CS181 ASSIGNMENT 3

ASHOK CUTKOSKY AND TONY FENG

Problem 1.

a. This is the probability that y lies in a box of sidelength 2ϵ centered at x. This is just the ratio of the area of this box to the are of the whole box: $p = (2\epsilon)^M$.

b. This probability is the ratio of the volume of the intersection of a box of sidelength 2ϵ centered at x with the box of sidelength 1 with the volume of the box of sidelength 1. The volume of this intersection is at most the volume of the box of sidength 2ϵ, so the ratio is at most $(2\epsilon)^M = p$.

c.
$$||x - y|| = \sqrt{\sum (x_i - y_i)^2} \geq \sqrt{\max(x_i - y_i)^2} = \max |x_i - y_i|$$
so that $||x-y|| \geq \max |x_i - y_i|$. The probability of being within distance ϵ of x is the volume of the ball of radius ϵ about x. Since distance under this euclidean metric is at most distance under the previous metric, this ball has volume at most the volume of the ball under the previous metric, which is at most p.

d. Let p be the probability that for a given point x, a randomly drawn point y will be within ϵ of x. Then the probability that for a given point x a set of N independently randomly drawn points y will all NOT be within ϵ of x is $(1-p)^N$. Thus the probability that the nearest neighbor of x is within ϵ of x is $1 - (1-p)^N$.

We seek a lower bound on N such that $1 - (1-p)^N \geq 1 - \delta$, i.e. $(1-p)^N \leq \delta$. An upper bound for p is not useful for this purpose, so we aren't sure how to use (c), but what follows is a different, valid argument.

Now p is just the intersection of the ball of radius ϵ about x and the box of sidelength 1. The ball of radius ϵ about x contains all the points of distance ϵ from x along each of the m principle axes, and so also contains the polyhedron given by the convex closure of these points. An M-dimensional pyramid with base area A and height h has volume Ah/M. Thus if A_M is the volume of this polyhedron in M-dimensions, we have $A_M = 2\epsilon A_{M-1}/M$, with $A_1 = 2\epsilon$. Thus $A_M = 2^M \epsilon^M / M!$.

The volume of the intersection of the ball of radius ϵ and the box is minimized when x is in a corner of the box, in which case the volume is given by the volume of the ball of radius ϵ divided by 2^M. Thus $p \geq \epsilon^M / M!$. Thus the probability that a nearest neighbor of x is witnih ϵ of x is $1 - (1-p)^N \geq 1 - (1 - \epsilon^M/M!)^N$ and so $N \geq \log \delta / \log(1 - \epsilon^M/M!)$.

e. Note that $\log(1 - \epsilon^M/M!)$ approaches 0 very rapidly as M increases so that the lower bound on N increases very rapidly. Intuitively, this says that points become very far apart very quickly as the dimension increases and so HAC, which relies on distances between points, might become less effective as all these distances become very large.

Problem 2.

a. ML chooses parameters
$$\theta_{ML} = \text{argmax}_\theta \, p(\mathcal{D} \mid \theta)$$

On the other hand, MP chooses

$$\theta_{MAP} = \text{argmax}_\theta \, p(\theta \mid \mathcal{D}) = \text{argmax}_\theta \, p(\mathcal{D} \mid \theta) p(\theta).$$

Also, assign

$$Pr(\theta|D) = \frac{Pr(D,\theta)}{Pr(\theta)} = \frac{Pr(D|\theta)Pr(\theta)}{\int_\Theta Pr(D \mid \theta)Pr(\theta)d\theta}$$

Then

$$\begin{aligned} Pr_{ML}(x|D) =& Pr(x|\theta_{ML}) \\ Pr_{MAP}(x|D) =& Pr(x|\theta_{MAP}) \\ Pr_{FB}(x|D) =& \int_\Theta Pr(x|\theta)Pr(\theta|D)d\theta \end{aligned}$$

b. The MAP method can be considered more Bayesian because it places greater emphasis on prior hypotheses, and updates these depending on the data, rather than basing itself completely on the data as ML does.

