Recap

- Bayes classifier uses a loss function to cpature costs of different errors.
- It is optimal for minimizing risk.
- There can be other criteria
- We saw minmax and Neymann-Pearson criteria.
- ROC is a convenient way to visualize the trade-off between false positive rate and false negative rate in a 2-class problem.

Recap

- We need to estimate class conditional densities to implement Bayes classifier.
- We considered parametric density estimation.
- Maximum likelihood (ML) estimation is one method to estimate parameters of a density function.
- ML estimate is the maximizer of Likelihood function:

$$L(\theta|\mathcal{D}) = \prod_i f(x_i|\theta)$$

- We saw examples of ML estimation for some densities.

One more example

- Suppose we have a discrete random variable, say, Z, that takes values $a_1, \cdots, a_M$ with probabilities $p_1, \cdots, p_M$.
- Given data in the form of *iid* realizations of this random variable, we want to estimate the parameters p_i.
- Note that the parameters satisfy: $p_i \geq 0$ and $\sum_i p_i = 1$.
- Intuitively the estimate of p_i should be fraction of data with value a_i.

- For our estimation, we represent the discrete random variable, Z, by an M-dimensional vector random variable $X = [X^1, \cdots, X^M]^T$.
- The idea is that if Z takes value a_i then we will represent it by X whose i^{th} component is one and all others are zero.
- So, the random vector X actually takes only M possible values, namely,
$[1,\ 0,\ \cdots,\ 0]^T$, $[0,\ 1,\ 0,\ \cdots, 0]^T$ etc.
- This is sometimes called '1 of M' representation or 'one-hot' representation.

- Thus, $X = [X^1, \cdots, X^M]^T$ satisfies: $X^i \in \{0, 1\}$ and $\sum_i X^i = 1$.
- Also now we have $p_i = \text{Prob}[X^i = 1]$.
- Now the mass function for X can be written as

$$f(x \mid p) = \prod_{i=1}^{M} p_i^{x^i},$$

$x = [x^1, \cdots, x^M]^T, \ x^i \in \{0, 1\}, \ \sum_i x^i = 1$

- Here, $p = (p_1, \cdots, p_M)^T$ is the parameter vector.

- Now the problem of estimating the parameters, p_i, becomes the following.
- We are given *iid* data

$$\mathcal{D} = \{x_1, \cdots, x_n\}$$

where $x_i = [x_i^1, \cdots, x_i^M]^T$ with $x_i^j \in \{0, 1\}$ and $\sum_j x_i^j = 1, \forall i$.
- The x_i are *iid* with

$$f(x_i|p) = \prod_{j=1}^{M} p_j^{x_i^j}$$

- We need to derive ML estimates for parameters p_i.

- The log likelihood function is given by

$$
\begin{aligned}
l(p \mid \mathcal{D}) &= \sum_{i=1}^{n} \ln(f(x_i \mid p)) \\
&= \sum_{i=1}^{n} \ln\left(\prod_{j=1}^{M} p_j^{x_i^j}\right) \\
&= \sum_{i=1}^{n} \sum_{j=1}^{M} x_i^j \ln(p_j)
\end{aligned}
$$

- We now want to find values for $p_i, \;\; i = 1, \cdots, M$, to maximize $l(p \mid \mathcal{D})$.
- But this is not an unconstrained maximization.
- We need to maximize l over only those p_i that satisfy $p_i \geq 0$ and $\sum_i p_i = 1$.
- Hence ML estimation of the parameters here becomes a constrained optimization problem as follows.

The constrained optimization problem is

$$\max_{p_i} \quad l(p \mid \mathcal{D}) = \sum_{i=1}^{n} \sum_{j=1}^{M} x_i^j \ln(p_j)$$

$$\text{subject to} \quad \sum_{i=1}^{M} p_i = 1$$

- We can solve this by the method of lagrange multipliers. (We have not explicitly included the non-negativity constraint).

- The lagrangian for this problem is given by

$$\sum_{i=1}^{n} \sum_{s=1}^{M} x_i^s \ln(p_s) + \lambda \left(1 - \sum_{s=1}^{M} p_s\right)$$

  where λ is the Lagrange multiplier.
- Now, we calculate the partial derivatives of the Lagrangian and equate them to zero to get the maximum.

- The lagrangian is

$$\sum_{i=1}^{n} \sum_{s=1}^{M} x_i^s \ln(p_s) + \lambda \left(1 - \sum_{s=1}^{M} p_s \right)$$

- Equating partial derivative w.r.t p_j to zero

$$\sum_{i=1}^{n} \frac{x_i^j}{p_j} - \lambda = 0, \; j = 1, \cdots, M$$

- Solving this, we get

$$p_j = \frac{1}{\lambda} \sum_{i=1}^{n} x_i^j, \; j = 1, \cdots, M$$

- Now we use the constraint, $\sum_j p_j = 1$.

$$\sum_j p_j = \sum_j \frac{1}{\lambda} \sum_{i=1}^{n} x_i^j = 1$$

- Using this we can calculate λ as

$$\begin{aligned} \lambda &= \sum_{j=1}^{M} \sum_{i=1}^{n} x_i^j \\ &= \sum_{i=1}^{n} \sum_{j=1}^{M} x_i^j \\ &= n \end{aligned}$$

where last step follows because $\sum_j x_i^j = 1, \forall i$.

