Pattern Recognition

Instructor:
Dr. Sadaf Yasmin
Associate professor
COMSATS University Islamabad Attock Campus

Mixture Models

A finite mixture model (FMM) assumes that the data you observe do not come from just one distribution, but from a combination (mixture) of several distributions.

1. Each component distribution corresponds to a "group" or "sub-population".
2. You don’t know which data point belongs to which group (hidden membership).
3. Instead, you model the overall data as a weighted sum of component distributions.

Mixture Models

They are particularly suitable for modelling distributions where the measurements arise from separate groups, but individual membership is unknown.

A finite mixture model is a distribution with probability density function of the form

$$p(\boldsymbol{x}) = \sum_{j=1}^{g} \pi_j p(\boldsymbol{x}; \boldsymbol{\theta}_j)$$

Mixture Models

$$p(\boldsymbol{x}) = \sum_{j=1}^{g} \pi_j p(\boldsymbol{x}; \boldsymbol{\theta}_j)$$

where

- g is the number of mixture components (often referred to as the *model order*);
- $\pi_j \geq 0$ are the mixture component probabilities (also referred to as mixing proportions), which satisfy $\sum_{j=1}^{g} \pi_j = 1$;
- $p(\boldsymbol{x}; \boldsymbol{\theta}_j)$, $j = 1, \ldots, g$, are the *component density* functions (each of which depends on a parameter vector $\boldsymbol{\theta}_j$).

Mixture of Two Gaussian Distributions

Imagine you are measuring the height of people in a population.

But actually, your data comes from two groups:

