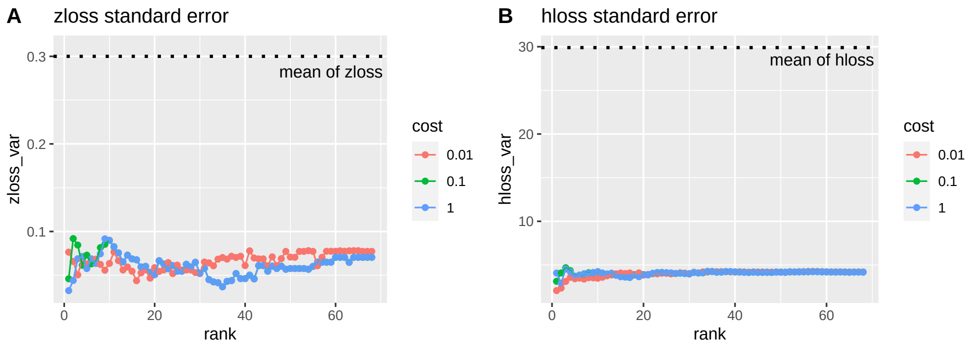

Figure 2: Standard deviation of each cross validation given a combination of (rank, cost). Dotted horizontal lines are mean of all loss values. A is when 0-1 loss is used and B when hinge loss.

2 Cross validation on brain binary IQ

For the data analysis, I labeled $y = 1$ when IQ is greater than 120 and $y = -1$ otherwise based on summary statistics: mean of IQ's is 119.4298 and median is 120. The number of individuals which are labeled as $y = 1$ is 56 and 58 for $y = -1$. I perform cross validation with cost values in $\{0.01, 0.1, 1\}$. I set the 10 multiple initializations to estimate the classifiers. The following is the cross validation result evaluated by 0-1 loss and hinge loss.

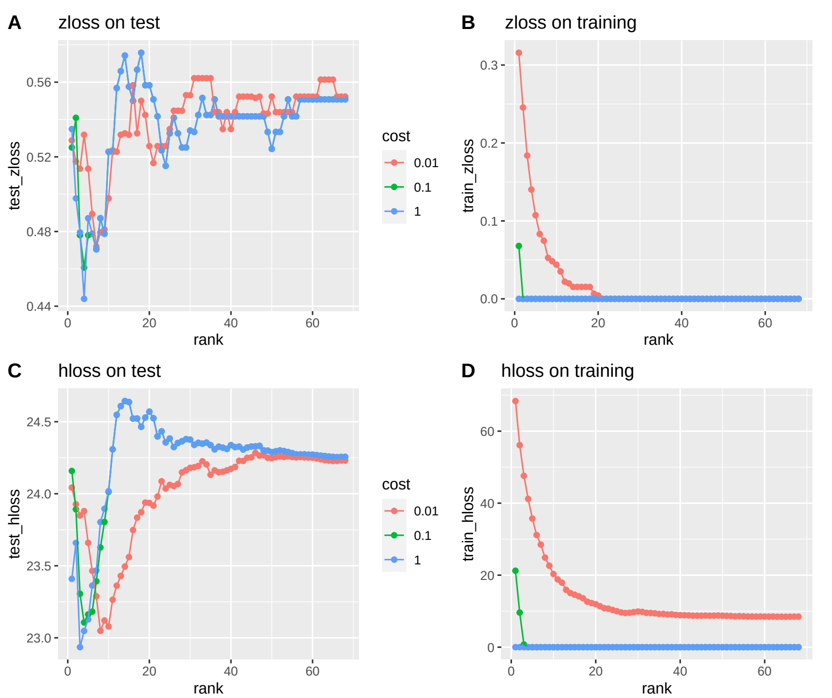

Figure 3: Cross validation results on the second data set. **A**: 0-1 loss on test datasets, **B**: 0-1 loss on training datasets, **C**: hinge loss on test datasets, and **D**: hingeloss on training datasets.

Unlike the first data application, we can see the clear optimal rank in both 0-1 loss case and hinge loss case. The optimal rank is 4 when 0-1 loss is used and 3 when hinge loss is used. In addition, there is no monotonic decreasing phenomenon when hinge loss used, which is observed in the Section

1. Similarly, we can observe that loss values are monotonic decreasing as the rank size increases converging to 0.

Similar to Section 1, I checked the standard error of cross validations at each combination of (rank, cost). Considering the magnitude of mean of loss values, I think each cross validation result is quite stable.

