Project: **Kernelization, Kernel Tricks**

Instructions:

1. The project is to be completed individually, but students are allowed to discuss with eachother.

2. Check the Assignment Schedule for the DUE date.

3. For each Exercise, submit the script, the output of running this script, and the answers to the questions (if applicable).

4. Submit via Moodle.

Programming Environment: For this project you may use R or Matlab.

Exercise 1: ***Generating the data sets***. Write a script (in R, Matlab, or SAS) that generates three data sets in a 2-dimensional space, defined as follows (see examples in http://cran.r-project.org/web/packages/kernlab/vignettes/kernlab.pdf):

(a) **BAD_kmeans**: The data set for which the kmeans clustering algorithm will not perform well.

(b) **BAD_pca**: The data set for which the Principal Component Analysis (PCA) dimension reduction method upon projection of the original points into 1-dimensional space (i.e., the first eigenvector) will not perform well.

(c) **BAD_svm**: The data set for which the *linear* Support Vector Machine (SVM) supervised classification method using two classes of points (positive and negative) will not perform well.

(d) Plot each data set in a 2-dimensional space.

Exercise 2: ***Evaluating the "badness" of the data mining methods***. Write a script that uses the BAD data set in Exercise 2, runs the corresponding data mining method, produces the output from the method, and evaluates how bad the performance of this method is. You may use various performance metrics to assess each method (e.g., the variance, precision, recall, F1 measure). Not all the metrics could equally apply to each of the technique. Reading the Performance Metrics chapter by Kanchana and John from the Practical Graph Mining with R book is strongly encouraged for performing this exercise. Also, the book web-site provides the R scripts to play with these metrics, if interested. Report the summary of the performance metrics used and the performance results obtained.

If you are an R user, here is an R script that could give you an idea of how to perform the projection of the data using principal components:

```
data (iris);
 iris;
 X = iris [ , 1:4];

pca = princomp (X, center=TRUE);
pca;
plot (pca); # screeplot
 loadings(pca);     # matrix of eigenvectors
 summary (pca); # check proportion of variance
```

```
P=pca$scores;     # projection of X onto eigenvectors
plot (P[ ,1], P[ ,2]);
points (P [1:50, 1], P[1:50,2], col="red");
points (P [51:100, 1], P[51:100,2], col="blue");
x11();
plot the same for X
```

Exercise 3: ***Kernelizing the methods.*** Write a script that uses the *kernalized* version of each of the data mining method in Exercise 3 (e.g., you may consider using ***kernlab*** and ***kkmeans*** packages in R for kernel SVM+PCA and kmeans, resp.).

(a) Choose at least two kernels for each of the methods.

(b) Use the same performance metrics as in Ex. 3, and compare the performance obtained by the methods after applying the kernel trick versus the original un-kernelized versions of the techniques.

(c) Do you observe the difference in performance when you use different kernels?

(d) What are the best performance results do you get by playing with different kernels and kernel parameters? Also, make sure to report the number of support vectors for the SVM (the good rule of thumb is to strive for no more than 35%-50% support vectors to avoid model overfitting.

For Matlab users, the following packages could be of help:

http://www.mathworks.com/matlabcentral/fileexchange/26182-kernel-k-means

http://www.mathworks.com/matlabcentral/fileexchange/27319-kernel-pca

http://www.mathworks.com/matlabcentral/answers/57232-how-to-use-svmtrain-with-a-custom-kernel-in-matlab

Exercise 4: ***Pipelining***. Dimension reduction is often used as the key data preprocessing step to other data mining techniques downstream of end-to-end data analysis. In this exercise we will use unsupervised kernel PCA as a preprocessing step to clustering. Later in the course, we will use *supervised dimension reduction methods* as a preprocessor to the supervised classification methods.

(a) Generalize your BAD_kmeans data set to very high-dimensional space (d>>2).

(b) Show that the kmeans clustering method does not perform well on that data.

(c) Apply the kernel PCA method to this high dimensional data and identify the number ($m<<d$) of principal components (i.e., eigenvectors) that provide a reasonably good low-dimensional approximation to your data (i.e., based on eigenvalue distribution). How much total variability of the data will be preserved upon using this low-dimensional representation?

(d) Project your original data onto the top m eigenvectors corresponding the largest eigenvalues.

(e) Run the kmeans clustering algorithm on the projected low dimensional data.

(f) Compare the performance of the kmeans on *d*-dimensional original data vs. the *m*-dimensional projected data. Has the performance improved?

(g) If you run the kernel kmeans clustering method on the original data, will get better/worse performance? Can you discuss the pros and cons of using kernel kmeans on the original data directly versus applying the kernel pca as the pre-processing step and then running the kmeans on the low-dimensional data.