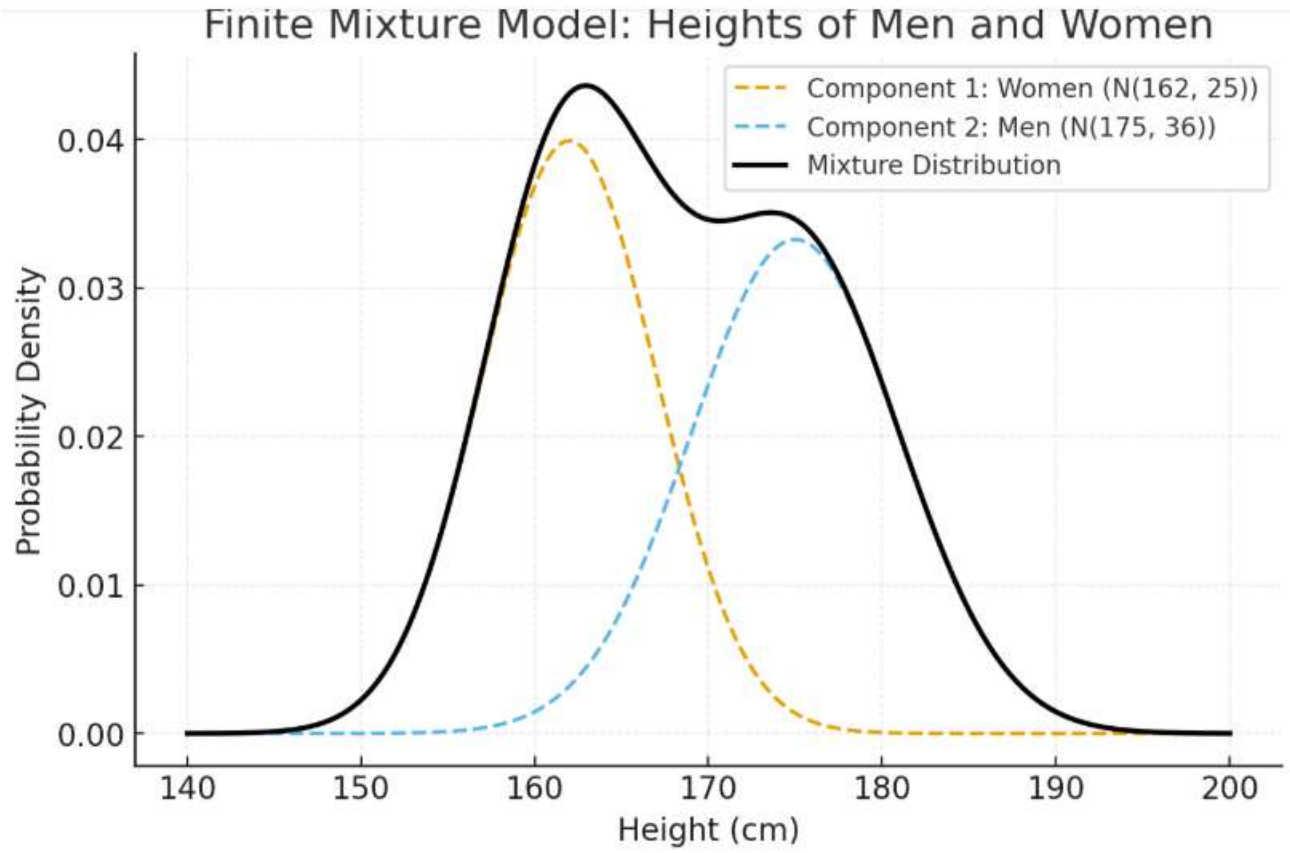
1. Adult women
2. Adult men

- Women's heights might follow a normal distribution with mean $\mu_1 = 162$ cm and variance $\sigma_1^2 = 25$.
- Men's heights might follow a normal distribution with mean $\mu_2 = 175$ cm and variance $\sigma_2^2 = 36$.
- Suppose the population is 50% women and 50% men.

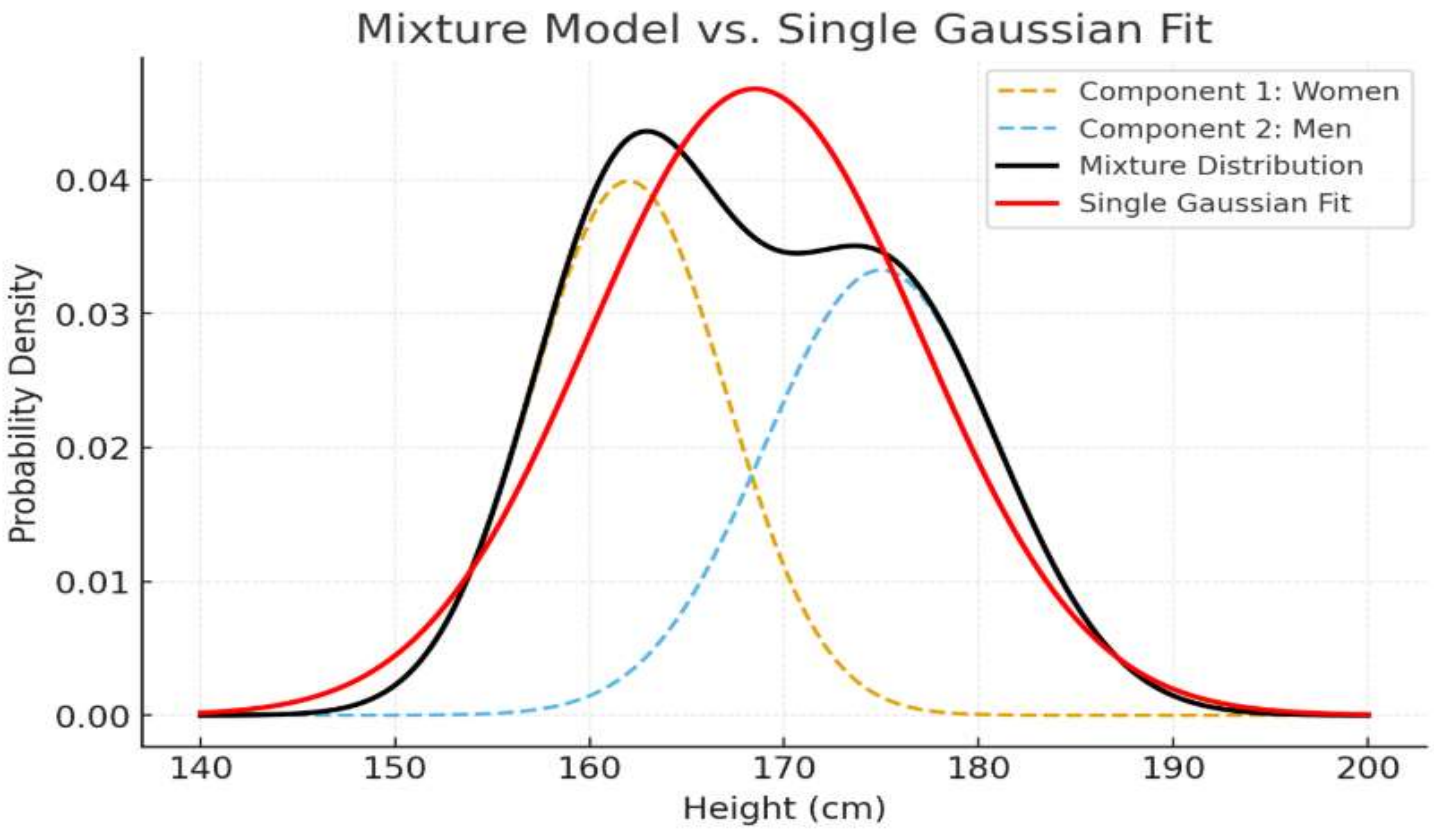
Then the mixture model PDF is:

$$p(x) = 0.5\,\mathcal{N}(x \mid 162, 25) \; + \; 0.5\,\mathcal{N}(x \mid 175, 36)$$

