Homework #6

RELEASE DATE: 12/21/2014

DUE DATE: 01/07/2015, 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.

There are two kinds of regular problems.

- *multiple-choice question (MCQ): There are several choices and **only one of them is correct**. You should choose one and only one.*
- *multiple-response question (MRQ): There are several choices and **none, some, or all of them are correct**. You should write down every choice that you think to be correct.*

*Some problems also come with **(+ . . .)** that contains additional todo items.*
If there are big bonus questions (BBQ), please simply follow the problem guideline to write down your solution, if you choose to tackle them.

> 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.

Decent 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. (MCQ) 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 θ is the usual logistic function. What is the gradient $\nabla F(A,B)$?

[a] $\frac{1}{N}\sum_{n=1}^{N}[-y_n p_n z_n, -y_n p_n]^T$

[b] $\frac{1}{N}\sum_{n=1}^{N}[-y_n p_n z_n, +y_n p_n]^T$

[c] $\frac{1}{N}\sum_{n=1}^{N}[+y_n p_n z_n, -y_n p_n]^T$

[d] $\frac{1}{N}\sum_{n=1}^{N}[+y_n p_n z_n, +y_n p_n]^T$

[e] none of the other choices

(+ derivation of your choice)

2. (MCQ) 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 the previous question, what is $H(F)$?

[a] $\frac{1}{N}\sum_{n=1}^{N}\begin{bmatrix} z_n^2 p_n(1-p_n) & z_n p_n(1-p_n) \\ z_n p_n(1-p_n) & p_n(1-p_n) \end{bmatrix}$

[b] $\frac{1}{N}\sum_{n=1}^{N}\begin{bmatrix} z_n^2 y_n(1-y_n) & z_n y_n(1-y_n) \\ z_n y_n(1-y_n) & y_n(1-y_n) \end{bmatrix}$

[c] $\frac{1}{N}\sum_{n=1}^{N}\begin{bmatrix} z_n^2 p_n(1-y_n) & z_n p_n(1-y_n) \\ z_n p_n(1-y_n) & p_n(1-y_n) \end{bmatrix}$

[d] $\frac{1}{N}\sum_{n=1}^{N}\begin{bmatrix} z_n^2 y_n(1-p_n) & z_n y_n(1-p_n) \\ z_n y_n(1-p_n) & y_n(1-p_n) \end{bmatrix}$

[e] none of the other choices

(+ derivation of your choice)

Kernel Regression Models

3. (MCQ) Recall that N is the size of the data set and d is the dimensionality of the input space. What is the size of matrix the get inverted in kernel ridge regression?

[a] $d \times d$

[b] $N \times N$

[c] $Nd \times Nd$

[d] $N^2 \times N^2$

[e] none of the other choices

(+ explanation of your choice)

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

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

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, penalizes the violations quadratically, just like the ℓ_2 loss SVM.

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

4. (MCQ) Which of the following is an equivalent 'unconstrained' form of (P_2)?

[a] $\min_{b,\mathbf{w}} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{n=1}^{N}(y_n - \mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n) - b)^2$

[b] $\min_{b,\mathbf{w}} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{n=1}^{N}(|y_n - \mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n) - b| - \epsilon)^2$

[c] $\min_{b,\mathbf{w}} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{n=1}^{N}(\max(\epsilon, |y_n - \mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n) - b|))^2$

[d] $\min_{b,\mathbf{w}} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{n=1}^{N}(\max(0, |y_n - \mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n) - b| - \epsilon))^2$

[e] none of the other choices

(+ explanation of your choice)

5. (MCQ) By a slight modification of the representer theorem presented in the class, the optimal $\mathbf{w}$ 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 the previous question 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 $s_n = \sum_{m=1}^{N}\beta_m K(x_n,x_m) + b$. What is $\frac{\partial F(b,\boldsymbol{\beta})}{\partial \beta_m}$?

