MVE440 Statistical Learning for Big Data
Home Exam

Peizheng Yang
peizheng@student.chalmers.se

June 14, 2019

1 Project 01-ST2, Class-related dimension reduction

1.1 Introduction

In this project, two techniques reduced-rank LDA and regularized-LDA were compared. The following questions were answered:

1. How to select λ in regularized-LDA?
2. What is impact of "the number of selected raw features" on the accuracy of the two methods above?
3. What is the impact of "the number of components selected" in the rank-reduced space on the accuracy of "reduced-rank LDA"?

1.2 Methods

1.2.1 Dataset

The real-world dataset used has 561 dimensions and more than 10,000 samples, with 6 categories of activities (walking, sitting, lying down, walking upstairs, etc.). Each dimension is a statistic value (mean, MSE etc.) of a time series captured by an accelerate meter. Highly correlated dimensions exist in this dataset.

1.2.2 What we did

Firstly, each dimensions of the raw dataset were normalized. Then the normalized data was applied on both methods.

1. **Regularized-LDA**
   Plot the LDA accuracy (y-axis) regarding the number of selected raw features ranging from $1 \sim 561$ (x-axis), with a set of λ values. The λ values selected $(10^{-5}, 0.1, 0.3, 0.5, 1.0, 5.0, 10.0, 100, 10000)$ had a wide range of magnitude, since we want to see how the parameter λ affects the LDA accuracy.
2. **Reduced-rank LDA**
   Plot the LDA accuracy (y-axis) regarding the number of selected raw features ranging from $1 \sim 561$ (x-axis), with a set of "the number of components selected in the rank-reduced space". In our case, there were only 5 components in the rank-reduced space, and we wanted to see how informative these components are in LDA.

1.3 Results

1.3.1 Regularized LDA

See Figure 1.

1. LDA fails when $\lambda = 0$, because the covariance matrix is rank-difficent. The accracy is very high when λ is a very small postive value (10^{-5}), and it goes down when increasing lambda. When $\lambda = 10000$, the accuracy drops to zero for every where on x-axis.
2. More raw features selected yields a higher accuracy.

1.3.2 Reduced-rank LDA

See Figure 2.

1. More raw features selected yields a higher accuracy.
2. More components selected in the "rank-reduced" space yields higher accuracy.

1.3.3 Regarding both

Both methods perform equally well on this particular dataset, both with optimal accuracy 96%.

1.4 Discussion

1.4.1 Take-home messages

LDA is the common part of reduced-rank LDA and regularized-LDA, but the two methods are different in which the data has been used. Reduced-rank LDA uses the "reduced-rank" data, while regularized-LDA uses the raw data. Regarding the two methods,

1. A small value of the regularization paramter λ is just enough to avoid the rank-deficient covariance matrix in LDA. If λ increases to a large magnitude then the accuracy gets worse.

2. Keep as many raw features as you can, and drop a few raw features might reduce the LDA accuracy.
3. Do not drop the components in the rank-reduced space when applying reduced-rank LDA. These components are very informative, and using all of them can maximize the LDA accuracy.

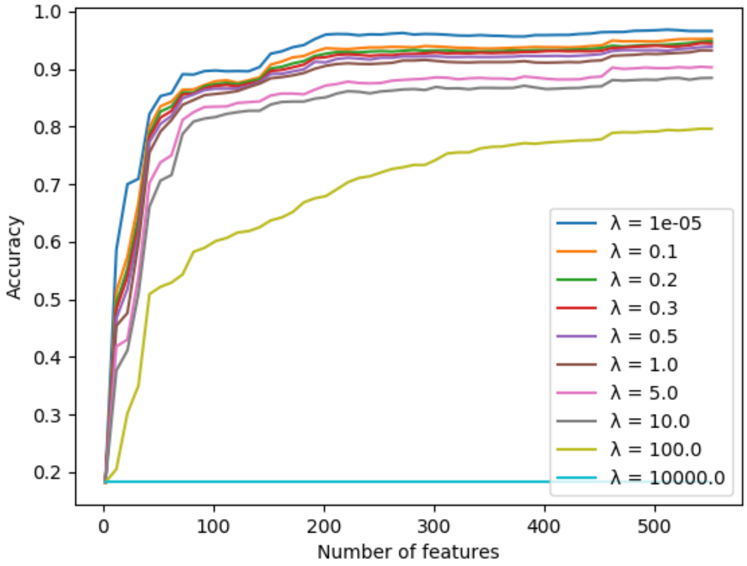


Figure 1: Project 01: Regularized LDA accuracy

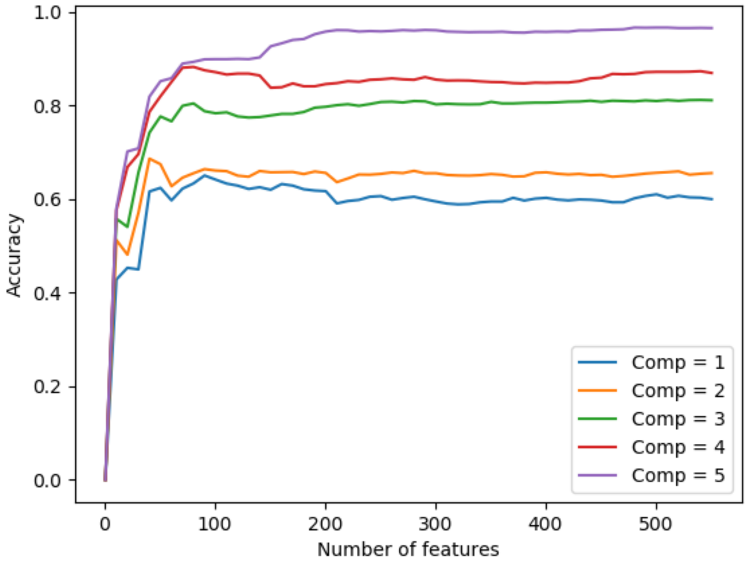


Figure 2: Project 01: Reduced-rank LDA accuracy

2 Project 02-ST2, Impact of cluster separation and relative location

2.1 Introduction

In this project, we explored how the clusters' locations and different clustering methods (k-means and k-medoids) has an impact on the clustering result. The following questions were answered.

1. How does K-means and K-medoids behave differently on the same dataset?
2. How does "scaling" affect the clustering prediction strength?
3. What is the difference between "clustering all the dimensions" and "clustering PC1 and PC2"?

2.2 Methods

2.2.1 Dataset

The real-world food dataset has 49 dimensions, and each dimension is a statistic value such as protein percentage or water etc. We manually selected out 6 very different types of food as 6 clusters, in order to satisfy the condition "clusters only depending on a part of the dimensions in a subspace".

2.2.2 What we did

Since we want to compare k-means and k-medoids in a comprehensive way, we did the following with each method.

1. Compute the prediction strength using both "scaled" and "unscaled" data, to see the impact of feature location. The optimal number of clusters is where the prediction strength reaches local maximum and above some threshold (which is 0.8 in our project).
2. Find the best clustering configuration (normalized or not, the optimal cluster numbers), then cluster the data with this configuration.
3. Cluster PC1 and PC2 with the same configuration, and compare this result with the result on the whole dataset.

2.3 Results

2.3.1 K-means

See Figure 3a, 3b and 4a-4d.

1. The cluster prediction strength plot is different between the scaled data and unscaled data. But the common trend is that increasing cluster number yields lower prediction strength.
2. Clustering result is partially indicative of the true category.
3. Clustering PC1 and PC2 is different from clustering the original dataset.

2.3.2 K-medoids

See Figure 3c, 3d and 5a, 5b.

