CSci 5521-001 Expectation – Maximization *v.1* Fall 2020

We sketch the Expectation Maximization Algorithm. This is an unsupervised algorithm that tries to learn a model to represent some given data. The input consists of some set of data samples $\mathbf{x}_1, \ldots, \mathbf{x}_n$ in d-dimensional space. We also start with a supposition that the data comes from k classes and that the distribution of data within each class follows a probability distribution of known form. What is not known are the specific parameters for those within-class probability distributions, the class priors, or anything that indicates membership of each sample in a particular class. Since the classes are unknown, they should really be called clusters.

In this writeup, we discuss the specific example where we have $k = 2$ classes A and B with unknown priors, and we assume the distribution of data within each class is gaussian with spherical covariances* but with unknown means and variances. So the unknown parameters we must learn are, for each class $\mathtt{C} = A, B$: the class prior $P_{\mathtt{C}}$, the within-class mean $\boldsymbol{\mu}_{\mathtt{C}}$, and the variance $\sigma^2_{\mathtt{C}}$.

In effect, we believe the data comes from two different classes or sources, and we would like to identify the properties of the individual classes $(\boldsymbol{\mu}_{\mathtt{C}}, \sigma^2_{\mathtt{C}})$, and the "prevalence" $P_{\mathtt{C}}$ for each class $\mathtt{C} = A, B$. For each sample datum $\mathbf{x}_i$, we would like to compute the relative "affiliation" of this datum to one class or the other. This takes the form of a "posterior" probability $\Pr(A|\mathbf{x}_i)$, $\Pr(B|\mathbf{x}_i)$.

The K-means algorithm can be considered a special case where the 'affiliations' of a datum is 'hard': either 1 or 0, the variance is fixed at 1 and the priors are equal. Only the class means float to maximize the 'probabilities'.

The difference with previous Bayes estimate of an unknown probability distribution/density is that we start with a fixed sample set which is re-used over and over. We also have a "multilayered" probability distribution in which we pretend that each data sample has been drawn by first selecting the class $\mathtt{C} = A$ or B with probability $P_{\mathtt{C}}$ and then selecting a point according to the within-class probability density $P(\mathbf{x}|\mathtt{C})$.

The expectation maximization Algorithm is a general paradigm which cycles between two steps: the E step and the M step. In the E step, we compute for each sample datum the probability of being generated by the current probability distributions $P_{\mathtt{C}}, \boldsymbol{\mu}_{\mathtt{C}}, \sigma^2_{\mathtt{C}}$, and infer posterior probabilities for the datum's class membership $\Pr(\mathtt{C}|\mathbf{x}_i)$. In the M step, we adjust the probability parameters to maximize the total probability of sample set given the class membership probabilities.

Problem Setup

1. $\mathbf{x}_1, \cdots, \mathbf{x}_n$ set of d-dimensional sample training data drawn independently and identically by first choosing a class A and B at random, then drawing a point according to that class's probability density. The probability distribution for points in class A is $p(\mathbf{x}|A)$ and the prior for A is P_A. The numerical parameters defining these distributions are denoted collectively by $\boldsymbol{\Theta}$. If the distributions are gaussian (with circular covariances), then the parameters are $\boldsymbol{\Theta} = \{\boldsymbol{\mu}_A, \sigma_A, P_A, \boldsymbol{\mu}_B, \sigma_B, [P_B]\}$, where $P_A + P_B = 1$.

   The goal is to find the values of the parameters $\boldsymbol{\Theta}$ that best explain the data. As a by-product, we would also like to learn the likely affiliation of each sample: which samples likely come from class A as opposed to B.

Preliminaries – Compute Probability of Observing Training Set Given $\boldsymbol{\Theta}$.

2. The total probability that all the points $\mathbf{x}_1, \cdots, \mathbf{x}_n$ could have been generated from these two ***Total Probability***

*by spherical gaussian, we mean that the covariance matrix is a multiple of the identity: $\Sigma = \sigma^2 I$.

