Intelligent Data Analysis

Week 8: Gaussian Mixture Models

Martin Russell

Objectives

- Review basic statistical modelling
- Review notion of probability density function (PDF)
- Revise the properties of Gaussian PDFs
- Multivariate Gaussian PDFs
- To introduce Gaussian Mixture Models (GMMs)
- Describe GMM parameter estimation – the E-M algorithm
- Introduce GMM supervectors – vector representation of continuous data

Discrete random variables

- Suppose that Y is a **random variable** which can take any value in a discrete set $X=\{x_1,x_2,\ldots,x_M\}$
- Suppose that $y_1,y_2,\ldots,y_N$ are samples of the random variable Y $C_1, C_2, \ldots, C_N$
- If c_m is the number of times that $y_n = x_m$ then an estimate of the probability that y_n takes the value x_m is given by:

$$P(x_m) = P(y_n = x_m) \approx \frac{c_m}{N}$$

Continuous Random Variables

- In most practical applications the data are not restricted to a finite set of values – they can take any value in real *N*-dimensional space
- Counting the number of occurrences of each value is no longer a viable way of estimating probabilities
- A **probability density function (PDF)** on *N*-dimensional space V is a function $p{:}V \longrightarrow R$ such that:

$$p(v) \geq 0, \forall v \in V, \int_v p(v)dv = 1$$

UNIVERSITY OF BIRMINGHAM

Continuous Random Variables

- A **random variable** X defined on V is governed by a probability density function p if, for any $U \subseteq V$

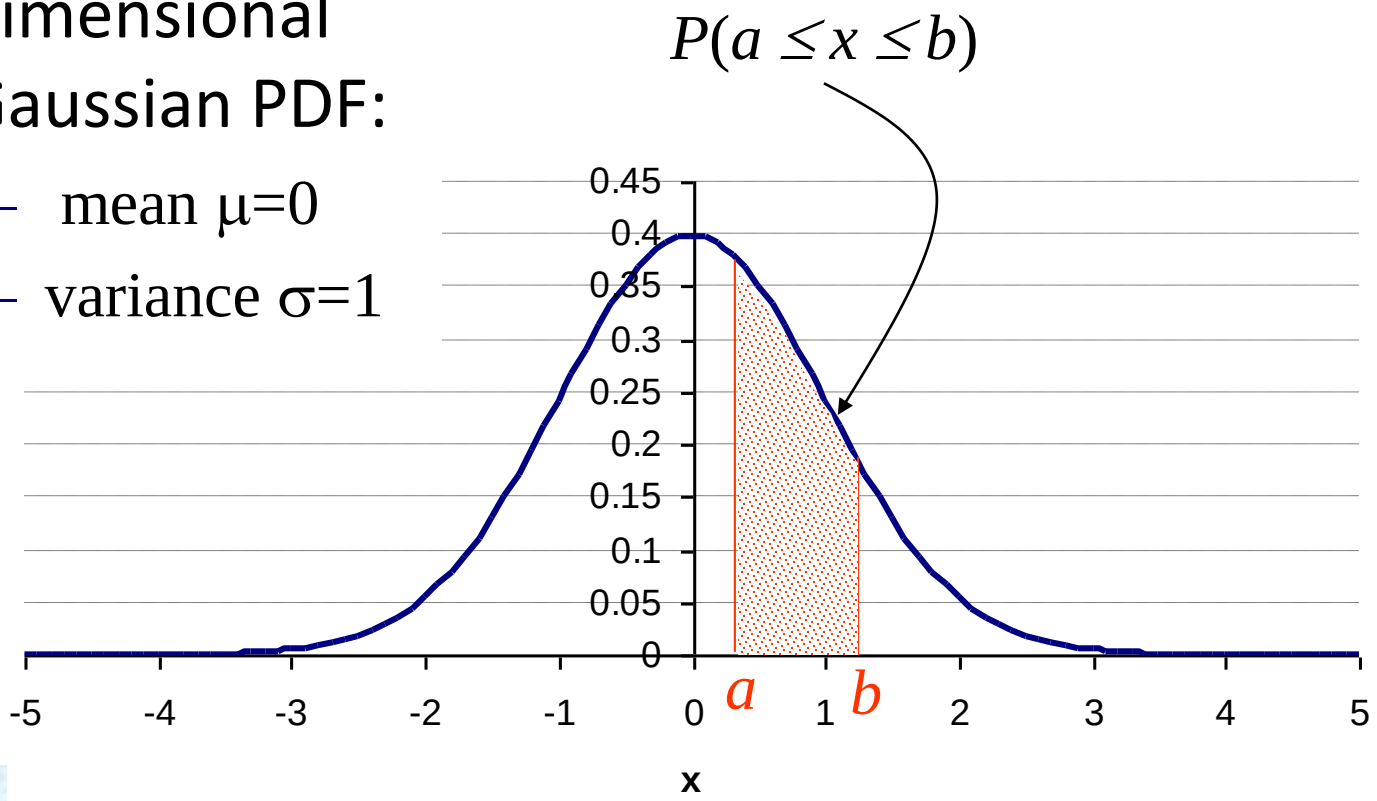
$$prob(X \in U) = \int_U p(v)dv = 1$$

Continuous Random Variables

- Familiar example is a **normal**, or **Gaussian** PDF
- A (scalar/univariate) Gaussian probability density function (PDF) is defined by two parameters – its **mean** μ and **variance** σ
- For a multivariate Gaussian PDF defined on a vector space, μ is the **mean vector** and σ is the **covariance matrix**

1-dimensional Gaussian PDF

- 'Standard' 1-dimensional Gaussian PDF:
  - mean $\mu=0$
  - variance $\sigma=1$



1-dimensional Gaussian PDF

- For a 1-dimensional Gaussian PDF p with mean μ and variance σ:

$$p(x) = p(x \mid \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{(x-\mu)^2}{2\sigma} \right)$$

Constant to ensure area under curve is 1

Defines 'bell' shape

Standard Deviation

- **Standard deviation** is the square root of the variance
- For a Gaussian PDF:
  - 68% of the area under the curve lies within one standard deviation (s.d.) of the mean
  - 95% of the area under the curve lies within two s.ds of the mean
  - 99% of the area under the curve lies within three standard deviations of the mean

Standard Deviation

- In other words, if $s = \sqrt{\sigma}$ then:

$$P(\mu - s \leq x \leq \mu + s) = 0.68$$
$$P(\mu - 2s \leq x \leq \mu + 2s) = 0.95$$
$$P(\mu - 3s \leq x \leq \mu + 3s) = 0.99$$

Multivariate Gaussian PDFs

- In the case where the random variable takes N-dimensional **vector** values the PDF is a **multivariate** Gaussian PDF and is given by:

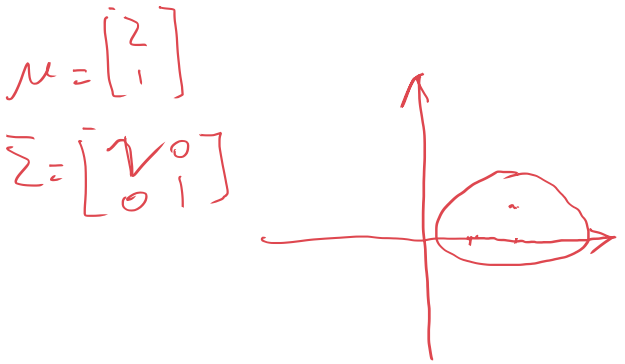
$$p(x) = \frac{1}{\sqrt{(2\pi)^N |\Sigma|}} \exp\left(\frac{-1}{2} (m - x)^T \Sigma^{-1} (m - x) \right)$$

where m is the N-dimensional **vector mean** and Σ is the $N \times N$ **covariance matrix**

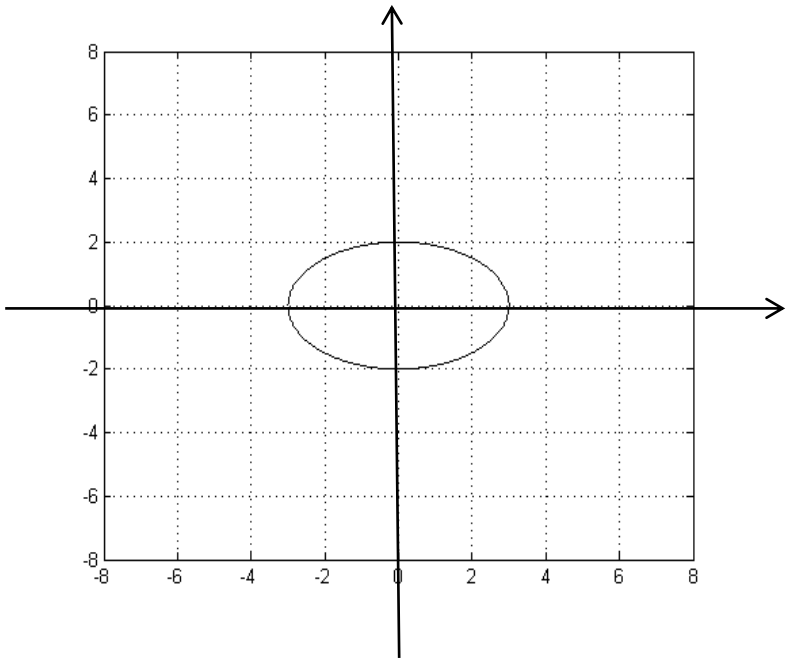
Visualising multivariate Gaussian PDFs

- It is simple to sketch a (1 dimensional) Gaussian PDF, using the 1, 2 and 3 standard-deviation rules and the value of $p(m)$
- A 2 dimensional Gaussian PDF can be sketched using MatLab's 3D plotting facility
- A simple way to visualize 1 2D Gaussian PDF is by drawing the **1-standard deviation contour** – the set of points that lie one standard deviation from the mean

