Homework #7

TA in charge: Ming-Feng Tsai, Room 301

RELEASE DATE: 12/25/2008

DUE DATE: 01/08/2009, 4:00 pm IN CLASS

TA SESSION: 01/07/2009, noon to 2:00 pm IN R106

Unless granted by the instructor in advance, you must turn in a hard copy of your solutions (without the source code) for all problems. For problems marked with (*), please follow the guidelines on the course website and upload your source code to designated places.

Discussions on course materials and homework solutions are encouraged. But you should write the final solutions alone and understand them fully. Books, notes, and Internet resources can be consulted, but not copied from.

Since everyone needs to write the final solutions alone, there is absolutely no need to lend your homework solutions and/or source codes to your classmates at any time. In order to maximize the level of fairness in this class, lending and borrowing homework solutions are both regarded as dishonest behaviors and will be punished according to the honesty policy.

You should write your solutions in English with the common math notations introduced in class or in the problems. We do not accept solutions written in any other languages.

7.1 Transforms: Explicit versus Implicit

Consider the following training set:

$$\mathbf{x}_1 = (1,0), y_1 = -1 \qquad \mathbf{x}_2 = (0,1), y_2 = -1 \qquad \mathbf{x}_3 = (0,-1), y_3 = -1$$

$$\mathbf{x}_4 = (-1,0), y_4 = +1 \qquad \mathbf{x}_5 = (0,2), y_5 = +1 \qquad \mathbf{x}_6 = (0,-2), y_6 = +1$$

$$\mathbf{x}_7 = (-2,0), y_7 = +1$$

(1) Use following nonlinear transformation of the input space $\mathcal{X}$ into another two-dimensional space Φ:

$$\phi_1(\mathbf{x}) = (\mathbf{x})_2^2 - 2(\mathbf{x})_1 - 1 \qquad \phi_2(\mathbf{x}) = (\mathbf{x})_1^2 - 2(\mathbf{x})_2 + 1$$

(a) (5%) Transform the training set into the Φ space.

(b) (10%) Write down the equation of the optimal separating "hyperplane" in the Φ space. Then, plot the transformed training points on the Φ plane as well as the boundary between the $+1$ and -1 regions, and mark the support vectors.

(c) (10%) Write down the equation of the corresponding nonlinear curve in the $\mathcal{X}$ space. Then, plot the original training points on the $\mathcal{X}$ plane as well as the boundary between the $+1$ and -1 regions, and mark the support vectors.

(2) Consider the same training set, but instead of explicitly transforming the input space $\mathcal{X}$, apply the (hard-margin) SVM algorithm with the dot product

$$K(\mathbf{x}, \mathbf{x}') = (1 + \mathbf{x} \cdot \mathbf{x}')^2$$

(which corresponds to a second-order polynomial transformation)

(a) (5%) Set up the optimization problem using $(\alpha_1, \cdots, \alpha_7)$ and numerically solve for them. What is the optimal α?

(b) (10%) Write down the equation of the corresponding nonlinear curve in the $\mathcal{X}$ space. Then, plot the original training points on the $\mathcal{X}$ plane as well as the boundary between the $+1$ and -1 regions, and mark the support vectors.

(3) (10%) Should the two nonlinear curves (and support vectors) found in the two subproblems above be the same? Why or why not? Make a comparison and briefly describe your findings.

7.2 A Leave-One-Out Bound of Support Vector Machine

Consider

(A)

$$\begin{aligned} \min_{\alpha} \quad & E_N(\alpha) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i,\mathbf{x}_j) - \sum_{i=1}^{N}\alpha_i \\ \text{s.t.} \quad & \sum_{i=1}^{N} y_i\alpha_i = 0 \\ & 0 \le \alpha_i \le C \end{aligned}$$

and

(B)

$$\begin{aligned} \min_{\beta} \quad & E_{N-1}(\beta) = \frac{1}{2}\sum_{i=1}^{N-1}\sum_{j=1}^{N-1}\beta_i\beta_j y_i y_j K(\mathbf{x}_i,\mathbf{x}_j) - \sum_{i=1}^{N-1}\beta_i \\ \text{s.t.} \quad & \sum_{i=1}^{N-1} y_i\beta_i = 0 \\ & 0 \le \beta_i \le C \end{aligned}$$

(1) (10%) Assume that α^* is an optimal solution for (A) with $\alpha_N^* = 0$. Let $\hat{\beta} = \left(\alpha_1^*, \alpha_2^*, \ldots, \alpha_{N-1}^*\right)$. Argue that $\hat{\beta}$ is a feasible vector for (B). That is, check that $\hat{\beta}$ satisfies all constraints of (B).

