EECS 545 HW 6

Due Wednesday, Nov. 2, at 11 pm via Gradescope.

1. Margins and Slack Variables (5 points)

Let $(\mathbf{w}, b, \boldsymbol{\xi})$ be the solution of the optimal soft-margin hyperplane based on training data $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)$. Show that if $\boldsymbol{x}_i$ is a margin error (see notes), then the distance from $\boldsymbol{x}_i$ to the hyperplane

$$\{\boldsymbol{x} : y_i(\mathbf{w}^T \boldsymbol{x} + b) = 1\}$$

is proportional to ξ_i, and give the constant of proportionality.

2. Support Vector Machines and Cross Validation (10 points)

Download the file `diabetes_scaled.mat` from Canvas. This contains variables `X` and `y` for a classification problem. There are 768 labeled feature vectors. The features are

```
1. Number of times pregnant
2. Plasma glucose concentration
3. Diastolic blood pressure (mm Hg)
4. Triceps skin fold thickness (mm)
5. 2-Hour serum insulin (mu U/ml)
6. Body mass index (weight in kg/(height in m)^2)
7. Diabetes pedigree function
8. Age (years)
```

The binary labels are

```
Binary label:
   +1 = tested positive for diabetes
   -1 = non-positive
```

Reserve the last 268 instances for testing. The goal is to minimize the probability of error on the test data. On the 500 training examples, train an SVM using the Cauchy kernel

$$k(\mathbf{u}, \mathbf{v}) = \left(1 + \frac{\|\mathbf{u} - \mathbf{v}\|^2}{\sigma^2}\right)^{-1},$$

using a 5-fold cross validation estimate of the probability of error to select the parameters C and σ. Report the selected parameters, the cross-validation error estimate at those parameters, the test error, and the number of support vectors of the final classifier. Also, submit your code.

Comments:

- Matlab users: we have provided you with the file `smo.m`, which implements the SMO algorithm mentioned in the notes. Type `help smo` to see how it works. If you are curious, take a look "under the hood" and see if you can understand what it is doing. Alteratively, Matlab provides an SVM solver in one of its toolboxes, or in Python, you can use the scikit-learn library and run "from sklearn.svm import SVC." Details of this function can be found at
`http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html`.

- For consistency of everyone's solutions, please search over the following grid of parameter values (in Matlab notation):

```
grid_sigma = 2.^(0:5);
grid_C     = 2.^(6:11);
```

  So you are searching over 36 parameter combinations.
- SVM solvers often have a tolerance parameters that determines the algorithm's termination criterion. For example, the provided `smo.m` routine has a tolerance parameter that yields good results when set to 0.01. Setting the parameter too small can lead to long run times, and setting it to large can lead to poor convergence. Try to find a middle ground.
- Please don't use randomization to select the folds, even though this is generally a good idea. These data have already been randomized. This way everyone should get a similar answer.
- Please also don't worry about making sure the class proprtions are the same within each fold. Once again this is generally a good idea, but in this problem it is not necessary.

3. PCA (5 points each)

Read over the proof of PCA based on the generalized Rayleigh quotient. Then solve the following problems. The first problem motivates a method for selecting k, as discussed at the end of the first set of notes on PCA.

a. Let $k \in \{0, 1, \ldots, d\}$ be arbitrary. Show that

$$\min_{\boldsymbol{\mu}, A, \{\boldsymbol{\theta}_i\}} \sum_{i=1}^{n} \|\boldsymbol{x}_i - \boldsymbol{\mu} - A\boldsymbol{\theta}_i\|^2 = n \sum_{j=k+1}^{d} \lambda_j,$$

where A ranges over all $d \times k$ matrices with orthonormal columns.

Hint: This is easy if you use properties of the trace operator.

b. Give a condition involving the spectral decomposition of the sample covariance matrix that is both necessary and sufficient for the subspace $\langle A \rangle$ in PCA to be unique.

