CS181 Assignment 3

Joy Ming and Alisa Nguyen (15 March 2013)

1 Problem 1

a. The probability that all of the M dimensions of $x - y$ are between $-\epsilon$ and ϵ is $\boxed{\rho = (2\epsilon)^M}$.
For each dimension i of χ, the probability that $|x_i - y_i| \leq \epsilon$ is equivalent to

$$\begin{gathered} P(|x_i - y_i| \leq \epsilon) = \\ P(-\epsilon \leq x_i - y_i \leq \epsilon) = \\ P(-\epsilon - x_i \leq -y_i \leq \epsilon - x_i) = \\ P(\epsilon + x_i \geq y_i \geq x_i - \epsilon) = \\ P(x_i - \epsilon \leq y_i \leq \epsilon + x_i) \end{gathered}$$

This distribution function is equivalent to $\int_{\epsilon+x_i}^{x_i-\epsilon} f(x)dx$, where $f(x)$ is the PDF of y_i, which we know to have a uniform distribution, so $f(x) = \frac{1}{b-a} = 1$. Thus, we get:

$$\begin{gathered} \int_{\epsilon+x_i}^{x_i-\epsilon} 1dx = \\ \epsilon + x_i - (x_i - \epsilon) = 2\epsilon \end{gathered}$$

Because we want to know the probability that all of the M dimensions of $x - y$ are between $-\epsilon$ and ϵ, we simply take $\Pi_{i=1}^{M} P(|x_i - y_i| \leq \epsilon) = (2\epsilon)^M$.

b. The probability of $max_m|x_m - y_m| \leq \epsilon$ is at most ρ because as shown in (a), ρ does not depend on x_i and thus holds for all x_i. In addition, logically, if x is the center point, the average distance from it to any other point y is at most $\frac{1}{2}$ for any one dimension. As x moves farther and farther away from the center, the average distance increases so that it becomes at most 1 in any one dimension. So, if x is not in the center $max_m|x_m - y_m|$ grows and is less likely to be less than ϵ, decreasing that probability so that it is less than ρ.

c. We will show that $||x - y|| \geq max_m|x_m - y_m|$.

$$\begin{gathered} ||x - y|| = \sqrt{\Sigma_{m=1}^{M}(x_m - y_m)^2} \\ \sqrt{\Sigma_{m=1}^{M}(x_m - y_m)^2} \geq max_m|x_m - y_m| \\ \Sigma_{m=1}^{M}(x_m - y_m)^2 \geq (max_m|x_m - y_m|)^2 \end{gathered}$$

This is true because the left side of the inequality includes the right side in its sum. $||x - y||$ is the total Euclidean distance between two points whereas $max_m|x_m - y_m|$ is only the distance between one dimension of two points. The left side must be larger.

If x is any point in χ, and y is a point in χ drawn randomly from a uniform distribution on χ, then the probabilty that $||x-y|| \leq \epsilon$ is also at most p because $||x-y||$ is greater than or equal to $max_m|x_m - y_m|$, making it less likely to be less than ϵ and thus giving it a probability lower than ρ of being less than ϵ.

d. Lowerbound on number N of points needed to guarantee that the nearest neighbor of point x will be within a radius ϵ of it is $\boxed{log\delta/log(1 - (2\epsilon)^M)}$.
For the nearest neighbor not to be within a radius ϵ, none of the neighbors can be within a radius ϵ.

The probability that any one neighbor is not within a radius ϵ of x is $1-(2\epsilon)^M$, so the probability that all the nighbors are not within a radius ϵ of x is equivalent to $(1-(2\epsilon)^M)^N$, where N is the number of neighbors. So, the probability that at least one neighbor is within a radius ϵ is 1 - that quantity. Since we want to guarantee with probability at least $1-\delta$ that the nearest neighbor will be within a radius ϵ of it, we can solve for a lower bound for N by setting the two equations equal to each other.

$$\begin{gathered} 1-\delta = 1-(1-(2\epsilon)^M)^N \\ 1-1+(1-(2\epsilon)^M)^N = \delta \\ (1-(2\epsilon)^M)^N = \delta \\ Nlog(1-(2\epsilon)^M) = log\delta \\ \boxed{N = log\delta / log(1-(2\epsilon)^M)} \end{gathered}$$

e. We can conclude that the effectiveness of the hierarchical agglomerative clustering algorithm in high dimensional spaces is ineffective as the dimension M grows because N would also grow too large and HAC would require too many N points to actually be effective. As M increases, the denominator of the lower bound for N decreases, thus leading to an increase in N overall. In addition, as covered in class, when the size of the dataset gets larger, the probability that two points from different clusters are closer to each other in terms of distance than two points from separate clusters converges to 1/2.