classes A B is

$$\Pr(\mathbf{X}|\mathbf{\Theta}) \stackrel{\text{def}}{=} \Pr(\mathbf{x}_1, \cdots, \mathbf{x}_n|\mathbf{\Theta}) = \Pr(\mathbf{x}_1|\mathbf{\Theta}) \cdot \Pr(\mathbf{x}_2|\mathbf{\Theta}) \cdot \ldots \cdot \Pr(\mathbf{x}_n|\mathbf{\Theta}) \tag{1}$$

because we assume the points were drawn independently. We expand each factor:

$$\begin{aligned} \Pr(\mathbf{x}_i|\mathbf{\Theta}) = [\Pr(\mathbf{x}_i \ \& \ A) + \Pr(\mathbf{x}_i \ \& \ B)] &= [\Pr(\mathbf{x}_i|A) \cdot P_A + \Pr(\mathbf{x}_i|B) \cdot P_B] \\ &= \left[P_A \frac{1}{\sigma_A^d (2\pi)^{(d/2)}} \exp\left(\frac{\|\mathbf{x}_i - \boldsymbol{\mu}_A\|^2}{-2\sigma_A^2} \right) + P_B \frac{1}{\sigma_B^d (2\pi)^{(d/2)}} \exp\left(\frac{\|\mathbf{x}_i - \boldsymbol{\mu}_B\|^2}{-2\sigma_B^2} \right) \right] \end{aligned} \tag{2}$$

where we omit the "$|\mathbf{\Theta}$" on the right since everything here depends on the given values $\mathbf{\Theta}$. Here we have used the assumption that the conditional probability within each class is a gaussian:

$$\Pr(\mathbf{x}_i|A) = \frac{1}{\sigma_A^d (2\pi)^{(d/2)}} \exp\left(\frac{\|\mathbf{x}_i - \boldsymbol{\mu}_A\|^2}{-2\sigma_A^2} \right), \tag{3}$$

(and similarly for B).

3. In the following, we will need the derivatives of the total probability (1) with respect to each parameter θ_j in $\mathbf{\Theta}$. We include here the formulas with short derivations. ***Derivative***

$$\begin{aligned} \frac{\partial \Pr(\mathbf{X}|\mathbf{\Theta})}{\partial \theta_j} &= \frac{\partial}{\partial \theta_j} \left(\Pr(\mathbf{x}_1|\mathbf{\Theta}) \cdot \Pr(\mathbf{x}_2|\mathbf{\Theta}) \cdot \ldots \cdot \Pr(\mathbf{x}_n|\mathbf{\Theta}) \right) \\ &= \Pr(\mathbf{X}|\mathbf{\Theta}) \cdot \left(\frac{\partial/\partial \theta_j \Pr(\mathbf{x}_1|\mathbf{\Theta})}{\Pr(\mathbf{x}_1|\mathbf{\Theta})} + \frac{\partial/\partial \theta_j \Pr(\mathbf{x}_2|\mathbf{\Theta})}{\Pr(\mathbf{x}_2|\mathbf{\Theta})} + \ldots + \frac{\partial/\partial \theta_j \Pr(\mathbf{x}_n|\mathbf{\Theta})}{\Pr(\mathbf{x}_n|\mathbf{\Theta})} \right) \end{aligned} \tag{4}$$

where the parameter θ_j is one of $\boldsymbol{\mu}_A, \sigma_A, P_A, \boldsymbol{\mu}_B, \sigma_B$. Here we show the derivatives of each factor *wrt* each parameter, where we omit the $(|\mathbf{\Theta})$ throughout to save clutter. The derivative (gradient) of the i-th factor *wrt* the mean is itself a vector of derivatives:

$$\begin{aligned} \partial/\partial \boldsymbol{\mu}_A \Pr(\mathbf{x}_i) &= \partial/\partial \boldsymbol{\mu}_A [\Pr(\mathbf{x}_i|A) \cdot P_A + \Pr(\mathbf{x}_i|B) \cdot P_B] = P_A \partial/\partial \boldsymbol{\mu}_A \Pr(\mathbf{x}_i|A) \\ &= P_A \Pr(\mathbf{x}_i|A)(\mathbf{x}_i - \boldsymbol{\mu}_A) \tfrac{1}{\sigma_A^2}, \end{aligned} \tag{5}$$

where the last equality comes from the gaussian distribution assumption.

$$\begin{aligned} \partial/\partial \sigma_a \Pr(\mathbf{x}_i) &= \partial/\partial \sigma_A [\Pr(\mathbf{x}_i|A) \cdot P_A + \Pr(\mathbf{x}_i|B) \cdot P_B] = P_A \partial/\partial \sigma_A \Pr(\mathbf{x}_i|A) \\ &= P_A \Pr(\mathbf{x}_i|A) \left[\frac{\|\mathbf{x}_i - \boldsymbol{\mu}_A\|^2}{\sigma_A^3} - \frac{d}{\sigma_A} \right] \\ &= \frac{P_A \Pr(\mathbf{x}_i|A)}{\sigma_A^3} \left[\|\mathbf{x}_i - \boldsymbol{\mu}_A\|^2 - d\sigma_A^2 \right] \end{aligned} \tag{6}$$