Example



- If $\Sigma = \begin{bmatrix} 9 & 0 \\ 0 & 4 \end{bmatrix}$, standard deviations in 'x' and 'y' directions are 3 and 2, respectively, and the 1 s.d. contour is an elipse:



Example 2:

- Now suppose $\Sigma = \begin{bmatrix} 7.75 & 2.17 \\ 2.17 & 5.25 \end{bmatrix}$ and $m = \begin{bmatrix} 2 \\ 4 \end{bmatrix}$
- Calculate the eigenvalue decomposition of Σ

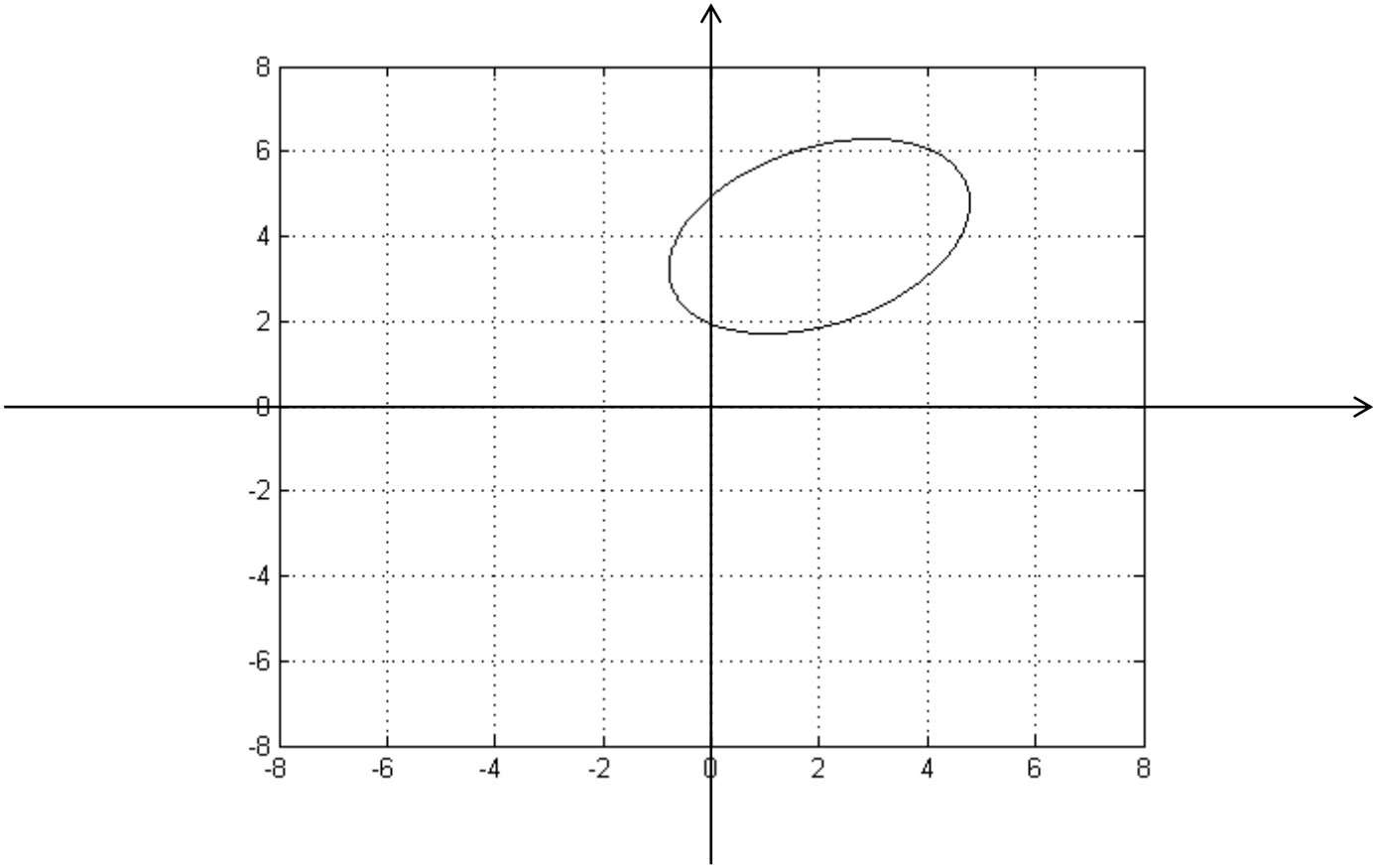
$$\Sigma = UDU^T = \begin{bmatrix} \frac{\sqrt{3}}{2} & \frac{-1}{2} \\ \frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} 9 & 0 \\ 0 & 4 \end{bmatrix} \begin{bmatrix} \frac{\sqrt{3}}{2} & \frac{1}{2} \\ \frac{-1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix}$$

Example 2 (continued)

- Note U is a rotation through 30^o
- Hence the one standard deviation contour is the same as in the previous example, but rotated through 30^o and translated by

$$m = \begin{bmatrix} 2 \\ 4 \end{bmatrix}$$

Example 2 (continued)



Fitting a Gaussian PDF to Data

- Suppose $y = y_1, \ldots, y_n, \ldots, y_N$ is a set of N data values
- For a Gaussian PDF p with mean μ and variance σ, define:

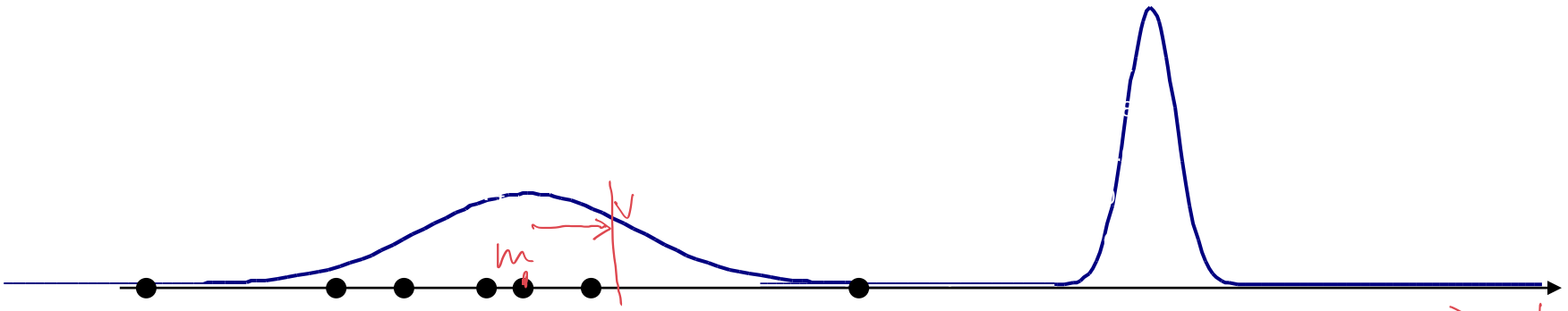
$$p(y \mid \mu, \sigma) = \prod_{n=1}^{N} p(y_n \mid \mu, \sigma)$$

- How do we choose μ and σ to maximise $p(y|\mu, \sigma)$?

Fitting a Gaussian PDF to Data

Good fit

Poor fit



$Y = \{y_1, y_2, \cdots, y_N\}$ GR. Intuition: choose m, v s.t. $P(Y)$ /max

max. $P(Y) = \prod_{n=1}^{N} P(y_n) = \prod_{n=1}^{N} \frac{1}{\sqrt{2\pi v}} e^{\left(\frac{(x-m)^2}{2v}\right)}$

MLE: $P(Y|m,v)$ → Likelihood

Bayes': $P(m,v|Y) = \frac{P(Y|m,v)P(m,v)}{P(Y)}$ ← posterior

$\log(P(Y)) = \sum_{n=1}^{N} \left[\log\left(\frac{1}{\sqrt{2\pi v}}\right) - \frac{(x-m)^2}{2v}\right]$

$\frac{d}{dm} = \sum_{n=1}^{N} \left[0 - \frac{1}{2v} 2(x-m)\cdot(-1)\right] = \frac{1}{N} \cdot \sum_{n=1}^{N} x_m$ → sample mean

Maximum Likelihood Estimation

- The 'best fitting' Gaussian maximises $p(y|\mu,\sigma)$.
- Terminology:
  - $p(y|\mu,\sigma)$, as a function of y is the **probability (density)** of y
  - $p(y|\mu,\sigma)$, a function of μ,σ is the **likelihood** of μ,σ
- Maximising $p(y|\mu,\sigma)$ with respect to μ,σ is **Maximum Likelihood (ML)** estimation of μ,σ

ML estimation of μ,σ

- Intuitively:
  - The ML estimate of μ should be the average value of $y_1, \ldots, y_N$, (the **sample mean**)
  - The ML estimate of σ should be the variance of $y_1, \ldots, y_N$. (the **sample variance**)
- This is true: $p(y|\ \mu, \sigma)$ is maximised by setting:

$$\mu = \frac{1}{N}\sum_{n=1}^{N} y_n, \sigma = \frac{1}{N}\sum_{n=1}^{N} (y_n - \mu)^2$$

Multi-modal distributions

- In practice the distributions of many naturally occurring phenomena do not follow the simple bell-shaped Gaussian curve
- For example, if the data arises from several difference sources, there may be several distinct peaks (e.g. distribution of heights of adults)
- These peaks are the **modes** of the distribution and the distribution is called **multi-modal**

