Problem Set: Support Vector Machines

Problem 1
Consider the following training data,

class	x_1	x_2
+	1	1
+	2	2
+	2	0
−	0	0
−	1	0
−	0	1

a) Plot these six training points. Are the classes {+, −} linearly separable?
b) Construct the weight vector of the maximum margin hyperplane by inspection and identify the support vectors.
c) If you remove one of the support vectors, does the size of the optimal margin decrease, stay the same, or increase?

Problem 2
Beginning with the optimization problem on slide 11 on the slides "SVM: Part 1," prove that the solution to the hard-margin optimization problem on page 13 provides a separating hyper-plane with maximum margin

a) Suppose that **w**, b is optimal for the hard margin optimization problem on page 13. We must show that **w**, b gives a hyperplane that maximizes the margin. First show that the margin for **w**, b (distance from hyperplane to nearest training example) is $1/||\mathbf{w}||$. To do this, you'll want to use the explicit expression derived in class for the distance between a training example $\mathbf{x}^{(i)}$, $y^{(i)}$ and the the hyperplane defined by **w** and b. You'll also need to make use of the fact **w**, b satisfy the constraints in the hard margin optimization problem (since it is feasible) and that meets at least one of the constraints with equality (since it is optimal).

b) Now let **z**, d be any other separable hyperplane, and let M denote its margin for the data set. Define $\mathbf{z}' = \mathbf{z}/||\mathbf{z}||M$ and $d' = d/||\mathbf{z}||M$. Show that **z'**, d' is a feasible solution for the hard-margin optimization problem, and therefore $||\mathbf{w}||^2 <= ||\mathbf{z}'||^2$ and hence $||\mathbf{w}|| <= ||\mathbf{z}'||$.

c) Use (a) and (b) to show that margin for **w**, b ($=1/||\mathbf{w}||$) is greater than or equal to the margin for **z**, d (= M).

Hint: Feel free to take a look at Andrew Ng's lecture notes. If you borrow material from those notes, be sure to use the notation used in class.

Problem 3
Consider a supervised machine learning problem with two features (x_1, x_2) and 4 training points $\underline{x}^{(1)}, \underline{x}^{(2)}, \underline{x}^{(3)}, \underline{x}^{(4)}$:

data	class	x_1	x_2
$\underline{x}^{(1)}$	+	0	0
$\underline{x}^{(2)}$	+	2	2
$\underline{x}^{(3)}$	–	0	2
$\underline{x}^{(4)}$	–	2	0

Denote w_0 & w_1 for the weights and b for the bias.

a) Argue that there is no solution that satisfies the constraints for the hard-margin SVM problem.
b) Show that there is a solution for the soft margin SVM problem. Explicitly provide such a solution (w_0,w_1,b, $\xi^{(1)}$, $\xi^{(2)}$, $\xi^{(3)}$, $\xi^{(4)}$).

Problem 4

a) For any two documents *x* and *z*, define *K*(*x, z*) to equal the number of unique words that occur in both *x* and *z* (i.e., the size of the intersection of the sets of words in the two documents). Is this function a kernel? Justify your answer. (Hint*:* *K*(*x, z*) is a kernel if there exists $\phi(x)$ such that *K*(*x, z*) = $\phi(x) \cdot \phi(z)$).

b) Assuming that $\mathbf{x} = [x_1, x_2]$, $\mathbf{z} = [z_1, z_2]$ (i.e., both vectors are two-dimensional) and $\beta > 0$, show that the following is a kernel:
$$K(\boldsymbol{x}, \boldsymbol{z}) = (1 + \beta \boldsymbol{x} \cdot \boldsymbol{z})^2 - 1$$
Do so by demonstrating a feature mapping $\Phi(\boldsymbol{x})$ such that $K(\boldsymbol{x}, \boldsymbol{z}) = \Phi(\boldsymbol{x}) \cdot \Phi(\boldsymbol{z})$.

Problem 5

In class we introduced the Multi-class SVM to generalize the binary SVM to multiclass classification. This involved introducing parameters $\mathbf{w}_k$ and b_k for each class k = 1,…,K (where K is the number of classes), and performing prediction for a new data point **x** using
$$\hat{y} = \arg\max_k \mathbf{w}_k \bullet \mathbf{x} + b_k$$

For this problem, prove that this is equivalent to the binary prediction rule sign($\mathbf{w} \bullet \mathbf{x}$ +b) in the case that K = 2. That is, suppose the data is separable and that $\hat{y} = \arg\max_{k \in \{1,2\}} \mathbf{w}_k \bullet \mathbf{x} + b_k)$ predicts the correct label for all data points **x**. Find **w**, *b* (as a function of $\mathbf{w}^{(1)}$, $b^{(1)}$, $\mathbf{w}^{(2)}$, and $b^{(2)}$) that gives an equivalent decision rule. As always, you must show all of your work to obtain full credit.

