Parameter Estimation Methods

Slides compiled by Sanghamitra De

Introduction

- In pattern recognition applications complete knowledge about the probabilistic structure of the problem is rarely present.
- Typically there is some vague, general knowledge about the situation, together with a number of design samples or training data — particular representatives of the patterns we want to classify.
- The problem, is to find some way to use this information to design or train the classifier.
- One approach to this problem is to use the samples to estimate the unknown probabilities and probability densities, and to use the resulting estimates as if they were true values.
- In typical supervised pattern classification problems, the estimation of the prior probabilities presents no difficulties.

Introduction

- However, in estimation of the class-conditional densities, the number of available samples always seems too small, and serious problems arise when the dimensionality of the feature vector x is large.
- If we know the number of parameters in advance and our general knowledge about the problem permits us to parameterize the conditional densities, then the severity of these problems can be reduced significantly.
- Assume that $p(x|\omega_i)$ is a normal density with mean μ_i and covariance matrix Σ_i, although we don't know the exact value of these quantities.
- So now we need to estimate values of μ_i and Σ_i
- Problem of parameter estimation is a classical one in statistics, and it can be approached in several ways.
- Two common and reasonable procedures are ***maximum likelihood estimation*** (MLE) and ***Bayesian estimation***.
- The two methods give identical results but are conceptually different.

Introduction

- Maximum likelihood and several other methods view the parameters as quantities whose values are fixed but unknown.
- The best estimate of their value is defined to be the one that maximizes the probability of obtaining the samples actually observed.
- Bayesian methods view the parameters as random variables having some known a priori distribution.
- Observation of the samples converts this to a posterior density, thereby revising our opinion about the true values of the parameters.
- In the Bayesian case, we shall see that a typical effect of observing additional samples is to sharpen the a posteriori density function, causing it to peak near the true values of the parameters. This phenomenon is known as ***Bayesian learning***.

Maximum Likelihood Estimation

- Advantages:
  - they nearly always have good convergence properties as the number of training samples increases.
  - maximum likelihood estimation often can be simpler than alternate methods, such as Bayesian techniques or other methods

Maximum Likelihood Parameter Estimation: General Principal

- Suppose we separate a collection of samples according to class, so that there are *c* data sets, D_1, ..., D_c, with the samples in D_j having been drawn independently as per the probability law $p(x|\omega_j)$
- Such samples are i.i.d. – independent and identically distributed random variables.
- Assume that $p(x|\omega_j)$ has a known parametric form, and so is determined uniquely by the value of a parameter Θ_j
- Find parameters that maximize probability of observations

- Assume c classes and
  - $p(x \mid \omega_j) \sim N(\mu_j, \Sigma_j)$
  - $p(x \mid \omega_j) \equiv p(x \mid \omega_j, \theta_j)$ where:

$$\theta = (\mu_j, \Sigma_j)$$

Maximum Likelihood Estimation

- Use n training samples in a class to estimate θ
- If D contains n independently drawn samples, $x_1, x_2, \ldots, x_n$

$$p(\mathcal{D} \mid \theta) = \prod_{k=1}^{n} p(x_k \mid \theta)$$

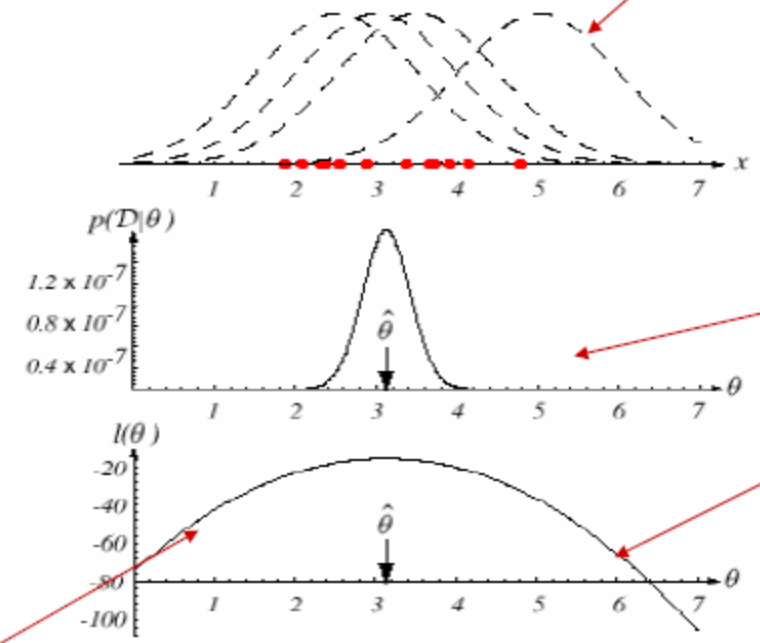
$p(\mathcal{D} \mid \theta)$ is called the likelihood of θ w.r.t. the set of samples

$\ell(\theta)$ is the log-likelihood of θ

- $l(\Theta) \equiv \ln p(D| \theta)$
- ML estimate of θ is, by definition the value that maximizes $p(D| \theta)$
- "It is the value of θ that best agrees with the actually observed training samples"

One-Dimensional Example

Four of infinite
no. of source distributions



Likelihood as a function of mean (peaks at mean)
Would be sharp peak with many samples

Log-likelihood function(also peaks at mean)

Maximizing the log-likelihood function

- Let $\theta = (\theta_1, \theta_2, \ldots, \theta_p)^t$ and let ∇_θ be the gradient operator

$$\nabla_\theta = \left[\frac{\partial}{\partial\theta_1}, \frac{\partial}{\partial\theta_2}, \ldots, \frac{\partial}{\partial\theta_p}\right]^t$$

- We define $l(\theta)$ as the log-likelihood function

$$l(\theta) = \ln p(D \mid \theta)$$

- Determine θ that maximizes the log-likelihood $\hat{\theta} = \arg\max_\theta \ell(\theta)$

- Set of necessary conditions for an optimum is:

$$\nabla_\theta l = \sum_{k=1}^{n} \nabla_\theta \ln p(x_k \mid \theta) = 0$$

MLE: The Gaussian Case: unknown μ

$p(x_i|\ \mu) \sim N(\mu, \Sigma)$

$$\ln p(x_k \mid \mu) = -\frac{1}{2}\ln\left[(2\pi)^d|\Sigma|\right] - \frac{1}{2}(x_k-\mu)^t\Sigma^{-1}(x_k-\mu)$$

$$and \quad \nabla_\mu \ln P(x_k \mid \mu) = \Sigma^{-1}(x_k-\mu)$$

$\theta = \mu$ therefore:

The ML estimate for μ must satisfy: $\sum_{k=1}^{n}\Sigma^{-1}(x_k-\hat{\mu}) = 0$

Multiplying by Σ :

$$\hat{\mu} = \frac{1}{n}\sum_{k=1}^{n} x_k$$

Just the sample mean

MLE: Gaussian Case - *unknown μ and Σ*

$\theta = (\theta_1, \theta_2) = (\mu, \sigma^2)$

Consider the univariate case where:
$\Theta_1 = \mu$ and $\Theta_2 = \sigma^2$

$$\ln p(x_k \mid \theta) = -\frac{1}{2}\ln 2\pi\theta_2 - \frac{1}{2\theta_2}(x_k - \theta_1)^2$$

$$\nabla_\theta l = \begin{pmatrix} \frac{\partial}{\partial \theta_1}(\ln p(x_k \mid \theta)) \\ \frac{\partial}{\partial \theta_2}(\ln p(x_k \mid \theta)) \end{pmatrix} = 0$$

$$\begin{cases} \frac{1}{\theta_2}(x_k - \theta_1) = 0 \\ -\frac{1}{2\theta_2} + \frac{(x_k - \theta_1)^2}{2\theta_2^2} = 0 \end{cases}$$

$$\begin{cases} \sum_{k=1}^{n} \frac{1}{\hat{\theta}_2}(x_k - \theta_1) = 0 & (1) \\ -\sum_{k=1}^{n} \frac{1}{\hat{\theta}_2} + \sum_{k=1}^{n} \frac{(x_k - \hat{\theta}_1)^2}{\hat{\theta}_2^2} = 0 & (2) \end{cases}$$

$$\mu = \sum_{k=1}^{n} \frac{x_k}{n} \quad ; \quad \sigma^2 = \frac{\sum_{k=1}^{n}(x_k - \mu)^2}{n}$$

$$\hat{\boldsymbol{\mu}} = \frac{1}{n}\sum_{k=1}^{n}\mathbf{x}_k$$

$$\widehat{\boldsymbol{\Sigma}} = \frac{1}{n}\sum_{k=1}^{n}(\mathbf{x}_k - \hat{\boldsymbol{\mu}})(\mathbf{x}_k - \hat{\boldsymbol{\mu}})^t.$$

For the multivariate case: MLE for the mean vector is the sample mean. MLE for the covariance matrix is the arithmetic average of the n matrices $(x_k - \hat{u})\ (x - \hat{u})^t$

MLE Bias

- ML estimate for σ^2 is biased (i.e. expected value of the sample variance is not equal to the true variance)

$$\mathcal{E}\left[\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^2\right]=\frac{n-1}{n}\sigma^2\neq\sigma^2.$$

- An elementary unbiased estimator for Σ is:

$$\mathbf{C}=\frac{1}{n-1}\sum_{k=1}^{n}(\mathbf{x}_k-\hat{\boldsymbol{\mu}})(\mathbf{x}_k-\hat{\boldsymbol{\mu}})^t,$$

MLE Bias

- Here **C** is the sample covariance matrix.
- If an estimator is unbiased for all distributions, then it is called *absolutely unbiased*.
- If the estimator tends to become unbiased as the number of samples become very large, then the estimator is *asymptotically unbiased*.

