This homework is individual work. All assignments are due at the beginning of class on the due date. Assignments turned in late will drop 10 points for each period of 24 hours for which the assignment is late. In addition, no assignments will be accepted after the solutions have been made available. Please include your cornell net id on your homework.

A. Statistical Learning Theory

1. Error Bounds (20 POINTS)

Consider the instance space X of points $\vec{x} = (x_1, x_2)$ with integer coordinates $([0..10], [0..10]) \subset N^2$. The task is to learn a concept $c : X \rightarrow \{0, 1\}$ which can be described as a rectangle $((a, b), (c, d)) \in N^2 \times N^2$. (a, b) is the left top corner of the rectangle, (c, d) is the lower right corner. An example is labeled positive, if it lies inside the rectangle or on it's boundary. It is negative, if it lies outside. The hypothesis space H are all rectangles over the instance space X.

You have a training sample of size n and your learning algorithm found a hypothesis that has zero training error. Give a bound for the prediction error of this hypothesis!

We can calculate $\|H\|$ in the following way: Unique rectangles can be generated by first selecting the left top corner $(i, j), i \in [0..N], j \in [0 : N]$ and then selecting the lower right corner $(k, l), k \in [0..i], l \in [j..N]$. From this we get:

$$\begin{aligned} \|H\| &= \sum_{i=0}^{N}\sum_{j=0}^{N}(i+1)(N-j+1) \\ &= \sum_{i=0}^{N}\sum_{j=0}^{N}(i+1)(j+1) \\ &= ((N+1)(N+2)/2)^2 \\ &= 4356 \end{aligned}$$

Using the formula derived in class, we can now say that with probability $> (1-\delta)$, prediction error will be less than ϵ where $\epsilon = \frac{1}{n}\left(ln(4356) + ln(\frac{1}{\delta})\right)$.

2. Infinite Hypothesis Spaces (20 POINTS)

In class we only derived generalization error bounds for finite hypothesis spaces. For infinite hypothesis spaces, these bounds cannot be applied directly, since they contain the number of hypothesis $|H|$ as a factor. The key trick to extend them to infinite hypothesis spaces is to consider the "effective" number of hypotheses, which is finite. To illustrate this, consider the following example.

Our hypothesis space are all intervals $[a, b]$ on the real line (i.e. the instance space $X = \Re$). A point x is classified positive, if $x \in [a, b]$, negative otherwise. Clearly, this hypothesis space is infinite. However, many of these hypotheses are "redundant" for any given sample.

Consider a sample $Sample = (x_1, ..., x_n)$ of n (distinct) points. Derive a formula for the number of hypotheses that classify the examples *in different ways*. The bound should hold for any such sample.

For the purpose of defining unique hypotheses intervals $[a, b]$ all points between two adjacent sample points x_i and x_{i+1} are equivalent (we are assuming, without loss of generality, that the points are ordered.) So we can count unique hypotheses

intervals as follows: To define non-empty intervals a can be selected in n different ways, where a is in the interval $[x_{i-1}, x_i), i \in [1..n]$ and $x_0 = -\infty$, and for each such a, b can be selected in $n - i + 1$ different ways (it could be in the intervals $[x_j, x_{j+1}], j \in [i..n]$ where $x_{n+1} = \infty$.) So we have $\frac{n(n+1)}{2}$ unique non-empty hypotheses and the empty hypotheses to get $\|H\| = \frac{n(n+1)}{2} + 1$.

B. Clustering

1. Hierarchical Agglomerative Clustering (20 POINTS)

Create by hand the clustering tree for the following sample of ten points in one dimension

$$Sample = (-2.2, -2.0, -0.3, 0.1, 0.2, 0.4, 1.6, 1.7, 1.9, 2.0). \tag{1}$$

Use single link clustering (i.e. $d(C_i, C_j) = min_{x \in C_i, x' \in C_j} ||x - x'||$). Based on the clustering tree, argue that three is the natural number of clusters.

After drawing the tree, it becomes clear that when there are three clusters, the width of each cluster is much smaller than the gap between any two clusters. This is not the case for any other number of clusters. Hence three is the natural number of clusters for this dataset.

2. K-Means as Greedy Search (20 POINTS)

The k-means algorithms (see Algorithm 1 in the handout) can be viewed as searching for the clustering $C_1, ..., C_k$ with mean vectors $\vec{\mu}_1, ..., \vec{\mu}_k$ that minimizes the following objective function, typically called the sum-of-squared-error criterion J_e:

$$J_e = \sum_{i=1}^{k} \sum_{\vec{x} \in C_i} ||\vec{x} - \vec{\mu}_i||^2 \tag{2}$$

Prove that in every iteration of the algorithm (before convergence), the objective function J_e never increases. In particular, prove that in each individual step

- "classify n samples according to the nearest $\vec{\mu}_i$" and
- "recompute $\vec{\mu}_i = \frac{1}{|C_i|} \sum_{\vec{x} \in C_i} \vec{x}$"

the objective function J_e never increases.

Step 1: Reclassifying to the nearest $\vec{\mu}_i$

Each point starts this step assigned to some mean $\vec{\mu}_i$. At this stage, the means do not change, so if a point is reassigned to a new mean, it must be closer to that new mean than it was to the old mean. So for these points, the objective function must decrease. The other points are not reassigned, and so for them the objective function is unchanged. Therefore the objective function cannot increase.

Step 2: Computing new means

We can take the partial derivative of the objective function, J_e with respect to each mean $\vec{\mu}_i$. If we set this to zero, as it must be at the minimum, we recover the result $\vec{\mu}_i = \frac{1}{|C|} \sum_{x \in C} \vec{x}$. We know this is the minimum and not the maximum because the second derivative of J_e with respect to $\vec{\mu}_i$ is positive. Hence by recomputing the means, we are in fact minimizing J_e.

3. Finding the Optimal k in k-Means (20 POINTS)

One shortcoming in k-means is that you have to specify the value of k. How about the following strategy for picking k automatically: try all possible values $k \in \{1, 2, ..., n\}$ and pick the k that minimizes J_e. Argue why this is a good / bad idea!

This is clearly a very bad idea because when $k = n$, each point is the mean of its own cluster and $J_e = 0$. This is the trivial solution that this approach would always give. (Note: the reason this happens is that for different values of k, J_e is in fact a different function, so it doesn't make sense to vary k.)