Homework 8

December 16, 2016

* For each question, give your answer. Justify your answer using a detailed explanation.

1 Understanding Decision Tree

Consider the "Golf Play" data used in the class, which is show in the following table.

	outlook	temperature	humidity	windy	play
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
3	overcast	hot	high	FALSE	yes
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
7	overcast	cool	normal	TRUE	yes
8	sunny	mild	high	FALSE	no
9	sunny	cool	normal	FALSE	yes
10	rainy	mild	normal	FALSE	yes
11	sunny	mild	normal	TRUE	yes
12	overcast	mild	high	TRUE	yes
13	overcast	hot	normal	FALSE	yes
14	rainy	mild	high	TRUE	no

(i) From this table, build up you ID3 tree using Information Gain. Show each step, how you choose an attribute to split.

(ii) Using Information Gain Ratio, which attribute is the one you choose to split for the first step?

(iii) Using Gini Index, construct your decision tree.

2 Learning Ensemble Methods

Ensemble methods in Matlab

Ensemble methods are already implemented in Matlab. For example, you may create an ensemble with the `fitensemble` function, with syntax

```
ens = fitensemble(X,Y,model,numberens,learners)
```

There are also many other functions. Your task is to read and learn how to use ensemble methods in Matlab, including bagging, boosting, random forest (subspace method in Matlab). There is no need for you to code and implement those methods, since it would be complex and time-consuming for you. But you should be able to use the off-the-shelf packages.

2.1 Gradient Boosting Machines

For a given loss function ℓ (for instance, least squares loss) and a hypothesis space $\mathcal{F}$ of regression functions (i.e. functions mapping from the input space to $\mathbb{R}$), the loss of the i-th datum is given by $\ell\{y_i, f(x_i)\} = \ell\{y_i, \hat{y}_i\}$, since $\hat{y}_i = f(x_i)$. Recall the general gradient boosting algorithm can be stated as follows:

1. Initialize $f_0(x) = 0$
2. For $m = 1$ to M: we would like to reduce ℓ over the data set through manipulating f.
   (1) First we compute the gradient of the loss function with respect to f

$$(g_m)_i = \frac{\partial}{\partial f}\ell\{y_i, f(x_i)\}\Big|_{f=f(x_i)}$$

   and get g_m.
   (2) Fit regression model to negative gradient $-g_m$:

$$p_m = \arg\min_{p\in\mathcal{F}} \sum_{i=1}^{N}[-(g_m)_i - p(x_i)]^2.$$

(3) Choose fixed stepsize $\nu_m = \nu \in (0, 1]$, or take a "line search strategy":

$$\nu_m = \arg\min_{\nu>0} \sum_{i=1}^{N} \ell\{y_i, f_{m-1}(x_i) + \nu p_m(x_i)\}.$$

(4) Take this step:

$$f_m(x) = f_{m-1}(x) + \nu_m p_m(x).$$

3. Return f_M which is our final hypothesis function.

In this problem, we will derive two special cases of the general gradient boosting framework: ℓ_2-Boosting (regression) and BinomialBoost (classification). For BinomialBoost, you can read Chapter 16.4 of Textbook "*Machine Learning, A Probabilistic Perspective*" to understand how Adaboost is derived.

Here you are required to derive the algorithm for ℓ_2-Boosting. Specifically, consider the regression framework. Suppose our loss function (recall least squares method) is given by

$$\ell(\hat{y}, y) = \frac{1}{2}(\hat{y} - y)^2,$$

and at the beginning of the m-th round of gradient boosting, we have the function $f_{m-1}(x)$. Give a description of the ℓ_2-Boosting algorithm in this case using the Boosting framework above.

3 Clustering

3.1 Hierarchical clustering

Consider the following set of singleton clusters:

$$\{\{6\}, \{8\}, \{10\}, \{20\}, \{26\}, \{30\}, \{32\}, \{33\}\}.$$

Recall that in performing Hierarchical Agglomerative Clustering (HAC), there are four linkages that are typically used in calculating inter-cluster distance: single linkage, complete linkage, average linkage and distance between centroids. Assume that in merging clusters ties are broken by merging the two clusters that result in the smallest sum for the points in the cluster. So, for instance, if we have the three clusters {1}, {2}, and {3}, the first two would be merged first since they result in a sum of 3, whereas merging the last two would result in a sum of 5.

Starting with the eight singleton clusters in the dataset above, perform HAC until you are left with one cluster. Do this with (a) single linkage, and (b) complete linkage. Draw the resulting two binary trees.

3.2 k-means initialization

Choosing initial centroids is a crucial issue for k-means. Contemporary methods include:

(a) Randomly choose initial centroids and run k-means algorithm, and then repeat many times and select the one with the smallest final *within-cluster variation*

$$W = \sum_{k=1}^{K} \sum_{z_i=k} \|x^{(i)} - \mu_k\|_2^2$$

where K is the number of clusters, $\mu_k, k = 1, ..., K$ are the centroids, z_i is the latent variable we mentioned in class and $z_i = k$ if the i-th point is in cluster k.

(b) Randomly choose the first centroid, and then choose the second centroid as the one with largest distance to the first point. Then for point i, we can compute its distances to all current centroids and record the smallest distance d_{ik}. Suppose we have chosen k centroids so far, then

$$d_{ik} = \min_{j=1,...,k} \{\|x^{(i)} - \mu_j\|\}.$$

The next centroid is the point with largest d_{ik}. Repeat this procedure until all centroids are selected.

(c) Some modern methods assigning probability to each point proportional to their distances to each centroids, such as *k-means++* algorithm.

Now, use strategy (b) and point {6} as the first point, determine the initial three centroids, and apply k-means algorithm to split the data in Part I into three clusters.

3.3 Determine number of clusters

Sometimes, using clustering algorithms, we might have no problem specifying the number of clusters K *ahead of time*. Usually, K is *implicitly defined* by methods such as cutting a hierarchical clustering tree at a given height. However, in most exploratory applications, the number of clusters K is *unknown* and we need to determine the "right" value of K.

Many criterion can be applied here to determine the best K, such as BIC (Bayesian Information Criterion). We introduce here another useful criterion widely used in clustering, CH

(Calinski-Harabasz) index, which is defined by *within-cluster variation* W (see § 3.2) and *between-cluster variation* B. While W measures how *spread apart* the points in one group are from each other, B measures how spread apart the clusters are from each other:

$$B = \sum_{k=1}^{K} n_k \|\mu_k - \bar{x}\|_2^2$$

where as before μ_k is the average of the n_k points in cluster k, and $\bar{x}$ is the overall average, i.e.,

$$\mu_k = \frac{1}{n_k} \sum_{z_i=k} x_i \quad \bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i \quad N = \sum_{i=1}^{K} n_k.$$

This criterion is implemented in Matlab function `evalclusters`. Now read the description of `evalclusters`, and run it for `fisheriris` data we have introduced in class. It's known that there are around 300 kinds of iris flowers in this world, and we assume there are *at most* 10 kinds in the data set. Use function `evalclusters` and k-means method, determine how many kinds of iris flowers are most probably in the data.