HW Assignment 3 (Due date: Nov 7, Friday)

1 Problems

1. [**Kernel Techniques**, 20 + 20 points]
   Show that if $k_1(\mathbf{x}, \mathbf{y})$ and $k_2(\mathbf{x}, \mathbf{y})$ are valid kernel functions, then $k(\mathbf{x}, \mathbf{y}) = k_1(\mathbf{x}, \mathbf{y})k_2(\mathbf{x}, \mathbf{y})$ is also a valid kernel.

   [Bonus] Show that if A is a symmetric positive semidefinite matrix, then $k(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T A\mathbf{y}$ is a valid kernel.
   Hint: the Linear Algebra Background linked from the course web site describe some useful properties of symmetric matrices.

2. [**Kernel Nearest Neighbor**, 20 points]
   Exercise 6.3, page 320.

3. [**Distance-Weighted Nearest Neighbor**, 20 points]
   We have seen how to use kernels to formulate a distance-weighted nearest neighbor algorithm, when the labels are binary. Formulate a kernel-based, distance-weighted nearest neighbor that works for K classes, where $K \geq 2$.

4. [**Matrix Properties**, 20 points]
   Exercise 6.24, page 322.

5. [**Max Margin Hyperplanes**, 20 points]
   Exercise 7.2, page 357.

6. [**SVM Regression**, 20 points]
   Exercise 7.7, page 357.

7. [**Large Margin Perceptron**, 30 points]
   Let $\mathbf{u}$ be a current current vector of parameters and $\mathbf{x}$ and $\mathbf{y}$ two training examples such that $\mathbf{u}^T(\mathbf{x} - \mathbf{y}) < 1$. Use the technique of Lagrange multipliers to find a new vector of parameters $\mathbf{w}$ as the solution to the convex optimization problem below:

   minimize:
   $$J(\mathbf{w}) = \frac{1}{2}\|\mathbf{w} - \mathbf{u}\|^2$$

   subject to:
   $$\mathbf{w}^T(\mathbf{x} - \mathbf{y}) \geq 1$$

8. [**Feature Selection**, 150 points]
   In this exercise, you are asked to run an experimental evaluation of linear SVMs and the nearest neighbor algorithm on the problem of categorizing text documents in the presence of many irrelevant features.

   (a) For this assignment, you are supposed to work with the Dexter Data Set from the UCI Machine Learning Repository. The description webpage is at:
   http://archive.ics.uci.edu/ml/datasets/Dexter
   The actual dataset is located at:
   http://archive.ics.uci.edu/ml/machine-learning-databases/dexter

   Read the description of the dataset. Download the training data and the validation data, together with their labels.

   (b) Explain which of the filter methods discussed in class can be used directly for this dataset. Use each of these methods to rank the features, and train the algorithms using the first N features, where $N = 1, 5, 10, 20, 50, 100, 200, 300, ..., 1000,$ $2000, ..., 20000$. In the rare case that a feature has value 0 for all the examples, set the value of the filter method for that feature to be 0. This means the feature will be put last in the ranking. Also, do not forget to use the absolute value of the test for ranking. After the feature selection step, normalize the training and validation feature vectors, dividing by their Euclidean norm.

   (c) For each learning algorithm, plot the accuracy obtained on the validation data for all values of N. Create a separate graph for each filter method, showing plots for the following four learning algorithms: linear SVMs with a default capacity parameter $C = 1$ and k-nearest neighbor using Euclidean distance, for $k \in \{1, 5, 10\}$. Use the graphs to compare the algorithms and the methods you used for feature selection. When you plot the graphs, for the horizontal axis make sure you use equally spaced tics between the numbers in the list $N = 1, 5, 10, 20, 50, 100, 200, 300, ..., 1000,\ 2000, ..., 20000$. This means that the distance between 1 and 5 should be the same as the distance between 100 and 200 and the distance between 19000 and 20000.

   (d) Select the best performing SVM-filter-N combination and kNN-filter-N combination and run them without doing feature vector normalization. Compare with the results obtained using normalization.

2 Tools

You are free to use MATLAB, R, or packages written in C++/Java/Python to complete this assignment. For your convenience, I recommend using SVMLIGHT, a C implementation, or LIBSVM which has both Java and C++ implementations. You can implement your own k-Nearest Neighbor algorithm; alternatively you may use the `Weka` implementation in `weka.classifiers.lazy.IBk` or the `scikit-learn` implementation in `sklearn.neighbors.KNeighborsClassifier`.

3 Submission

Please turn in a hard copy of your homework report at the beginning of class on the due date. Electronically submit a directory that has your working code, data, run traces, and a concise README file describing these before class. Create a gzipped, tar ball archive of your directory, and upload it on Blackboard by the due date.

For example, if the name is John Williams, creating the archive can be done using the following commands:

```
> tar cvf williams_john.tar williams_john

> gzip williams_john.tar
```

These two steps will create the file 'williams_john.tar.gz' that you can upload on Blackboard.

Please observe the following when handing in homework:

1. Structure, indent, and format your code well.
2. Use adequate comments, both block and in-line to document your code.
3. Type and nicely format the project report, including discussion points, tables, graphs etc. so that it is presentable and easy to read.
4. Working code and/or correct answers is only one part of the assignment. The project report, including discussion of the specific issues which the assignment asks about, is also a very important part of the assignment. Take the time and space to make an adequate and clear project report. On the non-programming learning-theory assignment, clear and complete explanations and proofs of your results are as important as getting the right answer.