Gaussian Mixture PDFs

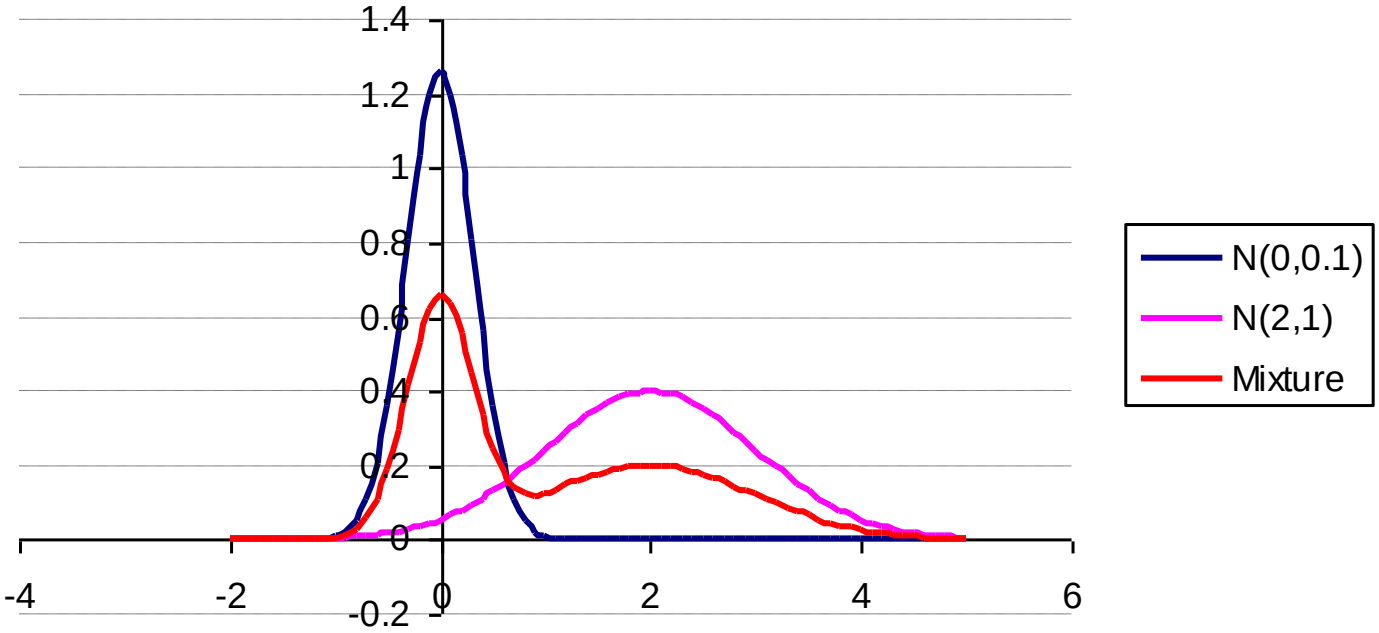
- Gaussian Mixture PDFs, or Gaussian Mixture Models (GMMs) used to model multi-modal and other non-Gaussian distributions.
- A GMM is just a weighted average of several Gaussian PDFs, called the **component** PDFs
- For example, if p_1 and p_2 are Gaussian PDFs, then

$$p(y) = w_1 p_1(y) + w_2 p_2(y)$$

defines a 2 component Gaussian mixture PDF

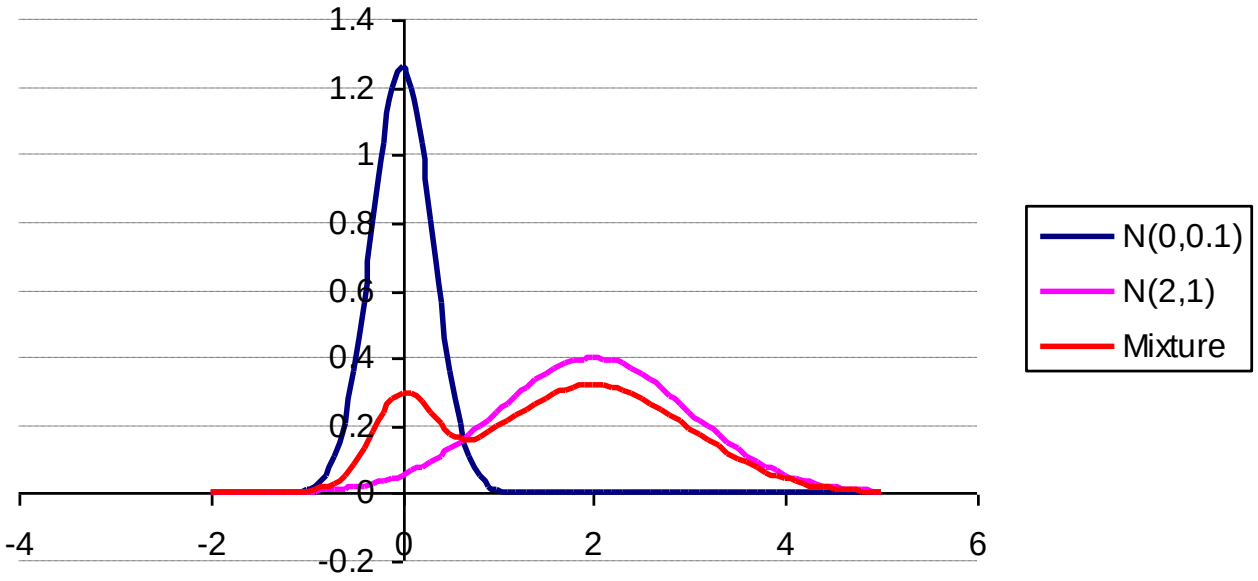
Gaussian Mixture - Example

- 2 component mixture model
  - Component 1: $\mu=0$, $\sigma=0.1$
  - Component 2: $\mu=2$, $\sigma=1$
  - $w_1 = w_2 = 0.5$



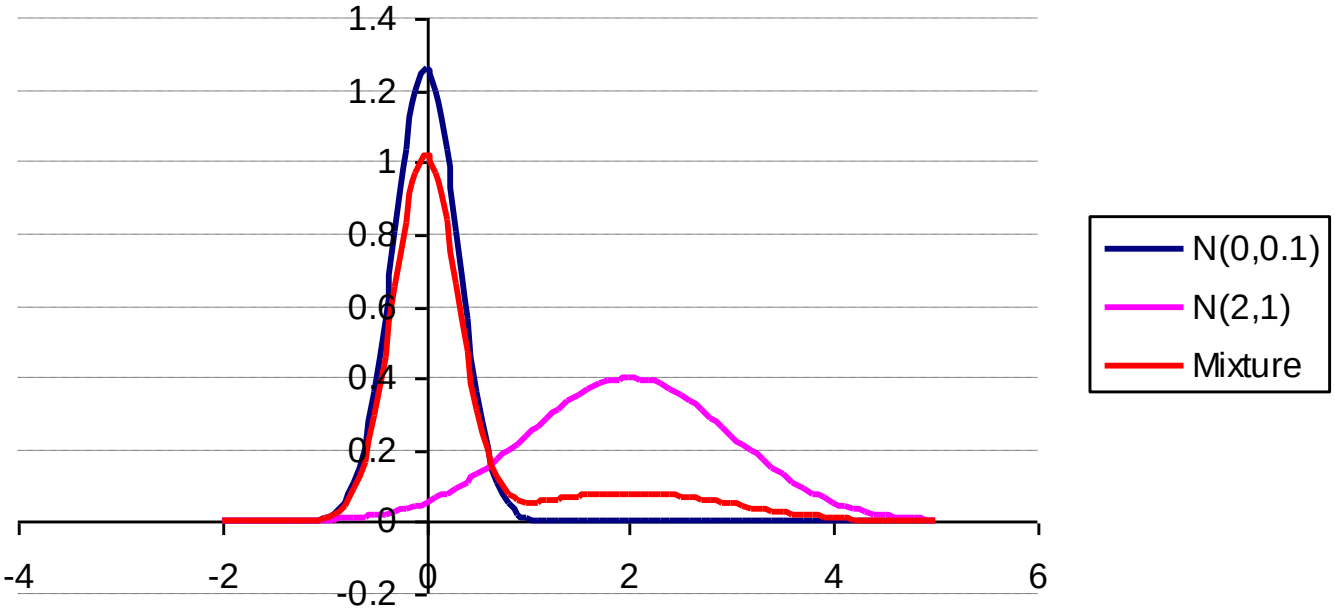
Example 2

- 2 component mixture model
  - Component 1: $\mu=0$, $\sigma=0.1$
  - Component 2: $\mu=2$, $\sigma=1$
  - $w_1 = 0.2$ $w_2=0.8$