Mixture of Two Gaussian Distributions



Mixture of Two Gaussian Distributions



Mixture of Normal + Poisson

Suppose we're studying a hospital:

1. Some data are continuous: e.g., blood pressure levels of patients (roughly Normal).
2. Some data are counts: e.g., number of times a patient visits the hospital per month (Poisson-like).

$$p(\boldsymbol{x}) = \sum_{j=1}^{g} \pi_j p(\boldsymbol{x}; \boldsymbol{\theta}_j)$$

Mixture of Normal + Poisson

Now, our observed dataset is a blend of these two groups (we don't know which patient belongs to which group).

We can write the mixture model:

$$p(x) = 0.6\,\mathcal{N}(x \mid 120, 15^2) \; + \; 0.4\,\text{Poisson}(x \mid \lambda = 3)$$

First component: models continuous blood pressure around mean 120.

Second component: models discrete hospital visit counts (skewed, integer values).

The mixture density adds both together, giving us a hybrid distribution.

Multivariate Normal Distribution

A multivariate normal distribution, i.e. $p(\boldsymbol{x}; \boldsymbol{\theta}_j)$ is the probability density for the multivariate normal distribution with mean vector $\boldsymbol{\mu}_j$ and covariance matrix $\boldsymbol{\Sigma}_j$, so that $\boldsymbol{\theta}_j = \{\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\}$.

The normal mixture model therefore has probability density function

$$p(\boldsymbol{x}) = \sum_{j=1}^{g} \pi_j N(\boldsymbol{x}; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$$

2.5.2 Mixture models for discrimination

Mixture models can be used in a discrimination problem by modelling each class conditional density as a mixture distribution

$$p(\boldsymbol{x}|\omega_j) = \sum_{r=1}^{g_j} \pi_{j,r} p(\boldsymbol{x}; \boldsymbol{\theta}_{j,r}) \tag{2.21}$$

where g_j is the number of components for class ω_j, $\{\pi_{j,r} \geq 0, r = 1, \ldots, g_j\}$ are the mixture component probabilities for class ω_j (satisfying $\sum_{r=1}^{g_j} \pi_{j,r} = 1$) and $\{\boldsymbol{\theta}_{j,r}, r = 1, \ldots, g_j\}$ are the parameter vectors for the component densities for class ω_j.

The class conditional densities (2.21) can be used within the Bayesian classifier as specified in Equation (2.1). Specifically, the discriminant rule is: assign $\boldsymbol{x}$ to ω_i if $d_i > d_j$, for all $j \neq i$, where

$$\begin{aligned} d_j &= p(\omega_j) p(x|\omega_j) \\ &= p(\omega_j) \left(\sum_{r=1}^{g_j} \pi_{j,r} p(\boldsymbol{x}; \boldsymbol{\theta}_{j,r}) \right) \end{aligned} \tag{2.22}$$

This method is widely applied in generative classifiers where the joint distribution $p(x, j)$ is modeled and then used for classification.

Mixture Models for Discrimination

Imagine a medical diagnosis scenario where a doctor wants to classify patients into two classes: those with a certain disease and those without. The symptoms and test results of patients can be quite varied, and within each class, the data distribution is not simple—it may be complex and multimodal because patients with the disease can have different subtypes or stages, and healthy patients may also show varied profiles.

To model this, each class conditional density is represented as a mixture of Gaussian distributions:

- For the disease class, the symptoms' distribution is modeled as a mixture of multiple Gaussian components, each representing a subtype of the disease.
- For the healthy class, the symptoms are modeled as another mixture of Gaussian components covering different healthy profiles.

Mixture Models for Discrimination

When a new patient's symptoms are observed, the model calculates the likelihood of these symptoms under each class's mixture model. The classification is made by comparing these likelihoods combined with prior probabilities of each class.