1. The cluster prediction strength has a local maximum at $k = 4$ on the scaled data.
2. Clustering result is partially indicative of the true category for "scaled data".
3. Clustering PC1 and PC2 is different from clustering the complete dataset for scaled data, while similar for unscaled data.

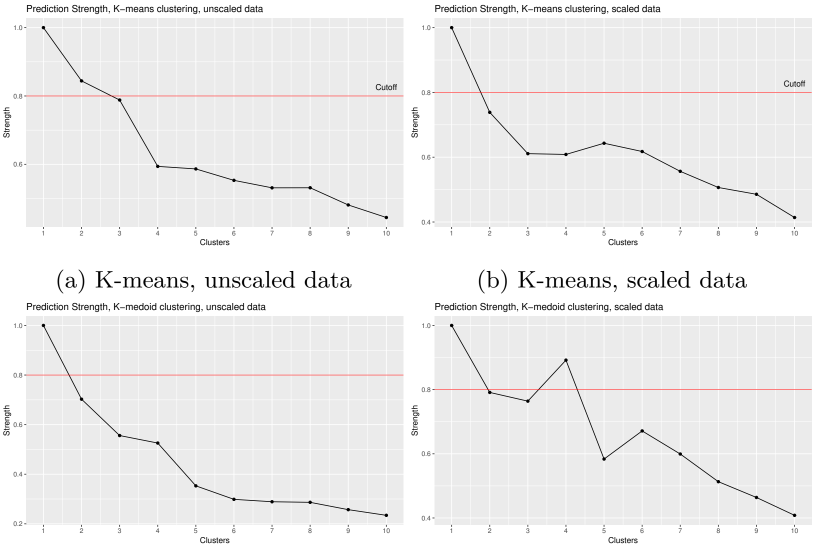
2.4 Discussion

2.4.1 Take-home message

1. K-means and K-medoids have different behaviour no matter in clustering result and prediction strength on the same data (scaling or not is different data).
2. Scaling or not scaling changes the distance between data points in the feature space, thus it has an impact on the clustering result.
3. Clustering on PC1 and PC2 is a different thing from clustering the complete data. No evidence shows that clustering PCA components is better or worse than clustering the whole dataset. Since the PCA drops information, be careful when interpret the clustering result based on PCA components.
4. No universal law that tells which clustering method is best. The best methods depends on the particular dataset to be analysed.

2.4.2 Future work

1. We only tried K-means and k-medoids algorithms to perform the clustering, while many other methods, such as DB-scan, GMM could be explored on this real datset. Since one of other groups found that “GMM work well even without separation when clusters are on different dimensions”, it is expected that GMM could make the clustering results more similar to the true clusters.
2. In our dataset, the true clusters are not balanced. We could explore how well the clustering is by sampling from each true clusters to make “balanced” case.



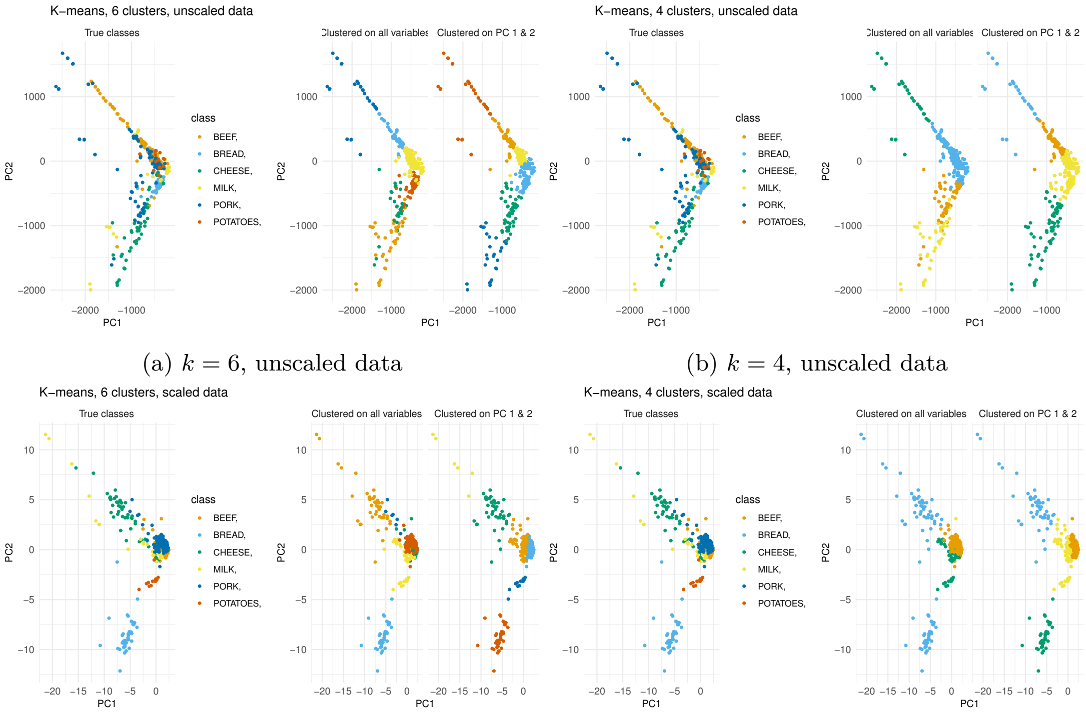
(a) K-means, unscaled data

(b) K-means, scaled data

(c) K-medoid, unscaled data

(d) K-medoid, scaled data

Figure 3: Prediction strength



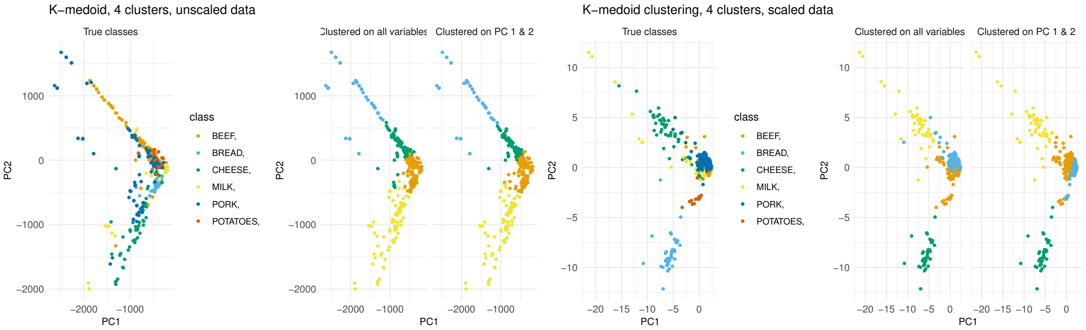
(a) $k = 6$, unscaled data

(b) $k = 4$, unscaled data

(c) $k = 6$, scaled data

(d) $k = 4$, scaled data

Figure 4: K-means clustering



(a) $k = 4$, unscaled data

(b) $k = 4$, scaled data

Figure 5: K-medoid clustering

3 Project 03-ST4, Lasso vs group lasso vs elastic net

3.1 Introduction

In this topic, we explored the effect of lasso, group lasso and elastic net, and also compare these method based on simulated dataset. We want to answer the following questions.

1. Given a sparse true model, how differently does ridge, lasso and elastic-net behave in estimating the true coefficients?
2. How can elastic net uncover a group of highly correlated variables?
3. How does group lasso affect the coefficients of a group of correlated variables?
4. Does label misspecification of group lasso affect the estimated coefficients?
5. In high dimensional settings $n < p$, can elastic-net find the true model?

3.2 Methods

3.2.1 Dataset

To make the true model flexible, we simulated our own dataset in this way. Firstly, we generated many columns with the same values, then defined several groups, and added different level of gaussian noise to different groups to adjust the group correlation strength. Secondly, responses were generated based on a simulated sparse true model.

