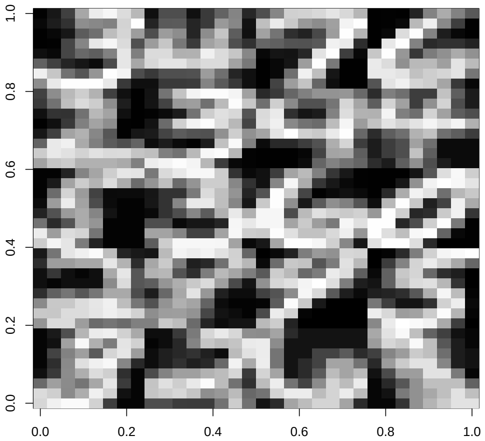
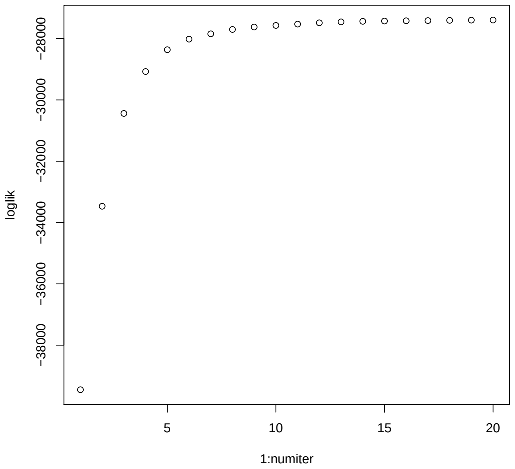
SMLDM HT 2014 - Part C Problem Sheet 4

1. Complete Question 6 of Problem Sheet 3 on implementing the EM algorithm to model a collection of handwritten digits using a mixture of products of Bernoullis.

   Answer: R Code:

```
X <- as.matrix(read.table("usps.txt"))
n <- dim(X)[1]
p <- dim(X)[2]
K <- 30
alpha <- 1
beta  <- 1
numiter <- 20
loglik <- matrix(0,1,numiter)

hinton <- function(X,n,m,x,y) {
  I <- matrix(0,x*n,y*m)
  for (i in 0:(n-1)) {
    for (j in 0:(m-1)) {
      I[i*x+(1:x),j*y+(1:y)] <- X[1+i+j*n,]
    }
  }
  image(I,col=grey(seq(0, 1, length = 256)))
}
# initialize E step
Q <- matrix(0,n,K)
for (i in 1:n) {
  c <- ceiling(runif(1)*K)
  Q[i,c] <- 1
}
for (iter in 1:numiter) {
  # actually do M step first, since E step initialized
  nk    <- matrix(1,1,n) %*% Q
  phikj <- (alpha + t(Q) %*% X) / ((2*alpha + t(nk)) %*% matrix(1,1,p))
  pik   <- (beta + nk) / (K*beta+n)

  # now E step
  Sik <- matrix(1,n,1)%*%log(pik)+X%*%log(t(phikj))+(1-X)%*%log(t(1-phikj))
  mi  <- matrix(apply(Sik,1,max),n,1)
  Sik <- Sik - (mi %*% matrix(1,1,K))
  Sik <- exp(Sik)
  si  <- Sik %*% matrix(1,K,1)
  Q   <- Sik / (si %*% matrix(1,1,K))

  # calculate log likelihood
  loglik[iter] <- sum(mi + log(si))

  hinton(phikj,4,5,8,8)
  Sys.sleep(.0)
}
dev.new()
plot(1:numiter,loglik)
```





(a) Yes, it takes approximately 20 iterations to converge.

(b) Yes, generally, but hard to see due to low resolution. A few digits seem over-represented, and some under.

(c) No, each run produce somewhat different answer, but generally qualitatively similar.

(d) Increasing number of clusters increases the log likelihood.

(e) 20 seems ok. Will need better methods to determine number of clusters. Judging from the cluster means, some digits are over-represented (too many clusters), but some are under (too few).

