Homework #6

RELEASE DATE: 12/10/2015

DUE DATE: 12/23/2015, BEFORE NOON

QUESTIONS ABOUT HOMEWORK MATERIALS ARE WELCOMED ON THE COURSERA FORUM.

Unless granted by the instructor in advance, you must turn in a printed/written 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. You are encouraged to (but not required to) include a README to help the TAs check your source code. Any programming language/platform is allowed.

Any form of cheating, lying, or plagiarism will not be tolerated. Students can get zero scores and/or fail the class and/or be kicked out of school and/or receive other punishments for those kinds of misconducts.

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 or Chinese with the common math notations introduced in class or in the problems. We do not accept solutions written in any other languages.

This homework set comes with 200 points and 20 bonus points. In general, every homework set would come with a full credit of 200 points, with some possible bonus points.

Descent Methods for Probabilistic SVM

Recall that the probabilistic SVM is based on solving the following optimization problem:

$$\min_{A,B} \quad F(A,B) = \frac{1}{N}\sum_{n=1}^{N} \ln\left(1+\exp\left(-y_n\left(A\cdot\left(\mathbf{w}_{\text{SVM}}^T\boldsymbol{\phi}(\mathbf{x}_n)+b_{\text{SVM}}\right)+B\right)\right)\right).$$

1. When using the gradient descent for minimizing $F(A,B)$, we need to compute the gradient first. Let $z_n = \mathbf{w}_{\text{SVM}}^T\boldsymbol{\phi}(\mathbf{x}_n)+b_{\text{SVM}}$, and $p_n = \theta(-y_n(Az_n+B))$, where $\theta(s) = \frac{\exp(s)}{1+\exp(s)}$ is the usual logistic function. What is the gradient $\nabla F(A,B)$ in terms of only y_n, p_n, z_n and N? Prove your answer.

2. When using the Newton method for minimizing $F(A,B)$ (see Homework 3 of Machine Learning Foundations), we need to compute $-(H(F))^{-1}\nabla F$ in each iteration, where $H(F)$ is the Hessian matrix of F at (A,B). Following the notations of Question 1, what is $H(F)$ in terms of only y_n, p_n, z_n and N? Prove your answer.

Kernel Ridge Regression

3. Assume that all $\mathbf{x}_n$ are different. When using the Gaussian kernel with $\gamma \to \infty$, what does the kernel matrix K used in kernel ridge regression look like? What is the optimal $\boldsymbol{\beta}$? Prove your answer.

Support Vector Regression

The usual support vector regression model solves the following optimization problem.

$$(P_1) \min_{b,\mathbf{w},\boldsymbol{\xi}^\vee,\boldsymbol{\xi}^\wedge} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{n=1}^{N}(\xi_n^\vee + \xi_n^\wedge)$$
$$\text{s.t.} \quad -\epsilon - \xi_n^\vee \le y_n - \mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n) - b \le \epsilon + \xi_n^\wedge$$
$$\xi_n^\vee \ge 0, \xi_n^\wedge \ge 0.$$

Usual support vector regression penalizes the violations $\xi_n^\vee$ and $\xi_n^\wedge$ linearly. Another popular formulation, called ℓ_2 loss support vector regression in (P_2), penalizes the violations quadratically, just like the ℓ_2 loss SVM introduced in Homework 1 of Machine Learning Techniques.

$$(P_2) \min_{b,\mathbf{w},\boldsymbol{\xi}^\vee,\boldsymbol{\xi}^\wedge} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{n=1}^{N}\left((\xi_n^\vee)^2 + (\xi_n^\wedge)^2\right)$$
$$\text{s.t.} \quad -\epsilon - \xi_n^\vee \le y_n - \mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n) - b \le \epsilon + \xi_n^\wedge.$$

4. Write down an equivalent 'unconstrained' form of (P_2) that is similar to page 10 of the "Support Vector Regression" lecture and prove the equivalence.

5. By a slight modification of the representer theorem presented in the class, the optimal $\mathbf{w}_*$ for (P_2) must satisfy $\mathbf{w}_* = \sum_{n=1}^{N}\beta_n\mathbf{z}_n$. We can substitute the form of the optimal $\mathbf{w}_*$ into the answer in Question 4 to derive an optimization problem that contains $\boldsymbol{\beta}$ (and b) only, which would look like

$$\min_{b,\boldsymbol{\beta}} F(b,\boldsymbol{\beta}) = \frac{1}{2}\sum_{m=1}^{N}\sum_{n=1}^{N}\beta_n\beta_m K(\mathbf{x}_n,\mathbf{x}_m) + \text{ something },$$

where $K(\mathbf{x}_n,\mathbf{x}_m) = (\boldsymbol{\phi}(\mathbf{x}_n))^T(\boldsymbol{\phi}(\mathbf{x}_m))$ is the kernel function. One thing that you should see is that $F(b,\boldsymbol{\beta})$ is differentiable to β_n (and b) and hence you can use gradient descent to solve for the optimal $\boldsymbol{\beta}$. For any $\boldsymbol{\beta}$, let $s_n = \sum_{m=1}^{N}\beta_m K(\mathbf{x}_n,\mathbf{x}_m) + b$. What is $\frac{\partial F(b,\boldsymbol{\beta})}{\partial \beta_m}$? Prove your answer.