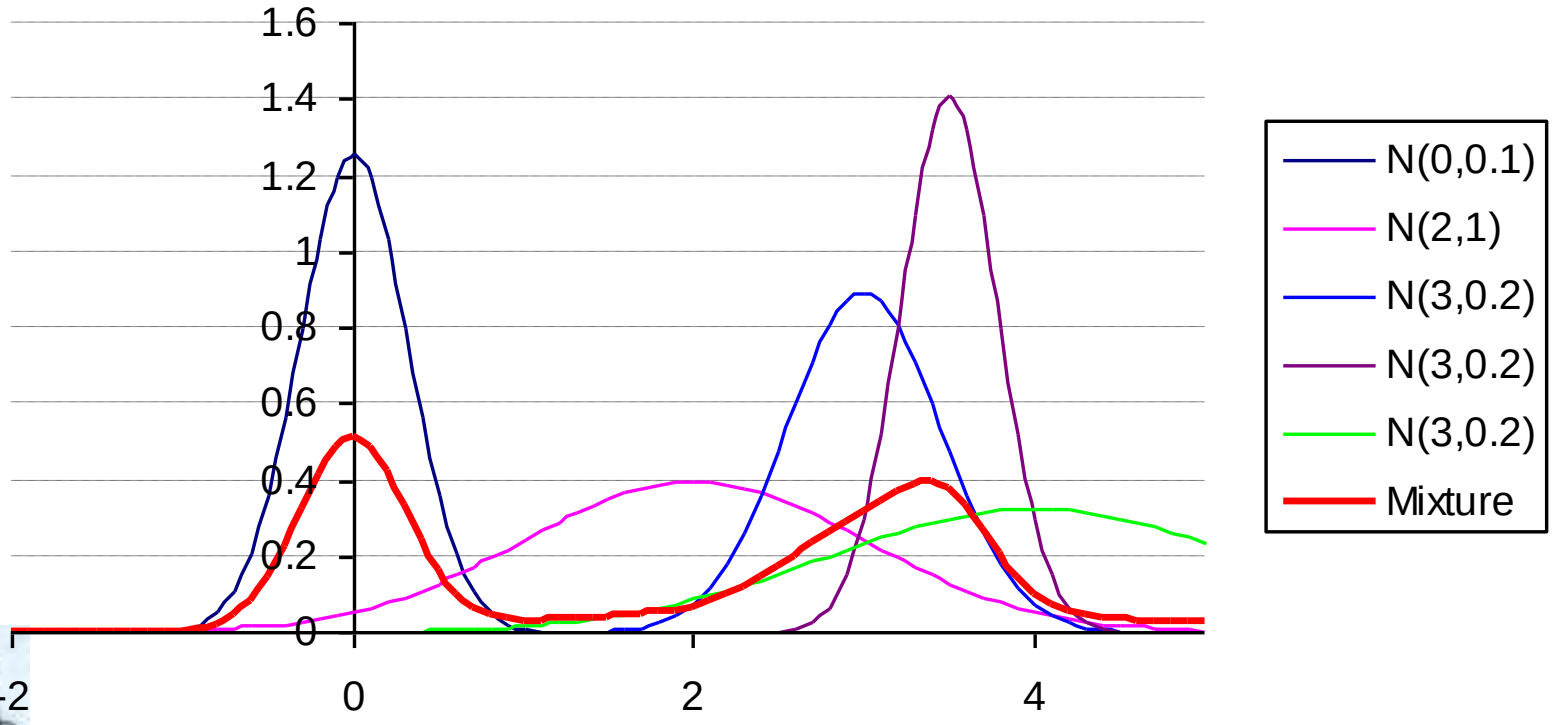
Example 3

- 2 component mixture model
  - Component 1: $\mu=0$, $\sigma=0.1$
  - Component 2: $\mu=2$, $\sigma=1$
  - $w_1 = 0.2$ $w_2=0.8$



Example 4

- 5 component Gaussian mixture PDF



Gaussian Mixture Model

- In general, an M component Gaussian mixture PDF is defined by:

$$p(y) = \sum_{m=1}^{M} w_m p_m(y)$$

where each p_m is a Gaussian PDF and

$$0 \leq w_m \leq 1, \sum_{m=1}^{M} w_m = 1$$

Relationship with Clustering

- Both model data using a set of centroids / means
- In clustering there is no parameter that specifies the ‘spread’ of a cluster. In a GMM component this is done by the covariance matrix
- In clustering we assign a sample to the closest centroid. In a GMM a sample is assigned to all components with varying probability.

Estimating the parameters of a Gaussian mixture model

- A Gaussian Mixture Model with M components has:
  - M means: $\mu_1, \ldots, \mu_M$
  - M variances $\sigma_1, \ldots, \sigma_M$
  - M mixture weights $w_1, \ldots, w_M$.
- Given $y = y_1, \ldots, y_T$, how do we estimate these parameters?
- i.e. how do we find a maximum likelihood estimate of $\mu_1, \ldots, \mu_M$, $\sigma_1, \ldots, \sigma_M$, $w_1, \ldots, w_M$?

Parameter Estimation

- If we knew which component each sample y_t came from, then parameter estimation would be easy:
  - Set μ_m to be average of samples that belong to m^{th} component
  - Set σ_m to be variance of samples that belong to m^{th} component
  - Set w_m to be proportion of samples that belong to m^{th} component
- But we **don't** know which component each sample belongs to.

The E-M Algorithm

- **Step 1**: Choose number of GMM components, M. and initial GMM parameters ($m_1,...m_M$, $\sigma_1,...,\sigma_M$ and $w_1,...,w_M$)
- **Step 2**: For each sample x_t and each GMM component m calculate $P(m|y_t)$ using Bayes theorem and current parameters (see next slide)
- **Step 3**: Define new estimate of m_i as:

$$\overline{m}_i = \frac{1}{P_i}\sum_{t=1}^{T} P(m \mid y_t) y_t \text{ where } P_i = \sum_{t=1}^{T} P(m \mid y_t)$$

E-M continued

- From Bayes' theorem:

Calculate from m^{th} Gaussian component

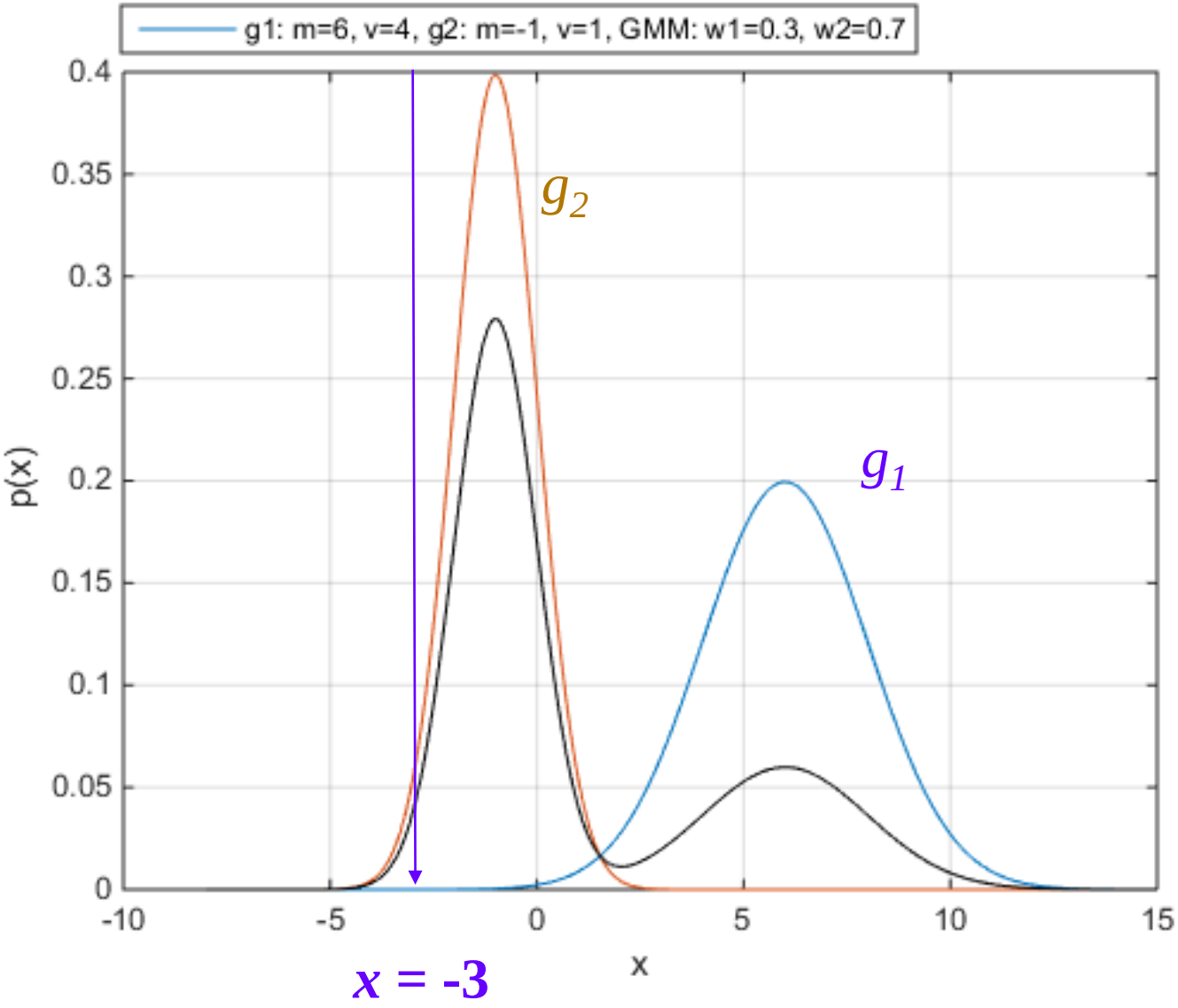
m^{th} weight

$$P(m \mid y_t) = \frac{p(y_t \mid m)P(m)}{p(y)} = \frac{p_m(y_t)w_m}{\sum_{k=1}^{M} p_k(y_t)w_k}$$