- Thus, we get the final ML estimate for p_j as

$$\hat{p}_j = \frac{1}{n} \sum_{i=1}^{n} x_i^j$$

- The final ML estimate for p_j is the fraction of times the j^{th} value occurs – intuitively clear.

- The distribution (or probability mass function) of any discrete random variable taking finitely many values, is specified by some M parameters like the p_i.
- Hence, what we presented is a general procedure to handle any discrete random variable.
- Here there is really no distinction between parametric and non-parametric estimation.
- Also, note that the a_i need not be numeric.

- Features that take only finitely many values are important in some machine learning problems. (sometimes called Categorical features)
- For example, search and ranking, document classification, spam filtering etc.
- For example, for sentiment analysis, we can use 'bag of words' as the feature vector.

- In such cases, each feature is a discrete random variable.
- We can estimate (marginal) distribution of features using our procedure.
- To implement Bayes classifier we need **joint** distribution of the feature vector.
- We can, e.g., assume features are independent (conditioned on the class label).
- Then, joint mass function is product of marginals.
- Often called, 'naive Bayes' classifier

Naive Bayes Classifier

- The class conditional density is modelled as

$$f(x|y=c,\theta) = \prod_{j=1}^{d} f(x_j|y=c,\theta_{jc})$$

  The joint density is product of marginals and each marginal has its own parameters.
- x_j can be binary and each marginal is Bernoulli with parameter θ_{jc}.
- x Could be Gaussian with diagonal covariance matrix

Example: Document Classification

- Can use a binary feature vector of dimension equal to the size of dictionary.
- x_i represents whether or not i^{th} word appears in the document.
- We can model: $\text{Prob}[x_j = 1 | y = c] = \theta_{jc}$.
- We can use MLE for the Bernoulli parameter, θ_{jc}.
- Finally, we can use Naive Bayes classifier.

- In this model, multiple occurrences of a word is ignored.
- We can change the model by taking x_i as the number of times i^{th} word appears in the document.
- We can then estimate the (marginal) distribution of each feature.
- Once again we can use Naive bayes classifier.

- Taking the raw word frequency (number of times a word occurs in a document) as the feature may not be satisfactory.
- Different documents may have different number of words; some words may appear in all documents.
- There are different ways to make useful features from word frequencies.

TF-IDF in document classification

- Let g_{ij} denote the number of times word i appears in document j
- Term frequency of word (or term) i in document j can be defined as
$$\mathsf{TF}_{ij} = \frac{g_{ij}}{\max_k g_{kj}}$$
This is always between 0 and 1 and would not be affected by different documents having different number of words.
- We can use these as the features.
- Still, term frequency may not adequately characterize utility of a word for the document classification.

TF-IDF in document classification

- Even if TF_{ij} is high, if the word is uniformly present in all documents, it may not be useful.
- The inverse document frequency of word i is defined by

  $$\mathrm{IDF}_i = \log_2(N/n_i)$$

  where N is the total number of documents and word i appears in n_i of them.
- The TF-IDF representation of a document j is the vector whose i^{th} component is given by $\mathrm{TF}_{ij}\mathrm{IDF}_i$.
- In all cases (binary features, term frequency or TF-IDF features) we can use Naive Bayes classifier.

Fitting probability models to data

- We have considered examples of simple density functions.
- In general, one can use ML estimation to fit any parameterized density model to data.
- One can use it for learning generative models or discriminative models.

Ingradients of ML estimation

- We choose a model (or model class): $f(x|\theta)$
- Value of θ specifies a particular model in this family of models
- We have data: $\{x_1, x_2, \cdots, x_n\}$.
- We learn a specific model by estimating $\hat{\theta}_n$ as

$$\hat{\theta}_n = \arg\max_{\theta} \sum_{i=1}^{n} \log(f(x_i|\theta))$$

- Since the data is all that we have, we can say that distribution given by the data is

$$f_{\text{data}}(x_i) = \frac{1}{n},\ i = 1, \cdots, n$$

- We are getting our generalization abilities by saying we want to capture it using a member from the family of models $f_\theta(x) \triangleq f(x|\theta)$.
- Goodness of any θ can be measured by asking how 'close' is the distribution f_θ to f_{data}.