3.2.2 What we did

To answer question 1, 2, 3, we simulated a dataset with one highly correlated group, with a set of sparse coefficients, and only one non-zero coeffient belongs to the highly correlated group, then

1. Apply ridge regression, lasso, and elastic net and plot the estimated coefficients.
2. Apply elastic-net on this dataset. Adjust the correlation strength of this highly correlated group to see how elastic-net behave.
3. Apply group lasso, and see the behaviour of the estimated coefficients of this highly correlated group.

To answer question 4, we simulated a dataset with two highly correlated groups (A and B). Misslabel the some variables of group A as group B. Then apply group lasso to this dataset.

To answer question 5, we simlated the dataset the same way as for question 1,2,3, but the only difference is that $n < p$ in this case.

3.3 Results

1. Given the sparse true model, ridge regression sets many non-zero coefficients. Lasso gives a sparse solution. And elastic net is in the middle of ridge regression and lasso. The sparsity of the estimated coefficients by elastic net depends on α.
2. Elastic net can set coefficients of a group of colinear variables to the similar non-zero value. While lasso tends to set only one non-zero coefficient, but this coefficient is not always where the true coefficient lies.
3. Group lasso activates the whole group where true non-zero coefficient exists given the correct group information. See Figure 6.
4. In some simulations, the mislabelled coefficients (group A mislabed as group B) are similar to the wrong group (group B), see Figure 7a. However, sometimes, the mislabelled coeffients are neither similar with group A nor group B, see Figure 7b.
5. The ability for Elastic net to find the true coefficients becomes weaker in $n < p$ setting. The estimated coefficients are more biased than large n setting.

3.4 Discussion

3.4.1 Take home messages

1. Lasso and ridge regression are in the two extremes. Elastic is the combination of two.
2. Elastic-net can uncover a group of colinear variables, while lasso only picks one non-zero coefficient from the group.
3. Given the correct group labelling, group lasso behaves similar to elastic-net in assign the whole group to similar values.

4. The mislabelled coefficients should be treated as another group, whose coefficients could be either different from group A or B.
5. In $n < p$ setting, elastic net is not able to find the true model. More data is required to estimated the correct coefficients.

3.4.2 Corrections on previous project slides

In the part ”change in group correlation”, the estimated coefficients (black) seemed to be lower than the true coefficients (red). This was caused by the samll range of y-axis. If we set y-lim up to 10, then all the estimated coefficients can be displayed. See Figure 8a,8b.

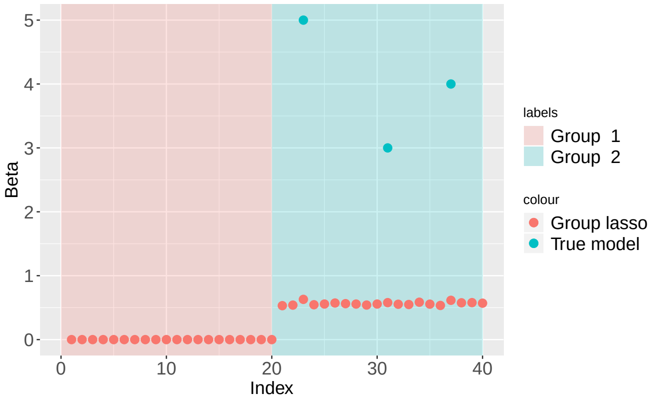
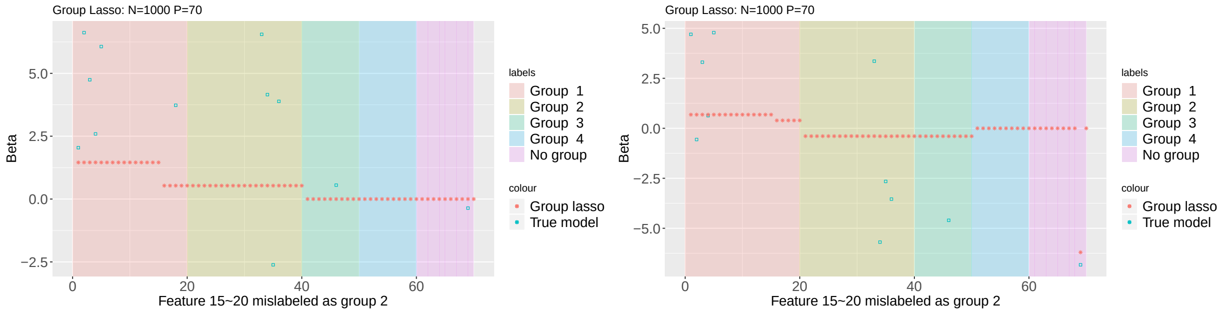
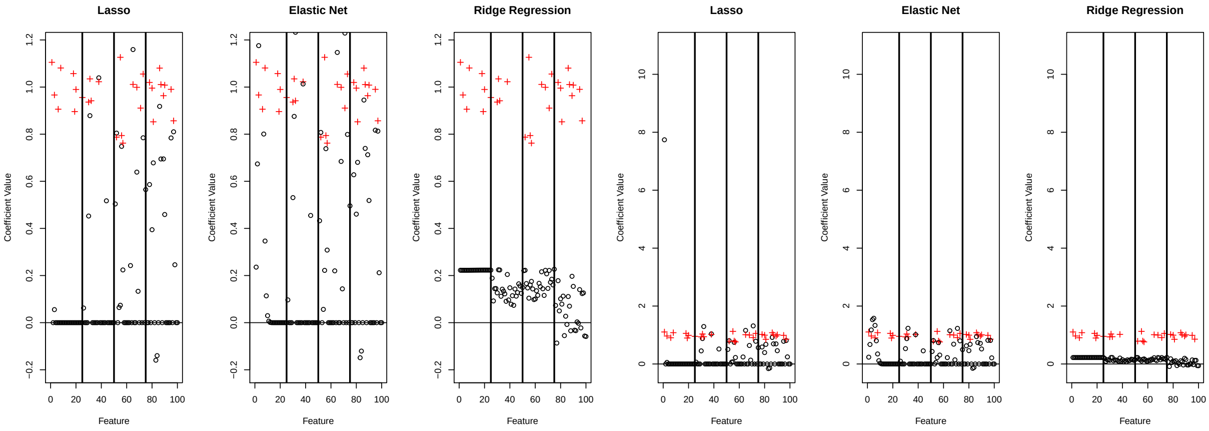


Figure 6: Group Lasso



(a) Mislabelled coefficents similar to the wrong group (b) Mislabelled coefficents different to the wrong group

Figure 7: Group lasso with mislabelling



(a) Before correction (b) After correction

Figure 8: Correction on previous slides

4 Project 04-ST2, Visualisation via dimension reduction

In this project, two methods of dimension reduction (tSNE, PCA) has been explored on simulated datasets. The following question was answered.

1. How does PCA and tSNE behave in cluster visualization? What is the difference between them?

4.1 Methods

4.1.1 Dataset

Two datasets were used to compare the behaviour of PCA and tSNE on different data. The first dataset is a simulated one with $3 \sim 12$ well-separated Gaussian-distributed clusters. This dataset has redundant features (which are irrelevant to any cluster). "Balanced or imbalanced data" and "with or without outliers" settings were both looked into. The second dataset is a 3D fish dataset.

4.1.2 What we did

1. Plot the PCA of 3 clusters and 12 clusters, and compare the two plots. Clusters are balanced and no outliers.
2. Plot the tSNE of from 3 clusters to 12 clusters, and compare the two plots. Clusters are balanced and no outliers.
3. Plot the PCA of 4 clusters with/without outliers, and compare.
4. Plot the tSNE of 12 clusters with/without outliers, and compare.
5. Plot the tSNE and PCA of the fish dataset, and compare.

