CS181 Assignment 3: Clustering and Parameter Estimation

Out Monday March 4th
Due Friday March 15th

Submit by **noon** via iSites dropbox.

General Instructions

You may work with **one other person** on this assignment. Each pair should turn in one writeup. This assignment consists of a theoretical component and an experimental component.

Problem 1

[20 Points] This question investigates the use of instance-based methods such as the HAC clustering algorithm in high dimensional spaces.

a. **[5 Points]** Let $\mathcal{X}$ be the M-dimensional unit hypercube, i.e., $\mathcal{X} = [0,1]^M$. Let $\boldsymbol{x} = [x_1, x_2, \cdots, x_M]$ be the point at the center of $\mathcal{X}$, and $\boldsymbol{y} = [y_1, y_2, \cdots, y_M]$ be a point in $\mathcal{X}$ drawn randomly from a uniform distribution on $\mathcal{X}$. For $\epsilon \in (0, \frac{1}{2})$, compute the probability ρ that $\max_m |x_m - y_m| \leq \epsilon$. That is, compute the probability that all of the M dimensions of $\boldsymbol{x} - \boldsymbol{y}$ are between $-\epsilon$ and ϵ.

b. **[3 Points]** Let $\boldsymbol{x}$ be some other point in $\mathcal{X}$, and $\boldsymbol{y}$ a point in $\mathcal{X}$ drawn randomly from a uniform distribution on $\mathcal{X}$. Based on your result from (a), argue (informally is OK) that the probability that $\max_m |x_m - y_m| \leq \epsilon$ is *at most* ρ.

c. **[3 Points]** Let $||\boldsymbol{x} - \boldsymbol{y}||$ denote the Euclidean distance between two points $\boldsymbol{x}$ and $\boldsymbol{y}$. Prove that $||\boldsymbol{x} - \boldsymbol{y}|| \geq \max_m |x_m - y_m|$. Given this, argue from (b) that if $\boldsymbol{x}$ is any point in $\mathcal{X}$, and $\boldsymbol{y}$ is a point in $\mathcal{X}$ drawn randomly from a uniform distribution on $\mathcal{X}$, then the probability that $||\boldsymbol{x} - \boldsymbol{y}|| \leq \epsilon$ is also at most ρ.

d. **[6 Points]** Using your result from (c), give a *lower bound* on the number N of points needed to guarantee, with probability at least $1 - \delta$, that the nearest neighbor of a point $\boldsymbol{x}$ will be within a radius ϵ of it. **[Hint:** think about what is required for the nearest neighbor *not* to be within a radius ϵ.]

e. **[3 Points]** What can you conclude about the effectiveness of the hierarchical agglomerative clustering algorithm in high dimensional spaces?

Problem 2

[25 Points] This question asks you to explore the maximum likelihood (ML), maximum *a posteriori* (MAP), and full Bayesian (FB) approach to estimating the parameters of probability models.

a. **[4 Points]** Denote data by $\mathcal{D}$ and parameters by θ. Given a prior $\Pr(\theta)$ and likelihood $\Pr(\mathcal{D} \mid \theta)$, what is the predictive distribution $\Pr(\boldsymbol{x} \mid \mathcal{D})$ for a new datum $\boldsymbol{x}$, for each of the ML, MAP and FB methods?

b. **[3 Points]** Discuss why MAP method can be considered to be "more Bayesian" than ML.

c. **[3 Points]** Consider the MAP method. Provide a practical advantage it enjoys, as a machine learning approach, over the ML method and another that it enjoys over the FB method.

d. **[5 Points]** Consider the soccer team example in the class notes for Lecture 11. Sketch or plot the Beta distribution for Beta(1,1), Beta(3,3) and Beta(2,5). Explain the intuition that is represented by adopting each of Beta(1,1), Beta(3,3) and Beta(2,5) as priors for the probability of a win.

e. **[3 Points]** Why is the Beta distribution useful when used together with MAP estimation for reasoning with Bernoulli (binary) random variables?

f. **[7 Points]** Look at the record of wins and losses for the Harvard football team in the 2011-12 season (games played in Fall 2011). Following the approach in the class notes, derive the estimate of whether or not the team will win next after the first three games of the 2012-13 season under each of the ML, MAP and FB approaches.