The derivative *wrt* the prior is

$$\begin{aligned} \partial/\partial P_A \textstyle\sum_i \Pr(\mathbf{x}_i) &= \partial/\partial p_A [\Pr(\mathbf{x}_i|A) \cdot P_A + \Pr(\mathbf{x}_i|B) \cdot P_B] \\ &= \Pr(\mathbf{x}_i|A) - \Pr(\mathbf{x}_i|B), \end{aligned} \tag{7}$$

where we have used $P_B = 1 - P_A$.

4. When we plug the individual derivatives (5) (6) into the derivative of the total probability (4), we have factors $P_A \Pr(\mathbf{x}_i|A)$ in the numerator in each term of (4) and $\Pr(\mathbf{x}_i)$ in the denominator. These combine to form the posterior probability ***total derivative***

$$\Pr(A|\mathbf{x}_i) = \frac{P_A \Pr(\mathbf{x}_i|A)}{\Pr(\mathbf{x}_i)}, \quad \text{and} \quad \frac{\Pr(A|\mathbf{x}_i)}{P_A} = \frac{\Pr(\mathbf{x}_i|A)}{\Pr(\mathbf{x}_i)}, \tag{8}$$

so that the derivatives of the total probability become

$$\partial/\partial \boldsymbol{\mu}_A \Pr(\mathbf{X}) = \Pr(\mathbf{X}) \cdot \sum_i \Pr(A|\mathbf{x}_i)(\mathbf{x}_i - \boldsymbol{\mu}_A) \,/\, \sigma_A^2, \tag{9}$$

$$\partial/\partial \sigma_A \Pr(\mathbf{X}) = \Pr(\mathbf{X}) \cdot \sum_i \Pr(A|\mathbf{x}_i) \left[\|\mathbf{x}_i - \boldsymbol{\mu}_A\|^2 - d\sigma_A^2 \right] \,/\, \sigma_A^3. \tag{10}$$

We have similar formulas for the derivative *wrt* $\boldsymbol{\mu}_B, \sigma_B$. Regarding the prior, we have

$$\begin{aligned} \partial/\partial P_A \Pr(\mathbf{X}) &= \Pr(\mathbf{X}) \cdot \sum_i \frac{\Pr(\mathbf{x}_i|A) - \Pr(\mathbf{x}_i|B)}{\Pr(\mathbf{x}_i)} \\ &= \Pr(\mathbf{X}) \cdot \sum_i \frac{\Pr(A|\mathbf{x}_i)}{P_A} - \frac{\Pr(B|\mathbf{x}_i)}{1 - P_A} \\ &= \Pr(\mathbf{X}) \cdot \sum_i \frac{\Pr(A|\mathbf{x}_i) - P_A}{P_A(1 - P_A)} \end{aligned} \tag{11}$$

E Step – Affiliations – Posterior Probabilities.

5. Given numerical values for the parameters $\boldsymbol{\Theta} = \{\boldsymbol{\mu}_A, \sigma_A, P_A, \boldsymbol{\mu}_B, \sigma_B, [P_B]\}$, compute the posterior probabilities (8): ***E Step***

$$\begin{aligned} P(A|\mathbf{x}_i) &= \frac{P_A}{\Pr(\mathbf{x}_i)} \Pr(\mathbf{x}_i|A) = \frac{P_A}{\Pr(\mathbf{x}_i)} \cdot \frac{1}{\sigma_A^d (2\pi)^{(d/2)}} \exp\left(\frac{\|\mathbf{x}_i - \boldsymbol{\mu}_A\|^2}{-2\sigma_A^2} \right) \\ P(B|\mathbf{x}_i) &= \frac{P_B}{\Pr(\mathbf{x}_i)} \Pr(\mathbf{x}_i|B) = \frac{P_B}{\Pr(\mathbf{x}_i)} \cdot \frac{1}{\sigma_B^d (2\pi)^{(d/2)}} \exp\left(\frac{\|\mathbf{x}_i - \boldsymbol{\mu}_B\|^2}{-2\sigma_B^2} \right) \end{aligned} \tag{12}$$

Again, the "$|\boldsymbol{\Theta}$" is omitted but should be in every item above. The quantity $P(A|\mathbf{x}_i)$ can be interpreted as a soft membership function, giving a measure of affinity of the point $\mathbf{x}_i$ to the class A. The requirement that $P(A|\mathbf{x}_i) + P(B|\mathbf{x}_i) = 1$ determines the scaling (denominator: $\Pr(\mathbf{x}_i)$).