Kullback-Leibler Divergence

- Let p and q be two densities (with p supported on a countable set).
- Then the KL divergence from p to q is defined by

$$KL(p||q) = -\sum_{x} p(x) \ln \left(\frac{q(x)}{p(x)} \right)$$

- It can be shown that $KL(p||q) \geq 0$ and it is zero only when the two distributions are identical.
- Given a family of distributions, the best one to approximate p could be the one which has least KL divergence from p.

- For the distributions f_θ and f_{data},

$$
\begin{aligned}
KL(f_{\text{data}}||f_\theta) &= -\sum_{i=1}^{n} f_{\text{data}}(x_i) \ln\left(\frac{f_\theta(x_i)}{f_{\text{data}}(x_i)}\right) \\
&= -\sum_{i=1}^{n} f_{\text{data}}(x_i) \ln(f_\theta(x_i)) + \text{const}
\end{aligned}
$$

- since $f_{\text{data}}(x_i) = 1/n$, minimizing this is same as maximizing

$$\sum_{i=1}^{n} \frac{1}{n} \ln(f_\theta(x_i))$$

- That is the ML estimate.

Mixture density estimation

- The last topic we consider under parametric estimation is that of mixture densities.
- In many cases we may not be able to capture the class conditional density using any standard density model.
- In such cases, often, modelling the class conditional density as a mixture of densities is helpful.

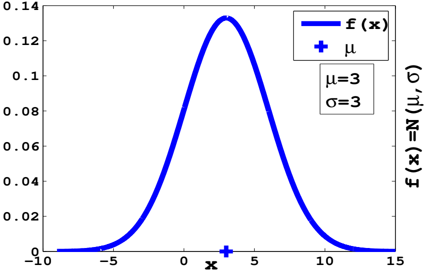
Mixture density model

- Consider a density model

$$f(x) = \sum_{k=1}^{K} \lambda_k f_k(x), \quad \lambda_k \geq 0, \text{ and } \sum_{k=1}^{K} \lambda_k = 1$$

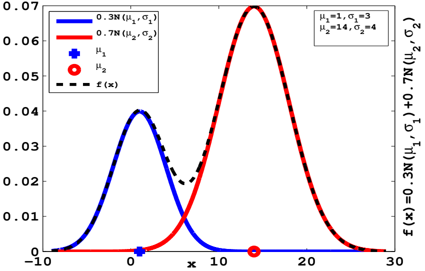
  where each f_k is a density function.
- Since each f_k is a density, given the conditions on λ_k, f is a convex combination of densities and hence is itself a density.
- Mixture densities are useful when data distribution is multimodal.

- Most standard densities are unimodal.
- For example, consider the normal density.



This is unimodal.

- Now let us consider a mixture of two normal densities



- This is a multimodal density
- When data density is multi-modal, we can often approximate it with mixture of gaussians.

ML estimation of mixture models

- Consider a mixture of normal densities

$$f(x \mid \theta) = \sum_{k=1}^{K} \lambda_k \, f_k(x)$$

  where each f_k is $\mathcal{N}(\mu_k, \, \Sigma_k)$.
- The parameter vector, θ, consists of all λ_k, which are called mixing coefficients, and all the parameters of the constituent densities, namely, $\mu_k, \, \Sigma_k, \, k = 1, \cdots, K$.

- Let $\mathcal{D} = \{x_1, \cdots, x_n\}$ be a sample of *n iid* data from this density.
- Then the likelihood function is

$$L(\theta \mid \mathcal{D}) = \prod_{i=1}^{n} \left[\sum_{k=1}^{K} \lambda_k \, f_k(x_i) \right]$$

Difficulty in estimating mixture density

- The log likelihood is given by

$$l(\theta \mid \mathcal{D}) = \sum_{i=1}^{n} \ln \left[\sum_{k=1}^{K} \lambda_k \, f_k(x_i) \right]$$

- Since there is a sum inside the log function, the densities f_k being from exponential family, does not simplify log likelihood.
- Maximizing log likelihood could become a difficult optimization problem.

Mixture of two one dimensional densities

- Consider one dimensional case with $K = 2$.
  Let for $j = 1, 2$,

$$\phi(x \mid \theta_j) = \frac{1}{\sigma_j \sqrt{2\pi}} \exp\left(-\frac{(x-\mu_j)^2}{2\sigma_j^2}\right), \quad \theta_j = (\mu_j,\ \sigma_j)$$

- The density model is

$$f(x \mid \theta) = \lambda_1\, \phi(x \mid \theta_1) + \lambda_2\, \phi(x \mid \theta_2)$$

where $\theta = (\theta_1,\ \theta_2,\ \lambda_1,\ \lambda_2)$

- The log likelihood is

$$l(\mathcal{D} \mid \theta) = \sum_{i=1}^{n} \ln(\lambda_1 \, \phi(x_i \mid \theta_1) + \lambda_2 \, \phi(x_i \mid \theta_2))$$