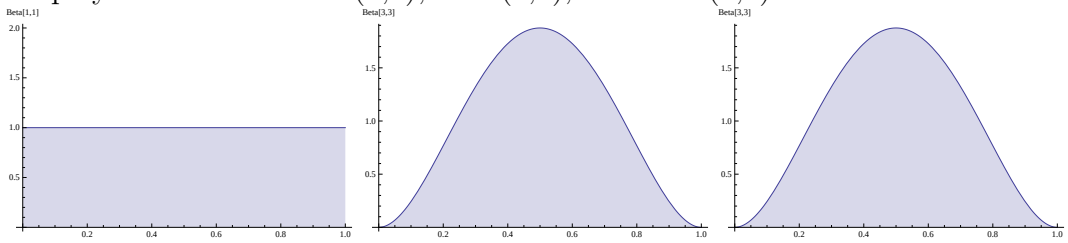
c. MAP enjoys the advantage of incorporating more knowledge about the world than the ML approach so that it is more resistant to overfitting. MAP enjoys the advantage of being computationally more tractable than the FB approach.

d. In the Bernoulli distribution, the parameter θ corresponds to the team's propensity to win, which can be thought of as "skill level." The intuition of Beta(1,1) is that each team's propensity to win is drawn from a uniformly random distribution. This is certainly a natural, if somewhat naive, hypothesis, based on the idea that skill is a kind of atomic unit.

The intuition of Beta(3,3) is that skills are drawn from a bell curve, with most people of similar skill and a few outliers in either direction. This certainly makes sense if one thinks of overall soccerskill resulting from the aggregation of many independent, randomic atomic units, with the end product normally distributed by the central limit theorem.

The intuition of Beta(2,5) is that skills are mostly drawn from a lower end, with a few high-skill outliers. One way of reasoning for this is that most people who play on the soccer team are likely to be quite good, so among those people represented on soccer teams we see the sort of "right side" of the bell curve distribution.

Displayed below are Beta(1,1), Beta(3,3), and Beta(2,5) in that order.



e. The Beta distribution is conjugate to the Bernoulli distribution, meaning that $P(\Theta \mid X)$ and $P(\Theta)$ have the same functional form. This property makes it very convenient to work with, since in the MAP formalism we deal with the expression $P(\Theta \mid \mathcal{D})$. If this takes a simple form, then it is easier to compute argmax_θ, for instance.

f. In Fall 2011, the Harvad football team won 9 games and lost once. Using the the ML model, the probability of the team winning any particular game is then just $\frac{9}{10}$, and so the probability of winning the next game is also $\frac{9}{10}$.

Now we calculate using the MAP model. Let's adopt the $\beta(5,3)$ prior. Then following the lecture notes,

$$\text{argmax}_\theta \, P(\theta|D) = \frac{5+9-1}{8+10-2} = \frac{13}{16}$$

so that the probability of winning the next game is $\frac{13}{16}$.

Now we use the FB model, again with the $\beta(5,3)$ prior. Then we have

$$\text{argmax}_\theta \, P(\theta|D) = \int_0^1 \theta p(\theta|D) d\theta = \frac{5+9}{8+10} = \frac{14}{18}$$

so that the probability of winning the next game is $\frac{14}{18}$.

Problem 3.

a. The loss function for vectors $x_1, \cdots, x_N$ with prototypes $\mu_1, \cdots, \mu_K$ and responsibilities r_{ij} is

$$\mathcal{L} = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} |x_n - \mu_k|^2 \tag{1}$$

The K-means updates are a kind of expectation-minimization for this quantity. The hypothesis requires $r_{nk} = 1$ or 0 for all n, k.

In the update step for the r_{nk}, we are simply assigning each datapoint to a cluster (by the binary form of the r_{nk}). The contribution of x_n to (1) is just the squared distance to its chosen cluster, so tautologically (1) is minimized when that cluster is chosen to have the least square distance.

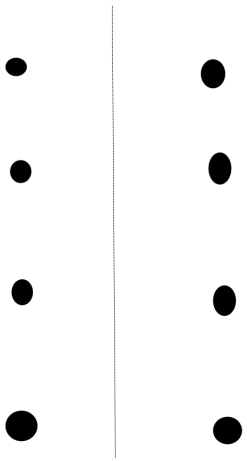
Now conside the update steps for the μ_k. Let $\mu_{k,d}$ be the dth component of μ_k. Then taking $\frac{\partial}{\partial mu_{k,d}}$ in (1) gives the equation

$$-2 \sum_{r_{nk}=1} (x_{n,d} - \mu_{k,d}) = 0 \implies \mu_{k,d} = \frac{1}{N_k} \sum_{r_{nk}=1} x_{n,d}$$

where $N_k = \#\{n \colon r_{nk} = 1\}$, and this is precisely the average of the dth coordinates of the points in cluster k.

b. These methods are related in that they both seek to present a *approximaton* of the data, which we can think of as a way of compressing or summarizing the data. K-means does so by replacing clusters of points by their means, and PCA does so by finding a projection.