Sum over all components

UNIVERSITY OF BIRMINGHAM

Example



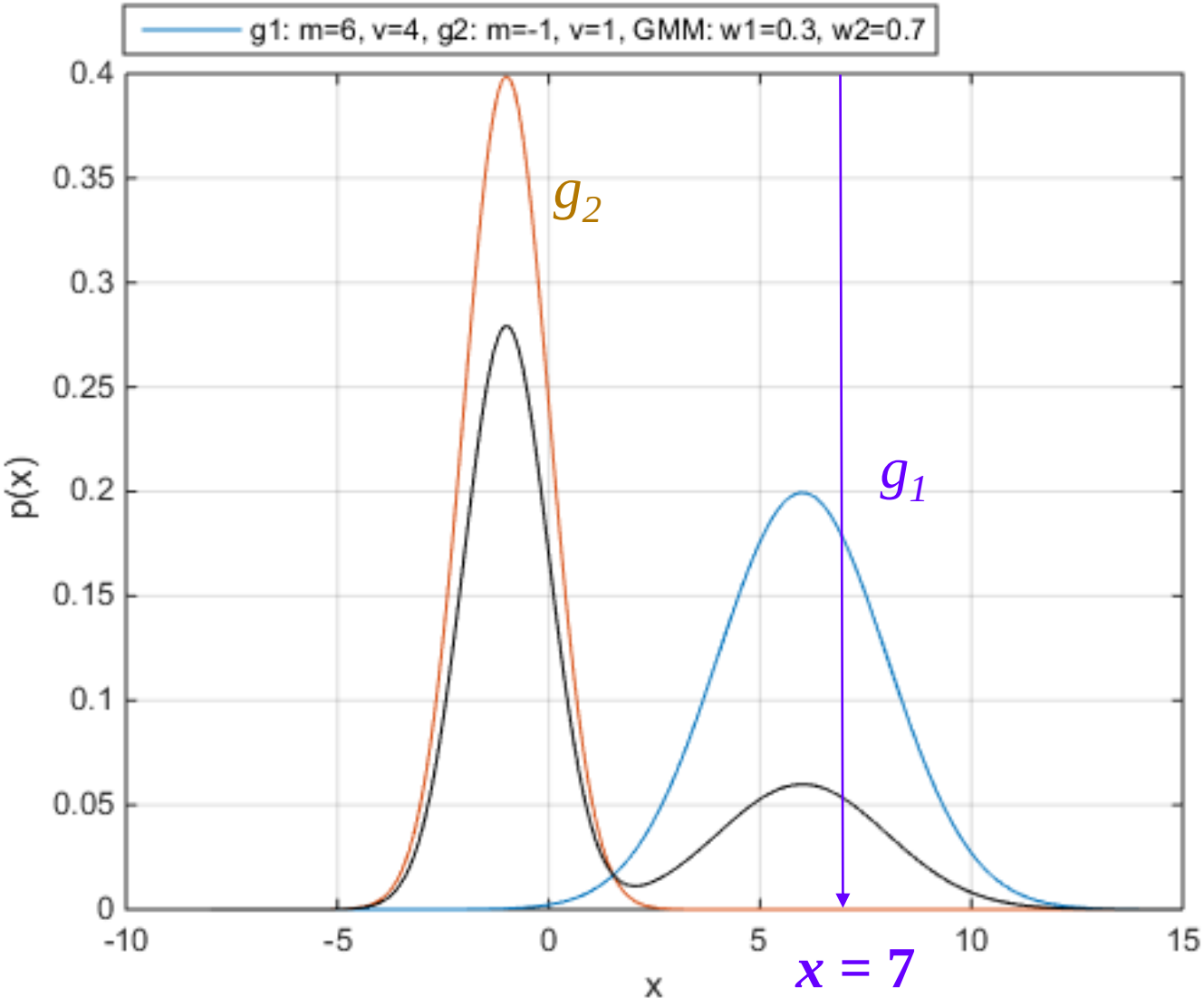
$$g_1(x) \approx 0$$

$$g_2(x) = 0.054$$

$$P(1 \mid x) \approx \frac{0 \times 0.3}{0 \times 0.3 + 0.054 \times 0.7} = 0$$

$$P(2 \mid x) \approx 1$$

Example



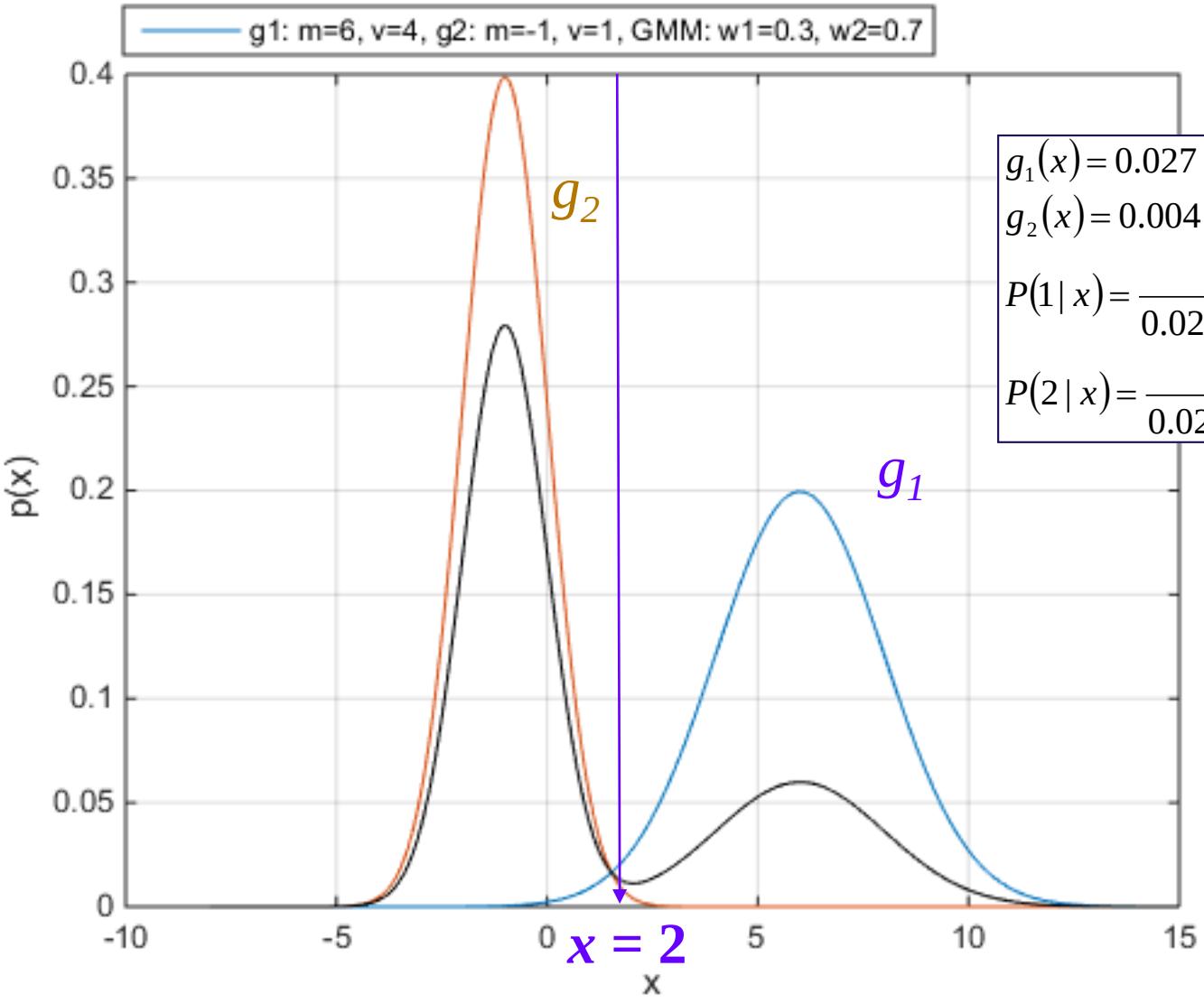
$$g_1(x) = 0.176$$

$$g_2(x) \approx 0$$

$$P(1 \mid x) \approx \frac{0.176 \times 0.3}{0.176 \times 0.3 + 0 \times 0.7} = 1$$

$$P(2 \mid x) \approx 0$$

Example



$$g_1(x) = 0.027$$

$$g_2(x) = 0.004$$

$$P(1 \mid x) = \frac{0.027 \times 0.3}{0.027 \times 0.3 + 0.004 \times 0.7} = 0.723$$

$$P(2 \mid x) = \frac{0.004 \times 0.7}{0.027 \times 0.3 + 0.004 \times 0.7} = 0.277$$

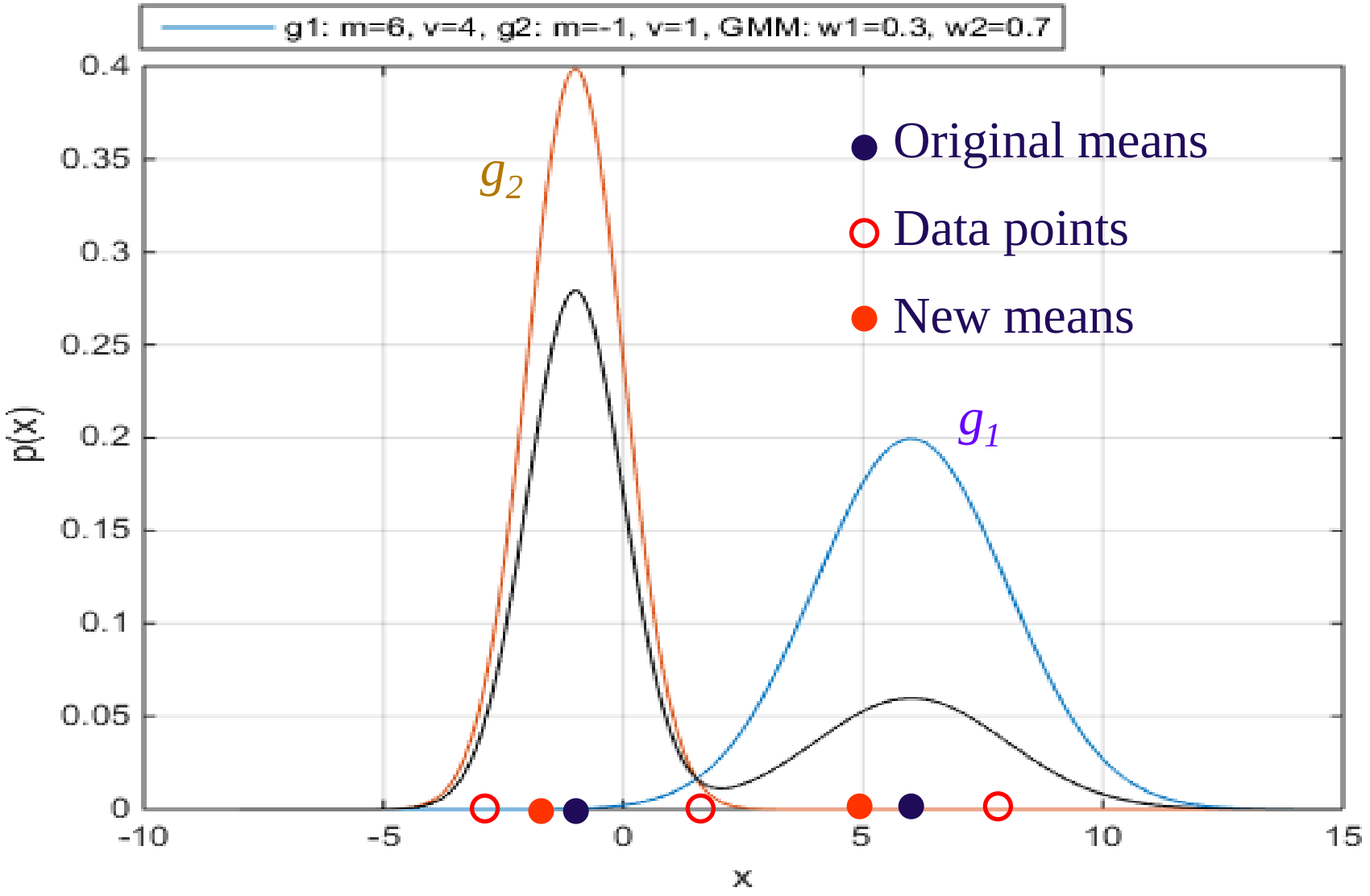
Example (continued)