2. Assume you have used some data $D = \{x_i, y_i\}_{i=1}^n$ where $x_i \in \mathbb{R}^p$ and $y_i \in \{1, 2, ..., K\}$ to train a classifier. We are interested in classifying a new input vector x_t. However we have only been able to collect $p-1$ features, say $(x_{t2}, ..., x_{tp})$ and x_{t1} is missing. Explain whether or not it is possible to use your classifier to classify this incomplete input vector in the following scenarios. If it is possible, how do you classify the incomplete test vector?

You may find the Gaussian identities cheat sheet useful if you are not very familiar with properties of the multivariate normals. You do not need to calculate integrals in this question.

(a) A naïve Bayes model, with

$$f(x|\phi_k) = \prod_{j=1}^{p} f(x_j|\phi_{kj}),$$

i.e. conditioned upon $Y = k$, you assume that the features are independent and feature x_j follows a distribution with density/probability mass function $f(x_j|\phi_{kj})$.

Answer: Yes. Since features are assumed independent in naïve Bayes, the probability of the observed test features is

$$\prod_{j=2}^{p} f(x_{tj}|\phi_{kj})$$

So that conditioned on these the posterior probability of $y_t = k$ is proportional to

$$\pi_k \prod_{j=2}^{p} f(x_{tj}|\phi_{kj})$$

i.e. we simply ignore feature 1.

(b) A LDA model, i.e.

$$f(x|\phi_k) = \mathcal{N}(x; \mu_k, \Sigma)$$

Answer: Yes. The conditional density under class k of $(x_{t2}, \dots, x_{tp})$ is simply a multivariate normal, with mean $\mu'_k = (\mu_{k2}, \dots, \mu_{kp})^\top$ and covariance matrix Σ' obtained from Σ by dropping the first row and column. The posterior probability of $y_t = k$ is just this times π_k and normalized.

(c) Generally, which conditions on $f(x|\phi_k)$ are necessary to allow us to implement easily, that is without using numerical integration, a probabilistic classifier like LDA and naïve Bayes in the presence of missing features?

Answer: Generally we need to be able to compute analytically the marginals of the class-conditional distributions, i.e.

$$f(x_t^{(J)}|\phi_k) = \int f((x_t^{(J)}, x_t^{(L)})|\phi_k) dx_t^{(L)}$$

where $x_t^{(J)}$ are the observed entries of x_t, and $x_t^{(L)}$ are the unobserved ones.

(d) A logistic regression model, i.e.

$$p(Y = y|X = x) = s(y(a + b^\top x))$$

where $y \in \{+1, -1\}$.

Answer: It would not directly be possible, since logistic regression does not model the full joint distribution so does not give predictions when the data vector is partially observed.

3. Consider using logistic regression to model the conditional distribution of binary labels $Y \in \{+1, -1\}$ given data vectors X.

(a) Suppose that the data is linearly separable, i.e. there is a hyperplane separating the two classes. Show that the maximum likelihood estimator is ill-defined.

Answer: Since the data is linearly separable, there is a scalar α and vector β such that $\alpha + \beta^\top X < 0$ whenever $Y = -1$ and $\alpha + \beta^\top X > 0$ whenever $Y = +1$. Let $c > 0$. the log likelihood at $a = c\alpha$, $b = c\beta$ is

$$\sum_{i=1}^{n} -\log(1 + \exp(-y_i(c\alpha + c\beta^\top x_i)))$$

Differentiating with respect to c,

$$\sum_{i=1}^{n} s(cy_i(\alpha + \beta^\top x_i))y_i(\alpha + \beta^\top x_i)$$

Noting that this is always positive, the log likelihood would be maximized only when $c \to \infty$.

(b) Suppose the first entry in X is binary, i.e. it takes on only values 0 or 1. Suppose that in the dataset, whenever $y_i = -1$, the corresponding entry $x_{i1} = 0$, but there are data cases with $y_i = +1$, and x_{i1} taking on both values. Show that the maximum likelihood estimator of b_1 is ∞, but that the dataset need not be linearly separable.