2 Problem 2

a. Given a prior distribution $Pr(\theta)$ and likelihood $Pr(D|\theta)$, the predictive distribution $Pr(x|D)$ for a new datum,

   (a) ML: $Pr(x|D) = Dist(\arg\max_\theta(\ln(Pr(D|\theta))))$,
   where $Dist()$ is a distribution applied to the θ we obtain with the given formula.

   (b) MAP: $Pr(x|D) = Dist(\arg\max_\theta(\ln(P(D|\theta)P(\theta))))$,
   where $Dist()$ is a distribution applied to the θ we obtain with the given formula.

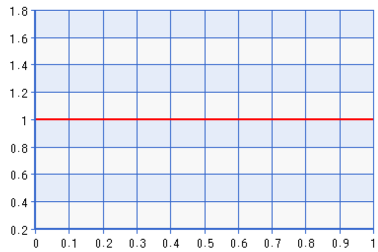
   (c) FB: $Pr(x|D) = \int p(x|\theta)P(\theta|D)d\theta$

b. MAP can be considered "more Bayesian" than ML because it takes into account the distribution of θ instead of assuming same weight or uniformity.

c. One advantage the MAP method enjoys over the ML method is that it accounts for the more likely distribution, as opposed to simply assuming uniformity, as with ML. FB, on the other hand, maintains a probability distribution is maintained over the set of all parameter values possible. However, because the normalizing factor contains an integration over all parameter values, it can be difficult to compute. This means that it also unnecessarily takes into account the less likely, meaning that FB is less practical than MAP to calculate. Therefore, MAP sits in the "sweet spot" of taking more into account in terms of distribution than ML but less excessively than FB.

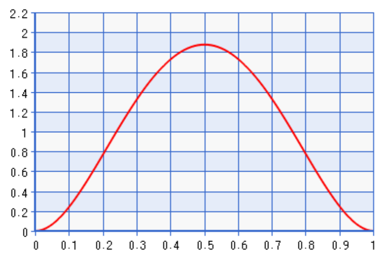
d. Soccer team example based on different Beta distributions as priors for the probability of a win, the different possible distributions are:

   - Beta(1,1)



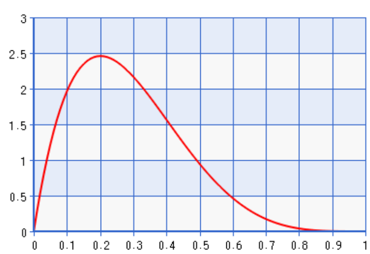
The Beta(1,1) distribution is equated with the Uniform(0,1) distribution, which is a number uniformly chosen between 0 and 1. Beta(1,1) is a uniform prior on θ which says there was just one positive and one negative example, essentially makes the probability of winning and losing equally likely. Often used as a "non-informative prior", this assumes uniformity and is not very descriptive in depicting the distribution of wins and losses otherwise.

- Beta(3,3)



Based on the equation for the mode of a variate with a beta distribution, $x = \frac{\alpha-1}{\alpha+\beta-2} = \frac{3-1}{3+3-2} = \frac{2}{4}$. This means that this distribution says that it is as likely to win as it is to lose as the distribution is symmetrical. However, this is more descriptive than simply a Beta(1,1) distribution, as seen in the plot because it looks more at the distribution over the entire space, with a lower variance as it makes values in the middle more likely.

- Beta(2,5)



In this case, the plot seems to be skewed toward the left, which the equation for the mode of a variate with a beta distribution show $x = \frac{\alpha-1}{\alpha+\beta-2} = \frac{2-1}{2+5-2} = \frac{1}{5}$ that this is true. This is more descriptive than Beta(1,1) in describing a distribution that is not necessarily uniform and it is also more more descriptive than Beta(3,3) in describing a distribution that is not necessarily as symmetrical.

e. The Beta distribution is the conjugate prior of the Bernoulli.

f. Under the ML approach

3 Problem 3

a. The K-means clustering objective is to minimize the sum of squared distances between prototype and data. The update steps can be derived by performing gradient descent on it by taking the partial derivative over all means and responsibilities. This means we can move along the negative gradient of the loss function, as shown in the lecture slides to converge to a local minimum. Each iteration of the K-means algorithm updates the prototypes to decrease the error, following gradient descent until the prototypes converge to a local minimum and the loop ends. Because each iteration does decrease the error, K-means is guaranteed to converge in the direction that decreases the error the most, or in the direction of steepest descent.