Bayesian estimation

Bayesian Estimation

- Although the answers we get by Bayesian estimation method will generally be nearly identical to those obtained by maximum likelihood, there is a conceptual difference:
  - ✓ In MLE $\boldsymbol{\theta}$, was supposed fix
  - ✓ In BE $\boldsymbol{\theta}$, is a random variable

  and training data allows us to convert a distribution on this variable into a posterior probability density.

Bayesian Estimation: the class-conditional densities

- The computation of posterior probabilities $P(\omega_i \mid \mathbf{x})$ lies at the heart of Bayesian classification
- Bayes' formula allows us to compute these probabilities from the Prior probabilities $P(\omega_i)$ and the class-conditional densities $p(x|\omega_i)$
- The best we can do is to compute $P(\omega_i|x)$ using all of the information at our disposal.
- Part of this information might be prior knowledge, such as knowledge of the functional forms for unknown densities and ranges for the values of unknown parameters.
- Part of this information might reside in a set of training samples.

Bayesian Estimation : the class-conditional densities

- If we again let *D* denote the set of samples, then we can emphasize the role of the samples by saying that our Goal is:
  compute $\mathbf{P(\omega_i \mid x, D)}$
- Given the sample *D*, Bayes formula can be written

$$P(\omega_i \mid \mathbf{x}, D) = \frac{p(\mathbf{x} \mid \omega_i, D) P(\omega_i \mid D)}{\sum_{j=1}^{c} p(\mathbf{x} \mid \omega_j, D) P(\omega_j \mid D)}$$

Bayesian Estimation : the class-conditional densities

- Now, assume that true values of prior probabilities are known.
- So, we substitute $P(\omega_i) = P(\omega_i \mid D)$.
- Also, as we are treating the supervised case, we can separate the training samples by class into c subsets D_1, …. D_c, with the samples in D_i belonging to ω_i.
- Now, the samples in D_i have no influence on $p(x| \omega_j, D)$ if $i \neq j$.
- So, we can work with each class separately, using only the samples in D_i to determine $p(x| \omega_i, D)$
- In conjunction to the assumption that prior probabilities are known, this gives us the equation:

Bayesian Estimation : the class-conditional densities

$$P(\omega_i|\mathbf{x},\mathcal{D}) = \frac{p(\mathbf{x}|\omega_i,\mathcal{D}_i)P(\omega_i)}{\sum_{j=1}^{c} p(\mathbf{x}|\omega_j,\mathcal{D}_j)P(\omega_j)}.$$

So, we now have ***c*** separate problems of the following form: use a set ***D*** of samples drawn independently according to the fixed but unknown probability distribution ***p(x)*** to determine **p(x|D)**. This is the central problem of Bayesian learning.

Bayesian Estimation : the parameter distribution

- Although the desired probability density **p(x)** is unknown, we assume that it has a known parametric form. The only thing assumed unknown is the value of a parameter vector **θ**.
- Also we have the fact that **p(x)** is unknown but has known parametric form by saying that the function **p(x|θ)** is completely known.
- Any information that is there about **θ** prior to observing the samples is assumed to be contained in a known prior density **p(θ)**.
- Observation of the samples converts this to a posterior density **p(θ|D)**, which, is sharply peaked about the true value of **θ**.

Bayesian Estimation : the parameter distribution

- So, our supervised learning problem is converted to an unsupervised density estimation problem.
- To this end, our goal is to compute **p(x|D)**, which is as close as we can come to obtaining the unknown **p(x)**.
- For this, we integrate the joint density **p(x, θ|D)** over **θ**.

$$p(\mathbf{x}|\mathcal{D}) = \int p(\mathbf{x}, \theta|\mathcal{D})\, d\theta,$$

where the integration extends over the entire parameter space.

- We can write **p(x, θ|D)** as the product **p(x|θ,D)p(θ|D).**
- Since the selection of **x** and that of the training samples in **D** is done independently, the first factor is merely **p(x|θ)**. That is, the distribution of **x** is known completely once we know the value of the parameter vector.
- So, above equation can be rewritten as:

$$p(\mathbf{x}|\mathcal{D}) = \int p(\mathbf{x}|\theta)p(\theta|\mathcal{D})\, d\theta.$$

Bayesian Estimation : the parameter distribution

- This equation links the desired class-conditional density $\mathbf{p(x|}\boldsymbol{D}\mathbf{)}$ to the posterior density $\mathbf{p(\theta|}\boldsymbol{D}\mathbf{)}$ for the unknown parameter vector.
- If $\mathbf{p(\theta|}\boldsymbol{D}\mathbf{)}$ peaks very sharply about some value $\hat{\boldsymbol{\theta}}$, we obtain $\mathbf{p(x|}\boldsymbol{D}\mathbf{)} \approx \mathbf{p(x|}\hat{\boldsymbol{\theta}}\mathbf{)}$, i.e., the result we would obtain by substituting the estimate $\hat{\boldsymbol{\theta}}$ for the true parameter vector.
- This result rests on the assumption that $\mathbf{p(x|\theta)}$ is smooth, and that the tails of the integral are not important.
- *These conditions are typically but not invariably the case.*
- In general, if we are less certain about the exact value of $\boldsymbol{\theta}$, this equation directs us to average $\mathbf{p(x|\theta)}$ over the possible values of $\boldsymbol{\theta}$.
- Thus, when the unknown densities have a known parametric form, the samples exert their influence on $\mathbf{p(x|}\boldsymbol{D}\mathbf{)}$ through the posterior density $\mathbf{p(\theta|}\boldsymbol{D}\mathbf{)}$.

Bayesian Parameter Estimation: Gaussian Case

Goal: Estimate θ using the a-posteriori density p(θ|D) & the desired probability density p(x|D) for the case where $p(\mathbf{x}|\boldsymbol{\mu}) \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$

- **The univariate case: p(μ | D)**

  μ is the only unknown parameter

$$p(x|\mu) \sim N(\mu, \sigma^2)$$

$$p(\mu) \sim N(\mu_0, \sigma_0^2)$$

(μ_0 and σ_0 are known!)

Bayesian Parameter Estimation: Gaussian Case

$$p(\mu \mid \mathsf{D}) = \frac{p(\mathsf{D} \mid \mu)\, p(\mu)}{\int p(\mathsf{D} \mid \mu)\, p(\mu) d\mu} \qquad \text{Bayes Formula}$$

$$= \alpha \prod_{k=1}^{k=n} p(x_k \mid \mu)\, p(\mu)$$

- α is the normalization factor dependent on D but independent of μ
- We know: $p(x_k \mid \mu) \sim N(\mu, \sigma^2)$ and $p(\mu) \sim N(\mu_0, \sigma_0^2)$
- Plugging in their gaussian expressions, extracting out factors which do not depend on μ (and getting them absorbed into constant α^n) yields:

$$p(\mu \mid D) = \alpha^n \exp\left(-\frac{1}{2}\left[\left(\frac{n}{\sigma^2} + \frac{1}{\sigma_0^2} \right)\mu^2 - 2\left(\frac{1}{\sigma^2} \sum_{k=1}^{n} x_k + \frac{\mu_0}{\sigma_0^2} \right)\mu \right] \right)$$

(from eq. 29 page 93)

Bayesian Parameter Estimation: Gaussian Case

- Observation: p(μ|D) is an exponential of a quadratic

$$p(\mu \mid D) = \alpha^n \exp\left(-\frac{1}{2}\left[\left(\frac{n}{\sigma^2} + \frac{1}{\sigma_0^2} \right)\mu^2 - 2\left(\frac{1}{\sigma^2}\sum_{k=1}^{n} x_k + \frac{\mu_0}{\sigma_0^2} \right)\mu \right] \right)$$

- It is normal! It is called a *reproducing density*

$$p(\mu \mid \mathsf{D}) \sim N(\mu_n, \sigma_n^2)$$

p(μ) is said to be a *conjugate prior*

Bayesian Parameter Estimation: Gaussian Case

$$p(\mu \mid D) = \alpha^n \exp\left(-\frac{1}{2}\left[\left(\frac{n}{\sigma^2} + \frac{1}{\sigma_0^2} \right)\mu^2 - 2\left(\frac{1}{\sigma^2}\sum_{k=1}^{n} x_k + \frac{\mu_0}{\sigma_0^2} \right)\mu \right] \right)$$

Identifying coefficients in the top equation with that of the generic Gaussian

$$p(\mu \mid D) = \frac{1}{\sqrt{2\pi}\sigma_n} \exp\left(-\frac{1}{2}\left[\frac{\mu - \mu_n}{\sigma_n} \right]^2 \right)$$

Yields expressions for μ_n and σ_n^2

$$\frac{1}{\sigma_n^2} = \frac{n}{\sigma^2} + \frac{1}{\sigma_0^2} \quad \text{and} \quad \frac{\mu_n}{\sigma_n^2} = \frac{n}{\sigma^2}\hat{\mu}_n + \frac{\mu_0}{\sigma_0^2}$$

Bayesian Parameter Estimation: Gaussian Case

Solving for μ_n and σ_n^2 yields:

$$\mu_n = \left(\frac{n\sigma_0^2}{n_0\sigma_0^2 + \sigma^2} \right) \hat{\mu}_n + \frac{\sigma^2}{n\sigma_0^2 + \sigma^2} \mu_0$$

$$and \quad \sigma_n^2 = \frac{\sigma_0^2 \sigma^2}{n\sigma_0^2 + \sigma^2}$$

From these equations we see as *n* increases:

- the variance decreases monotonically
- the estimate of p(μ|D) becomes more peaked, approaching a Dirac delta function as n approaches infinity. This behavior is called *Bayesian Learning*.