M Step – Find Maximum Likelihood Estimates of Parameters $\boldsymbol{\Theta}$.

6. Given the membership affinities in the form of the posterior probabilities, find the values of the parameters $\boldsymbol{\Theta} = \{\boldsymbol{\mu}_A, \sigma_A, P_A, \boldsymbol{\mu}_B, \sigma_B\}$ that would maximize the total probability $\Pr(\mathbf{X})$. We can find these by setting the derivatives (9), (10), (11) to zero. The results are ***M Step***

$$\sum_i \mathbf{x}_i \Pr(A|\mathbf{x}_i) = \boldsymbol{\mu}_A \sum_i \Pr(A|\mathbf{x}_i) \tag{13}$$

$$\sum_i \|\mathbf{x}_i - \boldsymbol{\mu}_A\|^2 \Pr(A|\mathbf{x}_i) = d\sigma_A^2 \sum_i \Pr(A|\mathbf{x}_i) \tag{14}$$

$$\sum_i \Pr(A|\mathbf{x}_i) = nP_A \tag{15}$$

We have analogous formulas for class B.

7. So we now have an iterative procedure. Each iteration starts with values for the parameters $\boldsymbol{\Theta} = \{\boldsymbol{\mu}_A, \sigma_A, P_A, \boldsymbol{\mu}_B, \sigma_B, P_B\}$, as well as the induced class probability distributions $\Pr(\mathbf{x}|A)$, $\Pr(\mathbf{x}|B)$. ***Procedure***

K0. In the special case of K-means, the priors are fixed to $P_A = P_B = 1/2$, the variances are fixed to $\sigma_A = \sigma_B = 1$. The within-class probability distributions become inversely proportional to just a measure of distance to the respective means: $\Pr(\mathbf{x}|A) = \exp(-\|\mathbf{x} - \boldsymbol{\mu}_A\|^2/2)$, $\Pr(\mathbf{x}|B) = \exp(-\|\mathbf{x} - \boldsymbol{\mu}_B\|^2/2)$. Hence computing the hardened posteriors is the same as simply assigning each datum to the nearest mean. Adjusting the means to maximize the probability is the same as computing the center separately for each class, since each datum is affiliated with exactly one class.

E. Compute the *expectation*: the new values of the joint probabilities using (2), and then the posteriors ("affiliations") $\Pr(A|\mathbf{x}_i)$, $\Pr(B|\mathbf{x}_i)$ using

$$P(A|\mathbf{x}_i) = \frac{\Pr(\mathbf{x}_i|A) \cdot P_A}{\Pr(\mathbf{x}_i)}, \quad P(B|\mathbf{x}_i) = \frac{\Pr(\mathbf{x}_i|B) \cdot P_B}{\Pr(\mathbf{x}_i)}. \tag{16}$$

K1. In the case of K-means, these affiliations are "hardened" to 1 or 0 depending on which is bigger, turning them into membership flags, or "indicator functions." In effect, each point $\mathbf{x}_i$ is assigned to the cluster represented by its closest center.

M. Given the new values for the posteriors, compute new values for the parameters $\boldsymbol{\Theta} = \{\boldsymbol{\mu}_A, \sigma_A, P_A, \boldsymbol{\mu}_B, \sigma_B, P_B\}$ to *maximize* the probability of the observed data. Define the quantities representing the total number of points in each cluster weighted by their degree of membership:

$$n_A = \sum_i \Pr(A|\mathbf{x}_i) \quad n_B = \sum_i \Pr(B|\mathbf{x}_i) \tag{17}$$