(2) (20%) Assume that β^* is an optimal solution for (B). That is, $E_{N-1}(\beta) \ge E_{N-1}(\beta^*)$ for all feasible vectors β. Prove that the $\hat{\beta}$ above satisfies

$$E_{N-1}(\hat{\beta}) = E_{N-1}(\beta^*).$$

In other words, $\hat{\beta}$ is also optimal for (B).

(3) (20%) Recall that #SV = (# of nonzero α_n^*). With the results in (1) and (2), prove that

$$\nu_c\Big(\text{SVM}, N\Big) \le \frac{\#SV}{N}.$$

That is, the leave-one-out error of SVM is upper bounded by the percentage of support vectors. You can use the fact that

$$\alpha_n^* = 0 \Longrightarrow y_n\Big(\big\langle \mathbf{w}^*, \phi(\mathbf{x}_n)\big\rangle - \theta^*\Big) \ge 1.$$

(*Note: see Homework 3 for the definition of* ν_c.)

7.3 Experiments with Linear Support Vector Machine (*)

Write a program to implement the linear Support Vector Machine by solving

$$\begin{aligned} \min_{\mathbf{w},\theta,\xi} \quad & \frac{1}{2}\langle \mathbf{w},\mathbf{w}\rangle + C\sum_{n=1}^{N}\xi_n \\ \text{s.t.} \quad & y_n\Big(\big\langle \mathbf{w},\mathbf{x}_n\big\rangle - \theta\Big) \ge 1 - \xi_n \\ & \xi_n \ge 0 \end{aligned}$$

(1) (50%) Let g_C be the decision function obtained when using C as the parameter. For $C = 0.01, 0.1, 1, 10, 100$, run the algorithm on the following set for training:

`http://www.csie.ntu.edu.tw/~htlin/course/ml08fall/data/hw7_3_1_train.dat`

and the following set for testing:

`http://www.csie.ntu.edu.tw/~htlin/course/ml08fall/data/hw7_3_1_test.dat`

For each setting, list the equation of g_C (in terms of the associated $\mathbf{w}$ and θ). Then, show $\nu(g_C)$, $\hat{\pi}(g_C)$, and $\frac{\#SV}{N}$. Briefly describe your findings.

(2) (50%) For $C = 0.01, 0.1, 1, 10, 100$, run the algorithm on the following set for training:

`http://www.csie.ntu.edu.tw/~htlin/course/ml08fall/data/hw7_3_2_train.dat`

and the following set for testing:

`http://www.csie.ntu.edu.tw/~htlin/course/ml08fall/data/hw7_3_2_test.dat`

For each setting, list the equation of g_C (in terms of the associated $\mathbf{w}$ and θ). Then, show $\nu(g_C)$, $\hat{\pi}(g_C)$, and $\frac{\#SV}{N}$. Briefly describe your findings.

(*Note: You can use any general-purpose packages for quadratic programming, but you cannot use any SVM-specific packages*)

7.4 Experiments with Nonlinear Support Vector Machine (*)

Write a program to implement the nonlinear Support Vector Machine by solving

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i,\mathbf{x}_j) - \sum_{i=1}^{N}\alpha_i \\ \text{s.t.} \quad & \sum_{i=1}^{N} y_i\alpha_i = 0 \\ & 0 \le \alpha_i \le C \end{aligned}$$

Run the algorithm on the following set for training:

`http://www.csie.ntu.edu.tw/~htlin/course/ml08fall/data/hw7_4_train.dat`

and the following set for testing:

`http://www.csie.ntu.edu.tw/~htlin/course/ml08fall/data/hw7_4_test.dat`

(1) (25%) Use the polynomial kernel $(1 + \mathbf{x}\cdot\mathbf{x}')^d$ with $d = 3, 6, 9$, and $C = 0.001, 1, 1000$. Let $g_{d,C}^{(1)}$ be the decision function obtained when using (d, C) as the parameter. For each (d, C) combination, show $\nu\left(g_{d,C}^{(1)}\right)$, $\hat{\pi}\left(g_{d,C}^{(1)}\right)$, and $\frac{\#SV}{N}$. Briefly describe your findings.

(2) (25%) Use the Gaussian-RBF kernel $\exp\left(\frac{-(\mathbf{x}-\mathbf{x}')^2}{2\sigma^2}\right)$ with $\sigma = 0.125, 0.5, 2$ and $C = 0.001, 1, 1000$. Let $g_{\sigma,C}^{(2)}$ be the decision function obtained when using (σ, C) as the parameter. For each (σ, C) combination, show $\nu\left(g_{\sigma,C}^{(2)}\right)$, $\hat{\pi}\left(g_{\sigma,C}^{(2)}\right)$, and $\frac{\#SV}{N}$. Briefly describe your findings.

(*Note: You can use any general-purpose packages for quadratic programming, but you cannot use any SVM-specific packages*)