Mehryar Mohri
Foundations of Machine Learning
Courant Institute of Mathematical Sciences
Homework assignment 3 - solution
Credit: Ashish Rastogi, Afshin Rostamizadeh
Ameet Talwalkar, and Eugene Weinstein.

1. **SVMs**:

   (a) Download and install `libsvm` from

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

5 points Download the `pendigits` data set. The task is to predict the digit label (0 − 9) based on the features computed over the digit image. The data is comma-delimited, with the last item being the label. Normalize the input data so that all feature values are between −1 and 1.

The binary `svm-scale` should be used to normalize the data.

15 points Train and test a SVM using polynomial kernels and 10-fold cross validation. For each setting of the polynomial degree $d = 1, 2, 3, 4$, plot the average error as the data set size is changed from 50 to 1000 data points (keep the first n points of the data set).

The accuracy plot appears in Figure 1.

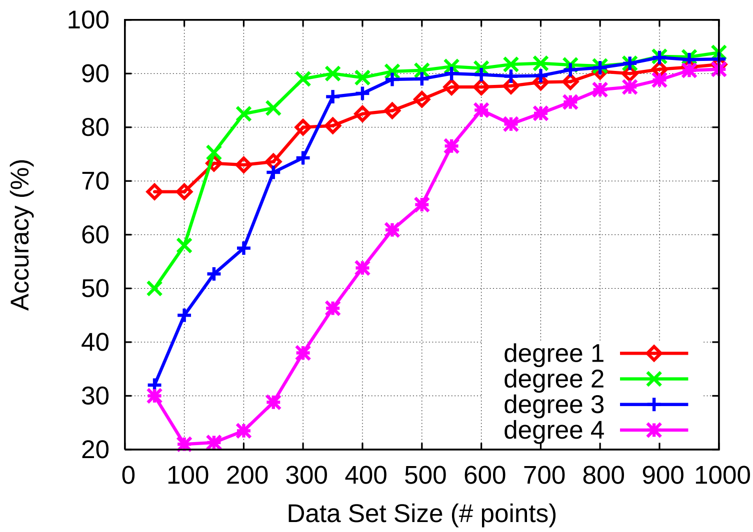

Figure 1: Accuracy achieved with polynomial kernels of varying degrees.

10 points Repeat the learning experiment with radial basis function (RBF) kernels. Use the script `grid.py` packaged with `libsvm` to do a sweep over the space of parameters (C, γ), where C is the SVM learning parameter and γ is the coefficient in the RBF kernel. Report the values of C and γ that yield the highest accuracy under 10-fold cross validation. Also report the accuracy achieved.

The following command should yield a good sweep of the parameter space:

```
grid.py -log2g -5,5,1 -v 10 -log2c -5,5,1 data.txt
```

This highest accuracy achieved with this sweep is 99.0%, and the optimal parameter setting is $C = 4$ and $\gamma = 0.5$.

10 points Let (C^*, γ^*) be the best parameters found in the previous exercise. With C fixed at C^*, plot the 10-fold cross-validation accuracy as the γ parameter is varied. The plot appears in Figure 2.

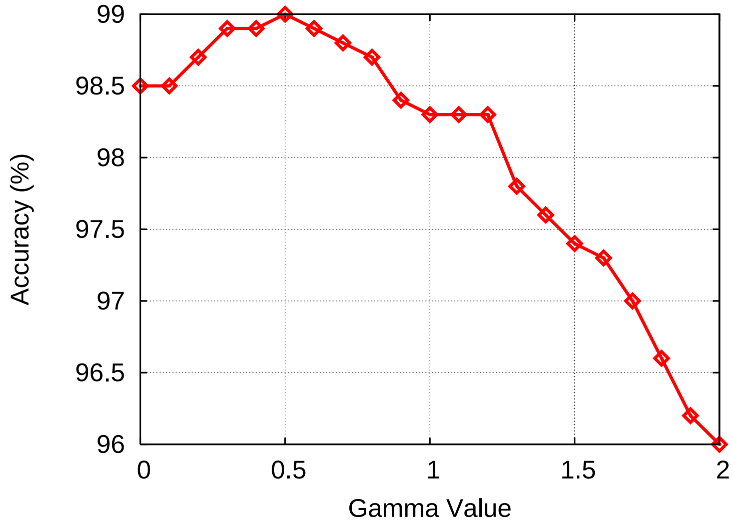

Figure 2: Accuracy achieved with RBF kernel for various settings of γ.

30 points Suppose you wish to use support vector machines to solve a learning problem where some training data points were more important than others. Assume each training point consists of a triplet (x_i, y_i, p_i), where $0 \le p_i \le 1$ is the importance of the ith point. Rewrite the primal SVM constrained optimization problem so that the penalty for mis-labeling a point x_i is scaled by the priority p_i. Then carry this modification through the derivation of the dual solution.