[a] $\sum_{n=1}^{N}\beta_n K(\mathbf{x}_n,\mathbf{x}_m) - 2C\sum_{n=1}^{N}[\![|y_n - s_n| \ge \epsilon]\!]\,(|y_n - s_n| - \epsilon)\text{sign}(y_n - s_n)K(\mathbf{x}_n,\mathbf{x}_m)$

[b] $\sum_{n=1}^{N}\beta_n K(\mathbf{x}_n,\mathbf{x}_m) + 2C\sum_{n=1}^{N}[\![|y_n - s_n| \ge \epsilon]\!]\,(|y_n - s_n| - \epsilon)\text{sign}(y_n - s_n)K(\mathbf{x}_n,\mathbf{x}_m)$

[c] $\sum_{n=1}^{N}\beta_n K(\mathbf{x}_n,\mathbf{x}_m) - 2C\sum_{n=1}^{N}[\![|y_n - s_n| \le \epsilon]\!]\,(|y_n - s_n| - \epsilon)\text{sign}(y_n - s_n)K(\mathbf{x}_n,\mathbf{x}_m)$

[d] $\sum_{n=1}^{N}\beta_n K(\mathbf{x}_n,\mathbf{x}_m) + 2C\sum_{n=1}^{N}[\![|y_n - s_n| \le \epsilon]\!]\,(|y_n - s_n| - \epsilon)\text{sign}(y_n - s_n)K(\mathbf{x}_n,\mathbf{x}_m)$

[e] none of the other choices

(+ derivation of your choice)

Blending and Bagging

6. (MCQ) 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 test squared 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 $\frac{1}{M}\sum_{m=1}^{M}(g_t(\tilde{\mathbf{x}}_m))^2 = s_t$. Which of the following equals $\sum_{m=1}^{M} g_t(\tilde{\mathbf{x}}_m)\tilde{y}_m$? Note that the calculation is the key to test set blending technique that the NTU team has used in KDDCup 2011.

[a] $\frac{M}{2}(+e_0 + s_t - e_t)$

[b] $\frac{M}{2}(+e_0 - s_t + e_t)$

[c] $\frac{M}{2}(-e_0 + s_t - e_t)$

[d] $\frac{M}{2}(-e_0 - s_t + e_t)$

[e] none of the other choices

(+ proof of your choice)

7. (MCQ) If bootstrapping is used to sample $N' = pN$ examples out of N examples and N is very large. Approximately how many of the N examples will not be sampled at all?

[a] $e^{-1} \cdot N$

[b] $e^{-p} \cdot N$

[c] $e^{-1/p} \cdot N$

[d] $(1 - e^{-p}) \cdot N$

[e] $(1 - e^{-1/p}) \cdot N$

(+ proof of your choice)

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 $g_t(\mathbf{x})$.

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

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

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. Which of the followings are true?

[a] $\mathcal{X}$ is of infinite size

[b] $g_{+1,1,L-1}$ is the same as $g_{-1,3,R+1}$

[c] $g_{s,i,\theta}$ is the same as $g_{s,i,\text{ceiling}(\theta)}$, where ceiling($\theta$) is the smallest integer that is greater than or equal to θ

[d] The number of different decision stumps equals the size of $\mathcal{X}$

[e] There are 22 different decision stumps for the case of $d = 2$ and $(L, R) = (1, 6)$

(+ explanation of your choices)

9. (MCQ) 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

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

Derive a simple equation that evaluates $K_{ds}(\mathbf{x}, \mathbf{x}') = \boldsymbol{\phi}_{ds}(\mathbf{x})^T\boldsymbol{\phi}_{ds}(\mathbf{x}')$ efficiently. Which of the following equation is correct?

[a] $K_{ds}(\mathbf{x}, \mathbf{x}') = d(R - L) - 2\|\mathbf{x} - \mathbf{x}'\|_1 + 2$

[b] $K_{ds}(\mathbf{x}, \mathbf{x}') = d(R - L) - 2\|\mathbf{x} - \mathbf{x}'\|_1 - 2$

[c] $K_{ds}(\mathbf{x}, \mathbf{x}') = 2d(R - L) - 4\|\mathbf{x} - \mathbf{x}'\|_1 + 2$

[d] $K_{ds}(\mathbf{x}, \mathbf{x}') = 2d(R - L) - 4\|\mathbf{x} - \mathbf{x}'\|_1 - 2$

[e] none of the other choices

(+ proof of your choice)

Theory of AdaBoost

10. (MCQ) After running the AdaBoost algorithm, which example $(\mathbf{x}_n, y_n)$ results in the largest $u_n^{(T+1)}$ value?

