# CS 475 Machine Learning: Homework 3
## Non-linear Methods
### Solution

## 1 Analytical (50 points)

**1) Decision Trees (12 points)** Consider a binary classification task with the following set of training examples:

| $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | classification |
|---|---|---|---|---|---|
| 0 | 1 | 1 | -1 | -1 | + |
| 0 | 1 | 1 | -1 | -1 | + |
| 0 | 1 | 1 | 1 | 1 | - |
| 0 | -1 | 1 | 1 | 1 | + |
| 0 | -1 | 1 | -1 | -1 | - |
| 0 | -1 | 1 | -1 | -1 | - |

(a) Can this function be learned using a decision tree? If so, provide such a tree (describe each node in the tree). If not, prove it.

(3 points) There is no overlap in samples. Thus we can always find a decision tree with perfect classification.

Example tre: We can use $x_2$ as the first split. Therefore Samples 1-3 are selected a subset, Samples 4-6 are selected another subset. For the first subset, we use $x_4$ as the second split. Therefore Samples 1-2 are selected as a subset with positive labels, and Sample 3 is selected as another subset with negative labels. A similar split can be made to the second subset, and we find a decision tree with perfect classification.

(b) Can this function be learned using a logistic regression classifier? If yes, give some example parameter weights. If not, why not.

(3 points) No. The data are not linearly separable.

(c) What is the entropy of the labels in the training data?

(3 points) We consider the label $Y$ as a Bernoulli random variable, and then have

$$H(Y) = -P(Y=1)\log(P(Y=1)) - P(Y=-1)\log(P(Y=-1)) = \log 2$$

(d) What is the information gain of $x_2$ relative to labels?

(3 points) We consider $X_2$ as a Bernoulli random variable, and then have

$$\begin{aligned}
H(Y|X_2) &= P(Y=1, X_2=1)\log\frac{P(X_2=1)}{P(Y=1,X_2=1)} \\
&+ P(Y=1, X_2=-1)\log\frac{P(X_2=-1)}{P(Y=1,X_2=-1)} \\
&+ P(Y=-1, X_2=1)\log\frac{P(X_2=1)}{P(Y=-1,X_2=1)} \\
&+ P(Y=-1, X_2=-1)\log\frac{P(X_2=-1)}{P(Y=-1,X_2=-1)} \\
&= \frac{1}{3}\log\frac{3}{2} + \frac{1}{6}\log 3 + \frac{1}{6}\log 3 + \frac{1}{3}\log\frac{3}{2} \\
&= \frac{2}{3}\log 3 - \log 2 + \frac{1}{3}\log 3 = \log 3 - \log 2.
\end{aligned}$$

Thus, we have

$$IG(Y|X) = H(Y) - H(Y|X) = 2\log 2 - \log 3.$$

**2) Decision Tree (12 points)** Let's investigate the accuracy of decision tree learning. We start by constructing a unit cube ($[0;1]\times[0;1]\times[0;1]$). We select $n$ samples from the cube, each with a binary label ($+1$ or $-1$), such that no two samples share either $x$, $y$, or $z$ coordinates. Each feature can be used multiple times in a decision tree. At each node we can only conduct a binary threshold split using one single feature.

(a) Prove that we can find a decision tree of depth at most $\log_2 n$, which perfectly labels all $n$ samples.

(6 points) Let us consider the first layer of the tree. Since all samples have no overlap at any coordinate, regardless the coordinate we select, we can always split the data into two sets with equal sizes by choosing a proper threshold. We then consider the second layer of the tree. Similar to the first layer, we can always split each subset of data into two smaller subsets with equal sizes by choosing a proper threshold. By induction eventually, we only need $\log_2 n$ layers in total such that all samples are divided in to $n$ subsets, where each one corresponds to one single sample. Thus, we perfectly labels all $n$ samples.

(b) If the samples can share either two coordinates but not all three, can we still learn a decision tree which perfectly labels all $n$ samples with the same depth as (a)? Why or why not?