Note that

$$n_A + n_B = \sum_i (\Pr(A|\mathbf{x}_i) + \Pr(B|\mathbf{x}_i)) = \sum_i (1) = n, \tag{18}$$

since $\mathbf{x}_i$ must come from cluster A or B. Then the update formulas derived from (13), (14), (15), respectively, are as follows:

$$\boldsymbol{\mu}_A = \frac{1}{n_A} \sum_i \Pr(A|\mathbf{x}_i)\mathbf{x}_i, \quad \boldsymbol{\mu}_B = \frac{1}{n_B} \sum_i \Pr(B|\mathbf{x}_i)\mathbf{x}_i, \tag{19}$$

$$\sigma_A^2 = \frac{1}{n_A d} \sum_i \Pr(A|\mathbf{x}_i) \cdot \|\mathbf{x}_i - \boldsymbol{\mu}_A\|^2, \quad \sigma_B^2 = \frac{1}{n_B d} \sum_i \Pr(B|\mathbf{x}_i) \cdot \|\mathbf{x}_i - \boldsymbol{\mu}_B\|^2, \tag{20}$$

$$P_A = n_A / n \quad P_B = n_B / n \tag{21}$$

Note: (19) (20) are just the Maximum Likelihood Estimates for the parameters of a gaussian, where each datum is weighted by its class affiliation. If the within-class distributions were something other than gaussian, we would obtain the analogous weighted MLE parameter estimates. The estimated priors (21) are the same regardless of the within-class distributions.

K2. In the case of K-means, only the means are updated by (19). Because of the hardening of cluster affiliations in step K1, this yields just the ordinary cluster centers.

Addendum - Update Priors with Multiple Clusters ***Update of Priors***

- If there are more than two distributions, the handling of the derivatives *wrt* the priors becomes more complicated, but the end result is the same update (21). The construction goes as follows. Let $\mathbf{p} = (P_A, P_B)^T$ be the vector of priors. The *gradient* of $\Pr(\mathbf{x}_i)$ *wrt* the vector $\mathbf{p}$ is

$$\begin{aligned} \nabla_{\mathbf{p}} \sum_i \Pr(\mathbf{x}_i) &= \nabla_{\mathbf{p}} [\Pr(\mathbf{x}_i|A) \cdot P_A + \Pr(\mathbf{x}_i|B) \cdot P_B] \\ &= \begin{bmatrix} \Pr(\mathbf{x}_i|A) \\ \Pr(\mathbf{x}_i|B) \end{bmatrix}. \end{aligned} \tag{22}$$

The gradient of the total probability is then also a vector:

$$\begin{aligned} \nabla_{\mathbf{p}} \Pr(\mathbf{X}) &= \Pr(\mathbf{X}) \cdot \sum_i \frac{1}{\Pr(\mathbf{x}_i)} \begin{bmatrix} \Pr(\mathbf{x}_i|A) \\ \Pr(\mathbf{x}_i|B) \end{bmatrix} \\ &= \Pr(\mathbf{X}) \cdot \sum_i \begin{bmatrix} \dfrac{\Pr(A|\mathbf{x}_i)}{P_A} \\ \dfrac{\Pr(B|\mathbf{x}_i)}{P_B} \end{bmatrix} = \Pr(\mathbf{X}) \cdot \begin{bmatrix} n_A \,/\, P_A \\ n_B \,/\, P_B \end{bmatrix}. \end{aligned} \tag{23}$$

Unfortunately we cannot simply set this to zero, but must incorporate the constraint $g(\mathbf{p}) = P_A + P_B = 1$ by using a Lagrange multiplier λ. The Kuhn-Tucker first order conditions are that the gradient of the objective must line up with the gradient of the constraint:

$$\nabla_{\mathbf{p}} \Pr(\mathbf{X}) = \Pr(\mathbf{X}) \begin{bmatrix} n_A \,/\, P_A \\ n_B \,/\, P_B \end{bmatrix} = \lambda \nabla_{\mathbf{p}} g(\mathbf{p}) = \lambda \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

We combine this with the constraint itself to yield the system of equations in the unknowns $P_A, P_B, \tilde{\lambda}$:

$$\begin{aligned} n_A \,/\, P_A &= \tilde{\lambda} \\ n_B \,/\, P_B &= \tilde{\lambda} \\ P_A + P_B &= 1, \end{aligned}$$

where $\tilde{\lambda} = \lambda \,/\, \Pr(\mathbf{X})$. The solution of this system is

$$\begin{aligned} P_A &= n_A \,/\, \tilde{\lambda} \\ P_B &= n_B \,/\, \tilde{\lambda} \\ \tilde{\lambda} &= n_A + n_B = n, \end{aligned}$$

yielding the update (21). This construction can easily be extended to any number of clusters.