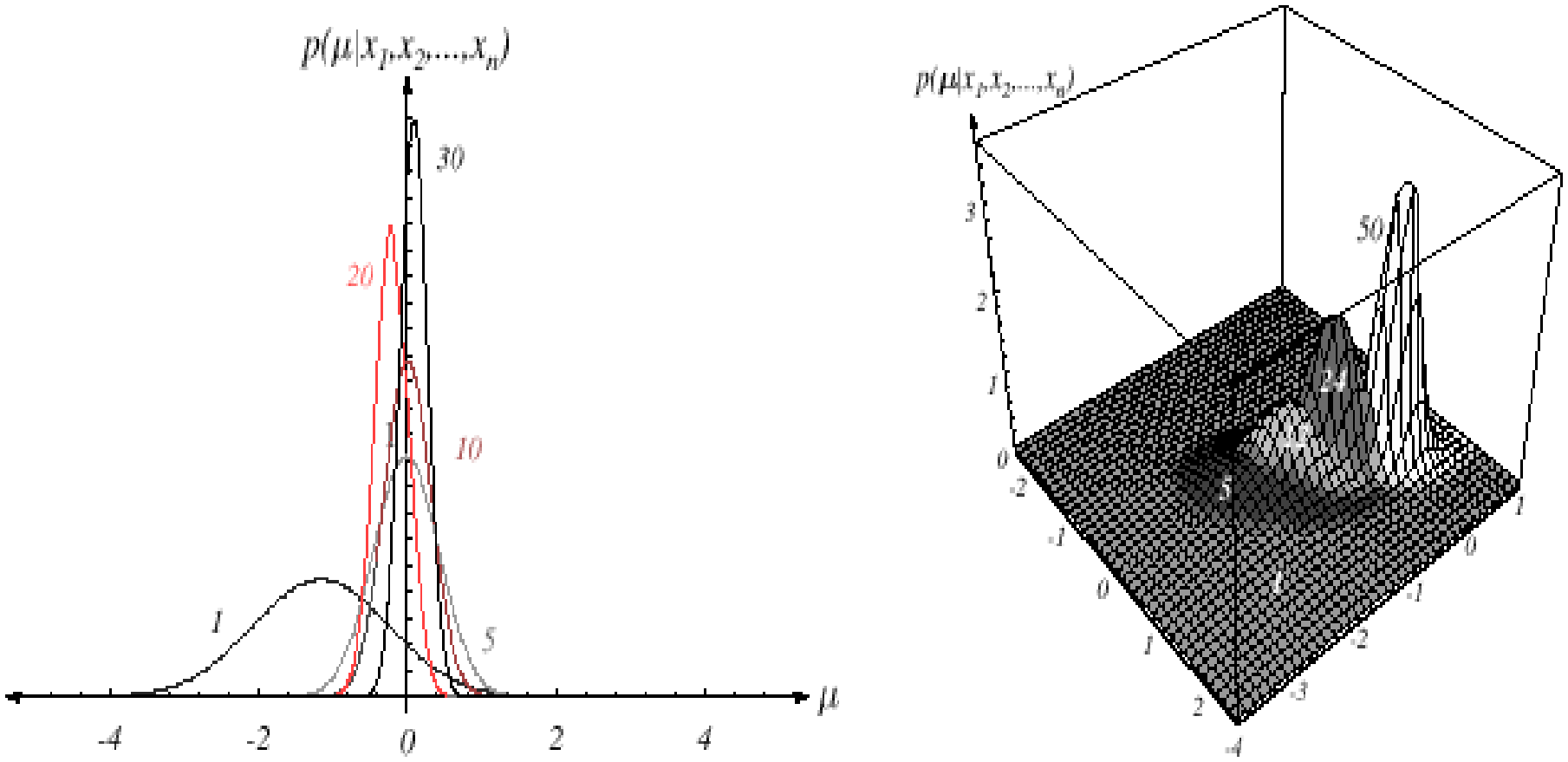


FIGURE 3.2. Bayesian learning of the mean of normal distributions in one and two dimensions. The posterior distribution estimates are labeled by the number of training samples used in the estimation. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Bayesian Parameter Estimation: Gaussian Case

- **The univariate case P(x | D)**
  - ✓ p(μ | D) computed (in preceding discussion)
  - ✓ p(x | D) remains to be computed!

$$p(x \mid \mathsf{D}) = \int p(x \mid \mu)\, p(\mu \mid \mathsf{D})\, d\mu \quad \text{is Gaussian}$$

It provides: $p(x \mid \mathsf{D}) \sim N(\mu_n, \sigma^2 + \sigma_n^2)$

We know σ^2 and how to compute μ_n and σ_n^2

Density $p(x|D)$ is the desired class-conditional density

$p(x|\omega_j, D_j)$ and together with prior probabilities $P(\omega_j)$ we obtain the Bayesian classification rule to design the classifier

$$\underset{\omega_j}{Max}\left[P(\omega_j \mid x, \mathsf{D}\right] \equiv \underset{\omega_j}{Max}\left[P(x \mid \omega_j, \mathsf{D}_j) P(\omega_j)\right]$$

Bayesian Parameter Estimation: Gaussian Case

- **The multivariate case:** (**Σ** is known but **μ** is not a direct generalization of the univariate case)
- As before assume: $p(x_k \mid \mu) \sim N(\mu, \Sigma)$ and $p(\mu) \sim N(\mu_0, \Sigma_0)$

Where Σ, Σ_0 and μ_0 are assumed to be known.

After observing a set ***D*** of ***n*** independent samples $x_1, \ldots, x_n$, Bayes formula is used to get:

$$p(\boldsymbol{\mu}|\mathcal{D}) = \alpha \prod_{k=1}^{n} p(\mathbf{x}_k|\boldsymbol{\mu}) p(\boldsymbol{\mu}) \qquad (40)$$

$$= \alpha' \exp\left[-\frac{1}{2}\left(\boldsymbol{\mu}^t(n\boldsymbol{\Sigma}^{-1} + \boldsymbol{\Sigma}_0^{-1})\boldsymbol{\mu} - 2\boldsymbol{\mu}^t\left(\boldsymbol{\Sigma}^{-1}\sum_{k=1}^{n}\mathbf{x}_k + \boldsymbol{\Sigma}_0^{-1}\boldsymbol{\mu}_0\right)\right)\right],$$

Which has the form:

$$p(\boldsymbol{\mu}|\mathcal{D}) = \alpha'' \exp\left[-\frac{1}{2}(\boldsymbol{\mu} - \boldsymbol{\mu}_n)^t \boldsymbol{\Sigma}_n^{-1}(\boldsymbol{\mu} - \boldsymbol{\mu}_n)\right]$$

Bayesian Parameter Estimation: Gaussian Case

- So, $p(\mu|D) \sim N(\mu_n, \sum_n)$ and we have a reproducing density.
- Equating the coefficients, we get analogs of eqns. in slide 28 as:

$$\Sigma_n^{-1} = n\Sigma^{-1} + \Sigma_0^{-1} \quad \& \quad \Sigma_n^{-1}\mu_n = n\Sigma^{-1}\hat{\mu}_n + \Sigma_0^{-1}\mu_0,$$

where $\hat{\boldsymbol{\mu}}_n$ is the sample mean

$$\hat{\boldsymbol{\mu}}_n = \frac{1}{n}\sum_{k=1}^{n} \mathbf{x}_k.$$

Bayesian Parameter Estimation: Gaussian Case

- With little manipulation, we get:

$$\boldsymbol{\mu}_n = \boldsymbol{\Sigma}_0\left(\boldsymbol{\Sigma}_0 + \frac{1}{n}\boldsymbol{\Sigma}\right)^{-1}\hat{\boldsymbol{\mu}}_n + \frac{1}{n}\boldsymbol{\Sigma}\left(\boldsymbol{\Sigma}_0 + \frac{1}{n}\boldsymbol{\Sigma}\right)^{-1}\boldsymbol{\mu}_0$$

