Homework #1

TA in charge: Han-Hsing Tu

RELEASE DATE: 09/25/2008

DUE DATE: 10/02/2008, 4:00 pm IN CLASS

TA SESSION: 10/01/2008, 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.

1.1 Simplified No-Free-Lunch Theorem

Let $\mathcal{X} = \{x_1, x_2, \ldots, x_N, x_{N+1}, \ldots, x_{N+M}\}$ and $\mathcal{Y} = \{0, 1\}$ (binary classification). Here the set of training examples is $D = \Big\{(x_n, y_n)\Big\}_{n=1}^{N}$, where $y_n \in \mathcal{Y}$, and the set of test inputs is $\Big\{x_{N+m}\Big\}_{m=1}^{M}$. The *Off-Training-Set error* (OTS) with respect to an underlying target f and a hypothesis g is

$$OTS(g, f) = \frac{1}{M} \sum_{m=1}^{M} I\Big[g(x_{N+m}) \neq f(x_{N+m})\Big].$$

(1) (10%) Say $f(x) = 1$ for all x and $g(x) = \begin{cases} 1, & \text{for } x = x_n,\ n < N + \frac{M}{3} \\ 0, & \text{otherwise} \end{cases}$. What is $OTS(g, f)$?

(2) (10%) How many $f\colon \mathcal{X} \to \mathcal{Y}$ can "generate" D in a noiseless setting? (i.e. $f(x_n) = y_n$ for $n = 1, 2, \ldots, N$.)

(3) (10%) Among those f in (2), for a fixed g and a given integer k between 0 and M, how many of those f satisfies $OTS(f, g) = \frac{k}{M}$?

(4) (10%) For a fixed g, if all those f in (2) are equally likely in probability, what is $\mathcal{E}\{OTS(f, g)\}$?

(5) (10%) A determistic algorithm A is defined as a procedure that takes D as an input, and outputs a hypothesis g as the decision function. Argue that for any two deterministic algorithms A_1 and A_2,

$$\mathcal{E}\Big\{OTS\big(f, A_1(D)\big)\Big\} = \mathcal{E}\Big\{OTS\big(f, A_2(D)\big)\Big\}.$$

You have now proved that "in a noiseless setting (f generates D), for a fixed D, if all possible f are equally likely, any two deterministic algorithms are the same in terms of $\mathcal{E}\{OTS\}$."

1.2 Bins and Marbles

Consider a sample of 10 marbles drawn from a bin with red and green marbles. The probability that any marble we draw is red is π (independently). For $\pi = 0.05$, $\pi = 0.5$, and $\pi = 0.8$, we address the probability of getting no red marbles ($\nu = 0$) in the following cases.

(1) (20%) We draw only one such sample. Compute the probability that $\nu = 0$.

(2) (15%) We draw 1000 independent samples. Compute the probability that (at least) one of the samples has $\nu = 0$.

(3) (15%) Repeat (2) for 1000000 independent samples.

1.3 Perceptron Learning (*)

Implement the perceptron learning algorithm (PLR) taught in class. Assume that

$$g^{(t)}(x) = \left\langle w^{(t)}, x \right\rangle - \theta^{(t)}$$

is the decision function obtained at the t-th iteration. Run your algorithm for 1000 iterations on the following training set (each row represents a pair of (x_n, y_n). The first column is $(x_n)_1$, the second one is $(x_n)_2$, and the third one is y_n):

`http://www.csie.ntu.edu.tw/~htlin/course/ml08fall/data/hw1_3_train.dat`

(1) (20%) Plot the training error $\nu\big(g^{(t)}\big)$ as a function of t and briefly state your findings.

(2) (20%) Test those $g^{(t)}$ on the following test set:

`http://www.csie.ntu.edu.tw/~htlin/course/ml08fall/data/hw1_3_test.dat`

Plot the test error $\hat{\pi}\big(g^{(t)}\big)$ as a function of t and briefly state your findings.

(3) (20%) Plot the training examples and the decision boundary "$g^{(1000)}(x) = 0$" in the same figure. Use different symbols to distinguish examples with different y_n. Briefly state your findings.

1.4 Perceptron Learning on Noisy Examples (*)

In this problem, you will use

`http://www.csie.ntu.edu.tw/~htlin/course/ml08fall/data/hw1_4_train.dat`

for training, and

`http://www.csie.ntu.edu.tw/~htlin/course/ml08fall/data/hw1_4_test.dat`

for testing.

(1) (20%) Repeat (1) and (2) in Problem 1.3 with the data sets above and briefly state your findings.

(2) (30%) Implement an algorithm that runs PLR while keeping "in its pocket" the best $g^{(t)}$ so far in terms of $\nu\big(g^{(t)}\big)$. That is, the "pocket" hypothesis,

$$g_*^{(t)} = g^{(\tau)}, \text{ where } \nu\Big(g^{(\tau)}\Big) \leq \nu\Big(g^{(s)}\Big) \text{ for } 1 \leq s \leq t,$$

is stored in addition to $g^{(t)}$. Run this algorithm with the data sets above. Repeat (1) and (2) in Problem 1.3 with $g_*^{(t)}$ instead of $g^{(t)}$. Briefly state your findings.

The new algorithm is a basic version of the so-called *pocket* algorithm for perceptron learning. See the following reference for details:

Gallant, S. I. (1990). Perceptron-based learning algorithms. *IEEE Transactions on Neural Networks*, vol. 1, no. 2, pp. 179-191.