Homework #7

TA in charge: Te-Kang Jan

RELEASE DATE: 12/28/2009

DUE DATE: 1/4/2009, 4:20 pm IN CLASS

TA SESSION: 12/31/2009 (**Thursday**), noon to 2:00 pm IN R106

Unless granted by the instructor in advance, you must turn in a written/printed 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 with the common math notations introduced in class or in the problems. We do not accept solutions written in any other languages.

7.1 Power of Adaptive Boosting

The adaptive boosting (AdaBoost) algorithm, as shown in the class slides, is as follows:

- Input: $\mathcal{D} = \{(\mathbf{x}_n, y_n)\}_{n=1}^N$.
- Set $u_n = \frac{1}{N}$ for all n.
- For $t = 1, 2, \cdots, T$,
  - Learn a simple rule h_t such that h_t solves
  $$h_t = \text{argmin}_h \sum_{n=1}^{N} u_n \cdot I[y_n \neq h(\mathbf{x}_n)].$$
  with the help of some base learner A_b.
  - Compute the weighted error $\epsilon_t = \frac{1}{\sum_{m=1}^N u_m} \sum_{n=1}^{N} u_n \cdot I[y_n \neq h_t(\mathbf{x}_n)]$ and the confidence
  $$\alpha_t = \frac{1}{2} \ln \frac{1-\epsilon_t}{\epsilon_t}$$
  - Emphasize the training examples that do not agree with h_t:
  $$u_n = u_n \cdot \exp\Big(-\alpha_t y_n h_t(\mathbf{x}_n)\Big).$$
- Output: combined function $H(\mathbf{x}) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})\right)$

In this problem, we will prove that AdaBoost can reach $E_{\text{in}}(H) = 0$ if T is large enough and every hypothesis h_t satisfies $\epsilon_t \leq \epsilon < \frac{1}{2}$.

(1) (8%) Let $U^{(t-1)} = \sum_{n=1}^{N} u_n$ at the beginning of the t-th iteration. What is $U^{(0)}$?

(2) (8%) According to the AdaBoost algorithm above, for $t \geq 1$, prove that

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

(3) (8%) By the result in (2), prove that $E_{\text{in}}(H) \leq U^{(T)}$.

(4) (8%) According to the AdaBoost algorithm above, for $t \geq 1$, prove that

$$U^{(t)} = U^{(t-1)} \cdot 2\sqrt{\epsilon_t(1-\epsilon_t)}.$$

(5) (8%) Using $0 \leq \epsilon_t \leq \epsilon < \frac{1}{2}$, for $t \geq 1$, prove that

$$\sqrt{\epsilon_t(1-\epsilon_t)} \leq \sqrt{\epsilon(1-\epsilon)}.$$

(6) (8%) Using $\epsilon < \frac{1}{2}$, prove that

$$\sqrt{\epsilon(1-\epsilon)} \leq \frac{1}{2} \exp\left(-2(\frac{1}{2}-\epsilon)^2\right).$$

(7) (8%) Using the results above, prove that

$$U^{(T)} \leq \exp\left(-2T(\frac{1}{2}-\epsilon)^2\right).$$

(8) (8%) Using the results above, argue that after $T = O(\log N)$ iterations, $E_{\text{in}}(H) = 0$.

7.2 Experiments with Adaptive Boosting (*)

(1) (6%) Prove that you can implement A_{ds} below in time $O(N \log N)$ instead of the brute-force implementation that takes $O(N^2)$.

(2) (10%) Implement the decision stump learning algorithm A_{ds}. That is, let

$$h_{s,i,\theta}(\mathbf{x}) = \text{sign}\Big(s \cdot (\mathbf{x})_i - \theta\Big),$$

where $s \in \{-1, +1\}$, $i \in \{1, 2, \ldots, d\}$, and $\theta \in \mathbb{R}$. Given a weighted training set $\mathcal{D} = \{(\mathbf{x}_n, y_n, u_n)\}_{n=1}^{N}$,

$$A_{ds}(\mathcal{D}) = \underset{h_{s,i,\theta}}{\text{argmin}} \sum_{n=1}^{N} u_n \cdot I\big[y_n \neq h_{s,i,\theta}(\mathbf{x}_n)\big].$$

Run the algorithm on the following set for training (with $u_n = \frac{1}{N}$ for all N):

`http://www.csie.ntu.edu.tw/~htlin/course/ml09fall/data/hw7_train.dat`

and the following set for testing:

`http://www.csie.ntu.edu.tw/~htlin/course/ml09fall/data/hw7_test.dat`

Let g be the decision function returned from A_{ds}. Report $E_{\text{in}}(g)$ and $E_{\text{out}}(g)$. Briefly state your findings.

(3) (20%) 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/ml09fall/data/hw7_train.dat`

and the following set for testing:

`http://www.csie.ntu.edu.tw/~htlin/course/ml09fall/data/hw7_test.dat`

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