4.2 Results

4.2.1 The first "Gaussian" dataset

1. If clusters are balanced without outliers, PCA can perfectly visualize 3 clusters. However, the visualization of 12 clusters are not indicative the true clusters.
2. tSNE can handle the visualization of $3 \sim 12$ clusters, both with and without outliers. However, if we have unbalanced clusters, tSNE has a possibility of not well separating the true clusters if the clusters are close with each other.
3. PCA is sensitive the outliers. If outliers are far away from any cluster, the visualization of PCA becomes less indicative of the true clusters.
4. tSNE can handle the outliers better than PCA. The visualization can still disclose the true clusters.

4.2.2 The second fish dataset

PCA almost keep the original shape of the fish. However, the "fish shape" is broken by tSNE. Only some local patterns of the fish is still visible.

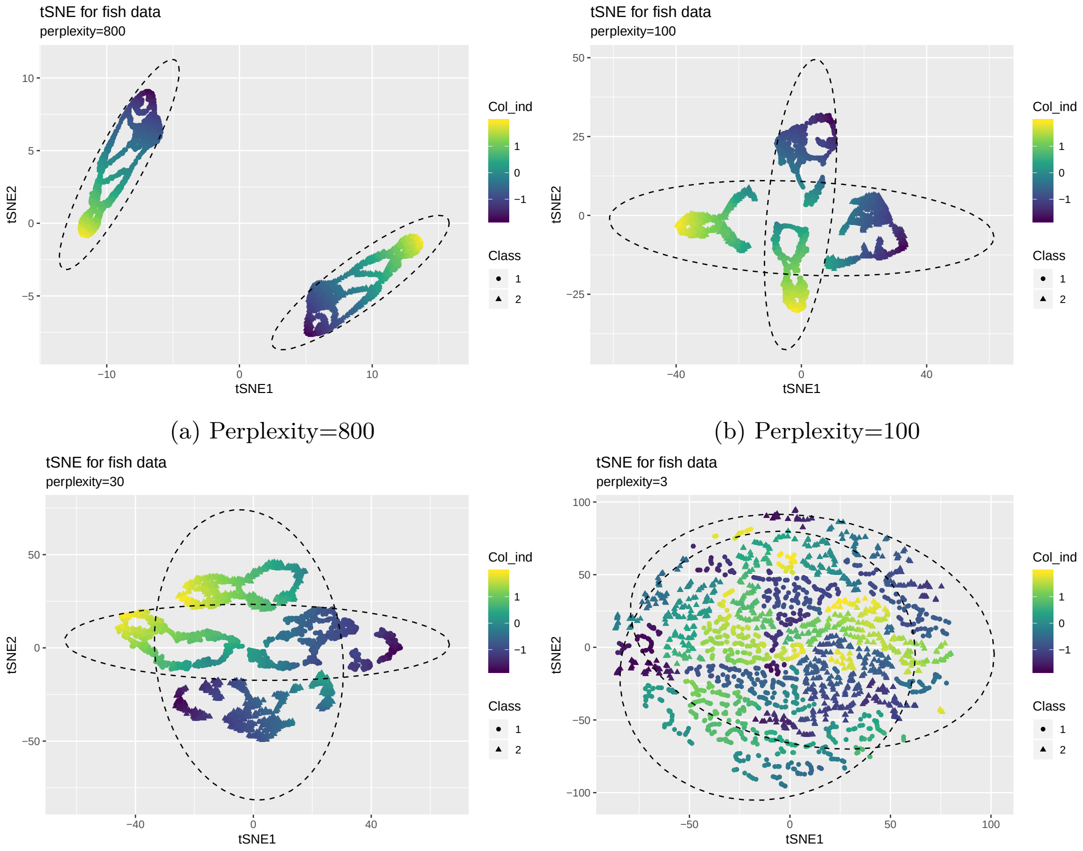
4.3 Discussion

4.3.1 Take-home messages from my group and other groups

1. PCA is a global dimension reduction method. It is suitable for visualizing only a few number of clusters (less than 4). PCA is senstive to outliers, and it can preserve the shape.
2. tSNE is a local dimension reduction method. It can handle more clusters than PCA. It is robust against outliers. However, tSNE is not honest to the original shape of the data. Also, the cluster distance in tSNE visualization cannot represent the distance in the original data.
3. Applying tSNE on a real world data does not always seperate the true clusters. Sometimes, fake patterns can be created by tSNE. We should be very careful when interpreting tSNE visualization.

4.3.2 Future work

1. Perplexity of tSNE is an important hyper-parameter. The larger perplexity means more global scope. Thus a well-preserved fish is expected with high perplexity. As indicated in Figure 9a $\sim$ 9d, higher perplexity makes tSNE capture more global pattern, while lower perplexity does the inverse.



(a) Perplexity=800

(b) Perplexity=100

(c) Perplexity=30

(d) Perplexity=3

Figure 9: tSNE on fish dataset with different perplexity

5 Task 1: Regression in big-n

5.1 Introduction

In this task, regression models were built for the four datasets respectively. Two main approaches are 1)variable selection & importance determination and 2)model fitting. The details of the approaches may differ from one dataset to another due to the difference in n and p, The key questions are following:

1. What methods can be used to select the important variables? How to determine the importance of each variables?
2. Are these variable selection methods applicable to big n and/or p case?
3. What is difference in small n and big n in computing the linear model coefficients?

5.2 Methods

The methods of variable selection are 1)elastic net, 2)stepwise feature selection, 3)variable importance by random forest and 4) relative importance by LMG. Among them only elastic net cannot gives the importance of the variable.

1. Lasso
   Lasso can do the variable selection by using L1 norm as penalization.
2. Stepwise feature selection
   In each step, a variable is either added to or removed from the set of selected variables based on a certain criterion. The variables are selected according to Akaike information criterion (AIC). The variable importance are given by the P-value of T-test. Variables with smaller P-value are more significant.
3. Random forest
   Random forest can give the importance measured by mean decrease accuracy (MDA). The prediction error (MSE for regression) of the out-of-bag samples is recorded, then the same is done after permuting each predictor variable. Larger MDA means higher variable importance.
4. LMG
   LMG is measured by R-squared value. There are $m!$ ordered sequences of the variables for a model with variables $x_1, ..., x_m$. When The variables are added one by one to the model according to a particular ordered sequence, and the proportion of the remaining variance accounted by each variable is logged [1]. The computation time roughly doubles for lmg when adding one more variable, but is virtually not affected by the change of sample size [2].

The approahes on the four datasets are following.

1. Standardize the dataset by colomns.
2. Try the four methods to do the variable selection and determine variable importance.
3. Fit the linear model by OLS using the selected variables.

Note that due to the n and p differs among the four datasets, some adaptions in variable selection methods may occur.

5.3 Results

5.3.1 Dataset A1

Dataset A1 has small n and small p. All of these variable selection methods can be performed on this dataset without computational power barrier.

