EE6550 Machine Learning, Spring 2016

Homework Assignment #2

Please turn in your solutions of the word problems to EECS Lab 608 and submit your programs, including source code, report and user manual, to iLMS both by 13:00 on May 9th Monday. You are encouraged to consult or collaborate with other students while solving the problems but you will have to turn in your own solutions and programs with your own words. Copying will not be tolerated. If you consult or collaborate, you must indicate all of your consultants or collaborators with their contributions.

Part I: Word problem set.

1. Problem 4.1 of Chapter 4 of the textbook.
2. Problem 4.4 of Chapter 4 of the textbook.
3. Problem 5.12 of Chapter 5 of the textbook.
4. Problem 5.13 of Chapter 5 of the textbook.
5. Problem 5.16 of Chapter 5 of the textbook.

Part II: Programming problem: Implementation of the sequential minimal optimization (SMO) algorithm for a kernel-based SVM-learning algorithm $\mathbb{A}$ for binary classification. Please study Problem 4.4 of Chapter 4 of the textbook on SMO.

Input:

1. A data file which contains a labeled training sample S. This labeled training sample is used to train the kernel-based SVM which will return a hypothesis h_S^{SVM} after n-fold cross-validation.
2. A data file which contains a labeled testing sample $\tilde{S}$. This labeled testing sample is used to evaluate the performance of the returned hypothesis h_S^{SVM} from the kernel-based SVM based on the labeled training sample S.
3. Choice of the kernel function $K(\mathbf{x}, \mathbf{x}')$. There are two choices:
   (a) The standard inner product kernel: $K(\mathbf{x}, \mathbf{x}') = \mathbf{x} \cdot \mathbf{x}'$.
   (b) The Gaussian kernel: $K(\mathbf{x}, \mathbf{x}') = \exp\left\{\frac{-\|\mathbf{x}-\mathbf{x}'\|^2}{2\sigma^2}\right\}$.
4. Choice of n-fold cross-validation, where $n = 5$ or $n = 10$. The free parameter vector $\boldsymbol{\theta}$ is C if the chosen kernel is the inner product kernel and (C, σ) if the chosen kernel is the Gaussian kernel. As discussed in Lecture 1, we use n-fold cross-validation to determine the best value of the free parameter vector $\boldsymbol{\theta}$.
   - Randomly partition a given training sample S of m labeled items into n subsamples or folds.
   - $((\omega_{i1}, c(\omega_{i1})), \ldots, (\omega_{im_i}, c(\omega_{im_i})))$: the ith fold of size m_i, $1 \leq i \leq n$.
     - Usually $m_i = \frac{m}{n}$ for all i.

- For any $i \in [1, n]$, the kernel-based SVM algorithm is trained on all but the ith fold to generate a hypothesis h_i, and the performance of h_i is tested on the ith fold.
- $\hat{R}_{CV}(\boldsymbol{\theta})$: the cross-validation error.

$$\hat{R}_{CV}(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{m_i}\sum_{j=1}^{m_i} 1_{h_i(\omega_{ij})\neq c(\omega_{ij})} = \frac{1}{m}\sum_{i=1}^{n}\sum_{j=1}^{m_i} 1_{h_i(\omega_{ij})\neq c(\omega_{ij})}.$$

- Choose a parameter vector $\boldsymbol{\theta}^*$ which minimize the cross-validation error $\hat{R}_{CV}(\boldsymbol{\theta})$.
- Train the kernel-based SVM algorithm with the best parameter setting $\boldsymbol{\theta}^*$ over the full training sample S of size m. The resulted hypothesis will be the returned hypothesis h_S^{SVM} from the kernel-based SVM algorithm.

Output:

1. The optimal value of the free parameter vector $\boldsymbol{\theta}$ up to the 2nd decimal point.
2. The hypothesis h_S^{SVM} returned by the kernel-based SVM algorithm.
   - If the chosen kernel is the inner product kernel, then

$$h_S^{SVM}(\mathbf{x}) = \text{sgn}(\mathbf{w}^{SVM} \cdot \mathbf{x} + b^{SVM})$$

where $\mathbf{w}^{SVM}$ is the weight vector returned by the SMO algorithm and

$$b^{SVM} = c(\mathbf{x}_j) - \mathbf{w}^{SVM} \cdot \mathbf{x_j} = -F_j^{SVM}$$

for any support vector $\mathbf{x}_j$ with $0 < \lambda_j^{SVM} < C$. Thus we have

$$h_S^{SVM}(\mathbf{x}) = \text{sgn}\left(c(\mathbf{x}_j) + \mathbf{w}^{SVM} \cdot (\mathbf{x} - \mathbf{x}_j)\right)$$

for any support vector $\mathbf{x}_j$ with $0 < \lambda_j^{SVM} < C$.

   - If the chosen kernel is the Gaussian kernel, then

$$h_S^{SVM}(\mathbf{x}) = \text{sgn}(\sum_{i=1}^{m} \lambda_i^{SVM} c(\mathbf{x}_i) K(\mathbf{x}_i, \mathbf{x}) + b^{SVM}),$$

where λ_i^{SVM} are the values of Lagrangian multipliers λ_i returned by the SMO algorithm and

$$b^{SVM} = c(\mathbf{x}_j) - \sum_{i=1}^{m} \lambda_i^{SVM} c(\mathbf{x}_i) K(\mathbf{x}_i, \mathbf{x}_j)$$

for any support vector $\mathbf{x}_j$ with $0 < \lambda_j^{SVM} < C$. Thus we have

$$h_S^{SVM}(\mathbf{x}) = \text{sgn}\left(c(\mathbf{x}_j) + \sum_{i=1}^{m} \lambda_i^{SVM} c(\mathbf{x}_i)(K(\mathbf{x}_i, \mathbf{x}) - K(\mathbf{x}_i, \mathbf{x}_j))\right)$$

for any support vector $\mathbf{x}_j$ with $0 < \lambda_j^{SVM} < C$.

3. Performance evaluation of the returned hypothesis h_S^{SVM} on the labeled testing sample $\tilde{S}$.

What to submit? You should submit the following items:

1. The electronic source code of your SMO algorithm. (It is recommended for you to use MatLab to write your programs.)
2. A printed report consisting of at least:
   (a) results with inner product kernel.
   (b) results with Gaussian kernel.
3. A user manual which should include instructions of
   (a) how to compile the source code;
   (b) how to run the algorithm.