Homework 1

1 Mistake Bounds

In this mistake bound model (see notes), show that halving algorithm may not be optimal. In particular, show that there exists a concept class $\mathcal{C}$ and an algorithm $\mathcal{A}$ such that $\mathrm{mistake}(\mathcal{A}, \mathcal{C})$ is lower than the mistake bound of the halving algorithm, $\mathrm{mistake}(\text{Halving}, \mathcal{C})$. On the other hand, does there exist a concept class $\mathcal{C}$, where the cardinality of $\mathcal{C}$ is some arbitrary power of 2, such that all algorithms make $\log |\mathcal{C}|$ mistakes, i.e. so the halving algorithm bound is the best possible (without taking into account the details of $\mathcal{C}$)?

Now suppose we are in a multi-class setting, where $\mathcal{Y} = \{1, 2, \ldots k\}$. At each round t, we see x_t and make some prediction, say $h_t(x_t)$, which is in $\mathcal{Y}$. At the end of the round, y_t is revealed and $\mathcal{A}$ makes a mistake if $h_t(x_t) \neq y_t$. Suppose the labels were actually produced by some function f in a given concept class $\mathcal{C}$, where each hypothesis in $\mathcal{C}$ is a mapping from $\mathcal{X}$ to $\mathcal{Y}$. In the multi-class setting, define:

$$\mathrm{mistake}(\mathcal{A}, \mathcal{C}) := \max_{f \in \mathcal{C}, T, x_{1:T}} \sum_{t=1}^{T} \mathbf{1}\left[h_t(x_t) \neq f(x_t)\right] \ .$$

Provide an algorithm $\mathcal{A}$ which sharply bounds $\mathrm{mistake}(\mathcal{A}, \mathcal{C})$ which is in terms of only the cardinality of $\mathcal{C}$.

2 The Multi-Class Perceptron

On round t, the learner is given an instance vector $\mathbf{x}_t \in \mathbb{R}^d$ and is required to predict a label out of a set of k predefined labels which we denote by $[k] = \{1, \ldots, k\}$. We denote the predicted label by $\hat{y}_t$. Again, the learner's goal is to minimize the number of prediction mistakes, M, it makes along its run, where:

$$M = \sum_{t=1}^{T} \mathbf{1}\left[\hat{y}_t \neq y_t\right] \ .$$

Again, the prediction of the algorithm at round t is determined by a hypothesis, $h_t : \mathbb{R}^d \to [k]$, where h_t is taken from a class of hypotheses $\mathcal{H}$. Here, we focus on the class of linear hypotheses. Formally, each $h \in \mathcal{H}$ is parameterized by a matrix of weights $W \in \mathbb{R}^{k \times d}$ and is defined to be:

$$h(\mathbf{x}) \ = \ \mathrm{argmax}_{j \in [k]} \left(W\mathbf{x}\right)_j \ ,$$

where $(W\mathbf{x})_j$ is the jth element of the vector obtained by multiplying the matrix W with the vector $\mathbf{x}$. Since each hypothesis is parameterized by a weight matrix, we refer to a matrix W also as a hypothesis. To evaluate the performance of a weight matrix W on an example $(\mathbf{x}, y)$ we check whether W makes a prediction mistake, namely determine if $\arg\max_j (W\mathbf{x})_j \neq y$.

This problem adapts the Perceptron algorithm for multiclass prediction (this adaptation is called Kesler's construction). We denote by W^t the weight matrix used by the Perceptron at round t. The Perceptron starts with the all zero matrix $W^1 = \mathbf{0}$ and updates it as follows

$$W^{t+1} \;=\; W^t + U^t \ ,$$

where $U^t \in \mathbb{R}^{k\times d}$ is the matrix defined by

$$U^t_{r,j} \;=\; x_{t,j}\left(\mathbf{1}\left[y_t = r\right] - \mathbf{1}\left[\hat{y}_t = r\right]\right) .$$

In other words, if there is no prediction mistake (i.e. $y_t = \hat{y}_t$), then there is no update (i.e. $W^{t+1} = W^t$), and if there is a prediction mistake, then $\mathbf{x}_t$ is added to the y_tth row of the weight matrix and subtracted from the $\hat{y}_t$th row of the matrix.