Variable selection and importance

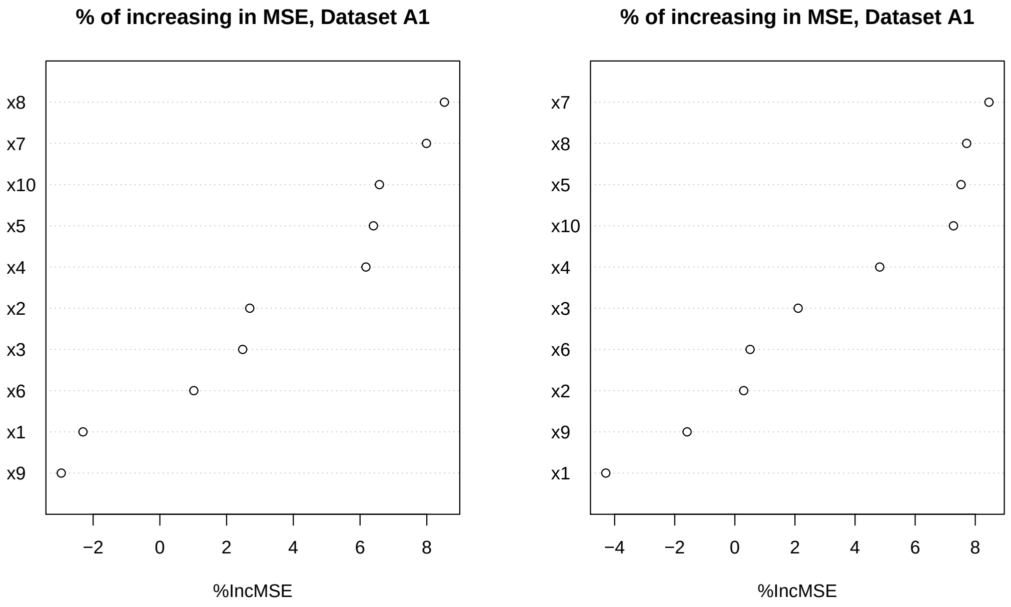
1. Lasso
   The selected features are x4, x5, x7, x8, x10.
2. Stepwise feature selection
   The model suggested by AIC is $y \sim x_2 + x_3 + x_4 + x_5 + x_7 + x_8 + x_{10}$. The P-value and significance of these selected variables are in Table 1. The variables with small P-values are not consistent with the varibles selected by elastic net.

	Estimate	Std. Error	t value	Pr(>|t|)	Signif
(Intercept)	0.0680	0.1643	0.414	0.67990	
x2	0.2536	0.1765	1.437	0.15421	
x3	0.3547	0.1881	1.885	0.06255	.
x4	-0.5971	0.1682	-3.550	0.00061	***
x5	-0.3702	0.1898	-1.950	0.05417	.
x7	-0.5294	0.1856	-2.852	0.00536	**
x8	-0.4857	0.1881	-2.582	0.01139	*
x10	0.5496	0.1746	3.148	0.00222	**

Table 1: Stepwise feature selection on A1

3. Random forest feature selection
   The number of trees is 500, with iid sampling size 50%. Larger value means higher variable importance. The variable importance is measured by the MDA. The randomness of this method is shown in Figure 10a, 10b.

   Even the order of importance is not fixed with multiple runnings, the first 5 important variables are always x4, x5, x7, x8, x10, and the least important 2 variables are always x1 and x9.



(a) Example a

(b) Example b

Figure 10: %IncMSE (MDA) on dataset A1. The left and right figure has different results.

4. Feature selection by LMG
   The relative importance by "LMG" is in Table 2. x4, x5, x7, x8, x10 have much more importance than the other variables.

Variables	lmg
x4	0.246889531
x7	0.216994106
x8	0.164310407
x5	0.146056651
x10	0.141962881
x3	0.028665306
x2	0.027122487
x6	0.011525049
x9	0.011113038
x1	0.005360544

Table 2: LMG on dataset A1

Final model

On dataset A1, Lasso, random forest and LMG give consistent variable selection. The final model of dataset A1 consists variable x4, x5, x7, x8, x10. The model fitted by Ordinary Least Squares regression (OLS) is in Table 3.

(Intercept)	x4	x5	x7	x8	x10
0.06799995	-0.61824424	-0.34718506	-0.43802018	-0.34764378	0.43998293

Table 3: The final model of A1

5.3.2 Dataset A2

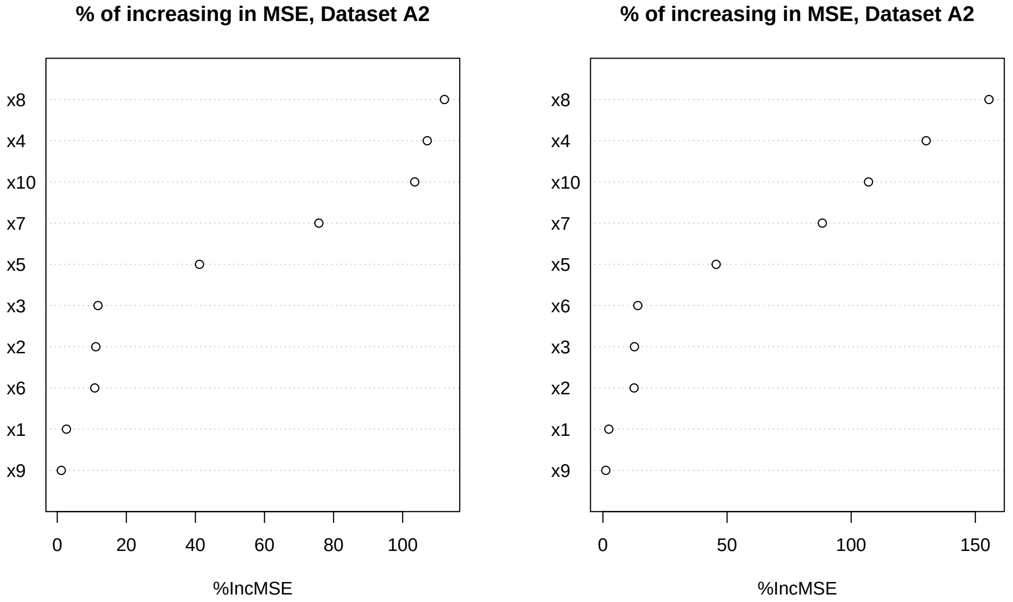
Dataset A2 is a case with large n and small p.

1. Lasso
   The selected features are x4, x5, x7, x8, x10, the same as A1.
2. Stepwise feature selection
   The suggested model base on AIC is the same as A1 ($y \sim x_2 + x_3 + x_4 + x_5 + x_7 + x_8 + x_{10}$). The P-values of x4, x5, x7, x8, x10 are extremely small compared to x2 and x6, from Table 4. The variables with small P-values are consistent with the variables selected by Lasso.

	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	-0.003923	0.004755	-0.825	0.4094	
x2	-0.011964	0.005372	-2.227	0.0260	*
x4	-0.447022	0.004756	-94.000	<2e-16	***
x5	-0.246616	0.005081	-48.535	<2e-16	***
x6	-0.010832	0.005387	-2.011	0.0444	*
x7	-0.362349	0.005091	-71.169	<2e-16	***
x8	-0.621172	0.005085	-122.152	<2e-16	***
x10	0.465964	0.005384	86.542	<2e-16	***

Table 4: Stepwise feature selection on dataset A1

3. Random forest feature selection
   To save computation time, the parameter of the trees are set as `ntree=100` and `sampsize = 0.2*nrow(X)`. The variable importance is in Figure 11a 11b. By running multiple times, it is found that the ranking of the five most important variables (x8, x4, x10, x7, x5) is more stable than in dataset A1.



(a) Example a

(b) Example b

Figure 11: %IncMSE on dataset A2. The left and right figure has different results.

4. Feature selection by LMG
   The importance ranking is different from A1, but the five most important variables are still x4, x5, x7, x8, x10, as shown in Table 5.

	lmg
x8	0.354764642
x7	0.170773991
x10	0.157174373
x4	0.155083515
x5	0.108628455
x3	0.032370869
x6	0.009952969
x2	0.009285615
x1	0.001047632
x9	0.000917941

Table 5: LMG on dataset A2

Final model
In dataset B2, four methods above give consistent variable selection. The final model of dataset A2 consists variable x4, x5, x7, x8, x10, the same as A1. The model fitted by (OLS) is in Table 6.

