MASSACHUSETTS INSTITUTE OF TECHNOLOGY
Department of Electrical Engineering and Computer Science
6.036—Introduction to Machine Learning
Spring Semester 2016

Assignment 2, Issued: Friday, Feb. 12 Due: Friday, Feb. 19

Perceptron Convergence Rates

1. **Mistake Bounds**

   Consider a set of n labeled training data points $\{(x^{(t)}, y^{(t)}), t = 1, \ldots, n\}$, where each $y^{(t)} \in \{-1, +1\}$ is the label for the point $x^{(t)}$, a d-dimensional vector defined as follows:

$$x_i^{(t)} = \begin{cases} \cos(\pi t) & i = t \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

   Recall the no-offset perceptron algorithm, and assume that $\theta \cdot x = 0$ is treated as a mistake, regardless of label. Assume in this problem that $n = d$ and that we initialize $\theta = 0$ as usual.

   (a) Consider the case $d = 2$. For each assignment of labels $y^{(1)}, y^{(2)}$, sketch the θ that results by running the perceptron algorithm until convergence. Convince yourself that the algorithm will make 2 updates for any ordering (and labeling) of the feature vectors.

   (b) Now consider the general case. Show that the no-offset perceptron algorithm will make exactly d updates to θ, regardless of the order in which we present the feature vectors, and regardless of how these vectors are labeled, i.e., no matter how we choose $y^{(t)}$, $t = 1, \ldots, t$.

   (c) What is the vector θ that the perceptron algorithm converges to based on this d-dimensional training set? Does θ depend on the ordering? How about the labeling?

   (d) Is the number of perceptron algorithm mistakes made, d, a violation of the $\frac{R^2}{\gamma^2}$ bound on mistakes we proved in class? Why or why not (Hint: $\|x^{(t)}\| = 1$ for all t, what is $\|\theta\|$?)

Passive-aggressive algorithm

We saw in lectures that the passive-aggressive (PA) algorithm (without offset) responds to a labeled training example (x, y) by finding θ that minimizes

$$\frac{\lambda}{2}\|\theta - \theta^{(k)}\|^2 + \text{Loss}_h(y\theta \cdot x) \tag{2}$$

where $\theta^{(k)}$ is the current setting of the parameters prior to encountering (x, y) and $\text{Loss}_h(y\theta \cdot x) = \max\{0, 1 - y\theta \cdot x\}$ is the hinge loss. We could replace the loss function with something else (e.g., the zero-one loss). The form of the update is similar to the perceptron algorithm, i.e.,

$$\theta^{(k+1)} = \theta^{(k)} + \eta\, yx \tag{3}$$

but the real-valued step-size parameter η is no longer equal to one; it now depends on both $\theta^{(k)}$ and the training example (x, y).

2. **Loss functions and decision boundaries**

   (a) Consider minimizing the function in eq. 2 with the hinge loss. What happens as the value of λ increases? If the λ is large, should the step-size of the algorithm (η) be large or small? Explain.

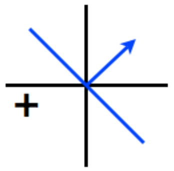
   (b) Consider minimizing the function in eq. 2 and the setting of our decision boundary plotted below. We ran our PA algorithm on the next data point in our sequence - a positively-labeled vector (indicated with a $+$). We plotted the results of our algorithm after the update, by trying out a few different variations of loss function and λ as follows:

   1) hinge loss and a large λ
   2) hinge loss and a small λ
   3) 0-1 loss and a large λ
   4) 0-1 loss and a small λ

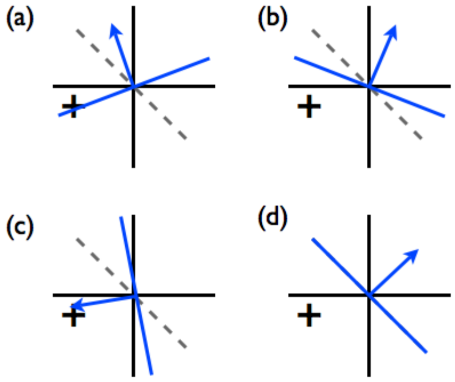
   Note that for a large λ, the $\frac{\lambda}{2}\|\theta - \theta^{(k)}\|^2$ term dominates and for a small λ, the $\text{Loss}_h(y\theta \cdot x)$ term dominates.

   Unfortunately, we forgot to label our result files, and want to find out which variation of algorithm corresponds to which update. Please help match up the 4 variations above with the resulting decision boundaries plotted in a-d below (the dotted lines correspond to the previous decision boundary, and the solid blue lines correspond to the new decision boundary; also, note that these are just sketches which ignore any changes to the magnitude of θ).

setting before update:



possible settings after update:



3. **Update equation, effect**

   (a) Suppose $\text{Loss}_h(y\theta^{(k+1)} \cdot x) > 0$ after the update ($\theta^{(k)} \rightarrow \theta^{(k+1)}$). What is the value of η that lead to this update? (Hint: you can simplify the loss function in this case).

   (b) Suppose we replace the perceptron algorithm in problem 1 with the passive aggressive algorithm. Will the number of mistakes made by the passive aggressive algorithm depend on feature vector ordering? Explain.