# Homework 3 Solutions

Each problem is worth 10 points. The average score was 16.7 out of 20.

**Problem 1.**

Let $x^{(i)}$ be the $i^{\text{th}}$ input misclassified by the perceptron, and let $w^{(i)}$ be the weight vector after the $i^{\text{th}}$ update. Thus, $w^{(i)} = x^{(1)} + x^{(2)} + \cdots + x^{(i)}$.

**(a)** We have $w^{(k)} = x^{(1)} + x^{(2)} + \cdots + x^{(k)}$. For the unknown vector $v$ whose existence is guaranteed by our second assumption, we have:

$$\begin{aligned} v \cdot w^{(k)} &= v \cdot x^{(1)} + v \cdot x^{(2)} + \cdots + v \cdot x^{(k)} \\ &\geq k\delta \end{aligned}$$

By Cauchy's inequality, we have $\|v\|^2 \, \|w^{(k)}\|^2 \geq (v \cdot w^{(k)})^2$. Combining this with the above inequality, we obtain:

$$\|v\|^2 \, \|w^{(k)}\|^2 \geq (v \cdot w^{(k)})^2 \geq k^2\delta^2$$

We therefore obtain the following lower bound on $\|w^{(k)}\|^2$:

$$\|w^{(k)}\|^2 \geq \frac{k^2\delta^2}{\|v\|^2}$$

**(b)** We have $w^{(i)} = w^{(i-1)} + x^{(i)}$. Note that $w \cdot x \leq 0$ for each input $x$ which is misclassified. We therefore have $w^{(i-1)} \cdot x^{(i)} \leq 0$ $(\forall i)$. We further note that $\|x^{(i)}\|^2 < M$ $(\forall i)$ by our first assumption. We can now write:

$$\begin{aligned} \|w^{(i)}\|^2 &= w^{(i)} \cdot w^{(i)} \\ &= (w^{(i-1)} + x^{(i)}) \cdot (w^{(i-1)} + x^{(i)}) \\ &= w^{(i-1)} \cdot w^{(i-1)} + 2w^{(i-1)} \cdot x^{(i)} + x^{(i)} \cdot x^{(i)} \\ &\leq w^{(i-1)} \cdot w^{(i-1)} + x^{(i)} \cdot x^{(i)} \\ &= \|w^{(i-1)}\|^2 + \|x^{(i)}\|^2 \\ &< \|w^{(i-1)}\|^2 + M \end{aligned}$$

Thus, $\|w^{(k)}\|^2 < kM$.

**(c)** Combining the inequalities from parts (a) and (b), we have:

$$\frac{k^2\delta^2}{\|v\|^2} \leq \|w^{(k)}\|^2 < kM$$

Therefore, $k < \frac{M\|v\|^2}{\delta^2}$. Thus, the number of updates is bounded, and so the perceptron algorithm converges.

**Problem 2.**

**(a)** The rank of the tree given in the Kearns/Vazirani text is 3.

**(b)** We show that the class of functions computed by rank $r$ decision trees is included in the class of functions computed by $r$-decision lists. We do this by induction on the height $h$ of a decision tree. We define the height of a decision tree to be the length of the longest simple path from the root decision node to any leaf decision node. For example, the height of the decision tree given in the book is 5.

**Base case:** $(h = 1)$ A height 1 decision tree corresponds to tree with at most 1 decision node. This decision tree has rank at most 1. (In fact, the rank of a decision tree is always at most the height of the tree.) It is easy to verify that such a decision tree can always be expressed as a 1-decision list.

**Inductive hypothesis:** We assume that for any decision tree of height less than $h$, the class of functions computed by rank $r$ decision trees is included in the class of functions computed by $r$-decision lists. We want to show this is also true for decision trees of height $h$.

**Inductive step:** Suppose decision tree $T$ has height $h$ and rank $r$, and its root node contains the decision $x$. Moreover, suppose that the left subtree has rank $r_L$ and the right subtree has rank $r_R$. Since both of these subtrees have height less than $h$, by the inductive hypothesis we know that the left subtree can be expressed as a $r_L$-decision list L, and the right subtree can be expressed as a $r_R$-decision list R. Without loss of generality, let's assume that $r_L \leq r_R$. Suppose that $L$ is:

$$(T_1, b_1), (T_2, b_2), \ldots, (T_k, b_k), b$$

Each $T_i$ is a conjunction of at most $r_L$ literals, and each $b_i$ is either 0 or 1, and $b$ is a default bit. Similarly, suppose $R$ is

$$(T'_1, b'_1), (T'_2, b'_2), \ldots, (T'_l, b'_l), b'$$

Each $T'_i$ is a conjunction of at most $r_R$ literals. Then the decision tree $T$ can be expressed as the following decision list:

$$(\overline{x}T_1, b_1), (\overline{x}T_2, b_2), \ldots, (\overline{x}T_k, b_k), (\overline{x}, b), (T'_1, b'_1), (T'_2, b'_2), \ldots, (T'_l, b'_l), b'$$

Note that terms $T'_1, T'_2, \ldots, T'_l$ are only evaluated when $x = 1$, since we have the decision "if $\overline{x}$ then $b$" before these terms in the decision list.

Let $r' = max(r_L + 1, r_R)$. Then this is a $r'$-decision list, since the decisions corresponding to decision list $L$ have one more literal, and the decisions corresponding to decision list $R$ stay the same. Now, if $r_L = r_R$, then the decision tree $T$ has rank $r = r_L + 1 = r_R + 1$, and $r' = r_L + 1 = r$, and thus we have a $r$-decision list. On the other hand, if $r_L < r_R$, then the decision tree $T$ has rank $r = r_R$, and $r' \leq r_R = r$, and thus the decision list above is a $r$-decision list.

This proves the inductive step, and we have shown that rank $r$ decision trees can be expressed as $r$-decision lists. Thus, we can PAC-learn rank $r$ decision trees by $r$-decision lists.