- So, given these initial estimates of g_1 and g_2, and data points $X = \{x_1, x_2, x_3\} = \{-3, 2, 7\}$, the new values of m_1 and m_2 are:

$$\overline{m}_1 = \frac{0 \times x_1 + 0.723 \times x_2 + 1 \times x_3}{0 + 0.723 + 1} = \frac{0 \times (-3) + 0.723 \times 2 + 1 \times 7}{1.723} = 4.9$$

$$\overline{m}_2 = \frac{1 \times x_1 + 0.277 \times x_2 + 0 \times x_3}{1 + 0.277 + 0} = \frac{1 \times (-3) + 0.277 \times 2 + 0 \times 7}{1.277} = -1.92$$

Example


g1: m=6, v=4, g2: m=-1, v=1, GMM: w1=0.3, w2=0.7
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
p(x)
-10
-5
0
5
10
15
x
g_2
g_1
Original means
Data points
New means

E-M and *k*-means clustering

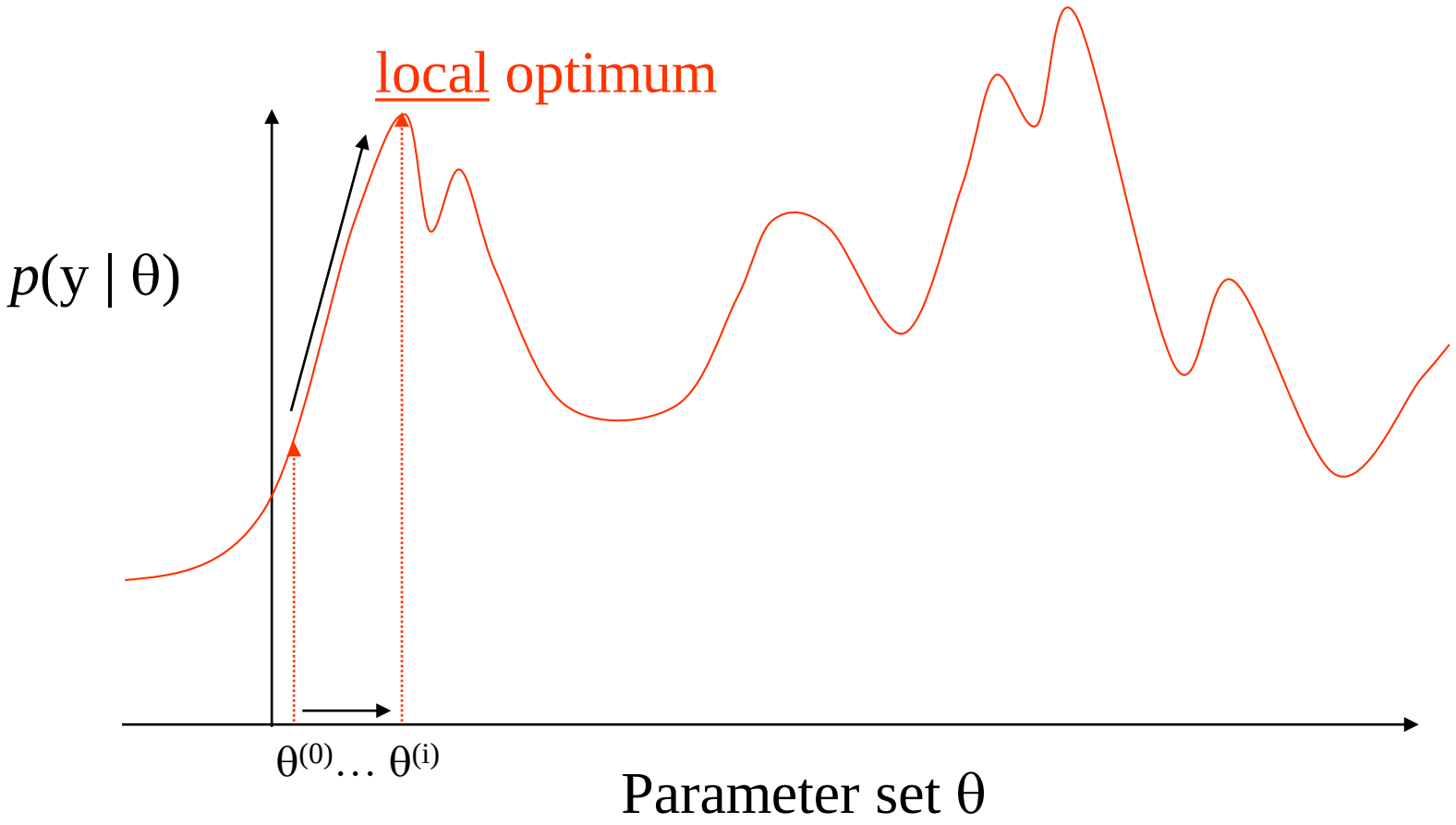
- Compare:
  - Estimating GMM component means in E-M, and
  - Estimating centroids in *k*-means clustering
- Notation
  - GMM component means $m_1,\ldots, m_N$
  - Cluster centroids $c_1,\ldots, c_N$
- Given a sample y
  - E-M: Calculate $P(n \mid y)$ for each GMM component n
  - *K*-means: Calculate $d(c_n,y)$ for each centroid c_n
- Reestimation
  - E-M: For each n, allocate $P(n|y_n)y_n$ to reestimation of m_n
  - *K*-means: Allocate all of y_n to the closest centroid ($\min\{d(c_n,y)\}$)

E-M and *k*-means clustering

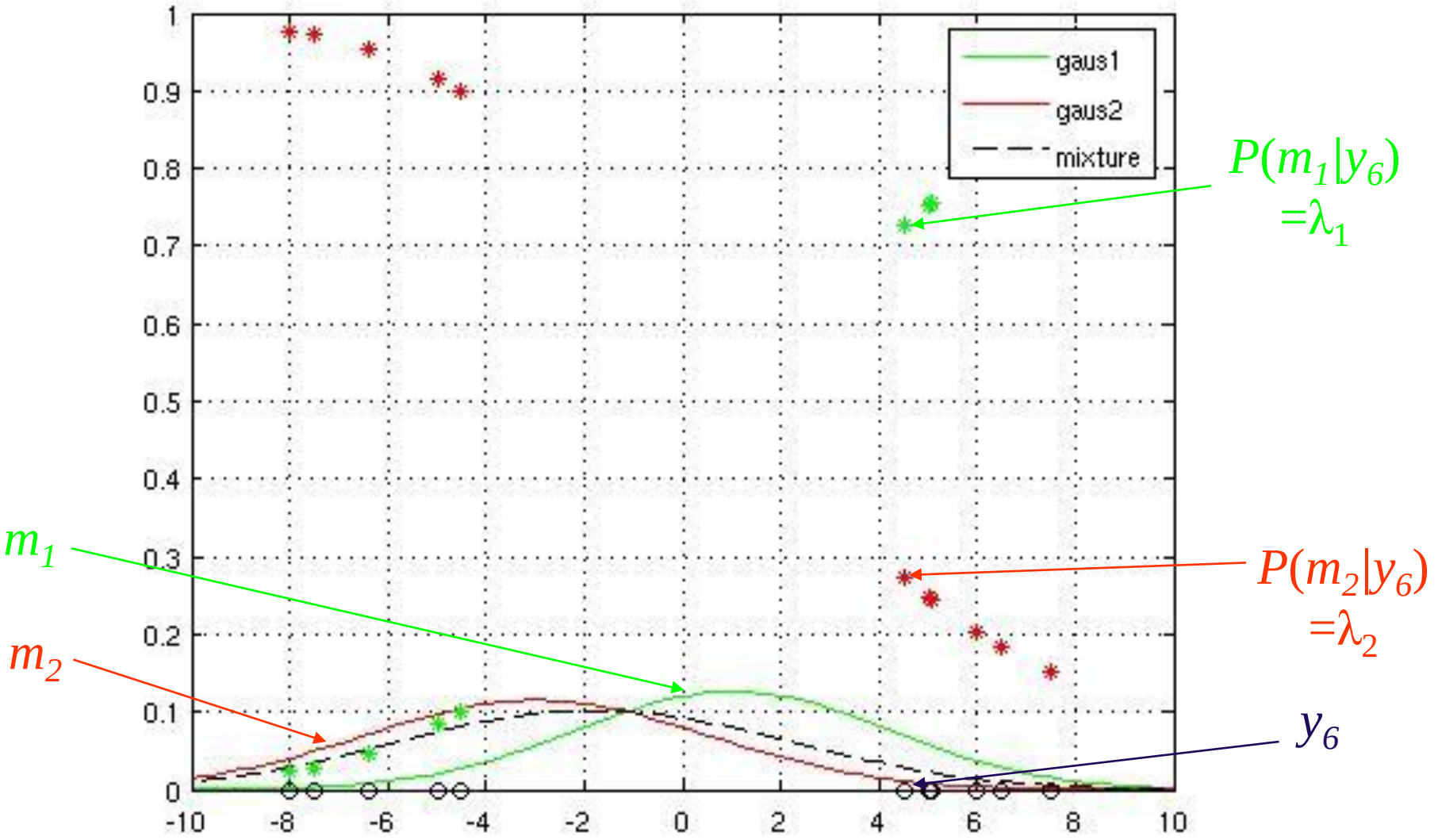
- In some implementations of E-M, y is used **only** to reestimate the mean m_n for the most probable GMM component n (i.e. $\max\{P(n|y)\}$
- If the GMM component variances are all equal, and all of the component weights w_n are equal, then the following are equivalent:
  - $n = \text{argmin}\{d(y,m_n)\}$ (m_n is closest centroid to x)
  - $n = \text{argmax}\{P(n|y)\}$ (i.e. n is the most probable GMM component)