- We need to maximize this with respect to θ.
- Let us calculate the partial derivatives of l.
- First note that

$$\frac{\partial \, \phi(x \mid \theta_j)}{\partial \mu_s} = \frac{\partial \, \phi(x \mid \theta_j)}{\partial \sigma_s} = 0, \quad \text{if} \quad j \neq s.$$

Recall that

$$\phi(x \mid \theta_j) = \frac{1}{\sigma_j\sqrt{2\pi}} \exp\left(-\frac{(x-\mu_j)^2}{2\sigma_j^2}\right), \quad \theta_j = (\mu_j,\ \sigma_j)$$

By differentiation we get, for $j = 1, 2$,

$$\begin{aligned}
\frac{\partial\, \phi(x \mid \theta_j)}{\partial \mu_j} &= \phi(x \mid \theta_j)\, \frac{(x-\mu_j)}{\sigma_j^2} \\
\frac{\partial\, \phi(x \mid \theta_j)}{\partial \sigma_j} &= \phi(x \mid \theta_j) \left[\frac{(x-\mu_j)^2}{\sigma_j^3} - \frac{1}{\sigma_j}\right]
\end{aligned}$$

By differentiation we got, for $j = 1, 2$,

$$\begin{aligned}
\frac{\partial\, \phi(x \mid \theta_j)}{\partial \mu_j} &= \phi(x \mid \theta_j)\, \frac{(x - \mu_j)}{\sigma_j^2} \\
\frac{\partial\, \phi(x \mid \theta_j)}{\partial \sigma_j} &= \phi(x \mid \theta_j)\, \left[\frac{(x - \mu_j)^2}{\sigma_j^3} - \frac{1}{\sigma_j} \right]
\end{aligned}$$

Now we have

$$\begin{aligned}
\frac{\partial\, l(\mathcal{D} \mid \theta)}{\partial \mu_j} &= \frac{\partial}{\partial \mu_j} \left(\sum_{i=1}^{n} \ln(\lambda_1\, \phi(x_i \mid \theta_1) + \lambda_2\, \phi(x_i \mid \theta_2)) \right) \\
&= \sum_{i=1}^{n} \frac{\lambda_j\, \phi(x_i \mid \theta_j)\, \frac{(x_i - \mu_j)}{\sigma_j^2}}{\lambda_1\, \phi(x_i \mid \theta_1)\, +\, \lambda_2 \phi(x_i \mid \theta_2)}
\end{aligned}$$

We get a similar expression for $\frac{\partial\, l(\mathcal{D} \mid \theta)}{\partial \sigma_j}$

- Define γ_{ij}, $i = 1, \cdots, n$, $j = 1, 2$,

$$\gamma_{ij} = \frac{\lambda_j \, \phi(x_i \mid \theta_j)}{\lambda_1 \, \phi(x_i \mid \theta_1) \, + \, \lambda_2 \phi(x_i \mid \theta_2)}$$

- Then we get

$$\begin{aligned}
\frac{\partial \, l(\mathcal{D} \mid \theta)}{\partial \mu_j} &= \sum_{i=1}^{n} \gamma_{ij} \, \frac{(x_i - \mu_j)}{\sigma_j^2} \\
\frac{\partial \, l(\mathcal{D} \mid \theta)}{\partial \sigma_j} &= \sum_{i=1}^{n} \gamma_{ij} \left[\frac{(x_i - \mu_j)^2}{\sigma_j^3} - \frac{1}{\sigma_j} \right]
\end{aligned}$$

- By equating the partial derivatives to zero, we get

$$\sum_{i=1}^{n} \gamma_{ij} \frac{(x_i - \mu_j)}{\sigma_j^2} = 0 \quad \Rightarrow \quad \mu_j = \frac{\sum_{i=1}^{n} \gamma_{ij}\, x_i}{\sum_{i=1}^{n} \gamma_{ij}}$$

$$\sum_{i=1}^{n} \gamma_{ij} \left[\frac{(x_i - \mu_j)^2}{\sigma_j^3} - \frac{1}{\sigma_j} \right] = 0 \quad \Rightarrow \quad \sigma_j^2 = \frac{\sum_{i=1}^{n} \gamma_{ij}\, (x_i - \mu_j)^2}{\sum_{i=1}^{n} \gamma_{ij}}$$

- Hence the ML estimates satisfy, for $j = 1, 2$,

$$\mu_j = \frac{\sum_{i=1}^n \gamma_{ij} \, x_i}{\sum_{i=1}^n \gamma_{ij}}$$

$$\sigma_j^2 = \frac{\sum_{i=1}^n \gamma_{ij} \, (x_i - \mu_j)^2}{\sum_{i=1}^n \gamma_{ij}}$$

- First, we like to note that these are not really estimates.
- The RHS in the above equations depends on the unknown parameter values.
- The solution to these give the ML estimates.
- There is an interesting structure here.

$$
\begin{aligned}
\mu_j &= \frac{\sum_{i=1}^{n} \gamma_{ij} \, x_i}{\sum_{i=1}^{n} \gamma_{ij}} \\
\sigma_j^2 &= \frac{\sum_{i=1}^{n} \gamma_{ij} \, (x_i - \mu_j)^2}{\sum_{i=1}^{n} \gamma_{ij}}
\end{aligned}
$$

- These are similar to the 'sample mean estimates'.
- It is a sample mean with 'weight' γ_{ij} for x_i.
  γ_{ij} are sometimes called responsibility coefficients.
- If there is only one component in the mixture, these become the usual ML estimates.

