

Homework 2 - Theory

*Lecture: Prof. Adam Klivans**Keywords: Perceptron, SGD, Boosting*

Instructions: Please either typeset your answers (L^AT_EX recommended) or write them very clearly and legibly and scan them, and upload the PDF on edX. Legibility and clarity are critical for fair grading.

1. **[10 points]** Consider running the Perceptron algorithm on a training set S arranged in a certain order. Now suppose we run it with the same initial weights and on the *same* training set but in a different order, S' . Does Perceptron make the same number of mistakes? Does it end up with the same final weights? If so, prove it. If not, give a counterexample, i.e. an S and S' where order matters.
2. **[10 points]** We have mainly focused on squared loss, but there are other interesting losses in machine learning. Consider the following loss function which we denote by $\phi(z) = \max(0, -z)$. Let S be a training set $(x^1, y^1), \dots, (x^m, y^m)$ where each $x^i \in \mathbb{R}^n$ and $y^i \in \{-1, 1\}$. Consider running stochastic gradient descent (SGD) to find a weight vector w that minimizes $\frac{1}{m} \sum_{i=1}^m \phi(y^i \cdot w^T x^i)$. Explain the explicit relationship between this algorithm and the Perceptron algorithm. Recall that for SGD, the update rule when the i^{th} example is picked at random is

$$w_{\text{new}} = w_{\text{old}} - \eta \nabla \phi(y^i w^T x^i).$$

Note: You do not need to be overly concerned about the discontinuity at $\phi(0)$, so you can ignore this when calculating the gradient for this problem.

3. **[6 points]** Here we will give an illustrative example of a weak learner for a simple concept class. Let the domain be the real line, \mathbb{R} , and let \mathcal{C} refer to the concept class of “3-piece classifiers”, which are functions of the following form: for $\theta_1 < \theta_2$ and $b \in \{-1, 1\}$, $h_{\theta_1, \theta_2, b}(x)$ is b if $x \in [\theta_1, \theta_2]$ and $-b$ otherwise. In other words, they take a certain Boolean value inside a certain interval and the opposite value everywhere else. For example, $h_{10, 20, 1}(x)$ would be $+1$ on $[10, 20]$, and -1 everywhere else. Let \mathcal{H} refer to the simpler class of “decision stumps”, i.e. functions $h_{\theta, b}$ such that $h(x)$ is b for all $x \leq \theta$ and $-b$ otherwise.
 - (a) Show formally that for any distribution on \mathbb{R} (assume finite support, for simplicity; i.e., assume the distribution is bounded within $[-B, B]$ for some large B) and any unknown labeling function $c \in \mathcal{C}$ that is a 3-piece classifier, there exists a decision stump $h \in \mathcal{H}$ that has error at most $1/3$, i.e. $\mathbb{P}[h(x) \neq c(x)] \leq 1/3$.
 - (b) Describe a simple, efficient procedure for finding a decision stump that minimizes error with respect to a finite training set of size m . Such a procedure is called an empirical risk minimizer (ERM).
 - (c) Give a short intuitive explanation for why we should expect that we can easily pick m sufficiently large that the training error is a good approximation of the true error, i.e. why we can ensure generalization. (Your answer should relate to what we have gained in

going from requiring a learner for \mathcal{C} to requiring a learner for \mathcal{H} .) This lets us conclude that we can weakly learn \mathcal{C} using \mathcal{H} .

4. **[10 points]** Consider an iteration of the AdaBoost algorithm (using notation from the video lecture on Boosting) where we have obtained classifier h_t . Show that with respect to the distribution D_{t+1} generated for the next iteration, h_t has accuracy exactly $1/2$.