(which, as in the univariate case, is a linear combination of $\hat{\boldsymbol{\mu}}_n$ and $\boldsymbol{\mu}_0$) and

$$\boldsymbol{\Sigma}_n = \boldsymbol{\Sigma}_0\left(\boldsymbol{\Sigma}_0 + \frac{1}{n}\boldsymbol{\Sigma}\right)^{-1}\frac{1}{n}\boldsymbol{\Sigma}.$$

The proof that $p(\mathbf{x}|\mathcal{D}) \sim N(\boldsymbol{\mu}_n, \boldsymbol{\Sigma} + \boldsymbol{\Sigma}_n)$ can be obtained as before by performing the integration

$$p(\mathbf{x}|\mathcal{D}) = \int p(\mathbf{x}|\boldsymbol{\mu})p(\boldsymbol{\mu}|\mathcal{D})\, d\boldsymbol{\mu}. \qquad (48)$$

Bayesian Parameter Estimation: Gaussian Case

- Since the sum of two independent, normally distributed vectors is again a normally distributed vector whose mean is the sum of the means and whose covariance matrix is the sum of the covariance matrices, we have:

$$p(\mathbf{x}|\mathcal{D}) \sim N(\boldsymbol{\mu}_n, \boldsymbol{\Sigma} + \boldsymbol{\Sigma}_n),$$

and the generalization is complete

Bayesian Parameter Estimation: General Theory

- P(x | D) computation can be applied to any situation in which the unknown density can be parameterized: the basic assumptions are:

  - ✓ The form of $P(x \mid \theta)$ is assumed known, but the value of θ is not known exactly
  - ✓ Our knowledge about θ is assumed to be contained in a known prior density $P(\theta)$
  - ✓ The rest of our knowledge θ is contained in a set D of n random variables $x_1, x_2, \ldots, x_n$ that follows P(x)

Bayesian Parameter Estimation: General Theory

The basic problem is:

"Compute the posterior density p($\boldsymbol{\theta}$ | D)" then "Derive p($\mathbf{x}$ | D)", where

$$p(\mathbf{x} \mid D) = \int p(\mathbf{x} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta} \mid D) d\boldsymbol{\theta}$$

Using Bayes formula, we have:

$$p(\boldsymbol{\theta} \mid \mathsf{D}) = \frac{p(\mathsf{D} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})}{\int p(\mathsf{D} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta}) d\boldsymbol{\theta}},$$

And by the independence assumption:

$$p(\mathsf{D} \mid \boldsymbol{\theta}) = \prod_{k=1}^{n} p(\mathbf{x}_k \mid \boldsymbol{\theta})$$

- This constitutes the formal solution to the problem.

Problems of Dimensionality

- Problems involving 50 or 100 features are common (usually binary valued)
- Note: microarray data might entail ~20000 real-valued features
- Classification accuracy dependant on
  - Dimensionality (amount of training data)
  - discrete vs continuous

Problems of Dimensionality

- Case of two class multivariate normal with the same covariance
  - $p(x|\omega_j) \sim N(\mu_j, \Sigma)$, $j=1,2$
  - Statistically independent features
  - If the priors are equal then:

$$P(error) = \frac{1}{\sqrt{2\pi}} \int_{r/2}^{\infty} e^{-u^2/2}\, du \qquad \text{(Bayes error)}$$

$$where: \quad r^2 = (\mu_1 - \mu_2)^t \Sigma^{-1} (\mu_1 - \mu_2)$$

r^2 is the squared Mahalanobis distance

$$\lim_{r \to \infty} P(error) = 0$$

Problems of Dimensionality

- If features are ***conditionally*** independent then:

$$\Sigma = diag(\sigma_1^2, \sigma_2^2, \ldots, \sigma_d^2)$$

$$r^2 = \sum_{i=1}^{i=d} \left(\frac{\mu_{i1} - \mu_{i2}}{\sigma_i} \right)^2$$

  - ✓ Do we remember what conditional independence is?

Problems of Dimensionality

- Most useful features are the ones for which the difference between the means is large relative to the standard deviation
  - ✓ Doesn't require independence
- Adding independent features helps increase *r* → reduce error
- Caution: adding features increases cost & complexity of feature extractor and classifier
- It has frequently been observed in practice that, beyond a certain point, the inclusion of additional features leads to worse rather than better performance:
  - ✓ we have the wrong model !
  - ✓ we don't have enough training data to support the additional dimensions

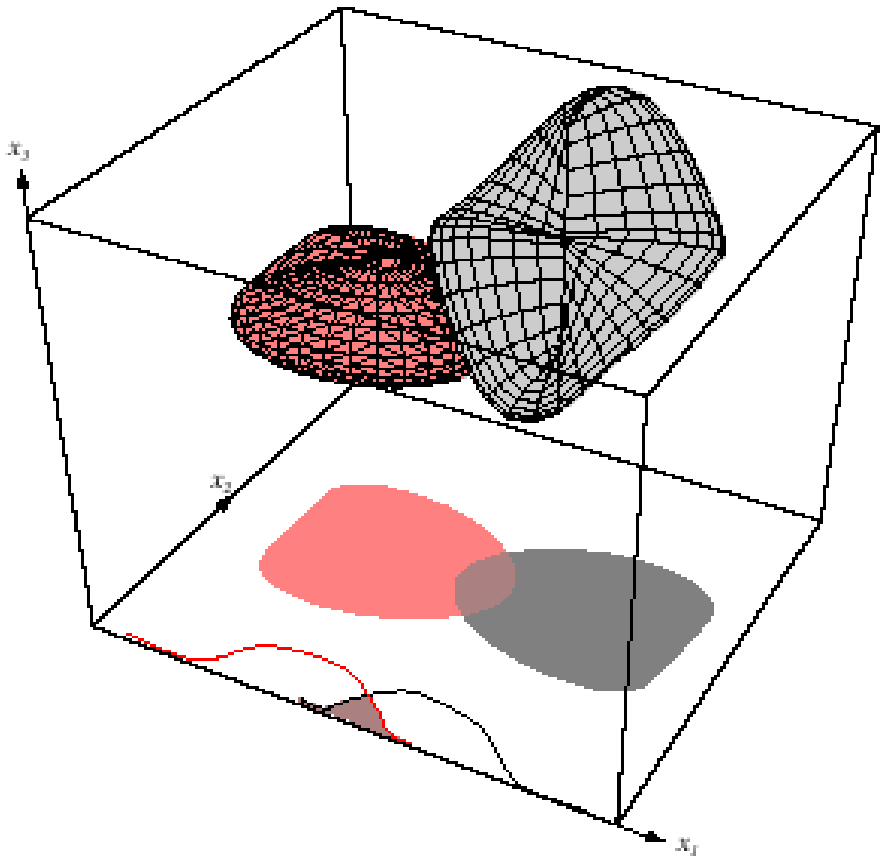


FIGURE 3.3. Two three-dimensional distributions have nonoverlapping densities, and thus in three dimensions the Bayes error vanishes. When projected to a subspace—here, the two-dimensional $x_1 - x_2$ subspace or a one-dimensional x_1 subspace—there can be greater overlap of the projected distributions, and hence greater Bayes error. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Computational Complexity

- Our design methodology is affected by the computational difficulty

  - "big oh" notation

    $f(x) = O(h(x))$ "big oh of h(x)"

    If: there exist constants c_0 and x_0 such that

    $$|f(x)| \leq c_0|h(x)| \text{ for all } x > x_0.$$

    (An upper bound on f(x) grows no worse than h(x) for sufficiently large x!)

    Example:

    $f(x) = 2+3x+4x^2$

    $g(x) = x^2$

    $f(x) = O(x^2)$

Computational Complexity

- "big oh" is not unique!

$f(x) = O(x^2)$; $f(x) = O(x^3)$; $f(x) = O(x^4)$

- "big theta" notation

$$f(x) = \theta(h(x))$$

If: there are constants x_0, c_1 and c_2 such that for $x > x_0$,

$f(x)$ always lies between $c_1 h(x)$ and $c_2 h(x)$.

So, $f(x) = \theta(x^2)$ but $f(x) \neq \theta(x^3)$

Computational Complexity

- Complexity of the ML Estimation
  - Gaussian priors in *d* dimensions classifier with *n* training samples for each of *c* classes
  - For each category, we have to compute the discriminant function

$$g(\mathbf{x}) = -\frac{1}{2}(\mathbf{x} - \overbrace{\hat{\boldsymbol{\mu}}}^{O(dn)})^t \overbrace{\hat{\boldsymbol{\Sigma}}^{-1}}^{O(d^2n)} (\mathbf{x} - \hat{\boldsymbol{\mu}}) - \overbrace{\frac{d}{2}\ln 2\pi}^{O(1)} - \underbrace{\frac{1}{2}\ln\left|\hat{\boldsymbol{\Sigma}}\right|}_{O(d^2n)} + \underbrace{\ln P(\omega)}_{O(n)}$$

Total = O(d^2n)

Total for *c* classes = O(cd^2n) ≅ O(d^2n)

- Cost increase when *d* and *n* are large!

