# **Homework Two**, due Tue 2/7 CSE 250B

1. *Perceptron over the line.* As we discussed in class, the perceptron will not converge if the data are not linearly separable. However, it is still possible to bound the number of mistakes the perceptron makes in some circumstances. Assume we have $n$ datapoints $(x_i, y_i)$ with $\|x_i\| = 1$.

   Assume also that all but $k$ of the points are linearly separable with margin $\gamma > 0$; that is, there exists a unit vector $w^*$ such that for at least $n - k$ of the points,

   $$y_i(w^* \cdot x_i) \geq \gamma.$$

   In this problem, you will prove that *in a single iteration*, the perceptron algorithm makes at most $T_0 = 1/\gamma^2 + k(1 + 2/\gamma)$ mistakes. (This bound is not tight, but it is convenient to use.)

   (a) First start by showing that if $T$ updates are performed during the single round of perceptron, then:

   $$w_T \cdot w^* \geq T\gamma - k(1 + \gamma).$$

   Also give an upper bound on $\|w_T\|^2$.

   (b) These upper and lower bounds specify a quadratic inequality in $\sqrt{T}$; any plausible value of $T$ must lie within the solution region of this inequality. Show that this solution region is strictly less than $T_0$ (there is no need to explicitly solve the inequality). Thus we can conclude $T \leq T_0$.

2. *Perceptron primary (challenge 2).* A voted-perceptron consists of $p$ individual perceptrons $w_i \in \mathbf{R}^d$, each with a weight $a_i$. A point $x$ is classified according to

   $$\operatorname{sgn}\left(\sum_{i=1}^{p} a_i \operatorname{sgn}(w_i \cdot x)\right).$$

   (We're assuming here that all data points have been preprocessed by adding an extra feature which is identically 1; hence we are only considering individual perceptrons which pass through the origin.)

   The particular form of a voted-perceptron gives it a more complex decision boundary than a single perceptron. Your challenge is to develop an algorithm for learning a voted-perceptron, given a data set and a value of $p$.

   (a) Design an algorithm which takes as input a labeled data set $(X, y)$ (with $x_i \in \mathbf{R}^d$ and $y_i \in \{+1, -1\}$), and also an integer $p$; and returns a voted perceptron $(W, a)$ with $p$ hyperplanes. You should not expect the data to be linearly separable.

   First give a clear and precise high-level description of your algorithm, and then write a Matlab/Octave function implementing it:

   ```
   [W,a] = votedperc(X,y,p)
   ```

   (b) While the study of prime numbers has fascinated mathematicians and mystics for millenia, suprisingly few have recognized the potential insight gained by applying perceptron-based classifiers to primality testing. It is time to turn this unturned stone.

   To this end, create training and test sets from the OCR data in which the prime digits $2, 3, 5, 7$ are combined into a single $+1$ category, and the composites $4, 6, 8, 9$ form the $-1$ category. Remember to add the extra identically-one feature to the data points.

   Run your algorithm on the training data for $p = 1, 2, 4, 8, 16$. In each case, compute the misclassification rate on the test data. Plot this rate as a function of $p$.

3. *More high-dimensional weirdness.* It is well-known that in $\mathbf{R}^d$, you can have at most $d$ vectors which are at right-angles to each other. That is, if unit vectors $x_1, \ldots, x_n \in \mathbf{R^d}$ have the property that:

$$x_i \cdot x_j \quad = \quad 0 \quad \text{for } i \neq j$$

then one can be certain that $n \leq d$. In this problem, you will show that for large $d$, it is nonetheless possible to find exponentially many vectors which are *almost* at right-angles to each other!

To this end, we will pick $n$ unit vectors $x_1, x_2, \ldots, x_n \in \mathbf{R}^d$ as follows:

- Write $x_i = (x_{i1}, x_{i2}, \ldots, x_{id})$.
- Pick each $x_{ik}$ to be either $1/\sqrt{d}$ or $-1/\sqrt{d}$, each with probability $1/2$ and independently of anything else.

(You might want to convince yourself that these $x_i$'s are actually unit vectors.)

(a) What is $\mathbf{E}(x_{ik}x_{jk})$ for $i \neq j$? What is $\mathbf{E}(x_i \cdot x_j)$ for $i \neq j$?

(b) Think of $c$ as a very small constant, like 0.01. Show that for any $i \neq j$,

$$\mathbf{P}(|x_i \cdot x_j| > c) \;\leq\; 2e^{-c^2 d/2}.$$

(c) Suppose $n$ is exponentially large in $d$, specifically $n = \frac{1}{2}e^{c^2 d/4}$. Show that with probability at least $3/4$, the vectors $x_1, \ldots, x_n$ are *all* almost at right-angles to each other, that is,

$$|x_i \cdot x_j| \leq c \quad \text{for all } i \neq j.$$