(6 points) No. We can construct such a counter example:

| $x_1$ | $x_2$ | $x_3$ | classification |
|---|---|---|---|
| 0.4 | 0.1 | 0.1 | + |
| 0.4 | 0.2 | 0.2 | - |
| 0.1 | 0.4 | 0.3 | + |
| 0.2 | 0.4 | 0.5 | - |
| 0.3 | 0.3 | 0.4 | + |
| 0.5 | 0.5 | 0.4 | - |
| 0.6 | 0.6 | 0.6 | + |
| 0.7 | 0.7 | 0.7 | - |

There are no overlapped samples, but regardless the coordinate we choose, we cannot split the data equally into two subsets with 4 samples.

**3) Adaboost (14 points)** There is one good example at $x = 0$ and two negative examples at $x = \pm 1$. There are three weak classifiers are

$$h_1(x) = 1 \cdot \mathbf{1}(x > 1/2) - 1 \cdot \mathbf{1}(x \leq 1/2),$$
$$h_2(x) = 1 \cdot \mathbf{1}(x > -1/2) - 1 \cdot \mathbf{1}(x \leq -1/2)$$
$$h_3(x) = 1.$$

1. Show that this data can be classified correctly by a strong classifier which uses only three weak classifiers.

   (8 points)

$$-h_1(x) + h_2(x) - h_3(x)$$

2. Calculate the first two iterations of AdaBoost for this problem. Are they sufficient to classify the data correctly?

   (4 points) The AdaBoost can only use positive weights. Two iterations only yield two correctly classified samples.

**4) Ensemble Methods (12 points)** Consider the following binary classification Boosting algorithm.

1. Given $\{\mathbf{x}_i, y_i\}_{i=1}^N$, number of iterations $T$, weak learner $f$.
2. Initialize $\mathcal{D}_0$ to be a uniform distribution over examples.
3. For each iteration $t = 1 \ldots T$:
   (a) Train a weak learner $f$ on the data given $\mathcal{D}_t$ to produce hypothesis $h_t$.
   (b) Compute the error of $h_t$ as $\epsilon_t = P_{\mathcal{D}_t}[h_t(\mathbf{x}_i) \neq y_i]$
   (c) Compute $\alpha_t = \frac{1}{2} \log \frac{1-\epsilon_t}{\epsilon_t}$
   (d) Update $\mathcal{D}$ as:
   $$\mathcal{D}_{t+1}(i) = \frac{\mathcal{D}_t(i)}{Z_t} \times \begin{cases} \exp(-\alpha_t + (T-t)/T) & \text{if } h_t(\mathbf{x}_i) = y_i \\ \exp(\alpha_t + (T-t)/T) & \text{otherwise} \end{cases}$$
4. Output final hypothesis $H(\mathbf{x}) = \text{sign}\left\{\sum_{t=1}^T \alpha_t h_t(\mathbf{x})\right\}$

$Z_t$ is a normalization constant so that $\mathcal{D}$ is a valid probability distribution.

Describe the difference between this algorithm and the AdaBoost algorithm we learned about in class. What problem of AdaBoost is this change designed to fix? How does changing the algorithm's user provided parameter affect this behavior?

We will accept two answers.

1) If you answer the problem as it is written: the modified update to $D$ does not change the resulting distribution. The additional term $(T-t)/T$ is eliminated when doing normalization.

2) If you instead change to

$$\mathcal{D}_{t+1}(i) = \frac{\mathcal{D}_t(i)}{Z_t} \times \begin{cases} \exp(-\alpha_t - (T-t)/T) & \text{if } h_t(\mathbf{x}_i) = y_i \\ \exp(\alpha_t + (T-t)/T) & \text{otherwise} \end{cases}$$

This has the effect of making small changes to the distribution in later iterations. By changing the distribution more slowly, we can reduce potential overfitting in later rounds. Viewed from another perspective, we are making a greater effort in early rounds to get examples correct.