Solution for Assignment 2: MLE, EM, Regression

10-701/15-781: Machine Learning (Fall 2004)

Due: Oct. 14th 2004, Thursday, In class,

Question 1. Maximum Likelihood Estimation (20pts)

Suppose X is a binary random variable that takes value 0 with probability p and value 1 with probability $1-p$. Let $X_1, \ldots, X_n$ be iid samples of X.

1.1 (5pts) Compute an MLE estimate of p (denote it by $\hat{p}$).

1.2 (5pts) Is $\hat{p}$ an unbiased estimate of p? Prove the answer.

1.3 (5pts) Compute the expected square error of $\hat{p}$ in terms of p.

1.4 (5pts) Prove that if you know that p lies in the interval $[\frac{1}{4}; \frac{3}{4}]$ and you are given only $n = 3$ samples of X, then $\hat{p}$ is an inadmissible estimator of p when minimizing the expected square error of estimation. (An estimator δ of a parameter θ is said to be *inadmissible* when there exists a different estimator δ' such that $R(\theta, \delta') \leq R(\theta, \delta)$ for all θ and $R(\theta, \delta') < R(\theta, \delta)$ for some θ, where $R(\theta, \delta)$ is a risk function and in this problem it is the expected square error of the estimator).

Answer:

1.1 $\hat{p} = \arg\max_p P(X_1, \ldots, X_n|p) = \arg\max_p \prod_{i=1}^n P(X_i|p) = \arg\max_p p^k(1-p)^{n-k} =$
$\arg\max_p log(p^k(1-p)^{n-k}) = \arg\max_p(k * logp + (n-k) * log(1-p))$,
where k is the number of 0's in $X_1, \ldots, X_n$
$\frac{d(k*logp+(n-k)*log(1-p))}{dp} = \frac{k}{p} - \frac{n-k}{1-p} = 0$
Hence, $k(1-p) - (n-k)p = 0$
$\hat{p} = \frac{k}{n}$

1.2 $E\{\hat{p}\} = E\{\frac{k}{n}\} = \frac{E\{k\}}{n} = \frac{np}{n} = p$,
where we used the fact that $E\{k\} = np$ because k is a binomial random variable.
Alternatively, $E\{k\} = E\{n - \sum_{i=1}^n X_i\} = n - \sum_{i=1}^n E\{X_i\} = n - n(1-p) = np$.

1.3 $E\{(\hat{p}-p)^2\} = E\{\hat{p}^2\} - 2 * E\{\hat{p}\}p + p^2 = \frac{E\{k^2\}}{n^2} - 2p^2 + p^2 =$
$\frac{Var\{k\}+E^2\{k\}}{n^2} - p^2 = \frac{np(1-p)+(np)^2}{n^2} - p^2 = \frac{p}{n}(1-p)$,
where we used the fact that $Var\{k\} = np(1-p)$ because k is a binomial random variable,
and $Var\{k\} = E\{k^2\} - E^2\{k\}$.

1.4 Consider another estimator $\tilde{p} = 1/2$.
$E\{(\tilde{p}-p)^2\} = (1/2-p)^2$
For $p = 1/2$ we have $E\{(\tilde{p}-p)^2\} = 0 < E\{(\hat{p}-p)^2\} = 1/12$
We now need to show $E\{(\tilde{p}-p)^2\} \leq E\{(\hat{p}-p)^2\}$ over $p \in [1/4; 3/4]$
$E\{(\tilde{p}-p)^2\} - E\{(\hat{p}-p)^2\} = (1/2-p)^2 - 1/3 * p(1-p) = 1/4 - 4/3p + 4/3p^2$
This is a parabola going up, so we need to show that it lies below or equal to zero for $p \in [1/4; 3/4]$
It is equivalent to showing that it is below or equal to 0 at boundary points.
In fact it is: at both $p = 1/4$ and $p = 3/4$ $1/4 - 4/3p + 4/3p^2 = 0$

Question 2. EM (25pts)

For the following questions, please give clear step by step derivation.