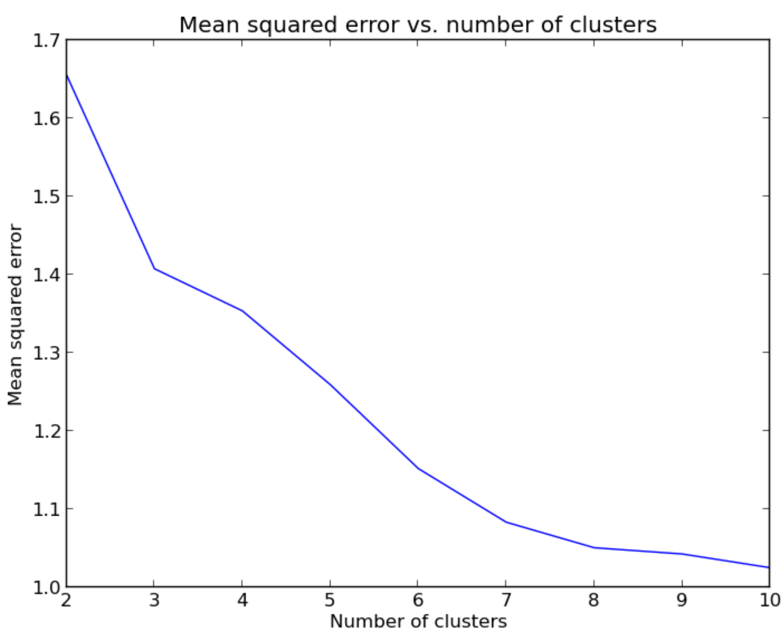
b. The principal components analysis (PCA), which looks to compress features using a linear reduction into lower dimension space, has an interpretation that looks to minimize the difference between data points through minimum square loss. One way they are related is that principal components are the continuous solutions to the discrete cluster membership indicators for K-means clustering. As PCA can be represented by a covariance matrix, K-means can be represneted by a spectral expansion of the data covariance matrix truncated at k-1 terms. The relaxed solution of k-means clustering is given by the PCA principal components, wher ethe PCA subspace spanned by the principal directions is identical to the cluster centriod subspace.
Some example situations and hypothetical data sets include:

- PCA appropriate
- K-means appropriate

4 Problem 4

a. K-means clustering algorithm

(a) Plot of mean squared error versus K



(b) If we were to chose the best K for this data based onthe plot generated in part (i) we would chose the value of K that minimizes the mean squared error, which in this case is 10. It is important to minimize mean squared error because it is a measure of the difference between estimated values and true values, following in line with objectives mentioned in problem 3.

b. HAC algorithm

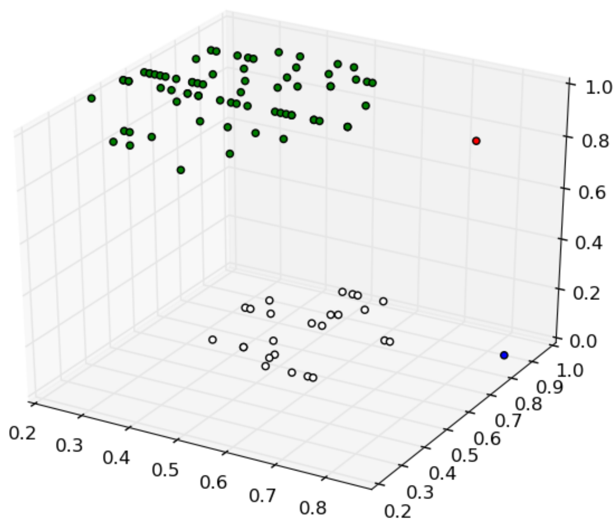
(a) Comparing clusters formed using *min* distance metric against clusters formed using *max* distance metric