(Intercept)	x4	x5	x7	x8	x10
-0.003922622	-0.447019869	-0.246582337	-0.362327618	-0.621171032	0.472377817

Table 6: The final model of A2

5.3.3 Dataset B1

Dataset B2 has big p and $n \approx p$. The challenge in this dataset lies in big p. As indicated by the results from A1 and A2, the variables primarily selected by Lasso are of importance. Thus, one possible way to make it computationally available is to determine the variable importance only on the importance selected by Lasso.

1. Lasso
   The features selected by Lasso are x13, x21, x45, x78, x168, x189, x253, x284, x333, x353, x387, x535, x600, x624, x682, x733, x779.
2. Stepwise feature selection (among variables selected by Lasso)
   The results are displayed in Table 7.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-0.01752	0.04726	-0.371	0.710937
x13	0.14808	0.04825	3.069	0.002208 **
x21	0.14224	0.04830	2.945	0.003308 **
x45	-0.31084	0.04764	-6.525	1.09e-10 ***
x78	0.15378	0.04748	3.239	0.001242 **
x168	0.14739	0.04774	3.088	0.002075 **
x189	0.33896	0.04863	6.971	5.78e-12 ***
x253	-0.16531	0.04766	-3.468	0.000546 ***
x284	0.28980	0.04773	6.072	1.80e-09 ***
x333	-0.15092	0.04760	-3.170	0.001569 **
x353	0.36746	0.04918	7.473	1.74e-13 ***
x387	-0.18681	0.04754	-3.929	9.12e-05 ***
x535	-0.16040	0.04751	-3.376	0.000765 ***
x600	-0.38262	0.04788	-7.992	3.72e-15 ***
x624	0.12931	0.04752	2.721	0.006617 **
x682	0.35245	0.04762	7.401	2.89e-13 ***
x733	-0.18097	0.04759	-3.803	0.000152 ***
x779	0.10732	0.04910	2.186	0.029065 *

Table 7: Stepwise feature selection on the features selected by Lasso of B1

3. Random forest feature selection (among variables selected by Lasso)
   The randomness can be annoying, however, it can be mitigated by averaging the importance of many random forests. `ntree=100` and `sampsize = 0.8*nrow(X)` are set in each forest. Table 8 (left) is the importance computed by averaging 20 random forests, and displayed by the descending order of %IncMSE. It is observed that from x353 down to x624 has a fixed ranking order with multiple runs.
4. Variable importance by LMG (among variables selected by Lasso)
   Table 8 (right) shows the relative importance given by LMG.

	%IncMSE
x353	0.200129985
x682	0.150596876
x600	0.142707432
x189	0.123722495
x45	0.103350672
x284	0.100562078
x387	0.060955618
x168	0.038198327
x13	0.028239253
x333	0.024050523
x78	0.023172824
x624	0.020487822
x21	0.012900036
x779	0.010838895
x253	0.010286235
x733	0.009210591
x535	0.000641365

	lmg
x600	0.13431753
x353	0.13304239
x189	0.12191592
x682	0.12012821
x45	0.09727861
x284	0.08838843
x387	0.03879753
x13	0.03574795
x333	0.0331039
x779	0.03130815
x21	0.02588753
x253	0.02584621
x733	0.02438033
x168	0.02379623
x78	0.02285483
x535	0.02216417
x624	0.02104207

Table 8: %IncMSE (left) and LMG (right) on dataset B1

Final model
The final model of dataset B1 consists variable selected by lasso, and they are x13, x21, x45, x78, x168, x189, x253, x284, x333, x353, x387, x535, x600, x624, x682, x733, x779. The coefficients are in Table 7, colomn "Estimate".

5.3.4 Dataset B2

Data B2 has both large p and n.

1. Lasso
   The features selected by Lasso are x13, x21, x45, x78, x139, x189, x245, x253, x284, x353, x387, x503, x535, x582, x600, x643, x682, x699, x733, x737.
2. Stepwise feature selection (among variables selected by Lasso)
   Similar with B1, only the features selected by Lasso are used. The results are displayed in Table 9.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-0.008868	0.004734	-1.873	0.061 .
x13	0.112025	0.004853	23.085	<2e-16 ***
x21	0.116064	0.004854	23.911	<2e-16 ***
x45	-0.243612	0.004734	-51.462	<2e-16 ***
x78	0.116787	0.004734	24.669	<2e-16 ***
x139	0.136940	0.004734	28.927	<2e-16 ***
x189	0.374089	0.004856	77.030	<2e-16 ***
x245	-0.133085	0.004734	-28.112	<2e-16 ***
x253	-0.145752	0.004737	-30.771	<2e-16 ***
x284	0.283667	0.004734	59.922	<2e-16 ***
x353	0.328178	0.004734	69.321	<2e-16 ***
x387	-0.226652	0.004734	-47.874	<2e-16 ***
x503	-0.090816	0.004808	-18.888	<2e-16 ***
x535	-0.177395	0.004734	-37.471	<2e-16 ***
x582	-0.175590	0.004734	-37.090	<2e-16 ***
x600	-0.404917	0.004737	-85.487	<2e-16 ***
x643	-0.049136	0.004808	-10.220	<2e-16 ***
x682	0.431548	0.004734	91.158	<2e-16 ***
x699	-0.081796	0.004734	-17.279	<2e-16 ***
x733	-0.151889	0.004734	-32.084	<2e-16 ***
x737	-0.138133	0.004734	-29.178	<2e-16 ***

Table 9: Stepwise feature selection on the features selected by Lasso of B2

3. Random forest (among variables selected by Lasso)
   Due to the big n, the similar method as Divide and conquer is applied. 200 subsets each of size $n_s = 10000$ were randomly sampled, and each of them is used to build a random forest. See Figure 12. The average importance of 200 random forests is displayed as Table 10 (left). Parallel computing was also used to accelerate the computation.

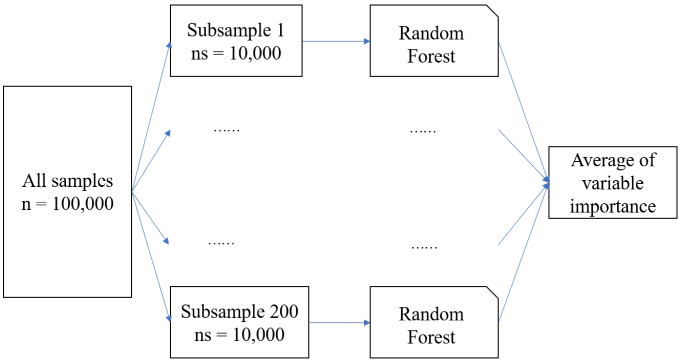


Figure 12: 200 subsets each of size $n_s = 10000$ were randomly sampled, and each of them is used to build a random forest. The average variable importance is computed from all these trees.

4. LMG (among variables selected by Lasso)
   Again, only the feature selected by Lasso are used. Results are displayed in Table 10 (right). The ranking order is similar with Table 10 (left).

	%IncMSE
x682	0.295107145
x600	0.245777452
x189	0.243779696
x353	0.148776675
x284	0.103472815
x45	0.070212181
x387	0.060996547
x535	0.031507245
x582	0.029802759
x733	0.021095214
x13	0.020993576
x21	0.020951176
x253	0.017766077
x139	0.016986162
x737	0.016898469
x245	0.016147966
x78	0.011647619
x503	0.008152217
x699	0.006042787
x643	0.003176385

	lmg
x682	0.179251809
x600	0.15491771
x189	0.144521137
x353	0.102852761
x284	0.076367574
x45	0.057995085
x387	0.051283004
x535	0.030890968
x582	0.029552355
x21	0.024134864
x13	0.023620353
x733	0.021810528
x253	0.018337402
x139	0.018131641
x737	0.018081165
x245	0.017196594
x78	0.013310964
x503	0.008052555
x699	0.006772682
x643	0.002918849

Table 10: %IncMSE (left) and lmg (right) on dataset B2

Final model
The final model of dataset B2 consists variable selected by lasso, and they are x13, x21, x45, x78, x139, x189, x245, x253, x284, x353, x387, x503, x535, x582, x600, x643, x682, x699, x733, x737. The coefficients are in Table 9, colomn "Estimate".