- Let us also find maximizers of log likelihood with respect to λ_j.
- Since we have a constraint $\lambda_1 + \lambda_2 = 1$, this is a constrained optimization.
- So, we need to equate to zero, the partial derivatives of
$$l(\mathcal{D} \mid \theta) + \eta(\lambda_1 + \lambda_2 - 1)$$
where η is the Lagrange multiplier.
- Recall

$$l(\mathcal{D} \mid \theta) = \sum_{i=1}^{n} \ln(\lambda_1 \, \phi(x_i \mid \theta_1) + \lambda_2 \, \phi(x_i \mid \theta_2))$$

- Hence

$$\frac{\partial l(\mathcal{D} \mid \theta)}{\partial \lambda_1} = \sum_{i=1}^{n} \frac{\phi(x_i \mid \theta_1)}{\lambda_1 \, \phi(x_i \mid \theta_1) \; + \; \lambda_2 \phi(x_i \mid \theta_2)}$$

Now $\frac{\partial(l(\mathcal{D} \mid \theta)+\eta(\lambda_1+\lambda_2-1))}{\partial\lambda_1} = 0$ implies

$$
\begin{aligned}
& \textstyle\sum_{i=1}^n \frac{\phi(x_i \mid \theta_1)}{\lambda_1 \phi(x_i \mid \theta_1) + \lambda_2\phi(x_i \mid \theta_2)} + \eta && = 0 \\
\text{or} \quad & \textstyle\sum_{i=1}^n \frac{\gamma_{i1}}{\lambda_1} + \eta && = 0 \\
\text{or} \quad & \lambda_1\eta = -\textstyle\sum_{i=1}^n \gamma_{i1} \Rightarrow \lambda_1 = -\frac{1}{\eta}\sum_{i=1}^n \gamma_{i1}
\end{aligned}
$$

- we get a similar equation for derivetive w.r.t. λ_2.
- Now, using $\lambda_1 + \lambda_2 = 1$, we get

$$
\eta = \eta(\lambda_1 + \lambda_2) = -\sum_{i=1}^{n}(\gamma_{i1} + \gamma_{i2}) = -n
$$

- Hence, the ML estimates for λ_j satisfy

$$\lambda_j = \frac{1}{n} \sum_{i=1}^{n} \gamma_{ij}$$

- Putting all these together we get

- The ML estimates for $\mu_j, \sigma_j, \lambda_j$, $j = 1, 2$, satisfy

$$\mu_j = \frac{\sum_{i=1}^n \gamma_{ij}\, x_i}{\sum_{i=1}^n \gamma_{ij}}, \quad \lambda_j = \frac{1}{n} \sum_{i=1}^n \gamma_{ij}$$

$$\sigma_j^2 = \frac{\sum_{i=1}^n \gamma_{ij}\, (x_i - \mu_j)^2}{\sum_{i=1}^n \gamma_{ij}}$$

- The structure of equations is interesting.
- These are not expressions for estimates.
- However, we can solve for estimates using, e.g., Gauss-Siedel iteration.

An Iterative Algorithm for Mixture Density Estimation

$$
\begin{aligned}
\mu_j^{(k+1)} &= \frac{\sum_{i=1}^{n} \gamma_{ij}^{(k)} x_i}{\sum_{i=1}^{n} \gamma_{ij}^{(k)}}, \quad \lambda_j^{(k+1)} = \frac{1}{n} \sum_{i=1}^{n} \gamma_{ij}^{(k)} \\
(\sigma_j^2)^{(k+1)} &= \frac{\sum_{i=1}^{n} \gamma_{ij}^{(k)} (x_i - \mu_j^{(k)})^2}{\sum_{i=1}^{n} \gamma_{ij}^{(k)}} \\
\gamma_{ij}^{(k+1)} &= \frac{\lambda_j^{(k+1)} \phi(x_i \mid \theta_j^{(k+1)})}{\sum_{j=1}^{2} \lambda_j^{(k+1)} \phi(x_i \mid \theta_j^{(k+1)})}
\end{aligned}
$$

- It is easy to generalize this to mixture of K Gaussians.

- What we have done so far is a special case of general procedure.
- We now look at this general procedure.

