Challenge Problem 3 Solution

Siddharth Batra

March 11, 2010

1 Decision Trees

1.1

$$P(y|x) = P_{l(x)}{}^{y}(1 - P_{l(x)})^{1-y}$$
$$P(y|x) = \phi^{y}(1-\phi)^{1-y}$$

$$\alpha(T) = \prod_{i=1}^{m} P(y^i|x^i)$$
$$\alpha(T) = \prod_{i=1}^{m} \phi^{y^i}(1-\phi)^{1-y^i}$$

$$\alpha_{\delta(l)}(T) = \prod_{i=1}^{m} \phi^{y^i}(1-\phi)^{1-y^i}$$
$$\alpha_{\delta(l)}(T) = \phi^{g}(1-\phi)^{m_l - g}$$

where:
$\phi = P_{l(x)}$
g is total good samples
m_l is total samples in $\delta(l)$
$(x^i, y^i) \epsilon \delta(l)$

Taking log and setting derivate to zero

$$\frac{\partial l_{\delta(l)}(T)}{\partial \phi} = \frac{g}{\phi} - \frac{m_l - g}{1-\phi} = 0$$
$$\phi = P_{l(x)} = \frac{g}{m_l}$$

1.2

The solution is inspired by Adaboosting and seeks to iteratively form a distribution over the samples which as the iterations progress gives more mass to samples that are harder to classify.

This distribution will be used for sampling examples in the first part and the update rule for the distribution is the solution for the second part.

1.2.1

$$(x^{(i)}, y^{(i)}) \sim P_{W_k}$$

1.2.2

$$W_{k+1}(j) = \frac{W_k(j)}{Z_k} \exp(-e_{b_k} y^{(j)} b_k(x'^{(j)}))$$

where:

Z_k is a normalization constant

Assuming labels +1 and -1 to make the math simpler, the above equation will give a smaller weight (0-1) to samples that are correct (-ve exponential) and will give a much larger weight (+ve exponential) to the mistakes while the current error of the classifier scales the weight according to the current performance.

2 Error bound for 1-nearest neighbour classifier

2.1

$$q(x) = p(Y = 1|X = x)$$
$$q(x) = \frac{p(X = x|Y = 1)P(Y = 1)}{P(X = x)}$$
$$q(x) = \frac{p(X = x|Y = 1)P(Y = 1)}{P(X = x|Y = 1)P(Y = 1) + P(X = x|Y = 0)P(Y = 0)}$$
$$q(x) = \frac{P_1(x)\theta}{P_1(x)\theta + P_0(x)(1 - \theta)}$$

2.2

Probability of misclassification -

$$
\begin{aligned}
&= P(X = x|Y = 1)P(Y = 0|X = x) + P(X = x|Y = 0)P(Y = 1|X = x) \\
&= P(X = x|Y = 1)(1 - q(x)) + P(X = x|Y = 0)q(x) \\
&= \frac{P(Y = 1|X = x)P(X = x)}{P(Y = 1)}(1 - q(x)) + \frac{P(Y = 0|X = x)P(X = x)}{P(Y = 0)}q(x) \\
&= \frac{q(x)P(X = x)(1 - q(x))}{\theta} + \frac{(1 - q(x))P(X = x)q(x)}{1 - \theta} \\
&= \frac{q(x)(1 - q(x))P(X = x)}{\theta(1 - \theta)}
\end{aligned}
$$

2.3

Probability of misclassification -

$$
\begin{aligned}
&= P(X = x|Y = 0)P(X = x'|Y = 1) + P(X = x|Y = 1)P(X = x'|Y = 0) \\
&= \frac{(1 - q(x))P(X = x)q(x')P(X = x') + q(x)P(X = x)(1 - q(x'))P(X = x')}{P(Y = 0)P(Y = 1)} \\
&= \frac{(1 - q(x))P(X = x)q(x')P(X = x') + q(x)P(X = x)(1 - q(x'))P(X = x')}{\theta(1 - \theta)} \\
&= P(X = x)P(X = x')\frac{(1 - q(x))q(x') + q(x)(1 - q(x'))}{\theta(1 - \theta)}
\end{aligned}
$$

2.4

Probability of misclassification with infinite samples -

$$
\begin{aligned}
&= P(X = x)P(X = x')\frac{(1 - q(x))q(x) + q(x)(1 - q(x))}{\theta(1 - \theta)} \\
&= 2P(X = x)P(X = x')\frac{(1 - q(x))q(x)}{\theta(1 - \theta)}
\end{aligned}
$$

2.5

$$2P(X = x)P(X = x')\frac{(1 - q(x))q(x)}{\theta(1 - \theta)} < 2\frac{q(x)(1 - q(x))P(X = x)}{\theta(1 - \theta)}$$

$$2P(X = x') < 2$$

$$P(X = x') < 1$$

which is true since the problem indirectly assumes:
$0 < P(X = x') < 1$

2.6

In the non-asymptotic case the number of training examples is not enough for the label probability distributions for any test point and its nearest neighbour to be the same. So for a test point x' the distributions of $q(x')$ and $q(x)$ where x is it's nearest neighbour are not the same, thereby invalidating the proof under this condition.

3 Hierarchical Clustering

The majority of the time complexity of Hierarchical Clustering is spent in maintaining the rule that at every time step the cluster centroids with the shortest distance must be merged.

Since this rule is relaxed the core idea of the solution is to use a LSH family for the Euclidean space to assert a probablistic notion of closeness rather than the deterministic notion that distance gives.

3.1

1. Take h hash functions to form a Euclidean family of functions and hash every point into a series of buckets with width a.

2. Iteratively or randomly chose any bucket and merge any two points in that bucket while removing redundant entries in the hash table.

3. Hash the new cluster centroid using the h hash functions and repeat.

As the book states, two points which have a distance $d >> a$ have a small chance of being hashed to the same bucket. More formally a Euclidean family is a $(a/2, 2a, 1/2, 1/3)$ LSH family.

Optionally, we can apply an AND construction with r rows followed by an OR construction with b bands to amplify the family described above.

3.2

Complexity of (1) is hn and complexity of (3) is h. Since the last two steps are performed at most n times and the first step is only performed once the overall complexity is $hn + hn = O(hn)$. This assumes that the complexity of (2) is negligible in comparison.

For the amplified case the complexity of (1) is $hrbn$ and complexity of (3) is hrb. The overall complexity for this case becomes $hrbn + hrbn = O(hrbn)$

3.3

The LSH Euclidean family can simply be swapped for a non-Euclidean family depending on which non-Euclidean distance metric is appropriate. For the more interesting issue of representing a cluster centroid, we can pick one of the points being clustered that is more representative of the cluster and delete the hash table entry of the other point.