Mehryar Mohri
Foundations of Machine Learning
Courant Institute of Mathematical Sciences
Homework assignment 3
April 5, 2013
Due: April 19, 2013

A. Kernels

1. Let $\mathcal{X}$ be a finite set. Show that the kernel K defined over $2^{\mathcal{X}}$, the set of subsets of $\mathcal{X}$, by

$$\forall A, B \in 2^{\mathcal{X}}, K(A, B) = \exp\Big(-\frac{1}{2}|A \Delta B|\Big),$$

   where $A \Delta B$ is the symmetric difference of A and B is PDS (*hint*: you could use the fact that K is the result of the normalization of a kernel function K'). Note that this could define a similarity measure for documents based on the set of their common words, or n-grams, or gappy n-grams, or a similarity measure for images based on some patterns, or a similarity measure for graphs based on their commong sub-graphs.

2. Let $\mathcal{X}$ be a finite set. Let K_0 be a PDS kernel over $\mathcal{X}$, show that K' defined by

$$\forall A, B \in 2^{\mathcal{X}}, K'(A, B) = \sum_{x \in A, x' \in B} K_0(x, x')$$

   is a PDS kernel.

3. Show that K defined by $K(x, x') = \frac{1}{\sqrt{1-(\mathbf{x} \cdot \mathbf{x}')}}$ for all $\mathbf{x}, \mathbf{x}' \in X = \{\mathbf{x} \in \mathbb{R}^N \colon \|\mathbf{x}\|_2 < 1\}$ is a PDS kernel. Bonus point: show that the dimension of the feature space associated to K is infinite (*hint*: one method to show that consists of finding an explicit expression of a feature mapping Φ).

B. Support Vector Machines

1. Download and install the `libsvm` software library from:

   `http://www.csie.ntu.edu.tw/~cjlin/libsvm/` ,

   and briefly consult the documentation to become more familiar with the tools.

2. Consider the `splice` data set

   `http://www.cs.toronto.edu/~delve/data/splice/desc.html`.

   Download the already formatted training and test files of that dataset from
   `http://www.cs.nyu.edu/~mohri/ml13/splice.train.txt`
   `http://www.cs.nyu.edu/~mohri/ml13/splice.test.txt`.

   Use the `libsvm` scaling tool to scale the features of all the data. The scaling parameters should be computed only on the training data and then applied to the test data.

3. Consider the corresponding binary classification which consists of distinguishing two types of splice junctions in DNA sequences using about 60 features. Use SVMs combined with polynomial kernels to tackle this problem.

   To do that, randomly split the training data into ten equal-sized disjoint sets. For each value of the polynomial degree, $d = 1, 2, 3, 4$, plot the average cross-validation error plus or minus one standard deviation as a function of C (let other parameters of polynomial kernels in `libsvm` be equal to their default values), varying C in powers of 10, starting from a small value $C = 10^{-k}$ to $C = 10^{k}$, for some value of k. k should be chosen so that you see a significant variation in training error, starting from a very high training error to a low training error. Expect longer training times with `libsvm` as the value of C increases.

4. Let (C^*, d^*) be the best pair found previously. Fix C to be C^*. Plot the ten-fold cross-validation error and the test errors for the hypotheses obtained as a function of d. Plot the average number of support vectors obtained as a function of d. How many of the support vectors lie on the margin hyperplanes?

5. Suppose we replace in the primal optimization problem of SVMs the penalty term $\sum_{i=1}^{m} \xi_i = \|\boldsymbol{\xi}\|_1$ with $\|\boldsymbol{\xi}\|_2^2$, that is we use the quadratic hinge loss instead. Give the associated dual optimization problem and compare it with the dual optimization problems of SVMs.

6. In class, we presented margin-based generalization bounds in support of the SVM algorithm based on the standard hinge loss. Can you derive similar margin-based generalization bounds when the quadratic hinge loss is used?

To do that, you could use instead of the margin loss function Φ_ρ defined in class the function Ψ_ρ defined by

$$\Psi_\rho(u) = \begin{cases} 1 & \text{if } u \leq 0 \\ \left(\frac{u}{\rho} - 1\right)^2 & \text{if } u \in [0, \rho] \\ 0 & \text{otherwise,} \end{cases}$$

and show that it is a Lipschitz function. Compare the empirical and complexity term of your generalization bound to those given in class using Φ_ρ.

C. Boosting

1. Let $\Psi : \mathbb{R} \to \mathbb{R}$ denote the function defined by $\Psi(u) = (1-u)^2 1_{u \leq 1}$. Show that Ψ is an upper bound on the binary loss function and that it is convex and differentiable. Use Ψ to derive a boosting-style algorithm as in the case of the exponential function used in AdaBoost using coordinate descent. Describe the algorithm in detail.

2. Implement that algorithm with boosting stumps and apply the algorithm to the same data set as question B with the same training and test sets. Plot the average cross-validation error plus or minus one standard deviation as a function of the number of rounds of boosting T by selecting the value of this parameter out of $\{10, 10^2, \ldots, 10^k\}$ for a suitable value of k, as in question B. Let T^* be the best value found for the parameter. Plot the error on the training and test set as a function of the number of rounds of boosting for $t \in [1, T^*]$. Compare your results with those obtained using SVMs in question B.