Now let us generalize the notion of the margin to the multiclass setting. In particular, for our sequence of examples $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_T, y_T)$, we would like to say that they are linearly separable with some margin γ. Assume there exists a $W^\star$ such that for all t, and for all $r \neq y_t$:

$$(W^\star \mathbf{x}_t)_{y_t} - (W^\star \mathbf{x}_t)_r \geq \gamma \, .$$

This difference $(W\mathbf{x}_t)_{y_t} - (W\mathbf{x}_t)_r$ is a generalization of the notion of *margin* from binary classification.

Now define the Frobenius norm as:

$$\|W^\star\|_F^2 = \sum_{r=1}^{k}\sum_{j=1}^{d}(W^\star_{i,j})^2 \ ,$$

which is just a more general notion of the Euclidean norm for a matrix.

Prove that the number of prediction mistakes of the multiclass Perceptron is at most,

$$M \;\leq\; \frac{2\|W^\star\|_F^2 ||x_{1:t}||^2}{\gamma^2} \ .$$

Here is an outline of the proof, which you are free to follow. It is convenient to use the notation:

$$\langle W^\star, W^t\rangle := \sum_{r=1}^{k}\sum_{j=1}^{d} W^\star_{r,j} W^t_{r,j} \ .$$

Again, the key idea of the proof is to look at how this quantity evolves over time. Define m_t to be 1 if a mistake occurs at time t and 0, else. Also note that $M = \sum_t m_t$. First provide the lower bound:

$$\langle W^\star, W^{t+1}\rangle \geq \langle W^\star, W^t\rangle + \gamma m_t$$

which implies that

$$\langle W^\star, W^{T+1} \rangle \geq \gamma M$$

Using Cauchy-Schwartz inequality we have

$$\langle W^\star, W^{T+1} \rangle \;\leq\; \|W^\star\|_F \, \|W^{T+1}\|_F \;\; .$$

Prove that:

$$\|W^{T+1}\|_F^2 \leq 2||x_{1:T}||^2 M$$

Now complete the proof.

3 Perceptron Can Make Too Many Mistakes

Let us define a d-dimensional data set

$$S(d) = \{(x_1, y_1), \ldots, (x_d, y_d)\}$$

as follows. Let $\mathbf{e}_i \in \mathbb{R}^d$ denote the unit vector with 1 in the ith position and zeroes elsewhere. For i in $\{1, \ldots, d\}$, define

$$x_i = \begin{cases} -\sum_{j=1}^{i-1} \mathbf{e}_j + \mathbf{e}_i & i \text{ is odd} \\ \sum_{j=1}^{i-1} \mathbf{e}_j - \mathbf{e}_i & i \text{ is even} \end{cases} ,$$

and

$$y_i = \begin{cases} +1 & i \text{ is odd} \\ -1 & i \text{ is even} \end{cases} .$$

- Run the perceptron algorithm on $S(d)$ by cycling through the data set until you find a $w \in \mathbb{R}^d$ that separates the positive examples from the negative ones. Let the number of mistakes made be $M(d)$. Plot $M(d)$ as a function of d for $d = \{1, \ldots, 8\}$.
- Prove that no matter which order we cycle through the data set $S(d)$, the perceptron algorithm will make at least $\Omega(2^d)$ mistakes before finding a separating hyperplane.

4 Learning Sparse Disjunctions

Suppose $\mathcal{X} = \{0, 1\}^d$. Let $i_1, \ldots, i_k$ be k indices such that the label y_t of x_t is $+1$ iff

$$x_{i_1} \vee x_{i_2} \vee \ldots \vee x_{i_k} = 1 \; .$$

- Show how to get a mistake bound of $O(k \ln d)$ using the halving algorithm.
- Define $\tilde{x}_t := (x_t, -1) \in \mathbb{R}^{d+1}$. Show that there is a non-negative $w^* \in \mathbb{R}^{d+1}$ with $\|w^*\|_1 = O(k)$ that gets a margin of 1 on the data set consisting of examples $(\tilde{x}_t, y_t)$.