5.4 Discussion

The common features of all of the four datasets is $n > p$. The difference is that the ratio n/p is different, and they are 10, 10^4, 1.25, 125 respectively. The results for feature selection is related with n/p. A larger ratio n/p yields more consistent variable selection results and variable importance determined by different methods. For example, the importance ranking by random forest in A2 is more stable than A1 (This also holds for B2 and B1), and the importance indicated by P-values of the A2 is more consistent with the importance measured by other methods.

Lasso is the best way to give a rough selection of variables among the four methods in the sense of computation availability. It provides a preliminary subset of important variables for the other three methods to determine the importance in dataset B1 and B2. Without the rough selection, the other 3 methods are difficult to implement due to large p. Lasso gives the same selection in A1 and A2 where $p = 10$. However, the set of variables selected in A1 by Lasso is different from $B2$. The reason for the difference could be that $n = 1000$ is insufficiently large compared to $p = 800$ in dataset B1. To prove this thought, a little experiment has been done. Several random subsets of B2 with two different size $n_{10\%} = 10000$ ($n/p = 12.5$) and $n_{1\%} = 1000$ ($n/p = 1.25$) were used to select features by lasso, the result displayed in Table 11 and 12 is that the feature selection by lasso based on a subset (where n/p is lower) is not stable. This could be an indication that the features may not be correctly selected in small n/p case such as dataset B1, but for the other datasets A1, A2 and B2, the variable selection by Lasso is more stable and trustworthy.

13 21 45 78 139 189 245 253 284 353 387 503 535 582 600 643 682 699 733 737
13 21 45 78 139 189 245 253 284 353 387 392 503 535 582 600 643 682 699 733 737

Table 11: Two trials of features Selected by lasso, based on 10% samples.

20 21 34 39 45 78 85 113 116 139 189 245 253 265 266 284 301 353 359 365 387 421 447 448 503 535 582 600 682 699
13 45 189 245 253 284 353 385 387 394 521 535 582 600 682 737

Table 12: Two trials of features Selected by lasso, based on 1% samples.

The other three variable selection methods works well in small p when n/p is sufficiently large. LMG is a non-random method. The flaw of this method is that it is unfriendly with large p, such as in B1 and B2. Thus, LMG on only the most important variables selected by Lasso is a good solution, since the importance of the non-important variables is not meaningful. The randomness of random forest is problem in determining the importance, however, it can be mitigated by averaging many random forests. The stepwise selection can not be directly implements in large p dataset. So that, for dataset with small p, all of these methods works well. For dataset with large p, it is advised to take pre-selection by Lasso first and then only measure the importance on the pre-selected variables.

If the dataset has extremely large n, computational barrier needs to be considered. Approaches such as Bootstrap of little bag (BLB) can be used in fitting the model, and the pattern of divide and conqueror can also be used in random forest to determine variable importance (in dataset B2). In R language, parallel computation can be also used for acceleration purpose.

6 High-dimensional clustering

6.1 Introduction

The aim of the task is to cluster the high-dimensional data with a proper method and measure the importance of each variables. The key problem is that, how to find the important variables in clustering for this large p dataset.

6.2 Methods

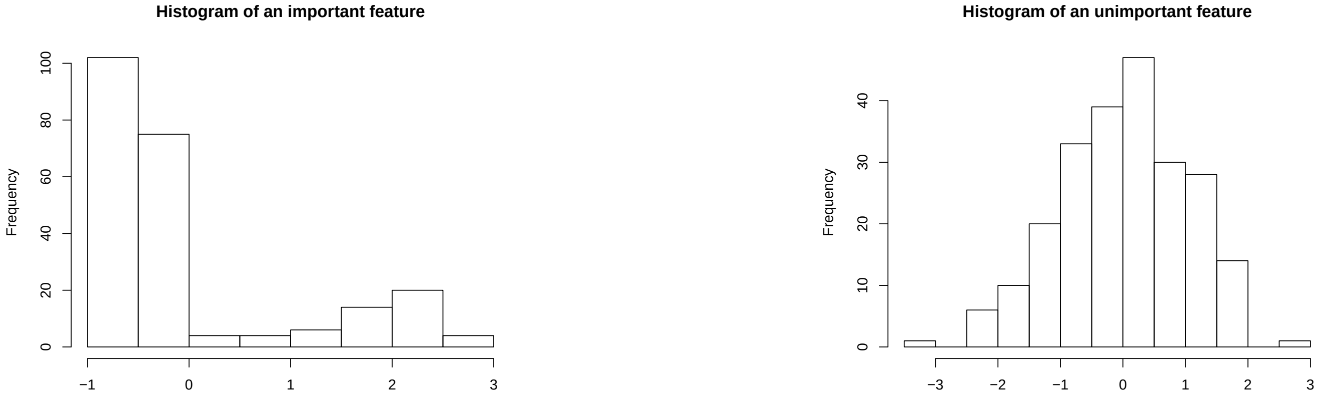
Since the traditional clustering methods that work well on $n > p$ dataset fail in $n < p$ setting, some adaptions must be introduced. The adaption could be either using high-dimensional clustering methods, or preprocessing the data by feature selection or transformation.

The procedure is as following:

1. Scale the whole dataset.
2. Do the first-round variable selection by Hartigan's dip-test [3], and keep the features with p-value less than 0.01.
3. Determine the optimal number of clusters by average Silhouette width based on the features from the first-round selection. Use GMM to cluster these features.
4. Do the second-round selection within the features selected by the first-round according to the method proposed in [4].
5. Use GMM to cluster the "most important features" from the second-round selection.

Several considerations about the procedure:

1. The first-round selection
   The aim of the first-round selection is to drop away the redundant features that may exisit in a high-dimensional dataset. If a feature is important for clustering, multi-model distribution is likely to exist in this feature. An example of multi-model and uni-model is shown in Figure 13a and 13b. Hartigan’ s dip-test is a method to find features with multi-modal distribution (also called "important features"), which answer the "which variables explain the final clustering best".
2. The second-round selection
   The aim of the second-round selection is to find the most clustering-relevant features within the features selected in the first round. The variable selection method for GMM clustering proposed in [4] can determine the clustering-relevant variables and the optimal number of clusters. However, the computational issue is that this method cannot directly handle the total 598 features. Thus, the first-round selection is necessary.
3. Traditional methods can work with the "important features". Among these methods, the model-based GMM is generally more flexible and more powerful than K-means in dealing with the ellipsoid shape of clusters.