4. Eigenfaces (5 points each)

In this exercise you will apply PCA to a modified version of the Extended Yale Face Database B. The modified database is available in the file `yalefaces.mat` on Canvas. The modification I made was simply to reduce the resolution of each image by a factor of $4 \times 4 = 16$ to hopefully avoid computational and memory bottlenecks.

For a whirlwind tour of the data, issue the following commands. In Matlab:

```
load yalefaces % loads the 3-d array yalefaces
for i=1:size(yalefaces,3)
    x = double(yalefaces(:,:,i));
    imagesc(x);
    colormap(gray)
    drawnow
    % pause(.1)
end
```

In Python:

```
import numpy as np
import scipy.io as sio
import matplotlib.pyplot as plt
import time

yale = sio.loadmat('yalefaces.mat')
yalefaces = yale['yalefaces']

for i in range(0,yalefaces.shape[2]):
    x = yalefaces[:,:,i]
    ax.imshow(x, extent=[0, 1, 0, 1])
    plt.imshow(x, cmap=plt.get_cmap('gray'))
    #time.sleep(0.1)
    plt.show()
```

Uncomment the `pause/sleep` command to slow things down a bit. What you will see are several different subjects (38 total) under a variety of lighting conditions.

a. By viewing each image as a vector in a high dimensional space, perform PCA on the full dataset. Hand in a plot the sorted eigenvalues (use the `semilogy` command in Matlab; `plt.semilogy` in Python) of the sample covariance matrix. How many principal components are needed to represent 95% of the total variation? 99%? What is the percentage reduction in dimension in each case? Useful commands in Matlab:`reshape, eig, svd, mean, diag`, and Python: `np.reshape, np.linalg.eig, np.linalg.svd, np.mean, np.diag`.

b. Hand in a 4×5 array of subplots showing principal eigenvectors ('eigenfaces') 0 through 19 as images, treating the sample mean as the zeroth order principal eigenvector. Comment on what facial or lighting variations some of the different principal components are capturing. Useful commands in Matlab: `subplot, imagesc, colormap(gray)`, in Python: `plt.imshow(x, cmap=plt.get_cmap('gray'))`, and for subplots a useful link is
`http://matplotlib.org/examples/pylab_examples/subplots_demo.html`

c. Submit your code.

PLEASE NOTE: For uniformity, please do **not** standardize the features as described in the "preprocessing" section of the PCA notes.

SUPPLEMENTAL EXERCISES

Not to be turned in.

1. Cross Validation Bias and Variance

Let $\mathbf{Z}^n = ((\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n))$ denote training data for a supervised learning problem. Let θ be a fixed value of the tuning parameter(s), and let $\widehat{f}_{\theta,\mathbf{Z}^n}$ denote the model learned by some supervised learning algorithm when applied to $\mathbf{Z}_n$.

Let us define the bias of K-fold cross-validation to be

$$B_{\theta,n} = \mathbb{E}_{\mathbf{Z}^n}[\widehat{R}_{CV}(\widehat{f}_{\theta,\mathbf{Z}^n})] - \mathbb{E}_{\mathbf{Z}^n}[R(\widehat{f}_{\theta,\mathbf{Z}^n})],$$

and the variance to be

$$V_{\theta,n} = \mathbb{E}_{\mathbf{Z}^n}\left[\left(\widehat{R}_{CV}(\widehat{f}_{\theta,\mathbf{Z}^n}) - \mathbb{E}_{Z^n}[\widehat{R}_{CV}(\widehat{f}_{\theta,\mathbf{Z}^n})]\right)^2\right].$$

Assume that the sequence $e_n = \mathbb{E}_{\mathbf{Z}^n}[R(\widehat{f}_{\theta,\mathbf{Z}^n})]$ decreases monotonically to a real number $e^* \geq 0$.

(a) Argue that $B_{\theta,n} < 0$, which means that CV tends to be optimistic. Also argue that the magnitude of the bias decreases as K increases (for fixed n).

(b) Aruge that CV is asymptotically unbiased, meaning that $B_{\theta,n} \to 0$ as $n \to \infty$ for fixed K.

(c) Argue that for fixed n, the variance of CV increases as K increases.