Problem Set 3

November 15, 2013

Due date: Mon, Dec 2, 2013 at 4pm.

Exercise 1 (30 points)

Assume a binary classification problem, where every data instance can belong to one of two possible classes: class A and class B.

1. Assume a meta-classifier that classifies an instance as follows: it asks n independent classifiers to classify the instance. If the majority of the independent classifiers classify the instance as class A, so does the meta-classifier. Otherwise, the meta-classifier classifies the instance as class B. If each one of the independent classifiers makes a classification error with probability p, what is the probability of error of the meta-classifier? (15 points)

2. Assume another meta-classifier that classifies an instance as class A, if there exists at least one independent classifier that classifies it as A. Otherwise, the meta-classifier classifies the instance as class B. What is the probability of error of the meta-classifier given that each independent classifier has probability of error p? (15 points)

Exercise 2 (30 points)

When building a decision tree, we select the best split node using an impurity measure. An example of impurity measure is the *entropy*. Consider node t in the decision tree and let $p(i \mid t)$ be the fraction of the records associated with node t and belonging to class i. Then, if there are c classes in total, we measure the impurity of t using entropy as follows:

$$H(t) = -\sum_{i=1}^{c} p(i \mid t) \log p(i \mid t).$$

1. Consider a node t in the decision tree that corresponds to a continuous feature (e.g., the salary). Assume that you want to partition the points that are in node t using k salary ranges $R_1, \ldots, R_k$ that are contiguous, non-overlapping and cover the same total salary range as t. Design an algorithm that finds these ranges and creates nodes $t_1, \ldots, t_k$ such that node t_i corresponds to range R_i and

$$H(t_1) + H(t_2) + \ldots + H(t_k)$$

is minimized. (15 points)

2. Compute the running time of this algorithm as a function of the number of points n_t that are associated with node t. (15 points)

Exercise 3 (20 points) Consider the graph that is described by the set of edges in the file `matrix.txt` that is available at `http://cs-people.bu.edu/cmav/cs565/matrix.txt`. Apply spectral partitioning techniques to partition the nodes of the graph into $k = \{2, \ldots, 20\}$ clusters. Plot the value of your objective function F as a function of the number of clusters k. For a partition into k groups the value of the objective function F_k is the number of edges in the original graph that have their endpoints in different clusters.

Describe your spectral algorithm for the partitioning.

Note: A spectral algorithm is expected to be using Fiedler vector computations.

Exercise 4: (30 points)

Let D the domain (or the universe) of n distinct objects, and let P be the set of distinct pairs of objects in D. Also, let σ_1, σ_2 be two rankings (permutations) of the elements in D. The Kendall's tau distance between two permutations is defined as follows: For each distinct pair $\{i, j\} \in P$ if i and j are in the same order in σ_1 and σ_2, then $K_{ij}(\sigma_1, \sigma_2) = 0$; if i and j are in the opposite order (such as i being ahead of j in σ_1 and j being ahead of i in σ_2), then $K_{ij}(\sigma_1, \sigma_2) = 1$. The Kendall's tau distance between σ_1 and σ_2 is given by $K(\sigma_1, \sigma_2) = \sum_{\{i,j\}\in P}(\sigma_1, \sigma_2)$.

Very often, instead of observing the whole ranking of the n objects we see only the sorted lists of the first k elements of the ranking. We call such list a top-k list. Let τ_1 and τ_2 be the top-k lists of two rankings of the elements in D. Then, we define the p-Kendall tau distance between τ_1 and τ_2 as follows. For a pair of objects $i, j \in D$ we the following cases.

1. If i and j both appear in τ_1 and τ_2 and are in the same order (such as i being ahead of j in both top-k lists), then $K^p_{ij}(\tau_1, \tau_2) = 0$.

2. If i and j both appear in τ_1 and τ_2, but in opposite order (such as i being ahead of j in τ_1 and j ahead of i in τ_2) then, $K^p_{ij}(\tau_1, \tau_2) = 1$.

3. If i and j both appear in one top-k list (say τ_1) and exactly one of i or j, say i, appears in the other top-k list (say τ_2), then if i is ahead of j in τ_1, then $K^p_{ij}(\tau_1, \tau_2) = 0$. Otherwise, $K^p_{ij}(\tau_1, \tau_2) = 1$. Intuitively, we know that i is ahead of j as far as τ_2 is concerned, since i appears in τ_2, but j does not.

4. If i, but not j, appears in one of the top-k lists (say τ_1) and j but not i appears in the other top-k list (say τ_2), then $K^p_{ij}(\tau_1, \tau_2) = 1$. Intuitively, we know that i is ahead of j as far as τ_1 is concerned and j is ahead of i as far as τ_2 is concerned.

5. If i and j both appear in one top-k list (say τ_1), but neither i nor j appears in the other top-k list (say τ_2). We call such pairs special pairs and we define $K^p_{ij}(\tau_1, \tau_2) = p$ with $0 \leq p \leq 1$.

We define the p-Kendall tau distance between two top-k lists to be: $K^p(\tau_1, \tau_2) = \sum_{\{i,j\}\in P_{\tau_1 \cup \tau_2}} K^p_{ij}(\tau_1, \tau_2)$, where $P_{\tau_1 \cup \tau_2}$ is the set of distinct pairs $\{i, j\} \in D_{\tau_1} \cup D_{\tau_2}$, (note that D_{τ_1} (D_{τ_2}) is the subset of elements from D that appear in τ_1 (resp. τ_2). You are asked to prove the following:

1. Prove that the Kendall's tau distance between two permutations σ_1 and σ_2, denoted by $K(\sigma_1, \sigma_2)$ satisfies the triangle inequality. (10 points)

2. Find the values of p for which the p-Kendall tau distance, K^p, satisfies the triangle inequality. (20 points)