- Our density model was

$$f(x \mid \theta) = \sum_{j=1}^{2} \lambda_j \, \phi(x \mid \theta_j)$$

(while we stick to 2-component mixture, it is easily generalized to K components).
- In our sample each x_i is drawn *iid* according to this distribution.

$$\text{density model:} \quad f(x \mid \theta) = \sum_{j=1}^{2} \lambda_j \, \phi(x \mid \theta_j)$$

- To generate x_i, we first choose a component density, with probabilities λ_j, and then generate it from the corresponding $\phi(x \mid \theta_j)$.
- If we knew which x_i are generated from which component density, then the estimation of all parameters is very easy.
- Let us first formalize this notion.

Missing Information

- Let random variables Z_{ij}, $i = 1, \cdots, n$, $j = 1, 2$, denote the information of which component density each sample comes from.
- For each i, $Z_{ij} = 1$ if x_i came from j^{th} component density.
- We would have $\sum_j Z_{ij} = 1, \ \forall i$.
- Also, we have

$$P[Z_{ij} = 1] = \lambda_j, \ \forall i; \quad \text{and} \quad f(x_i \mid Z_{ij} = 1) = \phi(x_i \mid \theta_j)$$

We can think of Z_{ij} as the 'missing information'.

- Let Z_i denote the vector with components Z_{ij}.
- Denote $\mathcal{D}^c = \{(x_1, Z_1), \cdots, (x_n, Z_n)\}$.
- Our data consists of only x_i. But suppose the sample data was $\mathcal{D}^c$.
- Then estimation is easy. For example,

$$\hat{\mu}_1 = \frac{\sum_{i=1}^n Z_{i1}\, x_i}{\sum_{i=1}^n Z_{i1}}, \quad \hat{\mu}_2 = \frac{\sum_{i=1}^n Z_{i2}\, x_i}{\sum_{i=1}^n Z_{i2}}$$

- These are very similar to earlier equations.

Complete and incomplete data

- The general situation is as follows.
- The data that we have is ‘incomplete’
- This is because of some ‘hidden’ or ‘missing’ data.
- If we are given the complete data then ML estimation is easy.
- In our example, x_i is the incomplete data.
- (x_i, Z_i) constitutes the complete data and Z_i constitute the missing or hidden or latent data/variables.

The EM Algorithm

- The EM algorithm is an efficient iterative procedure for ML estimation in such situations.
- The algorithm basically has two steps: 'Expectation' and 'Maximization'
- Hence the name of the algorithm.
- As per our notation, x_i, $i = 1, \cdots, n$ is the incomplete data and (x_i, Z_i), $i = 1, \cdots, n$ is the complete data.

- Let $f(x, Z \mid \theta)$ be the density for the complete data. That is, the complete data is n iid samples from this density model.
- Thus, the complete data log likelihood is

$$l(\theta \mid \mathcal{D}^c) = \ln \left(\prod_{i=1}^{n} f(x_i, Z_i \mid \theta) \right)$$

- As earlier, we would also denote $\mathcal{D}^c$ by $(\mathbf{x}, \mathbf{Z})$.
- Hence the complete data loglikelihood is also denoted by $\ln(f(\mathbf{x}, \mathbf{Z} \mid \theta))$.

- The two steps of EM algorithm are as follows:

E-step : Compute $Q(\theta,\ \theta^{(k)})$ which is expectation of the complete data loglikelihood w.r.t. the conditional distribution of hidden variables conditioned on incomplete data and current value of θ as $\theta^{(k)}$.

$$\begin{aligned} Q(\theta,\ \theta^{(k)}) &= E_{\mathbf{Z}|\mathbf{x},\theta^{(k)}} \ln(f(\mathbf{x}, \mathbf{Z} \mid \theta)) \\ &= \int \ln(f(\mathbf{x}, \mathbf{z} \mid \theta))\ f(\mathbf{z}|\mathbf{x}, \theta^{(k)}) d\mathbf{z} \end{aligned}$$

M-step : Compute next value of θ as $\theta^{(k+1)}$ by maximizing $Q(\theta,\ \theta^{(k)})$ over θ.

$$\theta^{(k+1)} = \arg\max_{\theta}\ Q(\theta,\ \theta^{(k)})$$

Example of EM

- Let us consider the example of estimating a two component Gaussian density.

$$f(x \mid \theta) = \sum_{j=1}^{2} \lambda_j \, \phi(x \mid \theta_j)$$

- The x_i, $i = 1, \cdots, n$, is the given data which is the incomplete data here.
- The Z_{ij}, $i = 1, \cdots, n$, $j = 1, 2$, that we defined earlier are the hidden variables or the missing data.
- Recall that Z_{ij} is the indicator whether or not x_i came from the j^{th} component of the mixture.