Answer: For a data vector x_i, let $z_i = (x_{i2}, \ldots, x_{ip})^\top$. Let $c = (b_2, \ldots, b_p)^\top$. The likelihood is

$$\begin{aligned} &\prod_{i:y_i=-1} s(-(a + b_1 x_{i1} + c^\top z_i)) \prod_{j:y_j=+1} s(a + b_1 x_{j1} + c^\top z_j) \\ = &\prod_{i:y_i=-1} s(-(a + c^\top z_i)) \prod_{j:y_j=+1} s(a + b_1 x_{j1} + c^\top z_j) \end{aligned}$$

Note that the likelihood is an increasing function of b_1, so the ML estimator is at $b_1 = \infty$.

It is sufficient to give an example of a dataset which is not linearly separable but satisfies the condition. Try $x_1 = (0,0)^\top$, $x_2 = (0,2)^\top$, $x_3 = (1,0)^\top$, $x_4 = (0,1)^\top$ along with $y_1 = y_2 = -1$ and $y_3 = y_4 = 1$.

Thus such overfitting behaviour which leads to ill-defined ML estimators can occur even when the dataset is not linearly separable.

4. The receiver operating characteristic (ROC) curve plots the sensitivity against the specificity of a binary classifier as a threshold for discrimination is varied. The larger the area under the ROC curve (AUC), the better the classifier is.

Suppose the data space is $\mathbb{R}$, the class-conditional densities are $f_0(x)$ and $f_1(x)$ for $x \in \mathbb{R}$ and for the two classes 0 and 1, and that the optimal Bayes classifier is to classify $+1$ when $x > c$ for some threshold c, which varies over $\mathbb{R}$.

(a) Give expressions for the specificity and sensitivity of the classifier at threshold c.

Answer: At a threshold c, the sensitivity is the true positive rate, which is:

$$\int_c^\infty f_1(x)dx$$

while the specificity is the true negative rate:

$$\int_{-\infty}^c f_0(x)dx$$

(b) Show that the AUC corresponds to the probability that $X_1 > X_0$, if data items X_1 and X_0 are independent and comes from class 1 and 0 respectively.