same variance $\cong$ distance metric

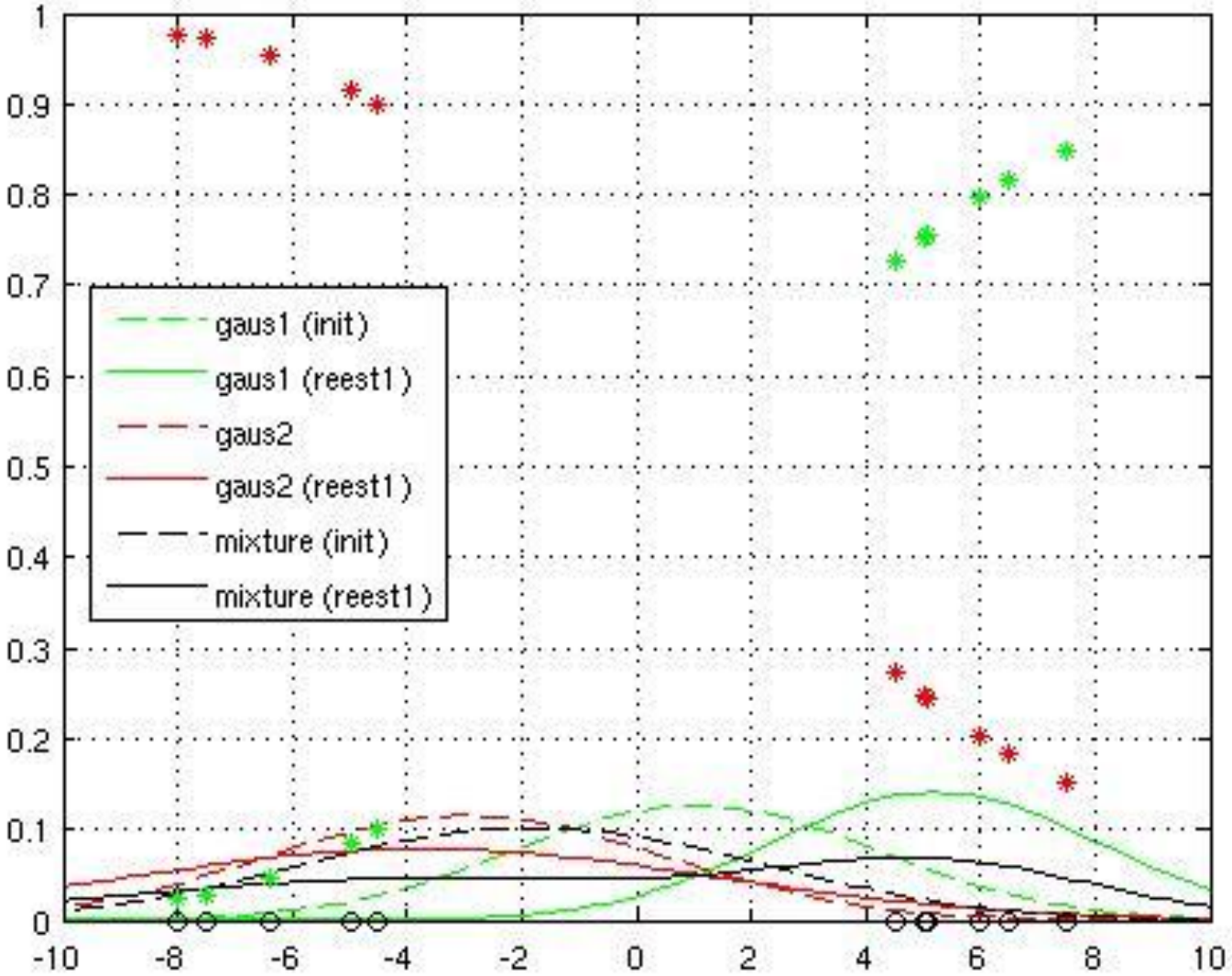
The E-M algorithm



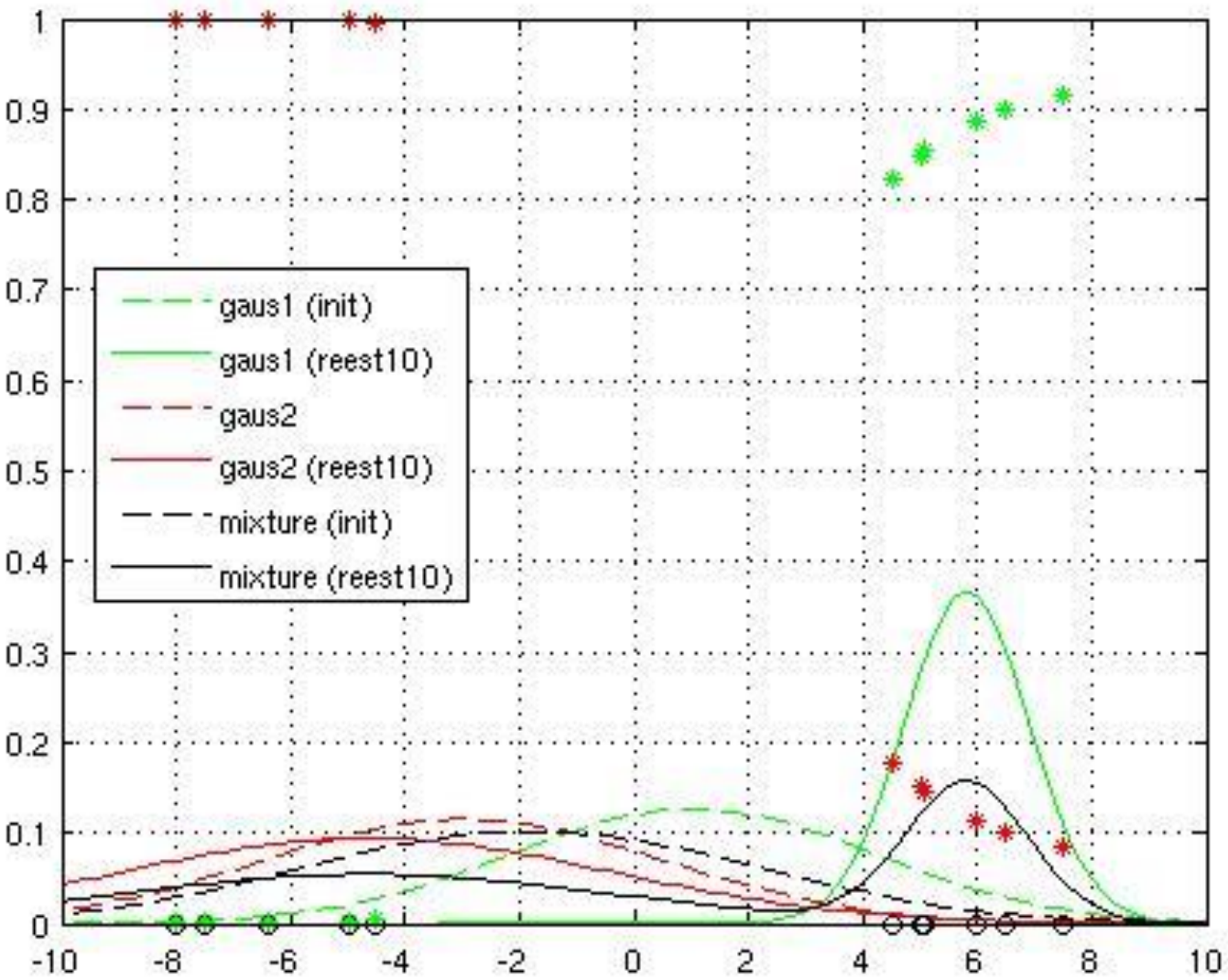
Example – initial model



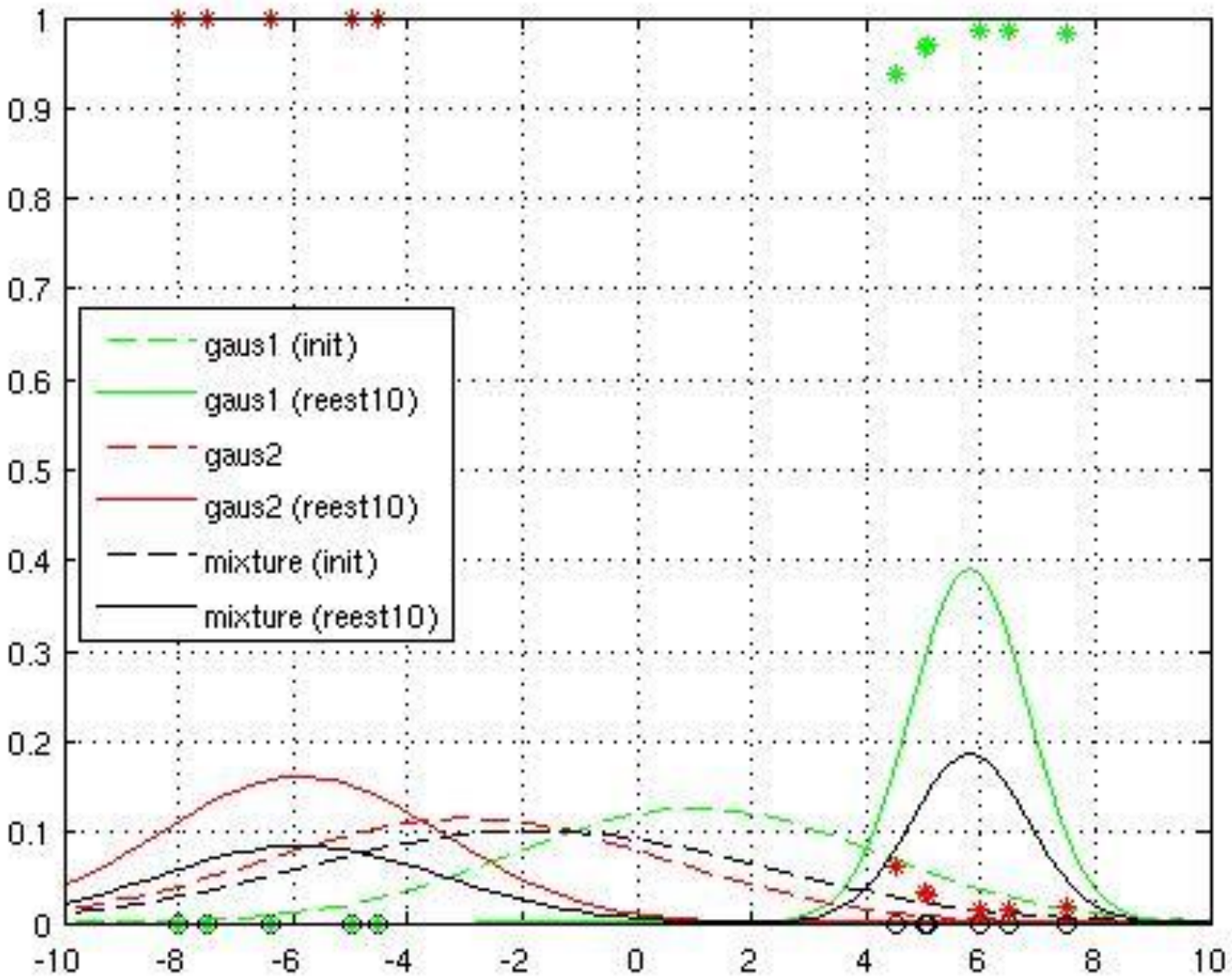
Example – after 1st iteration of E-M



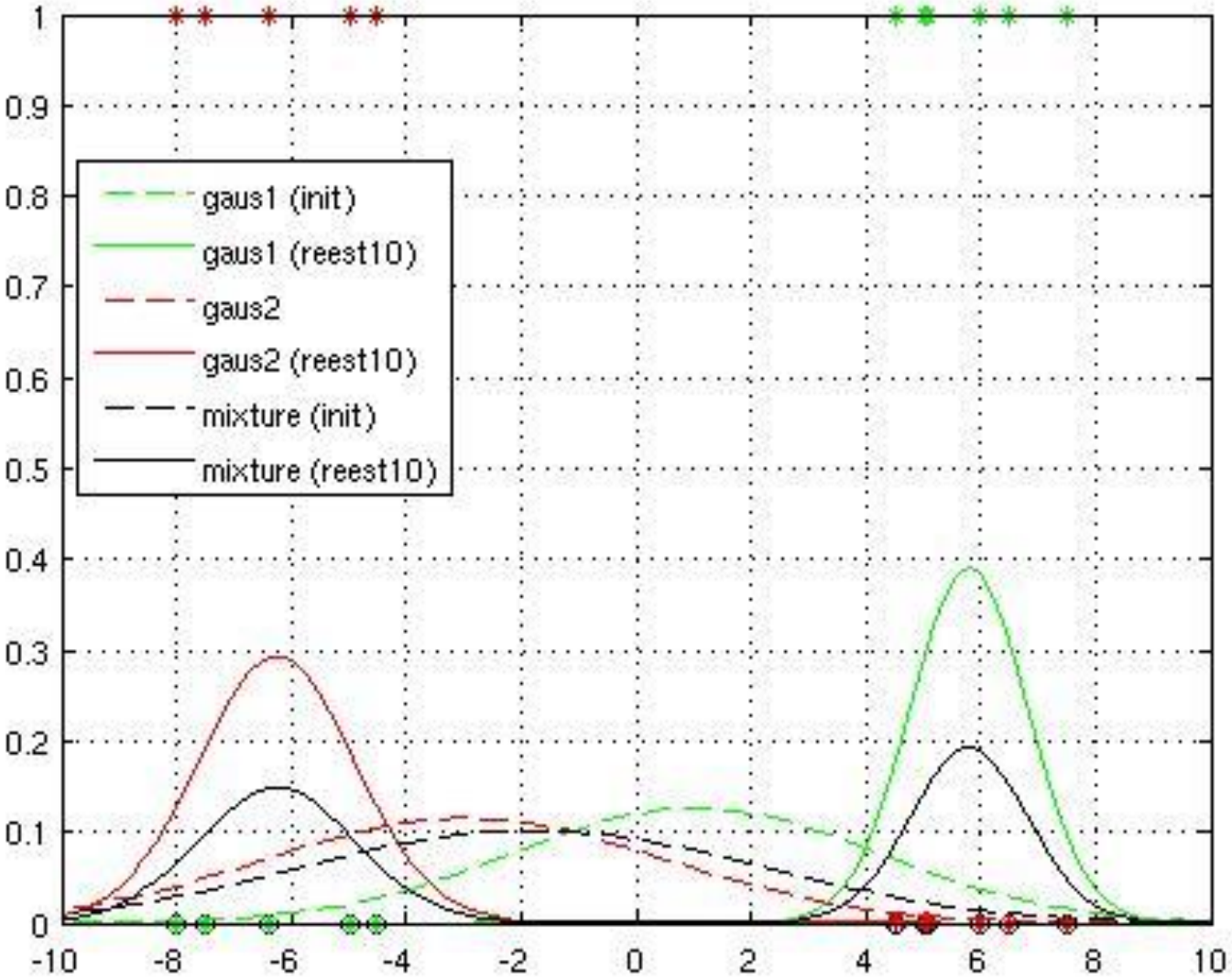
Example – after 2nd iteration of E-M



Example – after 4th iteration of E-M



Example – after 10th iteration of E-M



Summary so far ...

- Basic statistical modelling
- Probability distributions
- Probability density function
- Gaussian PDFs
- Multivariate Gaussian PDFs
- Gaussian mixture PDFs (GMMs)
- Maximum likelihood (ML) parameter estimation – the E-M algorithm
- Comparison of E-M for GMMs with k-means clustering

"Supervectors"

- Suppose that an item of data consists of a **variable length sequence** of vectors $Y = y_1, y_2 \ldots, y_t, \ldots, y_T$
- For example, Y could correspond to:
  - A recording of speech
  - The measurements from a pen during a signature
- Because the length T varies Y cannot be treated directly as a vector
- So methods from linear algebra (PCA) not applicable
- Analogy with text processing

Vectorization of continuous data

- Choose M the number of GMM components
- Apply the E-M algorithm together with the data set Y to create a M-component GMM $\boldsymbol{M}$
- The supervector representation $sup(Y)$ of Y is the M (number of GMM components) $\times$ N (matrix dimension) vector obtained by stacking the mean vectors of the components of $\boldsymbol{M}$

Example

- N=3, M=4 and means of components of $\boldsymbol{M}$ are:

$$m_1 = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}, m_2 = \begin{bmatrix} -3 \\ 4 \\ 1 \end{bmatrix}, m_3 = \begin{bmatrix} 10 \\ 12 \\ -12 \end{bmatrix}, m_4 = \begin{bmatrix} -1 \\ -3 \\ -2 \end{bmatrix}$$

- Then

$$sup(Y) = \left[\begin{array}{c} 1 \\ 2 \\ 1 \\ \hdashline -3 \\ 4 \\ 1 \\ \hdashline 10 \\ 12 \\ -12 \\ \hdashline -1 \\ -3 \\ -2 \end{array}\right]$$

Problems

