Assignment 4

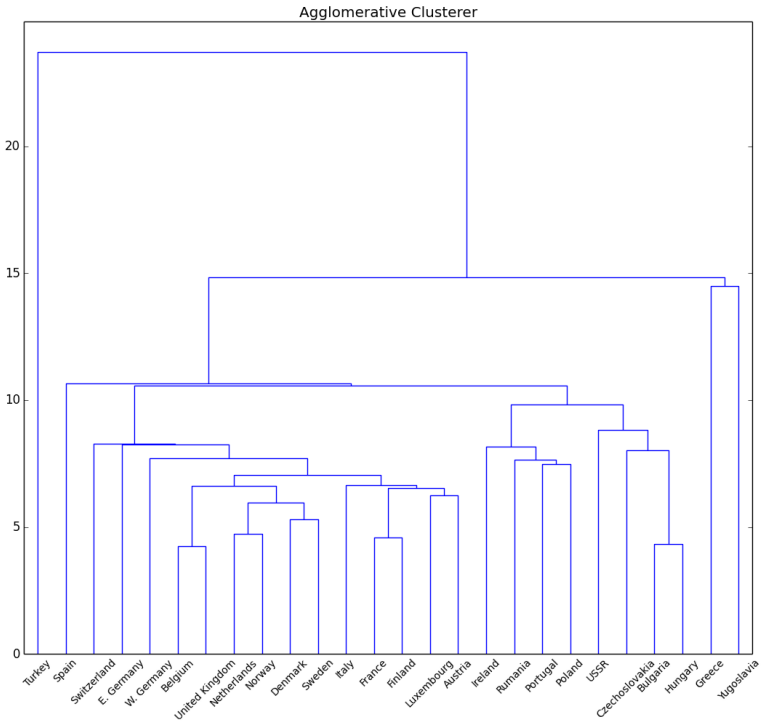
CS498 Applied Machine Learning

Jianshu Wang - March 5, 2018

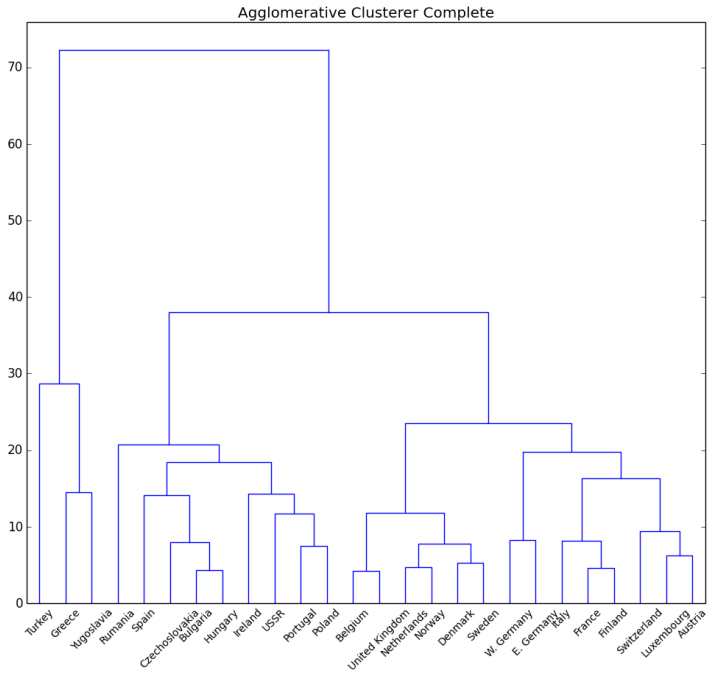
Introduction

Problem 1.1:

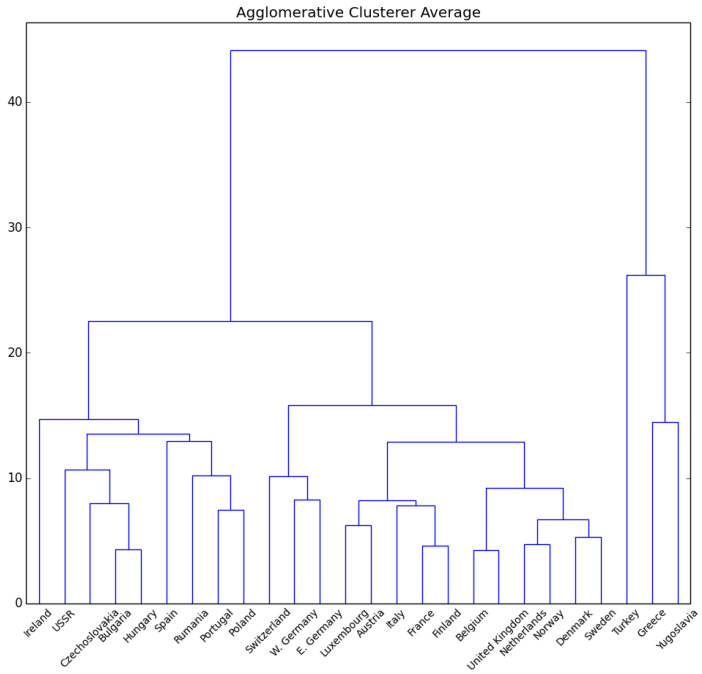
I got three figures and you can check them in my folder