2.1 (12pts) Suppose that the p.d.f. of a random variable X has a 2-component mixture form:

$$p_\alpha(x) = \alpha * p_1(x) + (1 - \alpha) * p_2(x) \tag{1}$$

One component is the density model $p_1(x)$ and the other component is the density model $p_2(x)$. We know both $p_1(x)$ and $p_2(x)$. We do not know α. Given that { $x_1, x_2, ..., x_n$ } are iid samples from the distribution of X, please give an EM algorithm for estimating α. (Describe the E-step and M-step clearly in your answer).

2.2 (13 pts) Suppose that $Y_1 \sim exp(1/\theta_1)$ and $Y_2 \sim exp(1/\theta_2)$, and $\theta_1 \neq \theta_2$. Y_1 and Y_2 are independent. Let $X = Y_1 + Y_2$ denote the sum of Y_1 and Y_2, Given that { $x_1, x_2, ..., x_n$ } are iid samples from the distribution of X.

- Derive an expression for the density of X in terms of θ_1 and θ_2
(Hint1: The density of Y_1 is $f_{\theta_1}(y) = \theta_1 e^{-\theta_1 y}$, similarly for Y_2)
(Hint2: You could first derive CDF of X, $F(x) = P(Y_1 + Y_2 < x) = \int_0^x \int_0^{x-y_1} f_{\theta_1}(y_1) f_{\theta_2}(y_2) dy_2 dy_1$)

- Derive the E-step and M-step, and give explicit expressions for the parameter updates in the EM process for computing the MLE of θ_1 and θ_2.

Answer:

2.1 The question is a simple case of Bilmes's paper Page 3 and 4. (Jeff. Bilmes, "A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models")

Let Y=$(y_1, y_2, ..., y_N)$ inform us which component "generated" each data item. $y_i \in \{1, 2\}$ for each i, and $y_i = k$ if the i^{th} sample was generated by the k^{th} mixture component.

E-step: We calculate the expectation of the complete likelihood:
$Q(\theta, \theta^{old}) = E_{P(Y|X,\theta^{old})}[log(L(\theta|X, Y))] = E_{P(Y|X,\theta^{old})}[\sum_{i=1}^N log(\alpha_{y_i} p_{y_i}(x_i|\theta_{y_i}))]$
*note: Y is a vector $(y_1, y_2, ..., y_N)$; And y_i only depends on x_i
$\therefore Q(\theta, \theta^{old}) = \sum_{i=1}^N \{E_{P(y_i|x_i,\theta^{old})}[log(\alpha_{y_i} p_{y_i}(x_i|\theta_{y_i}))]\} = \sum_{i=1}^N \{\sum_{y_i=1}^M [log(\alpha_{y_i} p_{y_i}(x_i|\theta_{y_i})) * p(y_i|x_i, \theta^{old})]\}$
*note: Then we could use l to substitute y_i
$Q(\theta, \theta^{old})$
$= \sum_{i=1}^N \{\sum_{l=1}^M [log(\alpha_l p_l(x_i|\theta_l)) * p(l|x_i, \theta^{old})]\}$
$= \sum_{i=1}^N \{\sum_{l=1}^M [log(\alpha_l) * p(l|x_i, \theta^{old})]\} + \sum_{i=1}^N \{\sum_{l=1}^M [log(p_l(x_i|\theta_l)) * p(l|x_i, \theta^{old})]\}$