- Different initial GMMs lead to different optimized GMMs and hence different supervectors
- Even if two GMMs are functionally identical, the order of components may be different
- This makes it difficult to **compare** supervectors for two different sequences
- The solution is to use a **Universal Background Model (UBM)**

Universal Background Model (UBM)

- Example: **speaker verification**
- To build a speaker verification system, start with recordings of many different speakers
- Build a **single** GMM, called the **Universal Background Model (UBM)** from **all** of the data from **all** of the speakers
- Think of the UBM as modelling the **inventory of speech sounds** averaged across a population od speakers

Universal Background Model (UBM)

- Given a new sequence Y_s from a speaker s use the E-M algorithm + UBM to create a GMM M_s. Use M_s to create $sup(Ys)$
- Think of M_s as the **speaker-specific** inventory of sounds for the speaker s
- Then $sup(Y_s)$ is a **vector representation** of the inventory of speech sounds for speaker s

Universal Background Model (UBM)

- Because they come from the **same** initial UBM, $sup(Y_s)$ and $sup(Y_r)$ are **comparable** for speakers s and r
- For example, $\cos(\theta)$, where θ is the angle between $sup(Y_s)$ and $sup(Y_r)$ is a measure of the similarity of speakers s and r – used in speaker verification systems

Supervectors

- A problem with supervectors is their **dimension**
- 512 component GMMs with 20 dimensional vectors are standard, resulting in 10,240 dimensional supervectors!
- **Dimension reduction** is an issue
- But, problems estimating a 10,240 $\times$ 10,240 covariance matrix, so PCA and LDA are **unreliable**
- State-of-the-art speaker verification systems use robust dimension reduction based on **i-vectors**

Summary

- Properties of Gaussian PDFs
- Gaussian Mixture Models (GMMs)
- Learning GMM parameters from data – the E-M algorithm
- Vector representation of continuous data
- GMM-Supervectors