Problem 3

[24 Points] In lecture, we discussed how K-means clustering can be viewed as a compression algorithm with a squared loss. That is, if we were forced to replace each datum with its prototype, how can we select the prototypes to do as well as possible?

a. **[12 Points]** Write down the K-means clustering objective, and show how the update steps can be derived by performing gradient descent on it.

b. **[12 Points]** As discussed in lecture, principal components analysis (PCA) also has a squared-loss compression interpretation. Explain how these two methods relate to each other, and describe situations and hypothetical data sets in each of them might be appropriate or inappropriate.

Problem 4

[75 Points] In this exercise, you will experiment with clustering algorithms on a dataset consisting of census data. You can find the following files in `http://www.seas.harvard.edu/courses/cs181/files/hw3.tar.gz`:

- `adults.txt`: The census data; there are 30717 instances.
- `adults-small.txt`: A subset of the census data containing only 3 attributes: age, education, and income. See `181adult.names` for explanations of how these attributes are represented.
- `181adult.names`: A description of the census data including a list of the 48 attributes (corresponding to 13 census categories). You should read this file to understand the data before you start. All feature values have been scaled to appropriate ranges. You don't need to modify the values further.
- `clust.py`: Support code for loading the census data.

The code should be run by typing `python clust.py <num-clusters> <num-examples>`. The support code for this assignment is minimal. You are welcome to write in a language of your choice, just be sure to indicate this in your write up. There are a total of 30717 examples in the dataset, but you do not need to use all of them when testing your code.

a. **[20 Points]** Implement the K-means clustering algorithm. Your program should print out the means of each of the clusters. You will want to generate random initial prototype vectors by sampling from the data points.

   (a) For *num-examples* = 1000 on the `adults` dataset (43 attributes), run the K-means algorithm with K = 2, 3, 4, 5, 6, 7, 8, 9, 10. For each value of K, compute the mean squared error (the mean of the squared distance of each point from its closest prototype vector). Provide a plot of this mean squared error versus K. [**Hint:** If your mean squared error results are not what you expect, be sure to re-run your code, since the initial parameters are random.]

   (b) If you were to choose the best K for this data based on the the plot you generated in part (i), what value of K might you choose? Justify your choice.

b. **[30 Points]** Implement the hierarchical agglomerative clustering (HAC) algorithm as well as the four distance metrics discussed in lecture: min, max, mean, and centroid. Your program should print out the means of each of the final clusters.

   (a) For *num-examples* = 100, $K = 4$, run HAC on the `adults-small` data set, which contains only 3 attributes. Compare the clusters formed using the *min* distance metric against the clusters formed using the *max* distance metric. For each distance metric, submit two things: (1) a table showing the number of instances in each cluster and (2) a scatterplot of the instances in 3-dimensions, clearly indicating which instances belong to which cluster. What differences do you notice between the clusters produced using the *min* versus the *max* distance metric? Does this make sense given the definition of the metrics?

   (b) For *num-examples* = 200, $K = 4$, run HAC on the `adults-small` data set, which contains only 3 attributes. Compare the clusters formed using the *mean* distance metric against the clusters formed using the *centroid* distance metric. For each distance metric, submit two things: (1) a table showing the number of instances in each cluster and (2) a scatterplot of the instances in 3-dimensions, clearly indicating which instances belong to which cluster. What differences do you notice between the clusters produced using the *mean* versus the *centroid* distance metric? Does this make sense given the definition of the metrics?

c. **[25 Points]** Implement the Autoclass clustering algorithm. Your program should print out the means of each of the clusters.

   (a) Run the Autoclass algorithm on the `adults` dataset (43 attributes), with *num-examples* = 1000, and $K = 4$. How many iterations does it take for your parameters to converge?

   (b) Submit a plot showing the log likelihood of the data versus number of iterations, for as many iterations as it takes to converge.

   (c) How does the run-time performance of Autoclass compare with the performance of K-means?

   (d) **[Extra Credit]** If you're really eager, you can run the same experiment as you did for K-means, varying $K = 2, 3, 4, \ldots, 10$ and plotting log likelihood against K. If you were to choose the best K for this data, what value of K would you choose? Why?