For our problem, $M = 1$, $\alpha_1 = alpha$ and $\alpha_2 = 1 - alpha$. Maximize the expression in terms of $alpha$,

$$\frac{\partial}{\partial \alpha}[\sum_{i=1}^N log(\alpha) * p(y_i = 1|x_i, theta^{old}) + \sum_{i=1}^N log(1 - \alpha) * p(y_i = 2|x_i, theta^{old})] = 0 \tag{2}$$

$$\frac{\sum_{i=1}^N p(y_i = 1|x_i, theta^{old})}{\alpha} = \frac{\sum_{i=1}^N p(y_i = 2|x_i, theta^{old})}{1 - \alpha}] \tag{3}$$

$$\alpha = \frac{\sum_{i=1}^N p(y_i = 1|x_i, theta^{old})]}{\sum_{i=1}^N p(y_i = 1|x_i, theta^{old}) + \sum_{i=1}^N p(y_i = 2|x_i, theta^{old})} \tag{4}$$

$$\alpha = \frac{\sum_{i=1}^{N} p(y_i = 1|x_i, theta^{old})]}{N} \tag{5}$$

$p(y_i = 1|x_i, theta^{old}) = \frac{\alpha p_1(x_i)}{\alpha p_1(x_i)+(1-\alpha)p_2(x_i)}$
$p(y_i = 2|x_i, theta^{old}) = \frac{(1-\alpha)p_2(x_i)}{\alpha p_1(x_i)+(1-\alpha)p_2(x_i)}$

2.2 First to get pdf of X:

- The density of Y_1 is $f_{\theta_1}(y) = \theta_1 e^{-\theta_1 y}$ similarly for Y_2)
- First derive CDF of X

$$F(x) = P(Y_1 + Y_2 < x) = \int_0^x \int_0^{x-y_1} f_{\theta_1}(y_1) f_{\theta_2}(y_2) dy_2 dy_1 \tag{6}$$

$$F(x) = \int_0^x \theta_1 e^{-\theta_1 y_1} \int_0^{x-y_1} \theta_2 e^{-\theta_2 y_2} dy_2 dy_1 \tag{7}$$

$$F(x) = 1 - \frac{\theta_2 e^{-\theta_1 x} - \theta_1 e^{-\theta_2 x}}{\theta_2 - \theta_1} \tag{8}$$

$$p(x) = \frac{\partial F(x)}{\partial x} = \frac{\theta_1 \theta_2}{\theta_2 - \theta_1}[e^{-\theta_1 x} - e^{-\theta_2 x}] \tag{9}$$

Then let us derive EM steps for estimating parameters: θ_1 and θ_2

- Way1: The same framework as the Reference paper and the first page of this note.
  - E step: Expectation of the complete likelihood
  The expectation of complete log-likelihood is:

$$Q(\theta, \theta^{old}) = E_{P(Y|X,\theta^{old})}[log(L(\theta|X, Y))] = E_{P(Y|X,\theta^{old})}[\sum_{i=1}^{N} log(f_{\theta_1}(y_{i,1}) f_{\theta_2}(y_{i,2}))] \tag{10}$$

$$Q(\theta, \theta^{old}) = E_{P(Y|X,\theta^{old})}\{\sum_{i=1}^{N}[log(\theta_1) - \theta_1 * y_{i,1} + log(\theta_2) - \theta_2 * y_{i,2}]\} \tag{11}$$

  - M step: Maximization of the above expectation
  Maximize the above Q, in terms of $theta_1$, $theta_2$, we could get:

$$\frac{\partial Q(\theta, \theta^{old})}{\partial \theta_1} = 0 \Rightarrow \theta_1 = \frac{n}{\sum_{i=1}^{n} E_{p(y_{i,1}|x_i,\theta^{old})}[y_{i,1}]} \tag{12}$$

$$\frac{\partial Q(\theta, \theta^{old})}{\partial \theta_2} = 0 \Rightarrow \theta_2 = \frac{n}{\sum_{i=1}^{n} E_{p(y_{i,1}|x_i,\theta^{old})}[y_{i,2}]} \tag{13}$$