Overfitting

- Dimensionality of model vs size of training data
  - **Issue: not enough data to support the model**
  - Possible solutions:
    - Reduce model dimensionality either by redesigning the feature extractor, by selecting an appropriate subset of the existing features, or by combining the existing features in some way
    - Make (possibly incorrect) assumptions to better estimate $\boldsymbol{\Sigma}$

Overfitting

- Estimate better $\boldsymbol{\Sigma}$
  - use data pooled from all classes
    - ✓ normalization issues
  - use pseudo-Bayesian form $\lambda\boldsymbol{\Sigma_0} + (1-\lambda)\boldsymbol{\Sigma_n}$ where Σ_0 is a prior estimate
  - by thresholding sample covariance matrix $\boldsymbol{\Sigma}$
    - ✓ reduces chance correlations
  - assume statistical independence
    - ✓ zero all off-diagonal elements

Shrinkage

- **Issue: not enough data to support the model**
- Shrinkage: weighted combination of common and individual covariances. If i is an index on the c categories in question, then:

$$\boldsymbol{\Sigma}_i(\alpha) = \frac{(1-\alpha)n_i\boldsymbol{\Sigma}_i + \alpha n\boldsymbol{\Sigma}}{(1-\alpha)n_i + \alpha n} \quad \text{for } 0 < \alpha < 1$$

- We can also shrink the estimate of the common covariances toward the identity matrix

$$\boldsymbol{\Sigma}(\beta) = (1-\beta)\boldsymbol{\Sigma} + \beta\mathbf{I} \quad \text{for} \quad 0 < \beta < 1$$

Gaussian Mixture Model (GMM)

- A **Gaussian Mixture Model (GMM)** is a parametric probability density function represented as a weighted sum of Gaussian component densities.
- GMMs are commonly used as a parametric model of the probability distribution of continuous measurements or features in a biometric system, such as vocal-tract related spectral features in a speaker recognition system.
- GMM parameters are estimated from training data using the iterative **Expectation-Maximization (EM)** algorithm or **Maximum A Posteriori (MAP)** estimation from a well-trained prior model.

GMM

- Gaussian distribution has significant limitations when it comes to modelling real data sets.
- Consider the example shown in next slide:
  - This is known as the 'Old Faithful' data set, and comprises 272 measurements of the eruption of the Old Faithful geyser at Yellowstone National Park in the USA.
  - Each measurement comprises the duration of the eruption in minutes (horizontal axis) and the time in minutes to the next eruption (vertical axis).
  - The data set forms two dominant clumps, and that a simple Gaussian distribution is unable to capture this structure, whereas a linear superposition of two Gaussians gives a better characterization of the data set.

GMM

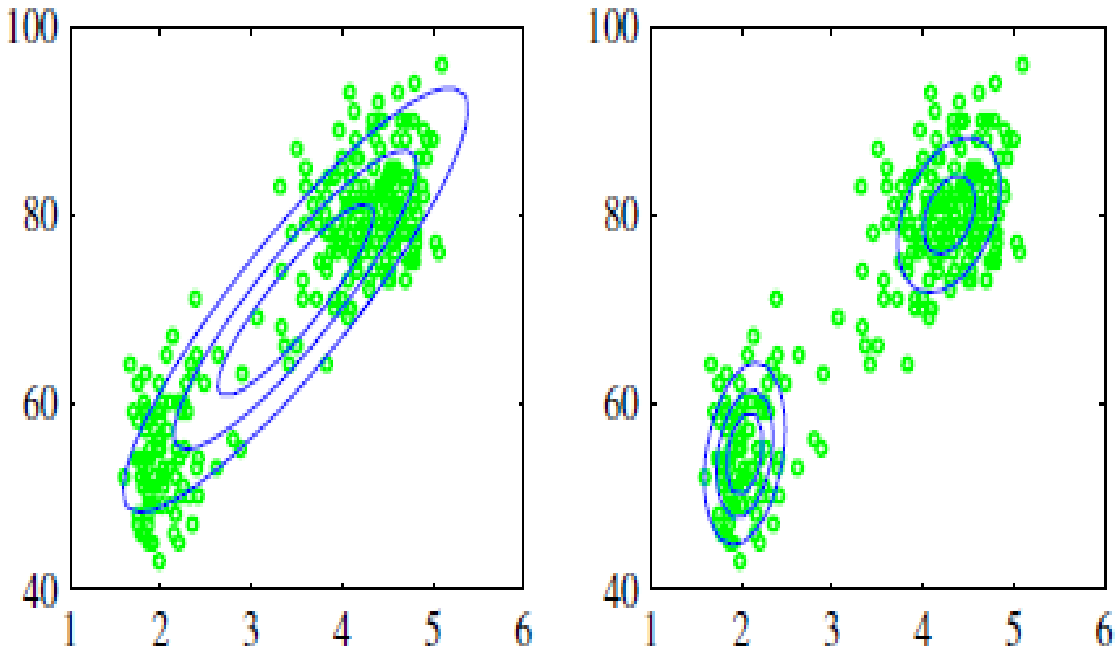


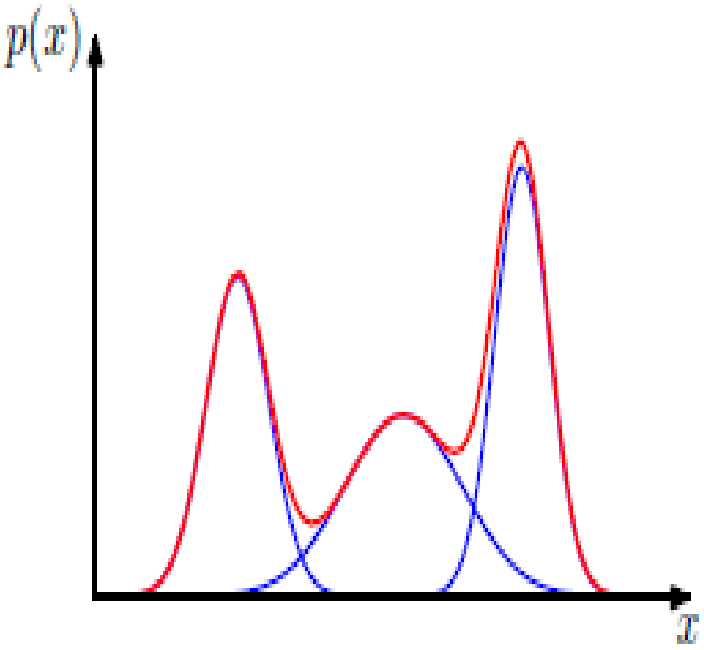
Figure 2.21 Plots of the 'old faithful' data in which the blue curves show contours of constant probability density. On the left is a single Gaussian distribution which has been fitted to the data using maximum likelihood. Note that this distribution fails to capture the two clumps in the data and indeed places much of its probability mass in the central region between the clumps where the data are relatively sparse. On the right the distribution is given by a linear combination of two Gaussians which has been fitted to the data by maximum likelihood using techniques discussed Chapter 9, and which gives a better representation of the data.

GMM

- Such superpositions, formed by taking linear combinations of more basic distributions such as Gaussians, can be formulated as probabilistic models known as *mixture distributions*.
- In the next figure it can be seen that a linear combination of Gaussians can give rise to very complex densities.
- By using a sufficient number of Gaussians, and by adjusting their means and covariances as well as the coefficients in the linear combination, almost any continuous density can be approximated to arbitrary accuracy.

GMM

Figure 2.22 Example of a Gaussian mixture distribution in one dimension showing three Gaussians (each scaled by a coefficient) in blue and their sum in red.



GMM

- Therefore consider a superposition of K Gaussian densities of the form:

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

- The above is called a *mixture of Gaussians*.
- Each Gaussian density $N(x|\mu_k,\Sigma_k)$ is called a *component* of the mixture and has its own mean μ_k and covariance Σ_k.
- Contour and surface plots for a Gaussian mixture having 3 components are shown in next figure
- Generally speaking, mixture models can comprise linear combinations of other distributions.