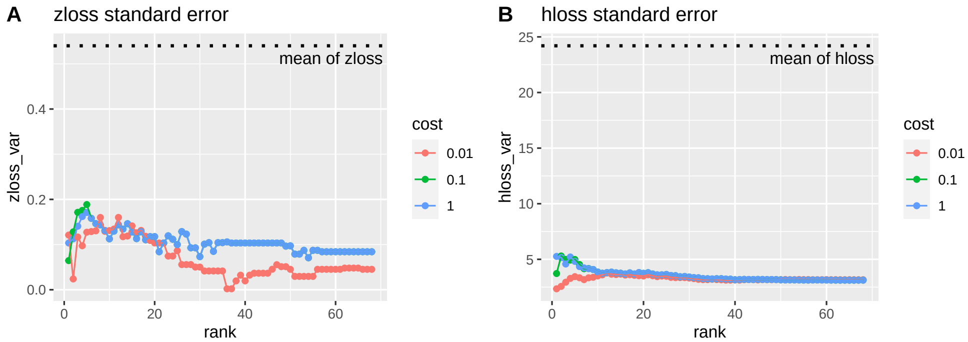

Figure 4: Standard deviation of each cross validation given a combination of (rank, cost). Dotted horizontal lines are mean all loss values. A is when 0-1 loss is used and B when hinge loss.

I perform the brain nodes clustering based on the above cross validation result. I calculate the coefficient $\boldsymbol{B}$ and devide it into $\boldsymbol{B} = \boldsymbol{B}_+ + \boldsymbol{B}_-$ where $\boldsymbol{B}_+$ is positive element of $\boldsymbol{B}$ and $\boldsymbol{B}_-$ negative. I plotted top 5% magnitude edges of $\boldsymbol{B}_+$ and $\boldsymbol{B}_-$ on brain nodes. The following shows the clustering results. It seems that connections in right hemisphere has positive effects on IQ but I cannot see clear tendency. One possible reason is that the brain connections and IQ does not have the direct linear relationship. From Figure 3, we can see that the least error rate of our model is around 0.44 which is quite high considering that the datset has binary responses. Therefore, I think we might need to use nonlinear kernel methods to have better performance.

training set

fixed some reasonable rank—> run a sequence of weighted classification —> probability estimation.

evaluate on test set evaluate goodness-of-fit on training set

KL(p_est, obs)

for a given pi, classification {y_1,…, y_n}

for each index i = 1,…, n

y_{i, weight 1} …., y_{i, weight H}

B: 68-by-68, roughly rank 4

B = P M Q^T, P is 68-by-4, Q is 4-by-68

run a kmeans on P weighted by (M)—> 4 groups of nodes (…)

first component, P[,1] vector 1-by-68,
second component, P[,2] vector 1-by-68,

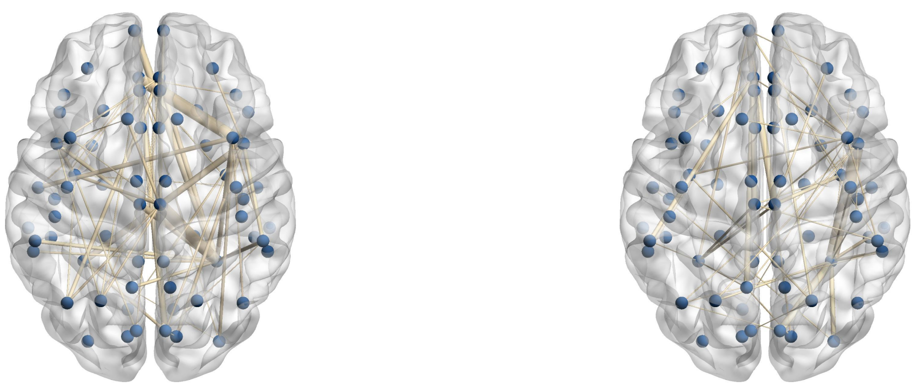

Figure 5: The right figure plots the edges that affect IQ positively, while the left figure shows the edges that affect IQ negatively.