CSL407 Machine Learning
Homework 4

Due on 10/10/2014, 11.55pm

Instructions: Upload to your moodle account one zip file containing the following. Please do not submit hardcopy of your solutions. In case moodle is not accessible email the zip file to the instructor at ckn@iitrpr.ac.in. Late submission is not allowed without prior approval of the instructor. You are expected to follow the honor code of the course while doing this homework.

1. **You are allowed to work in teams of size at most 2 for this homework. The submission of the zip file can be made by one of the team members.**
2. A neatly formatted PDF document with your answers for each of the questions in the homework. You can use latex, MS word or any other software to create the PDF. Include the names of the team members and the roll numbers in the pdf document.
3. Include a separate folder named as 'code' containing the scripts for the homework along with the necessary data files. Name the scripts using the problem number.
4. Include a README file explaining how to execute the scripts.
5. Name the ZIP file using the following convention **rollnumber1_rollnumber2_hwnumber.zip**

In this homework you will be experimenting with support vector machines (SVM) and compare its performance with your multi layer perceptron implementation from homework 3.

1. Matlab has an inbuilt quadratic optimizer `quadprog`. Use this package to implement the function `[alpha] = mysvmseparabledual(X, Y)`, which solves the dual SVM formulation for the linear separable case. Use a 10-fold stratified cross validation procedure to evaluate the performance of the classifier on the accompanying `dataset1.mat`. Fix the seed/state of the random number generator function to ensure that the experiments can be recreated.
   (a) Document the classifier in your report in terms of w and b
   (b) Plot the hyperplane corresponding to $\mathbf{w}^T\mathbf{x} + b = -1$ and $\mathbf{w}^T\mathbf{x} + b = 1$
   (c) Report the training time for the classifier.
2. We will now implement an SVM for the non-separable (in the linear sense) case along with the Kernel function.
   (a) Derive the dual formulation of the SVM for the non-separable case that uses the kernel function. The resulting quadratic optimization problem can be stated as

$$\max_{\alpha} \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{m,n=1}^{N} y_m y_n \alpha_m \alpha_n K(\mathbf{x}_m, \mathbf{x}_n) \tag{1}$$

   such that

$$0 \leq \alpha_n \leq C, n = 1, \ldots, N \tag{2}$$

$$\sum_{n=1}^{N} \alpha_n y_n = 0 \tag{3}$$

where C is the box constraint for soft margin and K is the kernel function. As with the linear separable case $\mathbf{w}$ can be expressed as

$$\mathbf{w} = \sum_{n=1}^{N} \alpha_n y_n \phi(\mathbf{x}_n) \tag{4}$$

ϕ is the feature transformation function that transforms the input $\mathbf{x}_n$ into a high dimensional data point. The kernel function computes the inner product between data points in this transformed high dimensional space. It can also be noted that

$$\begin{aligned}
\alpha_n = 0 \quad &\Rightarrow \quad y_n(\mathbf{w}^T\phi(\mathbf{x}_n) + b) \geq 1 && (5)\\
\alpha_n = C \quad &\Rightarrow \quad y_n(\mathbf{w}^T\phi(\mathbf{x}_n) + b) \leq 1 && (6)\\
0 < \alpha_n < C \quad &\Rightarrow \quad y_n(\mathbf{w}^T\phi(\mathbf{x}_n) + b) = 1 && (7)
\end{aligned}$$

Support vectors correspond to the data points that satisfy equation (7), i.e.. they lie on the margin. these data points can then be used to determine the value of the intercept term b.

(b) Implement the function `[alpha] = mysvmnonseparabledual(X, Y, K, C)`, which solves the dual SVM formulation for the non-separable case. Here K denotes the kernel function (kernel matrix), and C corresponds to the box constraint for the soft margin.

(c) Use a 10-fold stratified cross validation procedure to evaluate the performance of the classifier on the accompanying `dataset2.mat`. Experiment with a linear kernel and a quadratic polynomial kernel $K(\mathbf{x}_m, \mathbf{x}_n) = (1 + \mathbf{x}_m^T\mathbf{x}_n)^2$. Use $C = 1$. Report the classifier accuracy as well as the training time taken for the two kernels. Fix the seed/state of the random number generator function to ensure that the experiments can be recreated.

3. We will now compare the performance of your implementation with that of the default Matlab function for training SVM - `svmtrain`. Repeat the experiment in 2(c) using `dataset3.mat` using your implementation and the `svmtrain` function using the SMO method available in Matlab. Document the results for the two implementations in terms of training time and classification accuracy over the 10 folds.

4. Using the default Matlab function experiment with the digit classification dataset of homework 3. Use a 10 fold cross validation procedure to tune the parameters of the following SVM model

   - Box constraint parameter $C \in 10^{[-3:3]}$
   - Gaussian kernel function width - $\sigma \in 10^{[-3:3]}$

   (a) Report the average cross validated accuracy for the different choices of the parameters in the form of a table.

   (b) Design an experiment which allows you to compare the performance of the best SVM model against the best MLP model. State the conclusion on the performance after performing the experiment.