GMM

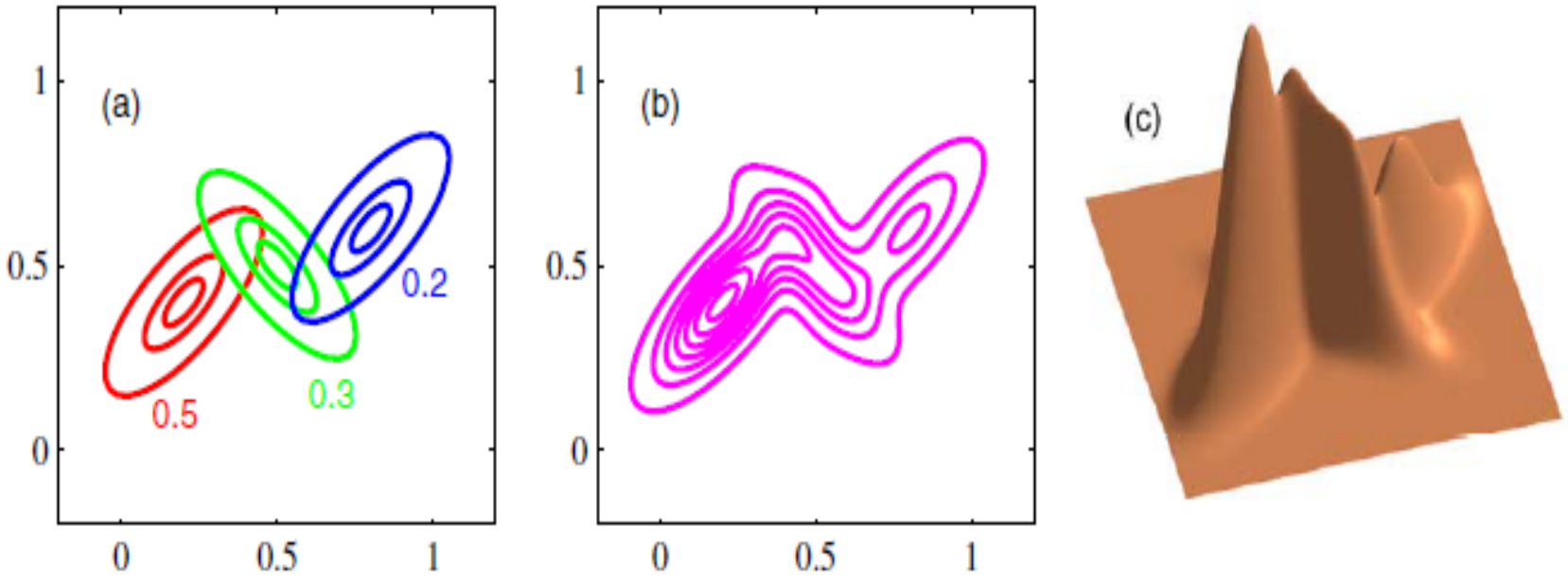


Figure 2.23 Illustration of a mixture of 3 Gaussians in a two-dimensional space. (a) Contours of constant density for each of the mixture components, in which the 3 components are denoted red, blue and green, and the values of the mixing coefficients are shown below each component. (b) Contours of the marginal probability density $p(\mathbf{x})$ of the mixture distribution. (c) A surface plot of the distribution $p(\mathbf{x})$.

GMM

- The parameters π_k in the equation (see previous slide) are called *mixing coefficients*.
- If we integrate both sides of the equation with respect to x, and note that both $p(x)$ and the individual Gaussian components are normalized, we obtain:

$$\sum_{k=1}^{K} \pi_k = 1.$$

- Also, the requirement that $p(x) \geq 0$, together with $N(x|\mu_k,\Sigma_k) \geq 0$, implies $\pi_k \geq 0$ for all k. Combining this with the condition above, we obtain:

$$0 \leqslant \pi_k \leqslant 1.$$

GMM

- Therefore, the mixing coefficients satisfy the requirements to be probabilities.
- From the sum and product rules, the marginal density is given by:

$$p(\mathbf{x}) = \sum_{k=1}^{K} p(k)p(\mathbf{x}|k)$$

- This is equivalent to eqn. in slide 53 in which we can view π_k = p(k) as the prior probability of picking the kth component, and the density $N(x|\mu_k,\Sigma_k)$ = $p(x|k)$ as the probability of x conditioned on k.
- NOTE: an important role is played by the posterior probabilities $p(k|x)$, which are also known as *responsibilities*.

GMM

- From Bayes' theorem these are given by:

$$\begin{aligned}\gamma_k(\mathbf{x}) &\equiv p(k|\mathbf{x}) \\ &= \frac{p(k)p(\mathbf{x}|k)}{\sum_l p(l)p(\mathbf{x}|l)} \\ &= \frac{\pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_l \pi_l \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)}.\end{aligned}$$

- The form of the Gaussian mixture distribution is governed by the parameters π, μ and Σ, where we have used the notation $\pi \equiv \{\pi_1, \ldots, \pi_K\}$, $\mu \equiv \{\mu_1, \ldots, \mu_K\}$ and $\Sigma \equiv \{\Sigma_1, \ldots \Sigma_K\}$.
- One way to set the values of these parameters is to use maximum likelihood.

GMM

- From eqn. in slide 53 the log of the likelihood function is given by

$$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$

  where $X = \{x_1, \ldots, x_N\}$.

- This situation is now more complex than with a single Gaussian, due to the presence of the summation over k inside the logarithm.
- As a result, the maximum likelihood solution for the parameters no longer has a closed-form analytical solution.
- One approach to maximizing the likelihood function is to use iterative numerical optimization techniques.
- Other alternative is to use *expectation maximization* (EM) algorithm.

Mixtures of Gaussians

- Recall that the Gaussian mixture distribution can be written as a linear superposition of Gaussians in the form

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k).$$

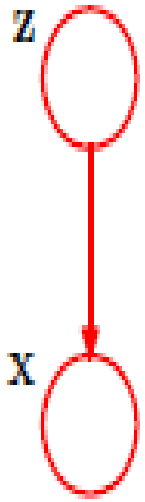
- We now turn to a formulation of Gaussian mixtures in terms of discrete latent variables.
- This will provide us with a deeper insight into this important distribution, and will serve to motivate the expectation-maximization algorithm.

Mixtures of Gaussians

- Let us introduce a K-dimensional binary random variable z having a 1-of-K representation in which a particular element z_k is equal to 1 and all other elements are equal to 0.
- The values of z_k therefore satisfy $z_k \in \{0, 1\}$ and $\Sigma_k z_k = 1$
- It can be seen that there are K possible states for the vector z according to which element is nonzero.
- We shall define the joint distribution p(x, z) in terms of a marginal distribution p(z) and a conditional distribution p(x|z), corresponding to the graphical model in next figure.

Mixtures of Gaussians

Graphical representation of a mixture model, in which the joint distribution is expressed in the form $p(\mathbf{x}, \mathbf{z}) = p(\mathbf{z})p(\mathbf{x}|\mathbf{z})$.



Mixtures of Gaussians

- The marginal distribution over z is specified in terms of the mixing coefficients π_k, such that: $p(z_k = 1) = \pi_k$

where the parameters $\{\pi_k\}$ must satisfy $0 \leq \pi_k \leq 1$ together with:

$$\sum_{k=1}^{K} \pi_k = 1$$

in order to be valid probabilities.

Since z uses a 1-of-K representation, we can also write this distribution in the form:

$$p(\mathbf{z}) = \prod_{k=1}^{K} \pi_k^{z_k}.$$

Mixtures of Gaussians

- Similarly, the conditional distribution of x given a particular value for z is a Gaussian:

$$p(\mathbf{x}|z_k = 1) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

- This can also be written in the form:

$$p(\mathbf{x}|\mathbf{z}) = \prod_{k=1}^{K} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_k}.$$

- The joint distribution is given by $p(\mathbf{z})p(\mathbf{x}|\mathbf{z})$, and the marginal distribution of $\mathbf{x}$ is then obtained by summing the joint distribution over all possible states of $\mathbf{z}$ to give:

$$p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{z})p(\mathbf{x}|\mathbf{z}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

where we have made use of previous two equations.

Mixtures of Gaussians

- Thus the marginal distribution of **x** is a Gaussian mixture of the form as in slide 59.
- If we have several observations $x_1, \ldots, x_N$, then, because we have represented the marginal distribution in the form:

$$p(x) = \Sigma_z \, p(x, z)$$

it follows that for every observed data point x_n there is a corresponding latent variable z_n.

- Thus, an equivalent formulation of the Gaussian mixture involving an explicit latent variable is found.
- Now, we can work with the joint distribution $p(\mathbf{x}, \mathbf{z})$ instead of the marginal distribution $p(\mathbf{x})$.
- This will lead to significant simplifications, specially through the introduction of the expectation-maximization (EM) algorithm.

Mixtures of Gaussians

- Another quantity that will play an important role is the conditional probability of z given x.
- We shall use $\gamma(z_k)$ to denote $p(z_k = 1|x)$, whose value can be found using Bayes' theorem:

$$\gamma(z_k) \equiv p(z_k = 1|\mathbf{x}) = \frac{p(z_k = 1)p(\mathbf{x}|z_k = 1)}{\sum_{j=1}^{K} p(z_j = 1)p(\mathbf{x}|z_j = 1)}$$

$$= \frac{\pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}.$$

Mixtures of Gaussians

- Here, π_k can be viewed as the prior probability of $z_k = 1$, and the quantity $\gamma(z_k)$ as the corresponding posterior probability once we have observed x.
- $\gamma(z_k)$ can also be viewed as the *responsibility* that component *k* takes for 'explaining' the observation x.

Expectation Maximization

- The EM algorithm
  - ✓ An excellent way of doing our unsupervised learning problem.
  - ✓ Many, many other uses, including inference of Hidden Markov Models
  - ✓ The expectation maximization (EM) algorithm, is a general technique for finding maximum likelihood solutions for probabilistic models having latent variables.