Dendrogram of single link



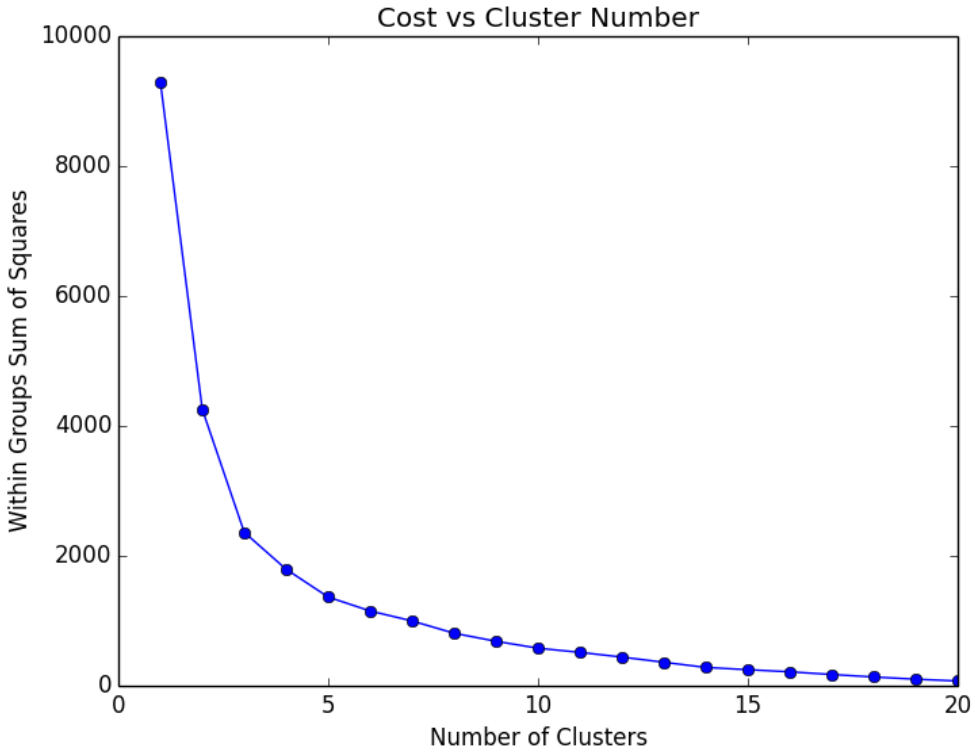
Dendrogram of complete link



Dendrogram of average link

Problem 1.2:

I would select k as 3 or 4, because when k is 3 or 4, they can both meet a sharp decrease in sum of error squares, and this means the newly added cluster can be well clustered from another one.



Cost value vs cluster number

Problem 2.1:

Here are my steps for this problem

1. First I gave labels for all the signal of each class, then I got all the file data and split them into training data 70% and testing data 30%
2. Then it is time to extract pieces from the datas I got 32*3 subarrays are my pieces, and then flat them in to a vector of 96 elements, now I get all the flat matrix for each signal and their labels in train set and test set
3. And now it is time to do K means clustering for train data set, I set my k to be 400. Then I got my cluster centers, I use sklearn.kmeans library in this step
4. Now I can generate the histogram vector by simply describing data piece with the closest cluster, so now I have the histogram vectors for train data set and the corresponding labels
5. Use train histograms and labels to train the randomforest classifier, I still used the packages in sklearn
6. And I used the same method to generate the histograms and corresponding labels for testing data
7. And use our classifier to predict each item in the testing data and calculation the accuracy

k = 400, depth = 16, estimator = 10, 32*3 subarray, I got accuracy 69.65%

Class Confusion Matrix:

```
[[ 3.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.]
 [ 0. 25.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  4.  0.  2.  0.  0.  0.  0.  1.  0.  0.  0.  0.]
 [ 0.  1.  0.  6.  0.  0.  0.  0.  0.  0.  0.  1.  0.  1.]
 [ 0.  0.  0.  0. 24.  0.  0.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  0.]
 [ 0.  1.  0.  0.  1.  0.  0.  9.  0.  1.  2. 11.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  3.  0.  0.  2.  1.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0. 23.  1.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  3. 17.  4.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0. 24.  0.  0.]
 [ 0.  0.  1.  0.  0.  0.  0.  0.  0.  2.  0.  0.  0.  0.]
 [ 0.  1.  0.  0.  0.  0.  0.  1.  0.  0.  2. 15.  0.  5.]]
```