Blending

6. Consider $T+1$ hypotheses $g_0, g_1, \cdots, g_T$. Let $g_0(\mathbf{x}) = 0$ for all $\mathbf{x}$. Assume that your boss holds a test set $\{(\tilde{\mathbf{x}}_m, \tilde{y}_m)\}_{m=1}^{M}$, where you know $\tilde{\mathbf{x}}_m$ but $\tilde{y}_m$ is hidden. Nevertheless, you are allowed to know the squared test error $E_{\text{test}}(g_t) = \frac{1}{M}\sum_{m=1}^{M}(g_t(\tilde{\mathbf{x}}_m) - \tilde{y}_m)^2 = e_t$ for $t = 0, 1, 2, \cdots, T$. Also, assume that $\frac{1}{M}\sum_{m=1}^{M}(g_t(\tilde{\mathbf{x}}_m))^2 = s_t$. In terms of all M, e_t, and s_t, how do you calculate $\sum_{m=1}^{M} g_t(\tilde{\mathbf{x}}_m)\tilde{y}_m$? Note that the calculation is the key to test set blending technique that the NTU team has used in KDDCup 2011. Prove your answer. (*Hint: Revisiting the bonus problems of HW3 may help!*)

7. Consider the case where the target function $f : [0,1] \to \mathbb{R}$ is given by $f(x) = x^2$ and the input probability distribution is uniform on $[0,1]$. Assume that the training set has only two examples generated independently from the input probability distribution and noiselessly by f, and the learning model is usual linear regression that minimizes the mean squared error within all hypotheses of the form $h(x) = w_1 x + w_0$. What is $\bar{g}(x)$, the expected value of the hypothesis, that the learning algorithm produces (see Page 10 of Lecture 207)? Prove your answer.

Boosting

8. Assume that linear regression (for classification) is used within AdaBoost. That is, we need to solve the weighted-E_{in} optimization problem.

$$\min_{\mathbf{w}} E_{\text{in}}^{\mathbf{u}}(\mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N} u_n(y_n - \mathbf{w}^T\mathbf{x}_n)^2.$$

The optimization problem above is equivalent to minimizing the usual E_{in} of linear regression on some "pseudo data" $\{(\tilde{\mathbf{x}}_n, \tilde{y}_n)\}_{n=1}^{N}$. Write down your pseudo data $(\tilde{\mathbf{x}}_n, \tilde{y}_n)$ and prove your answer. (*Hint: There is more than one possible form of pseudo data*)

9. Consider applying the AdaBoost algorithm on a binary classification data set where 99% of the examples are positive. Because there are so many positive examples, the base algorithm within AdaBoost returns a constant classifier $g_1(\mathbf{x}) = +1$ in the first iteration. Let $u_+^{(2)}$ be the individual example weight of each positive example in the second iteration, and $u_-^{(2)}$ be the individual example weight of each negative example in the second iteration. What is $u_+^{(2)}/u_-^{(2)}$? Prove your answer.

Kernel for Decision Stumps

When talking about non-uniform voting in aggregation, we mentioned that $\boldsymbol{\alpha}$ can be viewed as a weight vector learned from any linear algorithm coupled with the following transform:

$$\boldsymbol{\phi}(\mathbf{x}) = \Big(g_1(\mathbf{x}), g_2(\mathbf{x}), \cdots, g_T(\mathbf{x})\Big).$$

When studying kernel methods, we mentioned that the kernel is simply a computational short-cut for the inner product $(\boldsymbol{\phi}(\mathbf{x}))^T(\boldsymbol{\phi}(\mathbf{x}'))$. In this problem, we mix the two topics together using the decision stumps as our $g_t(\mathbf{x})$.

10. Assume that the input vectors contain only integers between (including) L and R.

$$\begin{aligned} & g_{s,i,\theta}(\mathbf{x}) = s \cdot \text{sign}\Big(x_i - \theta\Big), \\ \text{where} \quad & i \in \{1, 2, \cdots, d\}, d \text{ is the finite dimensionality of the input space,} \\ & s \in \{-1, +1\}, \theta \in \mathbb{R}, \text{ and } \text{sign}(0) = +1 \end{aligned}$$

Two decision stumps g and $\hat{g}$ are defined as the *same* if $g(\mathbf{x}) = \hat{g}(\mathbf{x})$ for every $\mathbf{x} \in \mathcal{X}$. Two decision stumps are different if they are not the same. How many different decision stumps are there for the case of $d = 2$, $L = 1$, and $R = 6$? Explain your answer.