EM Algorithm

- Consider a probabilistic model in which we collectively denote all of the observed variables by X and all of the hidden variables by Z.
- The joint distribution p(X,Z|θ) is governed by a set of parameters denoted by θ.
- Our goal is to maximize the likelihood function that is given by:

$$p(\mathbf{X}|\theta) = \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\theta).$$

EM Algorithm

- Next we introduce a distribution q(Z) defined over the latent variables.
- It is observed that, for any choice of q(Z), the following decomposition holds:

$$\ln p(\mathbf{X}|\theta) = \mathcal{L}(q,\theta) + \mathrm{KL}(q\|p)$$

where we have defined:

$$\mathcal{L}(q,\theta) = \sum_{\mathbf{Z}} q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{X},\mathbf{Z}|\theta)}{q(\mathbf{Z})} \right\}$$

$$\mathrm{KL}(q\|p) = -\sum_{\mathbf{Z}} q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{Z}|\mathbf{X},\theta)}{q(\mathbf{Z})} \right\}.$$

EM Algorithm

- Note that $\mathcal{L}(q, \theta)$ is a functional of the distribution q(Z), and a function of the parameters θ.
- The last two equations differ in sign and also that $\mathcal{L}(q, \theta)$ contains the joint distribution of X and Z while KL(q||p) contains the conditional distribution of Z given X.
- To verify the 1st equation of slide 69, we first make use of the product rule of probability to give:

$$\ln p(\mathbf{X}, \mathbf{Z}|\theta) = \ln p(\mathbf{Z}|\mathbf{X}, \theta) + \ln p(\mathbf{X}|\theta)$$

which we then substitute into the expression for $\mathcal{L}(q, \theta)$.

EM Algorithm

- This gives rise to two terms, one of which cancels KL(q||p) while the other gives the required log likelihood ln p(X|θ) after noting that q(Z) is a normalized distribution that sums to 1.

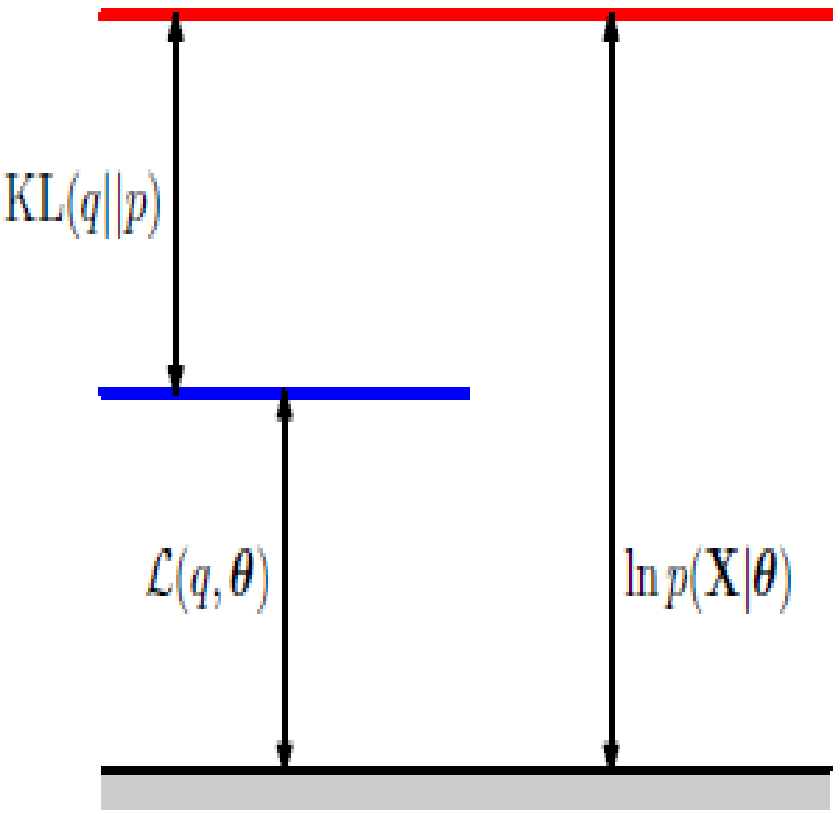
NOTE:

KL(q||p) is the Kullback-Leibler divergence between q(Z) and the posterior distribution p(Z|X, θ). Recall that the Kullback-Leibler divergence satisfies KL(q||p) ≥ 0, with equality if, and only if, q(Z) = p(Z|X, θ). So, $\mathcal{L}(q, \theta) \leq \ln p(X|\theta)$, in other words $\mathcal{L}(q, \theta)$ is a lower bound on ln p(X|θ)

- The EM algorithm is a two-stage iterative optimization technique for finding maximum likelihood solutions.
- The 1st equation of slide 69 can be used to define the EM algorithm and to demonstrate that it does indeed maximize the log likelihood.

EM Algorithm

Figure 9.11 Illustration of the decomposition given by (9.70), which holds for any choice of distribution $q(\mathbf{Z})$. Because the Kullback-Leibler divergence satisfies $\mathrm{KL}(q\|p) \geqslant 0$, we see that the quantity $\mathcal{L}(q, \boldsymbol{\theta})$ is a lower bound on the log likelihood function $\ln p(\mathbf{X}|\boldsymbol{\theta})$.

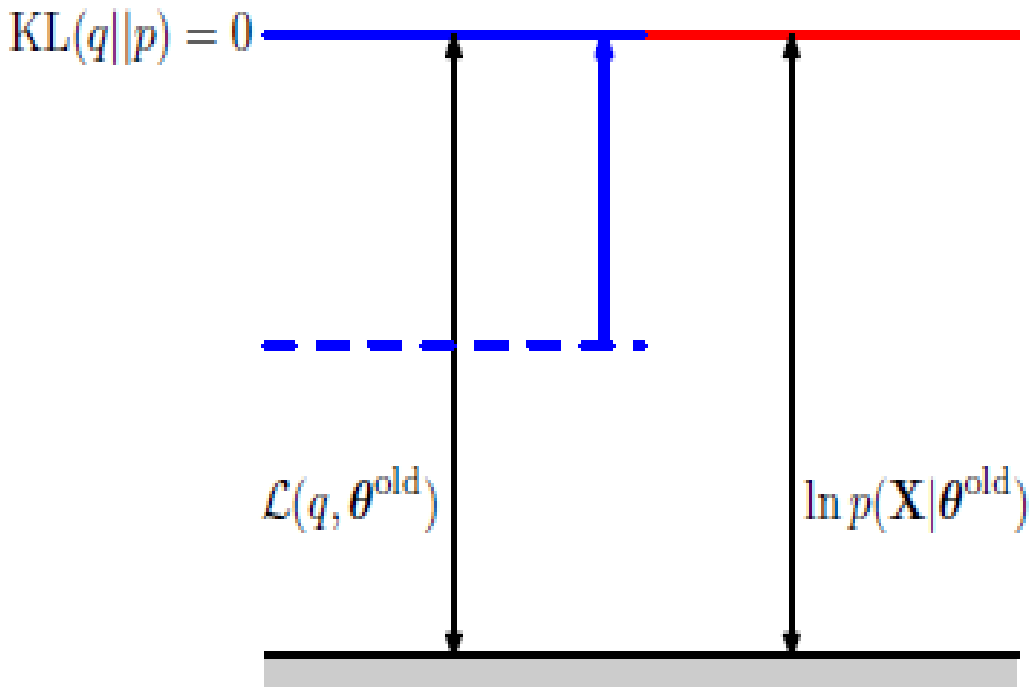


EM Algorithm

- Suppose that the current value of the parameter vector is θ^{old}.
- In the **E step**, the lower bound $\mathcal{L}(q, \theta^{old})$ is maximized with respect to $q(Z)$ while holding θ^{old} fixed.
- The solution to this maximization problem is easily seen by noting that the value of $\ln p(X|\theta^{old})$ does not depend on $q(Z)$ and so the largest value of $\mathcal{L}(q, \theta^{old})$ will occur when $q(Z)$ is equal to the posterior distribution $p(Z|X, \theta^{old})$.
- In this case, the lower bound will equal the log likelihood, as shown in the next figure.

EM Algorithm

Figure 9.12 Illustration of the E step of the EM algorithm. The q distribution is set equal to the posterior distribution for the current parameter values θ^{old}, causing the lower bound to move up to the same value as the log likelihood function, with the KL divergence vanishing.