Label vactor is: 'Brush_teeth' 'Climb_stairs' 'Comb_hair' 'Descend_stairs' 'Drink_glass' 'Eat_meat' 'Eat_soup' 'Getup_bed' 'Liedown_bed' 'Pour_water' 'Sitdown_chair' 'Standup_chair' 'Use_telephone' 'Walk'

Problem 2.2:

1st trial: I tried the k = 480, depth = 16, estimator = 10, 32*3 subarray, I got accuracy 70.65%

Class Confusion Matrix:

```
[[ 1.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  1.  0.  0.]
 [ 0. 23.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  2.]
 [ 0.  0.  4.  0.  1.  0.  0.  0.  0.  1.  1.  0.  0.  0.]
 [ 0.  5.  0.  3.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.]
 [ 0.  0.  0.  0. 23.  0.  0.  0.  0.  1.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  1.  0.  0. 16.  0.  0.  0.  8.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  3.  0.  2.  1.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0. 24.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  2.  0.  0. 22.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  4. 20.  0.  0.]
 [ 0.  0.  0.  0.  2.  0.  0.  0.  0.  1.  0.  0.  0.  0.]
 [ 0.  8.  0.  0.  0.  0.  0.  0.  0.  0.  9.  1.  0.  6.]]
```

Label vactor is: 'Brush_teeth' 'Climb_stairs' 'Comb_hair' 'Descend_stairs' 'Drink_glass' 'Eat_meat' 'Eat_soup' 'Getup_bed' 'Liedown_bed' 'Pour_water' 'Sitdown_chair' 'Standup_chair' 'Use_telephone' ‘Walk'

So a higher k can cause better accuracy

2nd trial: I tried the k = 480, depth = 16, estimator = 10, 36*3 subarray, I got accuracy 67.66%

Class Confusion Matrix:

```
[[ 1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  1.  0.  0.]
 [ 0. 24.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.]
 [ 0.  0.  3.  1.  0.  0.  0.  0.  0.  1.  2.  0.  0.  0.]
 [ 0.  5.  0.  4.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0. 23.  0.  0.  0.  0.  1.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  1.  0.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  0.]
 [ 0.  1.  0.  0.  1.  0.  0. 10.  0.  2.  1. 10.  0.  0.]
 [ 0.  0.  0.  0.  1.  0.  0.  0.  0.  0.  4.  1.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0. 24.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  1.  0.  1. 18.  4.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0. 24.  0.  0.]
 [ 0.  0.  0.  0.  3.  0.  0.  0.  0.  0.  0.  0.  0.  0.]
 [ 0. 11.  0.  0.  0.  0.  0.  0.  0.  0.  0.  9.  0.  4.]]
```

Label vactor is: 'Brush_teeth' 'Climb_stairs' 'Comb_hair' 'Descend_stairs' 'Drink_glass' 'Eat_meat' 'Eat_soup' 'Getup_bed' 'Liedown_bed' 'Pour_water' 'Sitdown_chair' 'Standup_chair' 'Use_telephone' 'Walk'

So a longer size subarray does not make a higher accuracy

3rd trial: I tried the k = 480, depth = 16, estimator = 10, 24*3 subarray, I got accuracy 67.66%

Class Confusion Matrix:

```
[[ 1.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  1.  0.  0.]
 [ 0. 23.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  1.]
 [ 0.  0.  3.  0.  1.  0.  0.  0.  0.  1.  0.  2.  0.  0.]
 [ 0.  4.  0.  4.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.]
 [ 0.  0.  0.  0. 23.  0.  0.  0.  0.  1.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  1.  0.  0. 16.  0.  0.  1.  7.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  3.  0.  1.  2.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0. 22.  0.  2.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  1. 14.  9.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  5. 19.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  2.  0.  0.  1.  0.]
 [ 0.  8.  0.  0.  0.  0.  0.  1.  0.  0.  0.  6.  0.  9.]]
```

Label vactor is: 'Brush_teeth' 'Climb_stairs' 'Comb_hair' 'Descend_stairs' 'Drink_glass' 'Eat_meat' 'Eat_soup' 'Getup_bed' 'Liedown_bed' 'Pour_water' 'Sitdown_chair' 'Standup_chair' 'Use_telephone' 'Walk'

So I think the size of different subarray does not change the accuracy