

58147 Machine Learning (Spring 2005)

Exercise 3 (Wednesday 16 February)

1. Calculate the value c_L for the Hellinger loss. Write out explicitly the condition for the prediction, as was done for square loss on page 64 of lecture notes.
2. Consider the Aggregating Algorithm for square loss, with $c = 2$ and $\eta = 1/2$. We know that, using the notation from page 64,

$$(y - \hat{y}_t)^2 \leq \Delta(y)$$

holds for $y \in \{-1, 1\}$ when $\eta = 1/2$ and $c = 2$. Show that it actually holds for all $y \in [-1, 1]$, using the same \hat{y} .

Hint: Write the condition as $f(y) \leq 1$ where $f(y) = \exp(\eta(y - \hat{y}_t)^2 - \eta\Delta(y))$. Show that $f''(y)$ is non-negative.

3. We generalise the Perceptron Algorithm by introducing a *learning rate* $\eta > 0$. The update becomes

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \eta\sigma_t y_t \mathbf{x}_t.$$

Further, we start the algorithm with $\mathbf{w}_1 = \mathbf{w}_{\text{init}}$ where the initial weights need not be zero. (Note that if we have $\mathbf{w}_{\text{init}} = \mathbf{0}$ then the learning rate does not affect the predictions $\text{sign}(\mathbf{w}_t \cdot \mathbf{x}_t)$.)

Assume some \mathbf{u} has margin 1 on the sample. Provide an alternative proof for the Perceptron Convergence Theorem by using

$$P_t = \frac{1}{2} \|\mathbf{u} - \mathbf{w}_t\|_2^2$$

as the potential function. The result should be that

$$\sum_{t=1}^T \sigma_t \leq \|\mathbf{u} - \mathbf{w}_{\text{init}}\|_2^2 X^2$$

for a suitable choice of η .

Hint: This is a fairly straightforward modification of the proof in the lecture notes. The value you get for η should be $1/c$ for the value c used in that proof.

4. One variant of the Perceptron would be to choose the new weight vector \mathbf{w}_{t+1} so that it has margin at least one but is as close to the old weight vector \mathbf{w}_t as possible.

In other words, let $H_t = \{\mathbf{w} \mid y_t \mathbf{w} \cdot \mathbf{x}_t \geq 1\}$. The proposed update can be written as

$$\mathbf{w}_{t+1} = \arg \min_{\mathbf{w} \in H_t} \|\mathbf{w} - \mathbf{w}_t\|_2.$$

Give a closed-form solution for \mathbf{w}_{t+1} . How does the distance from \mathbf{w}_t to the hyperplane $\{\mathbf{w} \mid \mathbf{w} \cdot \mathbf{x}_t = 1\}$ appear in your solution?

See the other side!

5. The purpose of this problem is to learn how to use R or Matlab for simple learning experiments. R is free software and should be available in all CS Linux machines (`/opt/R/bin/R`). You can download the software and documentation for our own machine at <http://www.r-project.org/>. Matlab is commercial software and available on some CS Linux machines (`/opt/matlab/bin/matlab`) and some IT Department machines (like vesuri).

Use matrix operations where possible. On R, you could use functions such as `matrix()`, `runif()`, `sum()`, `seq()`, `plot.default()`, `points()` and `%*%` (matrix product). On matlab, see `zeros()`, `rand()`, `sum()`, `linspace()`, `plot()` and `*`. Both programs have good online help.

Using R or Matlab, create a dataset as follows. Create a 200×2 matrix X of 200 instance vectors drawn uniformly at random from $[-1, 1]^2$. Label the instances using a linear classifier $(\mathbf{u}, b) = ((0.5, 0.5), 0)$ to get a 200×1 label vector \mathbf{y} . Split the data into two parts: the first 100 examples are the *training set* and the remaining 100 the *test set*.

Implement the Perceptron algorithm. Run it on the training set, iterating until there are no more mistakes, to obtain a hypothesis \mathbf{w} . (Assume fixed $b = 0$.) How many mistakes did the algorithm make? Use the classifier \mathbf{w} on the test data (without learning); how many mistakes does it make? Visualise the training data, final hypothesis and some of the intermediate hypotheses.

Repeat the experiment using 50-dimensional instances. For labeling the data, use the classifier $(\mathbf{w}, b) = ((0.5, 0.5, 0, \dots, 0), 0)$ How do the mistake counts change?