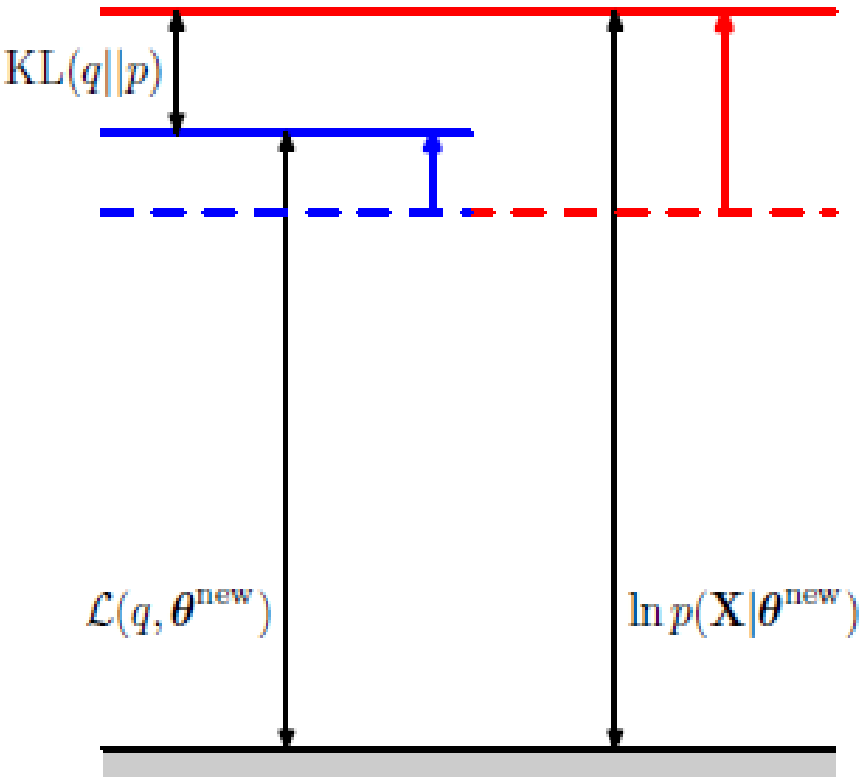


EM Algorithm

- In the subsequent **M step**, the distribution $q(Z)$ is held fixed and the lower bound $\mathcal{L}(q, \theta)$ is maximized with respect to θ to give some new value θ^{new}.
- This will cause the lower bound $\mathcal{L}$ to increase (unless it is already at a maximum), which will cause the corresponding log likelihood function to increase.
- Since the distribution q is determined using the old parameter values rather than the new values and is held fixed during the M step, it will not equal the new posterior distribution $p(Z|X, \theta^{new})$, and hence there will be a nonzero KL divergence.
- The increase in the log likelihood function is therefore greater than the increase in the lower bound, as shown in next figure.

EM Algorithm

Figure 9.13 Illustration of the M step of the EM algorithm. The distribution $q(\mathbf{Z})$ is held fixed and the lower bound $\mathcal{L}(q, \theta)$ is maximized with respect to the parameter vector θ to give a revised value θ^{new}. Because the KL divergence is nonnegative, this causes the log likelihood $\ln p(\mathbf{X}|\theta)$ to increase by at least as much as the lower bound does.



EM Algorithm

- If we substitute $q(Z) = p(Z|X, \theta^{old})$ into 2nd equation of slide 69, we see that, after the E step, the lower bound takes the form:

$$\begin{aligned} \mathcal{L}(q, \theta) &= \sum_{Z} p(Z|X, \theta^{old}) \ln p(X, Z|\theta) - \sum_{Z} p(Z|X, \theta^{old}) \ln p(Z|X, \theta^{old}) \\ &= \mathcal{Q}(\theta, \theta^{old}) + \text{const} \end{aligned} \tag{9.74}$$

where the constant is simply the negative entropy of the q distribution and is therefore independent of θ.

EM Algorithm

- In the M step, revised parameter estimate θ^{new} is determined by maximizing the function:

$$\theta^{\text{new}} = \arg\max_{\theta} \mathcal{Q}(\theta, \theta^{\text{old}}).$$

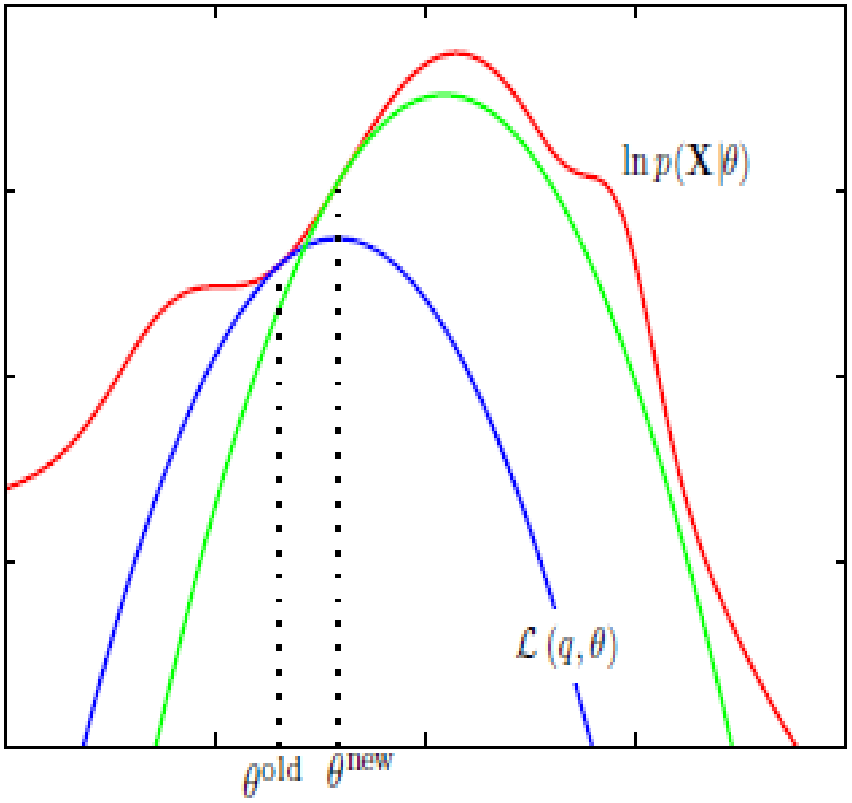
- In the definition of $Q(\theta, \theta^{old})$, the logarithm acts directly on the joint distribution $p(X,Z|\theta)$.

EM Algorithm

- So, in the M step, the quantity that is being maximized is the expectation of the complete-data log likelihood.
- Note that the variable θ over which we are optimizing appears only inside the logarithm.
- If the joint distribution $p(Z,X|\theta)$ comprises a member of the exponential family, or a product of such members, then we see that the logarithm will cancel the exponential and lead to an M step that will be typically much simpler than the maximization of the corresponding incomplete-data log likelihood function $p(X|\theta)$.

EM Algorithm

Figure 9.14 The EM algorithm involves alternately computing a lower bound on the log likelihood for the current parameter values and then maximizing this bound to obtain the new parameter values. See the text for a full discussion.



EM Algorithm (explanation of fig. in last slide)

- The **red curve** depicts the (incomplete data) log likelihood function whose value we wish to maximize.
- We start with some initial parameter value θ^{old}, and in the first E step we evaluate the posterior distribution over latent variables, which gives rise to a lower bound $\mathcal{L}(\theta, \theta^{(old)})$ whose value equals the log likelihood at $\theta^{(old)}$, as shown by the **blue curve**.
- Note that the bound makes a tangential contact with the log likelihood at $\theta^{(old)}$, so that both curves have the same gradient.
- This bound is a convex function having a unique maximum (for mixture components from the exponential family).
- In the M step, the bound is maximized giving the value $\theta^{(new)}$, which gives a larger value of log likelihood than $\theta^{(old)}$.
- The subsequent E step then constructs a bound that is tangential at $\theta^{(new)}$ as shown by the **green curve**.

EM for Gaussian Mixtures

Given a Gaussian mixture model, the goal is to maximize the likelihood function with respect to the parameters (comprising the means and covariances of the components and the mixing coefficients).

1) Initialize the means μ_k, covariances Σ_k and mixing coefficients π_k, and evaluate the initial value of the log likelihood.
2) **E step.** Evaluate the responsibilities using the current parameter values

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}.$$

EM for Gaussian Mixtures

3) **M step.** Re-estimate the parameters using the current responsibilities

$$\boldsymbol{\mu}_k^{\text{new}} = \frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{nk})\mathbf{x}_n$$

$$\boldsymbol{\Sigma}_k^{\text{new}} = \frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{nk})\left(\mathbf{x}_n-\boldsymbol{\mu}_k^{\text{new}}\right)\left(\mathbf{x}_n-\boldsymbol{\mu}_k^{\text{new}}\right)^{\mathrm{T}}$$

$$\pi_k^{\text{new}} = \frac{N_k}{N}$$

where

$$N_k = \sum_{n=1}^{N}\gamma(z_{nk}).$$

EM for Gaussian Mixtures

4) Evaluate the log likelihood:

$$\ln p(\mathbf{X}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$

and check for convergence of either the parameters or the log likelihood. If the convergence criterion is not satisfied return to step 2.

EM Algorithm

Algorithm 1 (Expectation-Maximization)

1 begin initialize $\theta^0, T, i = 0$
2 do $i \leftarrow i + 1$
3 E step : compute $Q(\theta;\, \theta^i)$
5 M step : $\theta^{i+1} \leftarrow \arg\max_{\theta} Q(\theta;\, \theta^i)$
6 until $Q(\theta^{i+1};\, \theta^i) - Q(\theta^i;\, \theta^{i-1}) \leq T$
7 return $\hat{\theta} \leftarrow \theta^{i+1}$
8 end

References

- https://www.geeksforgeeks.org/gaussian-mixture-model/
- Pattern Classification (2nd ed) by R. O. Duda, P. E. Hart and D. G. Stork, John Wiley & Sons, 2000
- Pattern Recognition and Machine Learning by Christopher M. Bishop, Springer.

End