Homework #6

RELEASE DATE: 12/19/2013

DUE DATE: 1/6/2014, BEFORE THE END OF CLASS

QUESTIONS ABOUT HOMEWORK MATERIALS ARE WELCOMED ON THE 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.

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 of ours would come with a full credit of 200 points, with some possible bonus points.

Kernel from 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 $g_t(\mathbf{x})$.

(1) (10%) Assume that the input vectors contain only integers between (including) L and R.

$$g_{s,i,\theta}(\mathbf{x}) = \text{sign}\Big(s \cdot x_i - \theta\Big).$$

Two decision stumps $g^{(1)}$ and $g^{(2)}$ are defined as the *same* if $g^{(1)}(\mathbf{x}) = g^{(2)}(\mathbf{x})$ for every $\mathbf{x} \in \mathcal{X}$. Two decision stumps are different if they are not the same. Argue that there are only finitely-many different decision stumps for $\mathcal{X}$ and list all of them for the case of $d = 2$ and $(L, R) = (1, 6)$.

(2) (10%) Let $\mathcal{G} = \{$ all different decision stumps for $\mathcal{X}$ $\}$. Since $\mathcal{G}$ is finite, we can 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.

The result can be easily extended to the case when $\mathcal{X}$ is an arbitrary box in $\mathbb{R}^d$ as well.

Power of Adaptive Boosting

Please consider the adaptive boosting (AdaBoost) algorithm shown on P.16 of Lecture 24. In the following problems, we will prove that AdaBoost can reach $E_{\text{in}}(G) = 0$ if T is large enough and every hypothesis g_t satisfies $\epsilon_t \leq \epsilon < \frac{1}{2}$.

(3) (10%) Let $U^{(t)} = \sum_{n=1}^{N} u_n^{(t)}$ at the beginning of the t-th iteration. According to the AdaBoost algorithm, for $t \geq 1$, prove that

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

(4) (10%) By the result in the previous problem, prove that $E_{\text{in}}(G) \leq U^{(T+1)}$.

(5) (10%) According to the AdaBoost algorithm above, for $t \geq 1$, prove that $U^{(t+1)} = U^{(t)} \cdot 2\sqrt{\epsilon_t(1-\epsilon_t)}$.

(6) (10%) Using $0 \leq \epsilon_t \leq \epsilon < \frac{1}{2}$, for $t \geq 1$, prove that $\sqrt{\epsilon_t(1-\epsilon_t)} \leq \sqrt{\epsilon(1-\epsilon)}$.

(7) (10%) Using $\epsilon < \frac{1}{2}$, prove that $\sqrt{\epsilon(1-\epsilon)} \leq \frac{1}{2}\exp\left(-2(\frac{1}{2}-\epsilon)^2\right)$.

(8) (10%) Using the results above, prove that $U^{(T+1)} \leq \exp\left(-2T(\frac{1}{2}-\epsilon)^2\right)$.

(9) (10%) Using the results above, argue that after $T = O(\log N)$ iterations, $E_{\text{in}}(G) = 0$.

Experiments with Adaptive Boosting

Implement the AdaBoost algorithm with decision stumps (i.e., use A_{ds} as A_b). Run the algorithm on the following set for training:

`http://www.csie.ntu.edu.tw/~htlin/course/ml13fall/hw6/hw6_train.dat`

and the following set for testing:

`http://www.csie.ntu.edu.tw/~htlin/course/ml13fall/hw6/hw6_test.dat`

Use a total of $T = 300$ iterations. Let $G_t(\mathbf{x}) = \text{sign}\left(\sum_{\tau=1}^{t} \alpha_\tau g_\tau(\mathbf{x})\right)$. Plot $E_{\text{in}}(G_t)$, $E_{\text{out}}(G_t)$, and $U^{(t)}$ (see the definition above) as functions of t on the same figure. Briefly state your findings.

(10) (10%, *) for $E_{\text{in}}(G_t)$

(11) (10%, *) for $E_{\text{out}}(G_t)$

(12) (10%, *) for $U^{(t)}$

(13) (10%, *) for your explanation of the findings

(14) (10%, *) Plot the training examples $(\mathbf{x}_n, y_n)$ and mark the positive/negative examples clearly. Then, mark the vectors with top 10 numerical values of $u_n^{(T)}$. Briefly state your findings.

Experiments with Unpruned Decision Tree (*)

(15) (10%, *) Implement the C&RT algorithm instroduced in class for numerical, non-missing features only, without pruning. Run the algorithm on the following set for training:

`http://www.csie.ntu.edu.tw/~htlin/course/ml13fall/hw6/hw6_train.dat`

and the following set for testing:

`http://www.csie.ntu.edu.tw/~htlin/course/ml13fall/hw6/hw6_test.dat`

Show your binary tree G (by graph or by writing down the if-then-else).

(16) (10%, *) Report $E_{\text{in}}(G)$ and $E_{\text{out}}(G)$.

(17) (10%, *) Implement the Bagging algorithm and couple it with your C&RT to make a preliminary random forest. Produce 100 trees with Bagging. Plot a histogram of $E_{\text{out}}(G_t)$, where each G_t is one of the trees. Compare your result with $E_{\text{out}}(G)$ above and report your findings.

(18) (10%, *) Let $\bar{G}_t$ be the uniform aggregation of the first t trees in your random forest above. Plot $E_{\text{in}}(\bar{G}_t)$ and $E_{\text{out}}(\bar{G}_t)$ on the same figure. Report your findings.

Yes, A Lighter Homework :-)

(19) (10%) Which one of our lectures do you like most? Why?

(20) (10%) Which one of our lectures do you like least? Why?

Bonus: Kernel Shifting

For a valid kernel K, consider a new kernel

$$\tilde{K}(\mathbf{x}, \mathbf{x}') = K(\mathbf{x}, \mathbf{x}') + c$$

for some positive c. It is not difficult to see that $\tilde{K}$ is also a valid kernel.

(21) (Bonus 10%) Argue that for the dual of soft-margin SVM, using $\tilde{K}$ instead of K yields exactly the same solution and exactly the same hypothesis.

(22) (Bonus 10%) Use the result above to simplify your kernel in (2).