- Way2: Use the simple thinking style of EM as the GMM slides.
  - E step: Get expected value of the hidden variables $y_{i,1}$
  See the following derivation for $E_{p(y_{i,1}|x_i,\theta^{old})}[y_{i,1}]$
  - M step: Based on the expected value of $y_{i,1}$, derive θ
  Due to $Y_1 \sim exp(\frac{1}{\theta_1})$, the MLE estimation of exponential model's parameter $\frac{1}{\theta_1}$ is the sample mean of Y_1, then we could get that:

$$\frac{1}{\theta_1} = \frac{1}{n}\sum_{i=1}^{n} E_{p(y_{i,1}|x_i,\theta^{old})}[y_{i,1}] \tag{14}$$

$$\theta_1 = \frac{n}{\sum_{i=1}^{n} E_{p(y_{i,1}|x_i,\theta^{old})}[y_{i,1}]} \tag{15}$$

Same for θ_2:

$$\theta_2 = \frac{n}{\sum_{i=1}^{n} E_{p(y_{i,1}|x_i,\theta^{old})}[y_{i,2}]} \tag{16}$$

In both the above two ways, we need to get the $E_{p(y_{i,1}|x_i,\theta^{old})}[y_{i,1}]$:
Y_1 is our hidden variable. Then for the given $\{ x_1, x_2, ..., x_n \}$ samples, there would be corresponding $\{ y_{1,1}, y_{2,1}, ..., y_{n,1} \}$.

$$p(y_{i,1}|x_i,\theta^{old}) = \frac{p(x_i, y_{i,1})|\theta^{old}}{p(x_i|\theta^{old})} = \frac{\theta_1^{old} e^{-\theta_1^{old} y_{i,1}} \theta_2^{old} e^{-\theta_2^{old}(x_i - y_{i,2})}}{p(x_i|\theta^{old})} \tag{17}$$

$$p(y_{i,1}|x_i,\theta^{old}) = (\theta_2^{old} - \theta_1^{old}) \frac{e^{y_{i,1}(\theta_2^{old} - \theta_1^{old})}}{e^{x_i(\theta_2^{old} - \theta_1^{old})} - 1} \tag{18}$$

$$E_{p(y_{i,1}|x_i,\theta^{old})}[y_{i,1}] = \int_0^{x_i} y_{i,1} p(y_{i,1}|x_i,\theta^{old}) dy_{i,1} = \frac{x_i e^{x_i(\theta_2^{old} - \theta_1^{old})}}{e^{x_i(\theta_2^{old} - \theta_1^{old})} - 1} - \frac{1}{\theta_2^{old} - \theta_1^{old}} \tag{19}$$

$$E_{p(y_{i,1}|x_i,\theta^{old})}[y_{i,2}] = E_{p(y_{i,1}|x_i,\theta^{old})}[x_i - y_{i,1}] = x_i - E_{p(y_{i,1}|x_i,\theta^{old})}[y_{i,1}] \tag{20}$$

Question 3. Gaussian mixtures (35pts)

In this problem you will implement a Gaussian mixture model algorithm and will apply it to the problem of clustering gene expression data. Gene expression measures the levels of messenger RNA (mRNA) in the cell. The data you will be working with is from a model organism called yeast, and the measurements were taken to study the cell cycle system in that organism. The cell cycle system is one of the most important biological systems playing a major role in development and cancer.

All implementation should be done in Matlab. At the end of each sub-problem where you need to implement a new function we specify the prototype of the function.

3.1 Download the file 'alphaVals.txt'. This file contains 18 time points (every 7 minutes from 0 to 119) measuring the log expression ratios of 745 cycling genes. Each row in this file corresponds to one of the genes. Also, download the file 'geneNames.txt' which contains the names of these genes. For some of the genes, we are missing some of their values due to problems with the microarray technology (the tools used to measure gene expression). These cases are represented by values greater than 100.

3.2 (17pts) Implement (in matlab) an EM algorithm for learning a mixture of five (18-dimensional) Gaussians. It should learn means, convariance matrices and weights for each of the Gaussian. You can assume, however, independence between the different data points, resulting in a diagonal covariance matrix. How can you deal with the missing data? Why is this correct? Plot the centers identified for each of the five classes. Each center should be plotted as a time-series of 18 time points. Hand this plot with your solutions.

