Spectral Clustering, PCA, LDA

Homework 2

0556623

Chieh Yu, Chen

陳傑宇

1. Visualization
   1. Use PCA to project all your data X\_train.csv onto 2D space . Use LDA to project all your data X\_train.csv onto 2D space.

|  |  |
| --- | --- |
| PCA | LDA |
|  |  |

* 1. Use different colors to show clusters obtained by RatioCut and NormalizedCut in 2D space (PCA projection).

|  |  |  |
| --- | --- | --- |
|  | RatioCut | NormalizedCut |
| Linear |  |  |
| RBF |  |  |
| Linear +RBF |  |  |

* 1. With all the data samples are shown by “dots”, the “support vectors” that you obtained from homework#1 should be shown with different symbols, e.g. square, triangle, cross. 4 figures in total (linear, polynomial, RBF, linear+RBF)

1. My Implementation Details
   1. <data\_loader.py>  
      as in homework 1, the code transfer data from .csv into .json format.
   2. <pca.py>  
      In function pca\_fit\_n\_transform():
      1. Computing the Covariance Matrix
      2. Computing eigenvectors and corresponding eigenvalues
      3. Sorting the eigenvectors by decreasing eigenvalues
      4. Choosing k eigenvectors with the largest eigenvalues
      5. Transforming the samples onto the new subspace
   3. <lda.py>  
      In function lda\_fit\_n\_transform():
      1. Calculate means for each class
      2. Compute the Covariance Matrix between classes
      3. Compute the Covariance Matrix within classes
      4. Computing eigenvectors and corresponding eigenvalues
      5. Sorting the eigenvectors by decreasing eigenvalues
      6. Choosing k eigenvectors with the largest eigenvalues
      7. Transforming the samples onto the new subspace
   4. <spectral\_slustering.py>   
      In function spectral\_clustering():
      1. construct similarity graph A (affinity matrix) by kernel function.
      2. compute Laplacian Cut
         1. mode==NCUT: L = D\*\*(-1/2) A D\*\*(-1/2)
         2. mode==RCUT: L = D - A
      3. compute the first k eigenvector of Laplacian Cut, U contains k eigenvectors
         1. mode==NCUT: get T by normalizing all rows of U to norm 1
         2. mode==RCUT: return U
      4. cluster all points with k-means algorithm in k-dims (k eigenvectors) space.
   5. <plot\_svm\_support\_vector.py>
2. Discussion

## Reference

1. <http://sebastianraschka.com/Articles/2014_pca_step_by_step.html>
2. <http://goelhardik.github.io/2016/10/04/fishers-lda/>
3. <https://en.wikipedia.org/wiki/Spectral_clustering>