DATA.STAT.770

Exercise set 10

Problem F1.

I have had serious issues installing the dimensionality reduction toolbox for Matlab (without success), and the relevant component analysis method for R caused errors regrading to the system singularity. Hence I used Neighborhood Components Analysis method implemented in Python package called *metric-learn*, https://pypi.org/project/metric-learn/

Output from program:

```
Nearest neighbor classification using euclidean distance:
accuracy = 95/105 = 0.9048

Nearest neighbor classification using mahalanobis distance:
accuracy = 103/105 = 0.981
```

I divided the Iris-dataset to training set (45 samples) and testing set (105 samples). In this case the NN with Mahalanobis distance outperforms the euclidean distance implementation, but due to the low dimensionality of the data, both methods performed reasonably well.

Problem F2.

Paper 1:

Zeng.F, Zhang.W, Zhang.S, Zheng.N, (2019), *Re-KISSME: A robust resampling scheme for distance metric learning in the presence of label noise.* In *Neurocomputing (Amsterdam), 2019-02-22, Vol.330, p.138-150.* Elsevier B.V. DOI: 10.1016/j.neucom.2018.11.009

In this paper, the authors proposed a robust resampling method for KISSME metric learning (Keep It Simple and Straightforward Metric) called Re-KISSME. Re-KISSME is a fully supervised metric learning method whose aim is to increase the robustness of the learned metric when there is a possibility that there is some label noise present in the training data. The traditional KISSME algorithm and it's variants computes the distances between samples based on Mahalanobis matrix, that is constructed by two covariance matrices. More precisely, KISSME tests the hypothesis, where H_0 denotes that sample pair is dissimilar and H_1 that they are similar. The test statistic is then formulated as follows:

$f(x_{ij}) = log\,\frac{p(x_{ij}|H_0)}{p(x_{ij}|H_1)}$, where $x_{ij} = x_i - x_j$ and p is a probability of a Gaussian distribution for equivalent hypothesis, say $p(x_{ij}|H_i) = \frac{1}{\sqrt{(2\pi)^d|\Sigma_i|}} exp(-\frac{1}{2}x_{ij}^T\Sigma_i^{-1}x_{ij})$. Hence the large value of f indicates that x_i and x_j are dissimilar and small value that they are similar. By taking out the constant terms, f can be reformulated as $f(x_{ij}) = x_{ij}^T(\Sigma_1^{-1} - \Sigma_0^{-1})x_{ij}$ which is the distance between two samples. Therefore KISSME rely on estimating the covariance matrices Σ_1^{-1} and Σ_0^{-1}. However, if there are incorrect labels present the estimates for covariance matrices might be highly biased, which leads to deviating distribution from the training samples and poor performance on the testing samples. Because simply removing of mislabeled samples might lead to information loss, the authors approach this problem by introducing a new resampling method that iteratively computes the covariance matrices. The iterative algorithm uses a Expectation Maximization algorithm to optimize a parameter set $\theta = \{M, k, b, r_{lm}\}$ that is used to characterize the noisy probability distribution. Especially M refers to Mahalanobis matrix that needs to be estimated.

Paper 2:

Qin, S., Song, S., Huang, G., Zhu, L., (2015), *Unsupervised neighborhood component analysis for clustering*. In Neurocomputing (Amsterdam), 2015-11-30, Vol.168, p.609-617. Elsevier B.V

In this paper, the authors proposed a novel unsupervised distance metric learning algorithm that doesn't use any label information while estimating the parameters for distance function. In traditional NCA, the class labels are needed to estimate the transformation matrix A and hence in UsNCA the class labels are replaced by a cluster indicator matrix $F = \{f_{ij}\}$, where $f_{ij} = 1$ if $x_i \in C_j$ and 0 otherwise. To prevent unbalanced clusters, the cluster indicator matrix is replaced by weighted cluster indicator matrix $H = F(F^T F)^{-1/2}$ and hence larger clusters will be down-weighted and smaller ones up-weighted. Finally the objective function for the UsNCA problem can be formulated as $\underset{H\,:\,H^T H = IA}{max}\ tr(H^T PH) - \lambda |A|_F^2$, where

$$P = \{p_{ij}\} = \left\{ \frac{exp(-|Ax_i - Ax_j|_2^2)}{\sum_{k \neq i} exp(-|Ax_i - Ax_k|_2^2)} \right\}.$$