The modified primal optimization problem can be written as

$$\begin{array}{rl} \text{minimize} & \frac{1}{2}||w||^2 + C\sum_{i=1}^{m} \xi_i p_i \\ \text{subject to} & y_i[w \cdot x_i + b] \ge 1 - \xi_i \end{array}$$

The Lagrangian holding for all $w, b, \alpha_i \ge 0, \beta_i \ge 0$ is then

$$\begin{aligned} L(w, b, \alpha) &= \frac{1}{2}||w||^2 + C\sum_{i=1}^{m} \xi_i p_i \\ &\quad - \sum_{i=1}^{m} \alpha_i[y_i(w \cdot x_i + b) - 1 + \xi_i] - \sum_{i=1}^{m} \beta_i \xi_i \end{aligned} \tag{1}$$

Then $\frac{\partial L}{\partial w}$ and $\frac{\partial L}{\partial b}$ are the same as for the regular non-separable SVM optimization problem. We also have $\frac{\partial L}{\partial \xi_i} = Cp_i - \alpha_i - \beta_i$. Thus to satisfy the KKT conditions we have for all $i \in [1, m]$,

$$w = \sum_{i=1}^{m} \alpha_i y_i x_i \tag{2}$$

$$\sum_{i=1}^{m} \alpha_i y_i = 0 \tag{3}$$

$$\alpha_i + \beta_i = C p_i \tag{4}$$

$$\alpha_i [y_i (w \cdot x_i + b) - 1 + \xi_i] = 0 \tag{5}$$

$$\beta_i \xi_i = 0 \tag{6}$$

Plugging Equation 2 into Equation 1, we get

$$\begin{aligned} L = {} & \frac{1}{2} || \sum_{i=1}^{m} \alpha_i y_i x_i ||^2 + C \sum_{i=1}^{m} \xi_i p_i - \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \\ & - \sum_{i=1}^{m} \alpha_i y_i b + \sum_{i=1}^{m} \alpha_i - \sum \alpha_i \xi_i - \sum_{i=1}^{m} \beta_i \xi_i \end{aligned} \tag{7}$$

Using Equation 4, we can simplify:

$$L = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} || \sum_{i=1}^{m} \alpha_i y_i x_i ||^2$$

meaning that the objective function is the same as in the regular SVM problem. The difference is in the constraints on the optimization. Recall that our dual form holds for $\beta_i \geq 0$. Using again equation 4, our optimization problem is to maximize L subject to the constraints:

$$\forall i \in [1, m], 0 \leq \alpha_i \leq C p_i \wedge \sum_{i=1}^{m} \alpha_i y_i = 0.$$

2. **Kernels**:

10 points Given a data set $x_1, \ldots, x_m$ and a kernel $k(x_i, x_j)$ with a Gram matrix K such that $k(x_i, x_j) = K_{ij}$, show that a map $\Phi(\cdot)$ can be given such that if K is positive semidefinite then $k(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j)$.

Because K is positive semidefinite, it can be diagonalized as $K = S \Lambda S^\top$ where Λ is a diagonal matrix of K's eigenvalues and S is the matrix of K's eigenvectors. Further decomposing, we get $K = S \Lambda^{1/2} \Lambda^{1/2} S^\top$. We then have

$$k(x_i, x_j) = K_{ij} = (S \Lambda S^\top)_{ij} = (\Lambda^{1/2} S_i) \cdot (\Lambda^{1/2} S_j)$$

where S_i is the ith eigenvector of K. Thus the kernel map $\Phi(x_i) = \Lambda^{1/2} S_i$ clearly satisfies the desired condition.

10 points Show the converse of the previous statement: that if there exists a mapping $\Phi(x)$, then the matrix K is positive semidefinite.

For any $\alpha_1, \dots \alpha_m \in \mathbb{R}^m$, we have

$$\sum_{i,j=1}^{m} \alpha_i \alpha_j K_{ij} = \left|\left|\sum_{i=1}^{m} \alpha_i \Phi(x_i)\right|\right|^2 \geq 0$$

10 points Let us define a *difference kernel* as $k(x, x') = ||x - x'||$ for $x, x' \in \mathbb{R}^m$. Show that this kernel is not positive definite symmetric (PDS).

Consider the Gram matrix defined as $K_{ij} = k(x_i, x_j)$. It is clear that K will have all zeros on the diagonal. Hence $\mathrm{tr}(K) = 0$. When $K \neq 0$, this means it must have at least one negative eigenvalue. Hence k is not PDS.

10 points The *cosine kernel* is defined as $k(x, x') = \cos \angle(x, x')$. Show that the cosine kernel is PDS.

Rewriting the cosine in terms of the dot product, we have

$$k(x, x') = \cos \angle(x, x') = \frac{x \cdot x'}{|x||x'|}$$

Thus, the cosine kernel is just a scaling of the standard dot product, which is a PDS kernel. Hence, the cosine kernel is also PDS.