11. Continuing from the previous question, let $\mathcal{G} = \{$ all different decision stumps for $\mathcal{X}$ $\}$ and enumerate each hypothesis $g \in \mathcal{G}$ by some index t. Define

$$\boldsymbol{\phi}_{ds}(\mathbf{x}) = \left(g_1(\mathbf{x}), g_2(\mathbf{x}), \cdots, g_t(\mathbf{x}), \cdots, g_{|\mathcal{G}|}(\mathbf{x})\right).$$

Derive a simple equation that evaluates $K_{ds}(\mathbf{x}, \mathbf{x}') = (\boldsymbol{\phi}_{ds}(\mathbf{x}))^T(\boldsymbol{\phi}_{ds}(\mathbf{x}'))$ efficiently and prove your answer.

Experiments with Adaptive Boosting.

For Questions 12-18, implement the AdaBoost-Stump algorithm as introduced in Lecture 208. Run the algorithm on the following set for training:

`hw2_adaboost_train.dat`

and the following set for testing:

`hw2_adaboost_test.dat`

Use a total of $T = 300$ iterations (please do not stop earlier than 300), and calculate E_{in} and E_{out} with the 0/1 error.

For the decision stump algorithm, please implement the following steps. Any ties can be arbitrarily broken.

(1) For any feature i, sort all the $x_{n,i}$ values to $x_{[n],i}$ such that $x_{[n],i} \leq x_{[n+1],i}$.

(2) Consider thresholds within $-\infty$ and all the midpoints $\frac{x_{[n],i}+x_{[n+1],i}}{2}$. Test those thresholds with $s \in \{-1, +1\}$ to determine the best (s, θ) combination that minimizes E_{in}^u using feature i.

(3) Pick the best (s, i, θ) combination by enumerating over all possible i.

For those interested, step 2 can be carried out in $O(N)$ time only!!

12. (*) Plot a figure for t versus $E_{\text{in}}(g_t)$. What is $E_{\text{in}}(g_1)$ and what is α_1?

13. From the figure in the previous question, should $E_{\text{in}}(g_t)$ be decreasing or increasing? Write down your observations and explanations.

14. (*) Plot a figure for t versus $E_{\text{in}}(G_t)$, where $G_t(\mathbf{x}) = \sum_{\tau=1}^{t} \alpha_\tau g_\tau(\mathbf{x})$. That is, $G = G_T$. What is $E_{\text{in}}(G)$?

15. (*) Plot a figure for t versus U_t, where $U_t = \sum_{n=1}^{N} u_n^{(t)}$. What is U_2 and what is U_T?

16. (*) Plot a figure for t versus ϵ_t. What is the minimum value of ϵ_t?

17. (*) Plot a figure for t versus $E_{\text{out}}(g_t)$ estimated with the test set. What is $E_{\text{out}}(g_1)$?

18. (*) Plot a figure for t versus $E_{\text{out}}(G_t)$ estimated with the test set. What is $E_{\text{out}}(G)$?

Experiment with Kernel Ridge Regression. Write a program to implement the kernel ridge regression algorithm from Lecture 206, and use it for classification (i.e. implement LSSVM). Consider the following data set

`hw2_lssvm_all.dat`

Use the first 400 examples for training and the remaining for testing. Calculate E_{in} and E_{out} with the 0/1 error.

Consider the Gaussian-RBF kernel $\exp\left(-\gamma\|\mathbf{x}-\mathbf{x}'\|^2\right)$. Try all combinations of parameters $\gamma \in \{32, 2, 0.125\}$ and $\lambda \in \{0.001, 1, 1000\}$.

19. (*) Among all parameter combinations, which combination results in the minimum $E_{\text{in}}(g)$? What is the corresponding $E_{\text{in}}(g)$?

20. (*) Among all parameter combinations, which combination results in the minimum $E_{\text{out}}(g)$? What is the corresponding $E_{\text{out}}(g)$?

.

Bonus: Power of Adaptive Boosting

In this problem, we will prove that AdaBoost can reach $E_{\text{in}}(G_T) = 0$ if T is large enough and every hypothesis g_t satisfies $\epsilon_t \le \epsilon < \frac{1}{2}$. Let U_t be defined as in Question 15. It can be proved (see Lecture 11 of Machine Learning Techniques) that

$$U_{t+1} = \frac{1}{N}\sum_{n=1}^{N} \exp\left(-y_n \sum_{\tau=1}^{t} \alpha_\tau g_\tau(\mathbf{x}_n)\right).$$

and $E_{\text{in}}(G_T) \le U_{T+1}$.

21. (10%) Prove that $U_1 = 1$ and $U_{t+1} = U_t \cdot 2\sqrt{\epsilon_t(1-\epsilon_t)} \le U_t \cdot 2\sqrt{\epsilon(1-\epsilon)}$.

22. (10%) Using the fact that $\sqrt{\epsilon(1-\epsilon)} \le \frac{1}{2}\exp\left(-2(\frac{1}{2}-\epsilon)^2\right)$ for $\epsilon < \frac{1}{2}$, argue that after $T = O(\log N)$ iterations, $E_{\text{in}}(G_T) = 0$.