Answer: Define the function

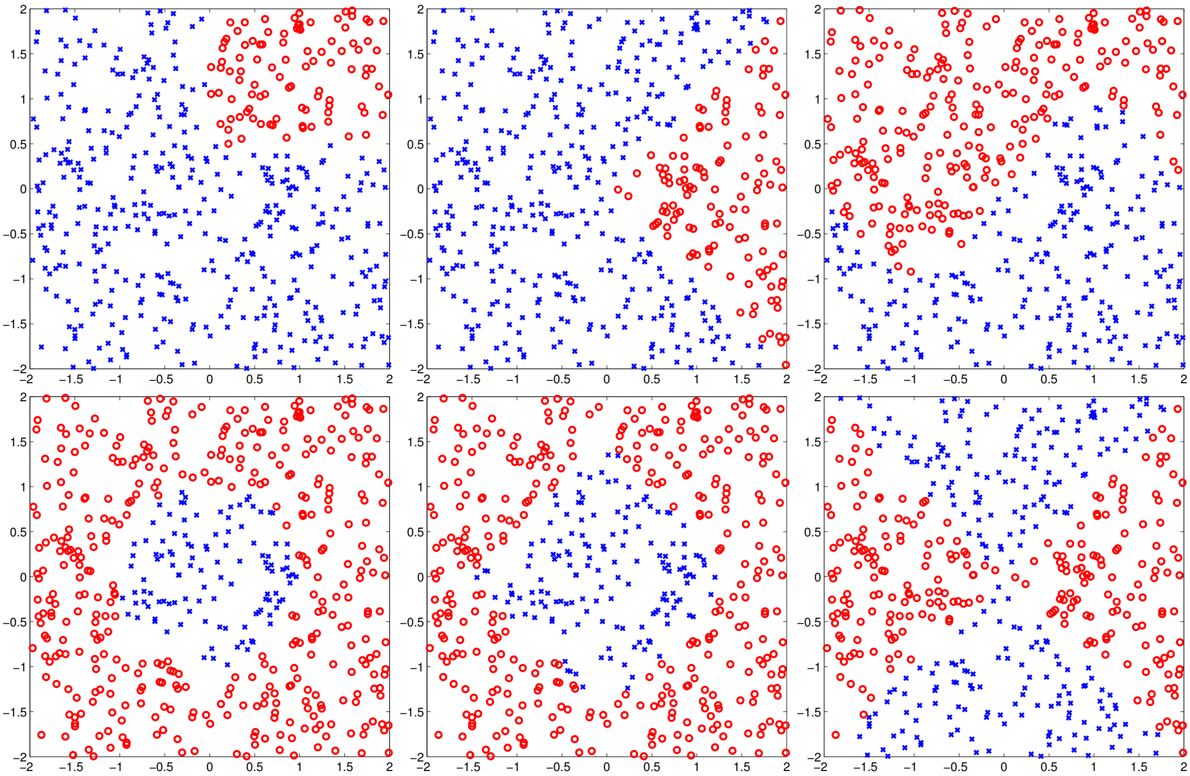
$$F_0(c) = \int_{-\infty}^c f_0(x)dx$$

which is the CDF of the 0 class so is invertible. At a specificity level s, the corresponding threshold is $F_0^{-1}(s)$ and so the AUC is

$$\begin{aligned}
&\int_0^1 \int_{F_0^{-1}(s)}^{\infty} f_1(x)dxds \\
=&\int_{-\infty}^{\infty} \int_z^{\infty} f_1(x)dx f_0(z)dz \qquad \text{by change of variable } s \mapsto F_0^{-1}(s) = z \\
=&\mathbb{P}(X_1 > X_0)
\end{aligned}$$

which is the probability of $X_1 > X_0$.

5. For each of the datasets below, find a non-linear function $\phi(x)$ which makes the data linearly separable, and the discriminant function (linear in $\phi(x)$) which will classify perfectly. Briefly explain your answer. You may assume, if a boundary looks like a straight line, or a function you are familiar with, that it is.



Answer: From left to right and top to bottom:

(a) Looks like we want $x_1 > 0$ and $x_2 > .5$. So use $\phi_1(x) = (\text{sign}(x_1), \text{sign}(x_2 - .5))^\top$. Then perfect classification can be obtained by $\text{sign}(x_1) + \text{sign}(x_2 - .5) \geq 2$.

(b) Looks like we want $x_1 < x_2$ and $x_1 > -x_2$. Use $\phi_2(x) = (\text{sign}(x_1 - x_2), \text{sign}(x_1 + x_2))^\top$ and classify by $-\text{sign}(x_1 - x_2) + \text{sign}(x_1 + x_2) \geq 2$.

(c) Looks like $x_2 < \sin(x_1)$, so $\phi_3(x) = (x_2, \sin(x_1))^\top$ and discriminate via $\sin(x_1) - x_2 > 0$.

(d) Looks like a circle, so we want $\sqrt{x_1^2 + x_2^2} > 1$. Use $\phi_4(x) = \sqrt{x_1^2 + x_2^2} > 1$.

(e) Looks like a diamond, so we want $|x_1| + |x_2| \leq 1$. Use $\phi_5(x) = |x_1| + |x_2|$.

(f) The two lines are $x_1 - x_2 = 0$ and $x_2 + x_1 = 0$. The red region are when $(x_1 - x_2)$ and $(x_2 + x_1)$ have different signs. So $\phi_6(x) = \text{sign}((x_1 - x_2)(x_2 + x_1))$.