i. Table showing number of instances in each cluster

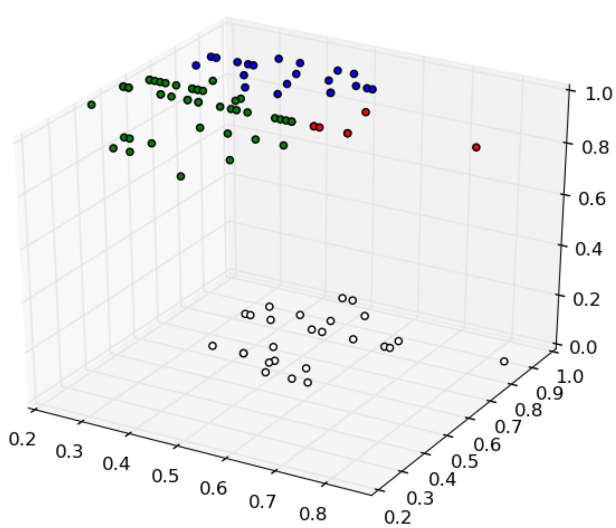
Metric	C1	C2	C3	C4
min	1	1	73	25
max	7	21	46	26

ii. Scatterplot of the instances in 3-dimensions

- Based on *min*:



- Based on *max*:



The *min* distance metric considers the distance between the two closest points in two separate clusters, but the *max* distance metric considers the distance between the two farthest points in the clusters. This results in the *min* measure creating more distinct and consolidated clusters than the *max* metric, which makes sense because by minimizing the smallest distance this brings the points closer to each other.

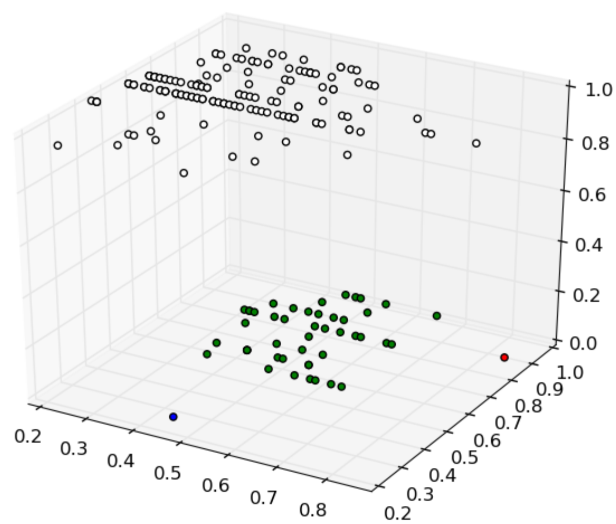
(b) Comparing clusters formed using *mean* distance metric against clusters formed using *centroid* distance metric

i. Table showing number of instances in each cluster

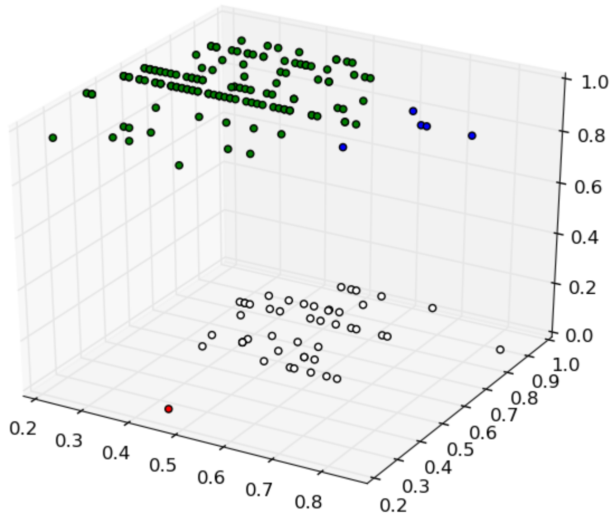
Metric	C1	C2	C3	C4
mean	1	1	46	152
cent	1	5	147	47

ii. Scatterplot of the instances in 3-dimensions

- Based on *mean*:



- Based on *cent*:



The *mean* distance metric looks at the average distances between each of the points in a cluster and *cent* looks at the distances to the center of a cluster. In the clusters based on *mean* the clusters are more distinct in a naive belief of what the clusters would look like whereas the *cent* is a little different and less distinct.

c. Autoclass clustering algorithm

(a) It takes x number of iterations for the parameters to converge.

(b) Plot of the log likelihood of the data versus number of iterations

(c) The run-time performance of auto-class seems to be much faster than that of K-means. (???)