(a) An feature (V321) with multi-model distribution, with P-value of Dip test 0.0015

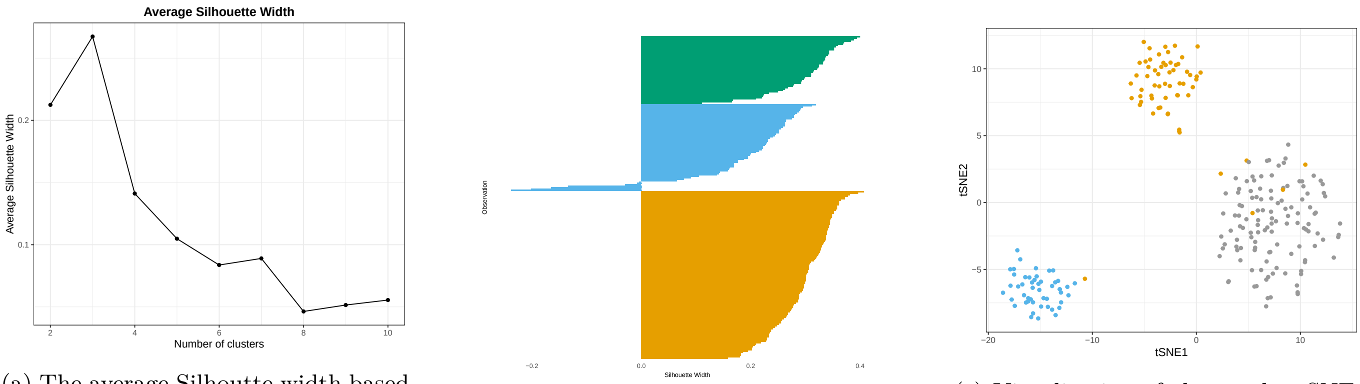
(b) An feature (V1) with uni-model distribution, with P-value of Dip test 0.9932

Figure 13: Histogram of features

6.3 Results

6.3.1 First-round selection

There are 17 features selected in the first-round by Hartigan’ s dip-test: V54, V123, V227, V275, V307, V321, V323, V373, V403, V404, V411, V424, V444, V519, V523, V547 and V584. The plot of “average Silhouette width” vs “number of clusters” is in Figure 14a. The optimal cluster number is 3 with average Silhouette width 0.267. The Silhouette width of 3 clusters is in Figure 14b. The visualization of clustering results is plotted by tSNE in Figure 14c (each color represents a cluster assignment by GMM). The cluster assignment is saved in `cluster_assignment_firstRound.csv`.



(a) The average Silhoutte width based on the 17 features.The optimal cluster number is 3 with average Silhouette width 0.267.

(b) Silhoutte width of each sample. The maximum is around 0.4.

(c) Visualization of clusters by tSNE. Colors correspond with clustering results.

Figure 14: first-round variable selection

To justify the importance of these 17 selected features, control group experiment has been done. The average of Silhoutte width of 10 bootstrap samples (each bootstrap sample has 17 features) from the total 598 features is in Figure 15. The average Silhoutte width in Figure 15 is lower than Figure 14a, which indicates the first-round selection does include the important variables for clustering.

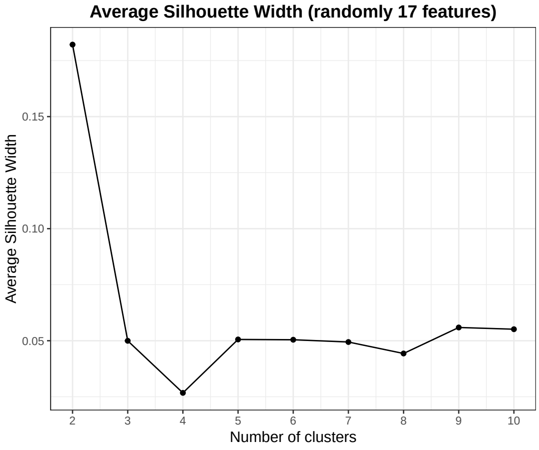
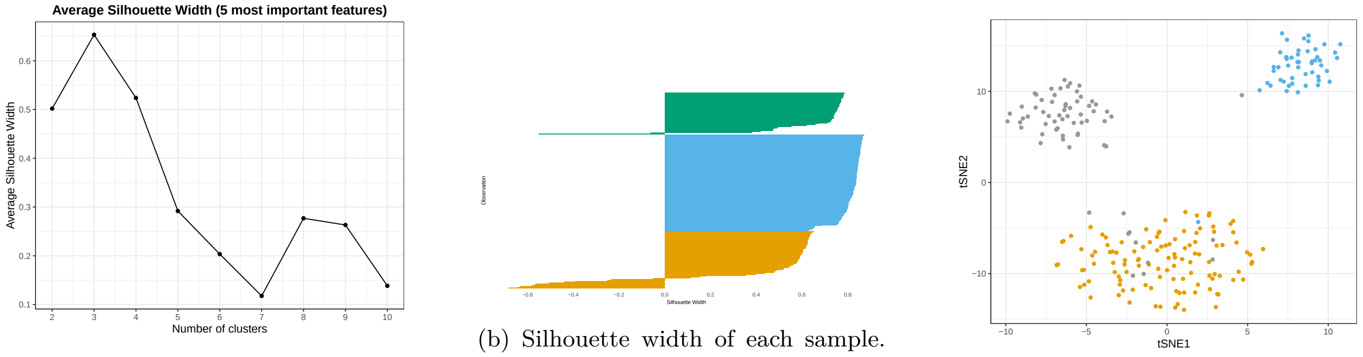


Figure 15: Silhoutte width based on 10 bootstrap features from the total 598 features in the first selection. Each bootstrap sample contains 17 features.

6.3.2 Second-round selection

The second-round selection selects 5 variables: V321, V411, V519, V523, V547. Similar as the first-round selection, the average Silhoutte width, Silhoutte width of each example and visualization of clusters are in Figure 16a, 16b, 16c. The average Silhoutte width on the five features is larger than the first-round. The clustering result (stored in `cluster_assignment_secondRound.csv`) is exactly the same as the first-round selection.

Again, to justify that the 5 variables from the second-round are of importance, the average Silhoutte width based on 10 bootstrap feature samples from the 17 features in the first selection is plotted in Figure 17. The bootstrapped average Silhoutte Width in Figure 17 is much lower than Figure 16a. Thus, it can be concluded that variable V321, V411, V519, V523, V547 are the most important variables that determines the clustering result.



(a) The average Silhoutte width based on the 5 features.The optimal cluster number is 3 with average Silhouette width 0.653.

(b) Silhouette width of each sample. The maximum is around 0.8.

(c) Visualization of clusters by tSNE. Colors correspond with the clustering results.

Figure 16: first-round variable selection

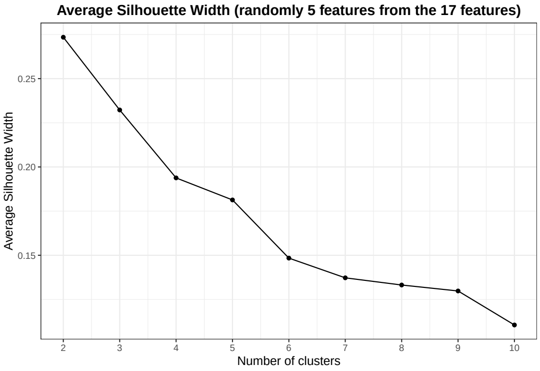


Figure 17: Silhoutte width based on 10 bootstrap features from the 17 features in the first selection. Each bootstrap sample contains 5 features.

6.4 Discussion

Traditional clustering methods usually fail on a large p dataset, so it is important to find the most relevant variables for clustering. Variable selection can be progressively done by two steps, not necessarily within one step. For example, in this task, the first-round variable selection by Hartigan' s dip-test screens out the most of the irrelevant variables to the clustering by testing the unimodality of each variable. The variables selected by the first round can be further selected by a second round with the method proposed in [4] to get the most important variables.

The importance of the selected variables can be justified by the control experiment of comparing the average Silhouette width of "bootstrapped" features and the selected features.

References

[1] William Kruskal. Relative importance by averaging over orderings. *The American Statistician*, 41(1):6–10, 1987.

[2] Ulrike Grömping et al. Relative importance for linear regression in r: the package relaimpo. *Journal of statistical software*, 17(1):1–27, 2006.

[3] John A Hartigan, Pamela M Hartigan, et al. The dip test of unimodality. *The annals of Statistics*, 13(1):70–84, 1985.

[4] Cathy Maugis, Gilles Celeux, and Marie-Laure Martin-Magniette. Variable selection for clustering with gaussian mixture models. *Biometrics*, 65(3):701–709, 2009.