- In the analysis of Winnow done in the class, we assumed we know $\|w^*\|_1$ to set η. Show that a similar analysis works even when we know only an *upper bound* W for $\|w^*\|_1$ and derive a mistake bound of

$$\frac{2\|x_{1:T}\|_\infty^2 \cdot W^2}{\gamma^2} \ln d \, .$$

  Conclude that this implies a mistake bound of $O(k^2 \ln d)$ in the setting of this problem.

We point out, however, that the original Winnow algorithm proposed by Littlestone is slightly different from our version and enjoys a mistake bound of $O(k \ln d)$ for this problem.

5 Gradient Descent with Less Apriori Knowledge

The gradient descent algorithm we provided in class assumed you knew (apriori) an upper bound on the maximal gradients of the cost functions and the time T. In particular, we set η as $\eta = \frac{D_2}{G_2}\sqrt{\frac{1}{T}}$, where it was assumed we knew a value of G_2 such that

$$||\nabla c_t(w_t)||_2 \leq G_2 \quad .$$

In this problem, we consider using the variable learning rate:

$$\eta_t = \frac{D_2}{\sqrt{\sum_{\tau=1}^{t} ||\nabla_t||^2}}$$

where again we are using the notation $\nabla_t = \nabla c_t(w_t)$. The update we follow, for $t \geq 1$ is:

$$w_{t+1} = \Pi_D[w_t - \eta_t \nabla_t]$$

where $\Pi_D[w]$ is the projection of w back into D,

Prove that the regret, for all times T, is bounded as follows:

$$R_T \leq 2D_2 \sqrt{\sum_{t=1}^{T} ||\nabla_t||^2}$$

Hence, we are able to run the algorithm with no knowledge of the upper bound G_2 nor the end time T (in fact, the guarantee holds for all times T). Notice that this bound is clearly less than $2D_2G_2\sqrt{T}$, but it could be much better.

We will proceed through a series of steps:

5.1 A Time Dependent Lemma

Let w^* be an arbitrary point in D. Prove that the decisions of the GD algorithm satisfy:

$$
\begin{aligned}
R_T &\leq \sum_{t=1}^{T} \nabla_t \cdot (w_t - w^*) \\
&\leq \frac{\|w_1 - w^*\|^2}{2\eta_1} - \frac{\|w_{T+1} - w^*\|^2}{2\eta_T} + \frac{1}{2}\sum_{t=1}^{T-1}\left(\frac{1}{\eta_{t+1}} - \frac{1}{\eta_t}\right)\|w_{t+1} - w^*\|^2 \\
&\qquad + \sum_{t=1}^{T}\frac{\eta_t}{2}||\nabla_t||^2 \\
&\leq \frac{D_2^2}{2\eta_1} + \frac{D_2^2}{2}\sum_{t=1}^{T-1}\left(\frac{1}{\eta_{t+1}} - \frac{1}{\eta_t}\right) + \sum_{t=1}^{T}\frac{\eta_t}{2}||\nabla_t||^2 \\
&= \frac{D_2^2}{2\eta_T} + \sum_{t=1}^{T}\frac{\eta_t}{2}||\nabla_t||^2
\end{aligned}
$$

This statement is a generalization of the statement we used for the fixed learning rate case.

5.2 A Technical Lemma

The following technical lemma is useful in the rest of proof. For any sequence of non-negative numbers $a_1, a_2, \dots a_T$ (with $a_1 > 0$), prove that:

$$\sum_{t=1}^{T}\frac{a_t}{\sqrt{\sum_{\tau=1}^{t} a_\tau}} \leq 2\sqrt{\sum_{t=1}^{T} a_t}$$

Hint: We proved this by induction and by using the fact that $\sqrt{1-x} \leq 1 - \frac{x}{2}$ for $0 \leq x \leq 1$. Feel free to prove this as you like.

5.3 Completing the Proof

Using this technical lemma, bound the second term above as follows:

$$\sum_{t=1}^{T}\eta_t||\nabla_t||^2 \leq 2D_2\sqrt{\sum_{t=1}^{T}||\nabla_t||^2}$$

Now complete the proof.