To see how PCA might fail to give a good summary where K-means succeeds, consider a figure of the form below, where the big dots represent a large cluster of data.



PCA would project onto the vertical line indicated, which fails to capture the important structural information that the data is split into two sides! On the other hand, K-means would probably (it's random, so we can't say for sure) be able to identify these clusters.

PCA is good for eliminating useless dimensions. Imagine trying to classify data as in problem 4, but we have features with some completely random binary components. PCA will tend to choose a line for which all these useless components all project onto the same point, which is good, because it eliminates useless features while preserving meaningful ones. However K-means clustering will tend to cluster within hyperplanes defined by the different discrete values of these useless random components, especially if those hyperplanes are separated by large distances.

In general, PCA fails to distinguish capture essentially convex ball-shaped clusters, while K-means fails to identify clusters that are not essentially ball-shaped convex regions.

Problem 4.

a. (a)

k	Mean squared distance
1	1.97
2	1.80
3	1.81
4	1.77
5	1.52
6	1.37
7	1.42
8	1.25
9	1.31
10	1.26

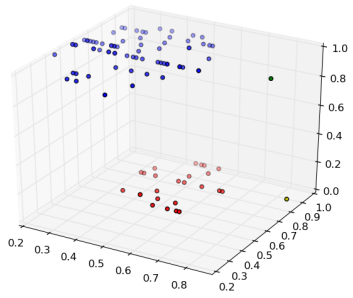
(b) It is difficult to tell, because the mean squared values always seem to be improving as the number of clusters increases (which should be the case). However,

the improvements seem to level off somewhat at 8 clusters, so that is about as reasonable a guess as we can make just from the above data.

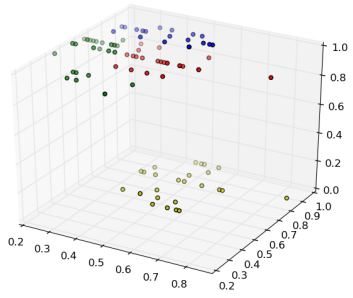
b. (a) Number of instances in each cluster by metric:

Cluster Number	Min metric	Max metric
0	73	19
1	25	23
2	1	32
3	1	26

Scatterplot of Min metric clusters:



Scatterplot of Max metric clusters:

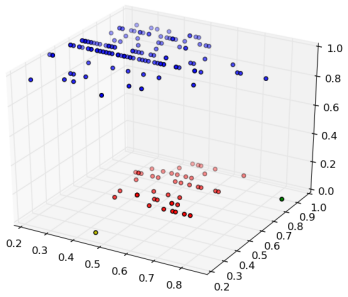


We see two hyperplanes, resulting from the fact that there is one discrete variable. The Min metric tends to separate out the clusters by discrete variables, which makes sense, while the Max metric has attempted to form convexish clusters. Visually, the Min metric looks better, but we should be careful not to be deceived by the nature of the data, as a small number of discrete variables will always look geometrically distinguished; this won't be so good in high dimensions.

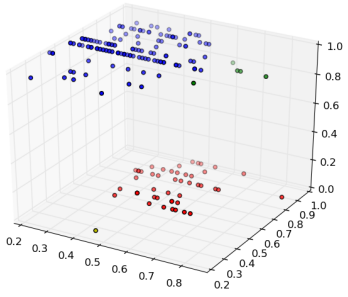
(b) Number of instances in each cluster by metric:

Cluster Number	mean metric	centroid metric
0	152	147
1	46	47
2	1	5
3	1	1

Scatterplot of mean metric clusters:

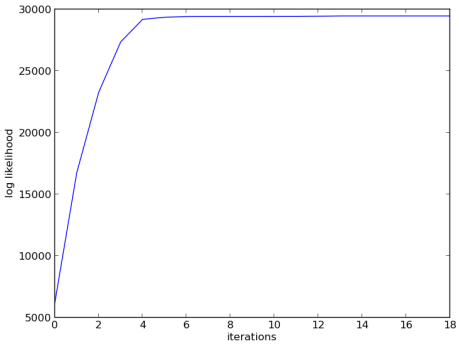


Scatterplot of centroid metric clusters:



These metrics behaved very similarly. The differences seem negligible. That's reasonable, considering the similarity of mean and centroid; both are middle ground between mean and max. These metrics distinguished hyperplanes of discrete variables even more starkly than did min.

c. (a) It takes autoclass 18 iterations to converge. The cluster sizes were 239,394,363 and 4.



(b)

(c) Autoclass takes about a minute to classify 1000 instances. K-means takes about a second. Thus K-means has a significantly better run-time than Autoclass.