Problem 6

The MNIST dataset is a database of handwritten digits. This problem will apply SVMs to automatically classify digits; the US postal service uses a similar optical character recognition (OCR) of zip codes to automatically route letters to their destination. The original dataset can be downloaded at *http://yann.lecun.com/exdb/mnist/*. For this problem, we randomly chose a subset of the original dataset. We have provided you with two data files, mnist_train.txt, mnist_test.txt. The training set contains 2000 digits, and the test set contains 1000 digits. Each line represents an image of size 28×28 by a vector of length 784, with each feature specifying a grayscale pixel value. The first column contains the labels of the digits, 0–9, the next 28 columns represent the first row of the image, and so on. We also provide a scripts written in Python, show show_img.py to show a single image; using these will help you have a better understanding of what the data looks like and how it is represented.

Using a Gaussian kernel, you will obtain less than 7% test error. Had you used more training data, SVM with Gaussian kernel can get down to 1.4% test error (degree 4 polynomial obtains 1.1% test error). With further fine-tuning (e.g., augmenting the training set by adding deformed versions of the existing training images), a SVM-based approach can obtain 0.56% test error [2]. The state-of-the-art, which uses a convolutional neural network, obtains 0.23% test error [1].

a. Read in mnist_train.txt, mnist_test.txt and transform them into feature vectors. Normalize the feature vectors so that each feature is in the range [-1, 1]. Since in this dataset each feature has minimum value 0 and maximum value 255, you can do this normalization by transforming each column $\vec{v}$ to $2\vec{v}/255 - 1$. The normalization step can be crucial when you incorporate higher-order features. It also helps prevent numerical difficulties while solving the SVM optimization problem.

b. We explore the use of non-linear kernels within Support Vector Machines. In this problem you will use **Python** and and the widely-used package **sklearn.SVM.SVC** (http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html) and explore some of its functions. The library is built on top of libsvm and implements the SMO algorithm, which performs block coordinate descent in the dual SVM [3,4]. Try the default settings, which uses the Gaussian kernel ('rbf') with γ=1/numfeatures, and C = 1. Make sure that each feature is scaled to [-1, 1]. Note that in the library, the Gaussian kernel is of the form $K(\vec{u}, \vec{v}) = \exp(-\gamma\|\vec{u} - \vec{v}\|^2)$(equivalent to what we showed in class when $\gamma = 1/2\sigma^2$) and the optimization problem is of the form

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{j=1}^{m} \xi_j \\ \textit{subject to} \quad & y_j(w \cdot x_j + b) \geq 1 - \xi_j \\ & \xi_j \geq 0. \end{aligned}$$

Train on the full training set. What is the test error?

c. Rather than using the default settings, we can choose the two parameters to be tuned (C and γ) using cross-validation. If you prefer, sklearn has a helper function for this purpose (http://scikit-learn.org/stable/modules/cross_validation.html) which is called cross_val_score. Report the 5-fold cross-validation error when γ and C are at their default settings. Finally, try different γ and C values to find a model with small cross-validation error. What were the best values that you found? What is the cross-validation error? What is the test error for this setting?

Problem 7 (Extra Credit)

Consider the optimization problem for the support vector machines with Lagrange multipliers:

$$\max_{\vec{\alpha}\geq 0} \min_{\vec{w},b} \frac{1}{2} \|\boldsymbol{w}\|^2 - \sum_i \alpha_i \left[\left(\boldsymbol{w} \cdot \boldsymbol{x}^{(i)} + b \right) y^{(i)} - 1 \right].$$

Show that it is equivalent to maximizing the dual problem:

$$max_{\alpha\geq 0 \, \sum_j \alpha_j y^{(j)}=0} \sum_j \alpha_j - \frac{1}{2} \sum_{i,j} y^{(i)} y^{(j)} \alpha_i \alpha_j (\boldsymbol{x}^{(i)} \cdot \boldsymbol{x}^{(j)})$$

Show all work in your derivation.

References

[1] Dan Ciresan, Ueli Meier, and J¨urgen Schmidhuber. Multi-column deep neural networks for classification. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3642–3649. IEEE, 2012.
[2] Dennis Decoste and Bernhard Sch¨olkopf. Training invariant support vector machines. *Machine Learning*, 46(1-3):161–190, 2002.
[3] Rong-En Fan, Pai-Hsuen Chen, and Chih-Jen Lin. Working set selection using second order information for training support vector machines. *The Journal of Machine Learning Research*, 6:1889–1918, 2005.
[4] John C. Platt. Fast training of support vector machines using sequential minimal optimization. In Bernhard Schölkopf, Christopher J. C. Burges, and Alexander J. Smola, editors, *Advances in kernel methods*, pages 185–208. MIT Press, 1999.