[a] $\text{argmax}_{1\le n\le N}(+y_n \sum_{t=1}^{T} \epsilon_t g_t(\mathbf{x}_n))$

[b] $\text{argmax}_{1\le n\le N}(-y_n \sum_{t=1}^{T} \epsilon_t g_t(\mathbf{x}_n))$

[c] $\text{argmax}_{1\le n\le N}(+y_n \sum_{t=1}^{T} \alpha_t g_t(\mathbf{x}_n))$

[d] $\text{argmax}_{1 \le n \le N} (-y_n \sum_{t=1}^{T} \alpha_t g_t(\mathbf{x}_n))$

[e] none of the other choices

(+ proof of your choice)

11. (MRQ) For the AdaBoost algorithm, let $U^{(t)} = \sum_{n=1}^{N} u_n^{(t)}$, the total example weights in the beginning of the t-th iteration, and $G_t(\mathbf{x}) = \text{sign}\left(\sum_{\tau=1}^{t} \alpha_\tau g_\tau(\mathbf{x})\right)$, the aggregated classifier until the t-th iteration. Which of the following is true?

[a] $U^{(1)} = 1$

[b] $E_{\text{in}}(g_t) \le U^{(t+1)}$ for all $t \ge 1$

[c] $E_{\text{in}}(G_t) \le U^{(t+1)}$ for all $t \ge 1$

[d] $U^{(t+1)} < U^{(t)}$ if $\epsilon_t < \frac{1}{2}$

[e] $U^{(t+1)} = U^{(t)}$ if $\epsilon_t = 1$

(+ proof of your choice)

Experiments of AdaBoost

Implement the AdaBoost algorithm (Page 16 of Lecture 208) with decision stumps. Run the algorithm on the following set for training:

`http://www.csie.ntu.edu.tw/~htlin/course/ml14fall/hw6/hw6_train.dat`

and the following set for testing:

`http://www.csie.ntu.edu.tw/~htlin/course/ml14fall/hw6/hw6_test.dat`

Use a total of $T = 300$ iterations. Let $G_t(\mathbf{x}) = \text{sign}\left(\sum_{\tau=1}^{t} \alpha_\tau g_\tau(\mathbf{x})\right)$, the aggregated classifier until the t-th iteration. Evaluate E_{in} and E_{out} (by the test set) using the 0/1 error.

12. (MCQ, *) Which of the following is true about $E_{\text{in}}(G_T)$?

[a] $E_{\text{in}}(G_T) = 0$

[b] $0 < E_{\text{in}}(G_T) < 0.1$

[c] $0.1 \le E_{\text{in}}(G_T) < 0.2$

[d] $0.2 \le E_{\text{in}}(G_T) < 0.3$

[e] $E_{\text{in}}(G_T) > 0.3$

13. (MCQ, *) Which of the following is true about $E_{\text{out}}(G_T)$?

[a] $E_{\text{out}}(G_T) = 0$

[b] $0 < E_{\text{out}}(G_T) < 0.1$

[c] $0.1 \le E_{\text{out}}(G_T) < 0.2$

[d] $0.2 \le E_{\text{out}}(G_T) < 0.3$

[e] $E_{\text{out}}(G_T) > 0.3$

14. (MCQ, *) Which of the following is true about $U^{(t)} = \sum_{n=1}^{N} u_n^{(t)}$?

[a] $U^{(T)} = 0$

[b] $0 < U^{(T)} < 0.1$

[c] $0.1 \le U^{(T)} < 0.2$

[d] $0.2 \le U^{(T)} < 0.3$

[e] $U^{(T)} > 0.3$

15. (MRQ, *) Which of the following statements are true in your experiment?