Here is the prototype of the matlab function you need to implement:

$$function[mu, s, w] = emcluster(x, k, ploton); \tag{21}$$

x is input data, where each row is an 18-dimensional sample. Values above 100 represent missing values. k is the number of desired clusters. $ploton$ is either 1 or 0. If 1, then before returning the function plots log-likelihood of the data after each EM iteration (the function will have to store the log-likelihood of the data after each iteration, and then plot these values as a function of iteration number at the end). If 0, the function does not plot anything. The function outputs mu, a matrix with k rows and 18 columns (each row is a center of a cluster), s is also k by 18, with each row being diagonal elements of the corresponding covariance matrix, and w is a column vector of size k, where $w(i)$ is a weight for ith cluster.

Answer: We have put a student code online. The implementation is pretty clear in terms of each step of the GMM iteration. A lot of students gave much more concise code than the one we chose. The purpose of the sample code is for those students who are not quite familiar with matlab coding.

The plot of the log-likelihood should be increasing.

The plots of the centers of each cluster should look like a sinusoid shape though with different phases (starting at a different point in the time series.

3.3 (3pts) How many more parameters would you have had to assign if we remove the independence assumption above? Explain.

Answer: The number of clusters times the number of covariances,
which is $k * ((d-1)+(d-2)+\ldots+1) = \frac{kd}{2}(d-1)$,
where $d = 18$ in our case.

3.4 (8pts) Suggest and implement a method for determining the number of Gaussians (or classes) that are the most appropriate for this data. Please confine the set of choices to values in between 2 and 7. (Hint: the method can use an empirical evaluation of clustering results for each possible number of classes). Explain the method.

Here is the prototype of the matlab function you need to implement:

$$function[k, mu, s, w] = clust(x); \tag{22}$$

x is input data, where each row is an 18-dimensional sample. Once again values above 100 represent missing values. k is the number of classes selected by the function. mu, s and w are defined as in 3.2.

Answer: This is essentially a model selection question. You could use different model selection ways to solve it.
- cross validation
- train-test
- Minimum description length (One student used. If you do not know it, do not bother)
- BIC (Some student used. If you do not know it, do not bother)

3.5 (5pts) Use the Gaussians determined in (d) to perform hard clustering of your data by finding, for each gene i the Gaussian j that maximizes the likelihood: $p(i|j)$. Use the function 'printSelectedGenes.m' to write the names of the genes in each of the clusters to a separate file.

Here is the prototype of the matlab function you need to implement:

$$function[c] = hardclust(x, k, mu, s, w); \tag{23}$$

x is defined as before. k, mu, s, w are the output variables from the function written in 3.4 and are therefore defined there. c is a column vector of the same length as the number of rows in x. For each row, it should indicate the cluster the corresponding gene belongs to. The function should also write out files as specified above. The filenames should be: clust1, clust2, ..., clustk.

Answer: For each data point, assign the cluster that has the maximum probability for this point.

3.6 (2 pts) Use compSigClust.m to perform the statistical significance test (everything is already implemented here, so just use the function). Hand in a printout with the top three categories for each cluster (this is the output of compSigClust.m).

Answer: Just run the code we provided on the cluster files you got above.

Question 4. Regression (20pts)

Linear regression models a real-valued output Y given an input vector X as

$$Y|X \sim Normal(\mu(X), \sigma^2)$$

where the mean is a linear function of the input: $\mu(X) = \beta^T X = \beta_0 + \beta_1 X_1 + ... + \beta_p X_p$

Logistic regression models a binary output Y by

$$Y|X \sim Bernoulli(\theta(X))$$

where the Bernoulli parameter is related to $\beta^T X$ by the logit transformation

$$logit(\theta(X)) \equiv \log\left(\tfrac{\theta(X)}{1-\theta(x)}\right) = \beta^T X$$

Given data $\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$, for each of the two regression models above , show that at the MLE $\hat{\beta}$

$$\sum_{i=1}^{n} x_i * y_i = \sum_{i=1}^{n} x_i * E[\text{ Y } |X = x_i, \beta = \hat{\beta}]$$