- By definition of Z_{ij}, we have

$$P[Z_{ij} = 1] = \lambda_j, \ \forall i; \quad \text{and} \quad f(x_i \mid Z_{ij} = 1) = \phi(x_i \mid \theta_j)$$

- Recall $Z_i = (Z_{i1}, \ Z_{i2})$. Hence

$$f(Z_i \,|\, \theta) = \prod_{j=1}^{2} (\lambda_j)^{Z_{ij}}, \quad \text{and} \quad f(x_i \,|\, Z_i, \theta) = \prod_{j=1}^{2} (\phi(x_i \,|\, \theta_j))^{Z_{ij}}$$

- Hence density of complete data is

$$f(x_i,\ Z_i \mid \theta) = f(x_i \mid Z_i, \theta) f(Z_i \mid \theta) = \prod_{j=1}^{2} (\lambda_j\, \phi(x_i \mid \theta_j))^{Z_{ij}}$$

- Thus complete data likelihood is

$$f(\mathbf{x},\ \mathbf{Z} \mid \theta) = \prod_{i=1}^{n} \left[\prod_{j=1}^{2} (\lambda_j\, \phi(x_i \mid \theta_j))^{Z_{ij}} \right]$$

- The complete data log likelihood is

$$\ln(f(\mathbf{x}, \mathbf{Z} \mid \theta)) = \sum_{i=1}^{n} \left[\sum_{j=1}^{2} Z_{ij} \ln(\lambda_j \, \phi(x_i \mid \theta_j)) \right]$$

- Note that we now have 'sum of log' rather than 'log of sum'
- It is easy to see how knowledge of the 'hidden' variables makes the ML estimation easy.

Example: E-step

- For the E-step, we have to take expectation of $\mathbf{Z}$ w.r.t. distribution conditioned on $\mathbf{x}$ at a given value of θ.
- We have, for any θ',

$$\begin{aligned} E[Z_{ij} \mid \mathbf{x}, \theta'] &= P[Z_{ij} = 1 \mid \mathbf{x},\ \theta'] = P[Z_{ij} = 1 \mid x_i,\ \theta'] \\ &= \frac{f(x_i \mid Z_{ij} = 1,\ \theta')\, P[Z_{ij} = 1|\theta']}{\sum_{j=1}^{2}\ f(x_i \mid Z_{ij} = 1,\ \theta')\, P[Z_{ij} = 1|\theta']} \\ &= \frac{\lambda_j\, \phi(x_i \mid \theta'_j)}{\sum_{j=1}^{2}\ \lambda_j\, \phi(x_i \mid \theta'_j)} \end{aligned}$$

- Thus, $E[Z_{ij} \mid \mathbf{x}, \theta'] = \gamma_{ij}(\theta')$ where

$$\gamma_{ij}(\theta') = \frac{\lambda_j \, \phi(x_i \mid \theta'_j)}{\sum_{j=1}^{2} \lambda_j \, \phi(x_i \mid \theta'_j)}$$

- This is the same γ_{ij} that we defined earlier.
- This notation emphasizes the fact that the value of γ_{ij} depends on the parameter vector.
- Now we need to do this expectation on the complete data log likelihood which is

$$\ln(f(\mathbf{x}, \mathbf{Z} \mid \theta)) = \sum_{i=1}^{n} \left[\sum_{j=1}^{2} Z_{ij} \ln(\lambda_j \, \phi(x_i \mid \theta_j)) \right]$$

- Thus, under the E-step, we get

$$\begin{aligned} Q(\theta,\ \theta^{(k)}) &= E_{\mathbf{Z}|\mathbf{x},\theta^{(k)}} \ \ln(f(\mathbf{x}, \mathbf{Z} \mid \theta)) \\ &= \sum_{i=1}^{n} \left[\sum_{j=1}^{2} E[Z_{ij} \mid \mathbf{x},\ \theta^{(k)}] \ln(\lambda_j \, \phi(x_i \mid \theta_j)) \right] \\ &= \sum_{i=1}^{n} \left[\sum_{j=1}^{2} \gamma_{ij}(\theta^{(k)}) \ \ln(\lambda_j \, \phi(x_i \mid \theta_j)) \right] \end{aligned}$$

Example: the M-step

- In the M-step, we find $\theta^{(k+1)}$ that maximizes (over θ),

$$
\begin{aligned}
Q(\theta,\ \theta^{(k)}) &= \sum_{i=1}^{n} \left[\sum_{j=1}^{2} \gamma_{ij}(\theta^{(k)})\ \ln(\lambda_j\, \phi(x_i \mid \theta_j)) \right] \\
&= \sum_{i=1}^{n} \sum_{j=1}^{2} \gamma_{ij}(\theta^{(k)}) \, [\ln(\lambda_j) \ - \ \ln(\sigma_j \sqrt{2\pi}) \\
&\qquad\qquad - \frac{(x_i - \mu_j)^2}{2\sigma_j^2}]
\end{aligned}
$$