[a] $E_{\text{in}}(g_t) \le E_{\text{in}}(G_T)$ for all $t = 1, 2, \ldots, T$

[b] $E_{\text{in}}(G_t)$ is non-increasing in t

[c] $E_{\text{out}}(G_t)$ is non-increasing in t

[d] $U^{(t)}$ is non-increasing in t

[e] α_t is non-increasing in t

Experiments with Unpruned Decision Tree

.

Implement the simple C&RT algorithm without pruning using the Gini index as the impurity measure as introduced in the class. For the decision stump used in branching, if you are branching with feature i and direction s, please sort all the $x_{n,i}$ values to form (at most) $N + 1$ segments of equivalent θ, and then pick θ within the median of the segment.

Run the algorithm on the following set for training:

`http://www.csie.ntu.edu.tw/~htlin/course/ml14fall/hw6/hw6_train.dat`

and the following set for testing:

`http://www.csie.ntu.edu.tw/~htlin/course/ml14fall/hw6/hw6_test.dat`

16. (MCQ, *) How many branch functions are there in the tree?

[a] 6

[b] 8

[c] 10

[d] 12

[e] 14

17. (MCQ, *) Which of the following is closest to the E_{in} (evaluated with 0/1 error) of the tree?

[a] 0.0

[b] 0.1

[c] 0.2

[d] 0.3

[e] 0.4

18. (MCQ, *) Which of the following is closest to the E_{out} (evaluated with 0/1 error) of the tree?

[a] 0.00

[b] 0.05

[c] 0.15

[d] 0.25

[e] 0.35

Now implement the Bagging algorithm and couple it with your decision tree above to make a preliminary random forest G_{RF}. Produce $T = 300$ trees with bagging. Repeat the experiment for 100 times and compute average E_{in} and E_{out} using the 0/1 error.

19. (MCQ, *) Which of the following is true about the average $E_{\text{out}}(G_{RF})$

[a] $0.13 \le E_{\text{out}}(G_{RF}) < 0.16$

[**b**] $0.10 \le E_{\text{out}}(G_{RF}) < 0.13$

[**c**] $0.07 \le E_{\text{out}}(G_{RF}) < 0.10$

[**d**] $0.04 \le E_{\text{out}}(G_{RF}) < 0.07$

[**e**] $E_{\text{out}}(G_{RF}) \le 0.04$

20. (MRQ, *) Let g_t be each tree in your random forest above, and G_t be the uniform aggregation of the first t trees. Which of the followings are true?

[**a**] $E_{\text{out}}(g_t) \ge E_{\text{out}}(G_{RF})$ for $t = 1, 2, \ldots, T$

[**b**] $E_{\text{out}}(G_t) \ge E_{\text{out}}(G_{RF})$ for $t = 1, 2, \ldots, T$

[**c**] the average of $E_{\text{in}}(g_t)$ is non-increasing with t

[**d**] the average of $E_{\text{out}}(g_t)$ is non-increasing with t

[**e**] the variance of $E_{\text{out}}(g_t)$ is non-increasing with t

Bonus: More about AdaBoost

21. (BBQ, 10 points) Prove or disprove the following claim: “when running the AdaBoost algorithm and getting $g_1, g_2, \ldots, g_T$, all those g_t’s are always different.”

22. (BBQ, 10 points) Prove or disprove the following claim: “any G found by AdaBoost-Stump can be equivalently expressed as a decision tree.”

Answer guidelines. First, please write down your name and school ID number.

Name: School ID:

Then, fill in your answers for MCQ, MRQ and BFQ in the table below.

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16
17	18	19	20

Lastly, please write down your solution to those (+ ...) parts and bonus problems, using as many additional pages as you want.

Each problem is of 10 points.

- For Problem with (+ ...), the answer in the table is of 3 score points, and the (+ ...) part is of 7 score points. If your solution to the (+ ...) part is clearly different from your answer in the table, it is regarded as a suspicious violation of the class policy (plagiarism) and the TAs can deduct some more points based on the violation.
- For Problem without (+ ...), the problem is of 10 points by itself and the TAs can decide to give you partial credit or not as long as it is fair to the whole class.