Answer:

4.1 For linear regression: (the general way is by 'Maximum Likelihood Estimation'.)

$$Y|X \sim Normal(\mu(X), \sigma^2)$$

We could write the log likelihood as:

$$LL = log(\prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} exp(-\frac{(y_i - \mu(x_i))^2}{2\sigma^2})) = \sum_{i=1}^{n} log(\frac{1}{\sqrt{2\pi}\sigma} exp(-\frac{(y_i - \beta^T x_i)^2}{2\sigma^2})) \tag{24}$$

$$\frac{\partial LL}{\partial \beta} = 0 \Rightarrow \frac{\partial \sum_{i=1}^{n} (y_i - \beta^T x_i)^2}{\partial \beta} = 0 \tag{25}$$

$$\Rightarrow \sum_{i=1}^{n} x_i * (y_i - \beta^T x_i) = 0 \Rightarrow \sum_{i=1}^{n} x_i * y_i = \sum_{i=1}^{n} x_i * (\hat{\beta}^T x_i) \tag{26}$$

Note: $E[\text{ Y } |X = x_i, \beta = \hat{\beta}] = \mu(x_i) = \hat{\beta}^T x_i \Rightarrow \sum_{i=1}^{n} x_i * y_i = \sum_{i=1}^{n} x_i * E[\text{ Y } |X = x_i, \beta = \hat{\beta}]$

4.2 For logistic regression: (the general way is still by 'Maximum Likelihood Estimation'.)
Logistic regression models a binary output Y by $Y|X \sim Bernoulli(\theta(X))$

$$\log\left(\tfrac{\theta(X)}{1-\theta(x)}\right) = \beta^T X \Rightarrow \theta(x) = \tfrac{\exp(\beta^T x)}{1+\exp(\beta^T x)} \text{ and } 1 - \theta(x) = \tfrac{1}{1+\exp(\beta^T x)}$$

We could write the log likelihood as:

$$LL = log(\prod_{i=1}^{n}\{\theta(x_i)^{y_i} * (1-\theta(x_i))^{1-y_i}\}) = \sum_{i=1}^{n}\{y_i * log(\theta(x_i)) + (1-y_i) * log(1-\theta(x_i))\}) \tag{27}$$

Plug $\theta(x_i)$ and $1-\theta(x_i)$ in, we get

$$LL = \sum_{i=1}^{n}\{y_i * (\beta^T x_i) - \log(1+\exp(\beta^T x_i))\} \tag{28}$$

$$\frac{\partial LL}{\partial \beta} = 0 \Rightarrow \sum_{i=1}^{n} x_i * y_i = \sum_{i=1}^{n} x_i * \frac{\exp(\beta^T x_i)}{1+\exp(\beta^T x_i)} \tag{29}$$

Note: by the property of Bernoulli distribution: $E[\text{ Y } |X = x_i, \beta = \hat{\beta}] = \theta(x_i) = \frac{\exp(\beta^T x_i)}{1+\exp(\beta^T x_i)}$

$$\Rightarrow \textstyle\sum_{i=1}^{n} x_i * y_i = \sum_{i=1}^{n} x_i * E[\text{ Y } |X = x_i, \beta = \hat{\beta}]$$

Note:
Actually in the above solutions, the full log likelihood function should look like the following first:

log-likelihood
$= log(\prod_{i=1}^{n}\{p(x_i, y_i)\})$
$= log(\prod_{i=1}^{n}\{p_{(Y|X)}(y_i|x_i) * p_X(x_i)\})$
$= log(\{\prod_{i=1}^{n} p_{(Y|X)}(y_i|x_i)\} * \{\prod_{i=1}^{n} p_X(x_i)\})$
$= log(\{\prod_{i=1}^{n} p_{(Y|X)}(y_i|x_i)\}) + log(\{\prod_{i=1}^{n} p_X(x_i)\})$
$= LL + LL_x$

Because LL_x do not involve with the parameter, in our maximization, we could just consider maximizing LL .