- This is now a simple optimization problem.

$$Q(\theta, \theta^{(k)}) = \sum_{i=1}^{n} \sum_{j=1}^{2} \gamma_{ij}(\theta^{(k)}) \left[\ln(\lambda_j) - \ln(\sigma_j \sqrt{2\pi}) - \frac{(x_i - \mu_j)^2}{2\sigma_j^2} \right]$$

$$\frac{\partial Q}{\partial \mu_1} = 0 \Rightarrow \sum_{i=1}^{n} \gamma_{i1}(\theta^k) \frac{(x_i - \mu_1)}{\sigma_1^2} = 0$$

- Hence we get

$$\mu_1^{k+1} = \frac{\sum_{i=1}^{n} \gamma_{i1}(\theta^{(k)})\, x_i}{\sum_{i=1}^{n} \gamma_{i1}(\theta^k)}$$

- This is same as the iterative algorithm we derived earlier.

$$Q(\theta, \theta^{(k)}) = \sum_{i=1}^{n} \sum_{j=1}^{2} \gamma_{ij}(\theta^{(k)}) \left[\ln(\lambda_j) - \ln(\sigma_j \sqrt{2\pi}) - \frac{(x_i - \mu_j)^2}{2\sigma_j^2} \right]$$

$$\frac{\partial Q}{\partial \sigma_1} = 0 \Rightarrow \sum_{i=1}^{n} \gamma_{i1}(\theta^{(k)}) \, [- \frac{1}{\sigma_1} + \frac{(x_i - \mu_1)^2}{\sigma_1^3}] = 0$$

- Hence we get

$$(\sigma_1^2)^{(k+1)} = \frac{\sum_{i=1}^{n} \gamma_{i1}(\theta^{(k)}) \, (x_i - \mu_1^{(k)})^2}{\sum_{i=1}^{n} \gamma_{i1}(\theta^k)}$$

- Once again same as earlier algorithm.

- Next, we need to find λ_j to maximize Q.
- However, we have the constraint $\lambda_1 + \lambda_2 = 1$.
- Hence we should solve

$$\frac{\partial(Q + \eta(\lambda_1 + \lambda_2 - 1)}{\partial \lambda_j} = 0$$

where η is the Lagrange multiplier.

$$Q(\theta, \theta^{(k)}) = \sum_{i=1}^{n} \sum_{j=1}^{2} \gamma_{ij}(\theta^{(k)}) \left[\ln(\lambda_j) - \ln(\sigma_j \sqrt{2\pi}) - \frac{(x_i - \mu_j)^2}{2\sigma_j^2} \right]$$

$$\frac{\partial(Q + \eta(\lambda_1 + \lambda_2 - 1)}{\partial \lambda_j} = 0 \Rightarrow \sum_{i=1}^{n} \gamma_{ij}(\theta^{(k)}) \frac{1}{\lambda_j} + \eta = 0$$

- Solving these and noting that $\sum_{i=1}^{n} \sum_{j=1}^{2} \gamma_{ij}(\theta^{(k)}) = n$ we get

$$\lambda_j^{(k+1)} = \frac{1}{n} \sum_{i=1}^{n} \gamma_{ij}(\theta^{(k)})$$

Thus we get

$$
\begin{aligned}
\mu_j^{(k+1)} &= \frac{\sum_{i=1}^n \gamma_{ij}^{(k)} x_i}{\sum_{i=1}^n \gamma_{ij}^{(k)}}, \quad \lambda_j^{(k+1)} = \frac{1}{n} \sum_{i=1}^n \gamma_{ij}^{(k)} \\
(\sigma_j^2)^{(k+1)} &= \frac{\sum_{i=1}^n \gamma_{ij}^{(k)} (x_i - \mu_j^{(k)})^2}{\sum_{i=1}^n \gamma_{ij}^{(k)}} \\
\gamma_{ij}^{(k+1)} &= \frac{\lambda_j^{(k+1)} \phi(x_i \mid \theta_j^{(k+1)})}{\sum_{j=1}^2 \lambda_j^{(k+1)} \phi(x_i \mid \theta_j^{(k+1)})} = \gamma_{ij}(\theta^{(k+1)})
\end{aligned}
$$

- So, this is actually the EM algorithm.

- To summarize, each iteration of the EM algorithm consists of two steps: given current value, $\theta^{(k)}$,

E-step : Compute $Q(\theta,\ \theta^{(k)})$ given by

$$Q(\theta,\ \theta^{(k)}) = E_{\mathbf{Z}|\mathbf{x},\theta^{(k)}}\left[\ln(f(\mathbf{x},\mathbf{Z}\mid\theta))\right]$$

M-step : Compute next value $\theta^{(k+1)}$ by

$$\theta^{(k+1)} = \arg\max_{\theta}\ Q(\theta,\ \theta^{(k)})$$

- Next question is: why does this procedure work?