Gene-SGAN identifies two distinct Alzheimer's disease subtypes that have different neuroanatomical patterns in brain structures and are associated with different sets of genetic variants (genetic drivers). These subtypes reflect potentially distinct underlying neuropathologic mechanisms. One subtype shows brain changes and genetic markers suggesting a particular pathway of neurodegeneration, while the other subtype exhibits a different spatial brain pattern and distinct genetic susceptibility factors

Parameter estimation for normal mixture models

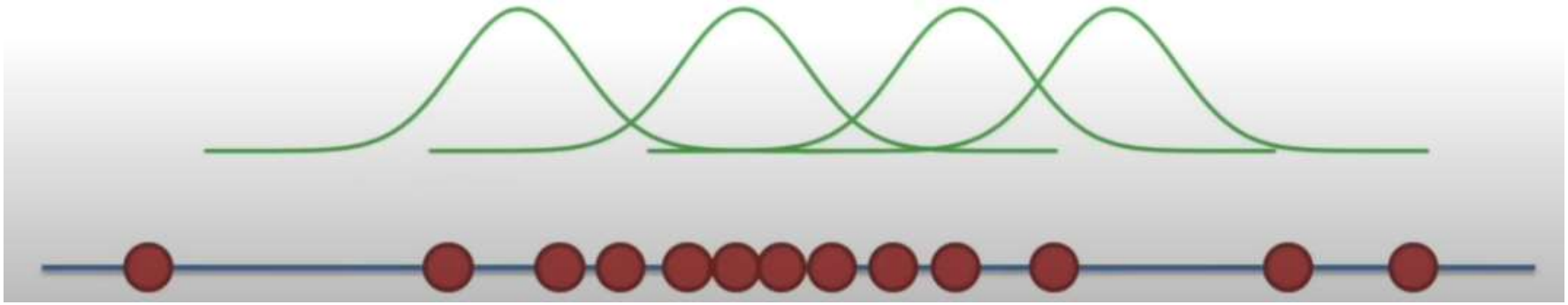
For a normal mixture model with pre-specified model order g, the parameters to be estimated are the mixture component probabilities $\{\pi_1, \ldots, \pi_g\}$ and the mixture component parameters $\boldsymbol{\theta}_j = \{\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\}$, $j = 1, \ldots, g$. The most prevalent technique for optimising the parameters given a set of n independent observations $\{\boldsymbol{x}\} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ (the training data) is an iterative procedure known as the Expectation Maximisation (EM) algorithm. The EM algorithm seeks to maximise the likelihood function

$$L(\pi_1, \ldots, \pi_g, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_g) = \prod_{i=1}^{n} \left\{ \sum_{j=1}^{g} \pi_j N(\boldsymbol{x}_i; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) \right\}$$

Maximum Likelihood

Once we settle on the shape, we have to figure out where to center the thing...

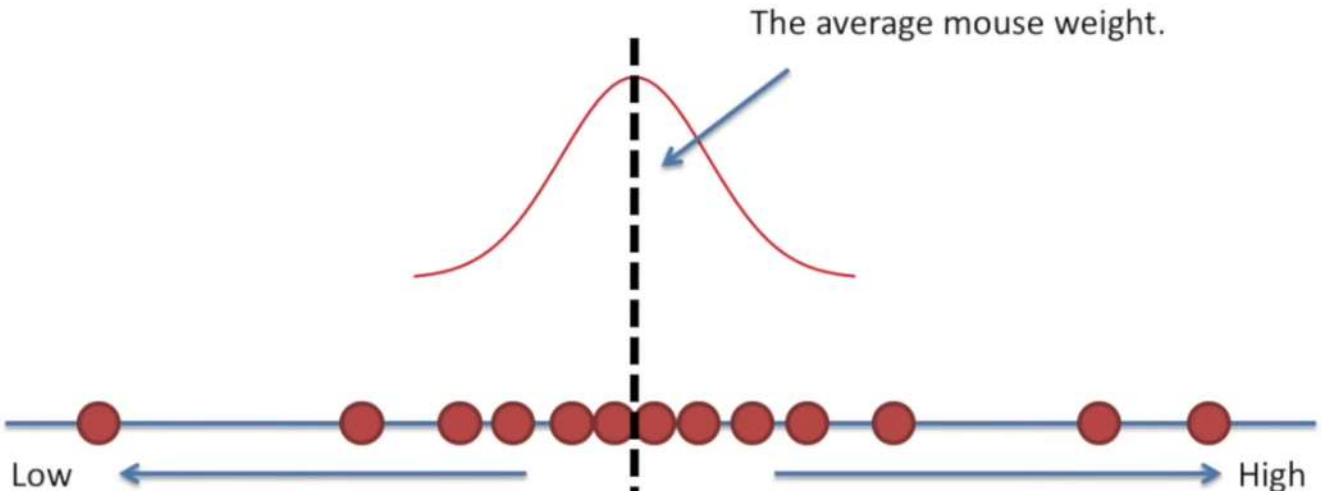
Is one location "better" than another?



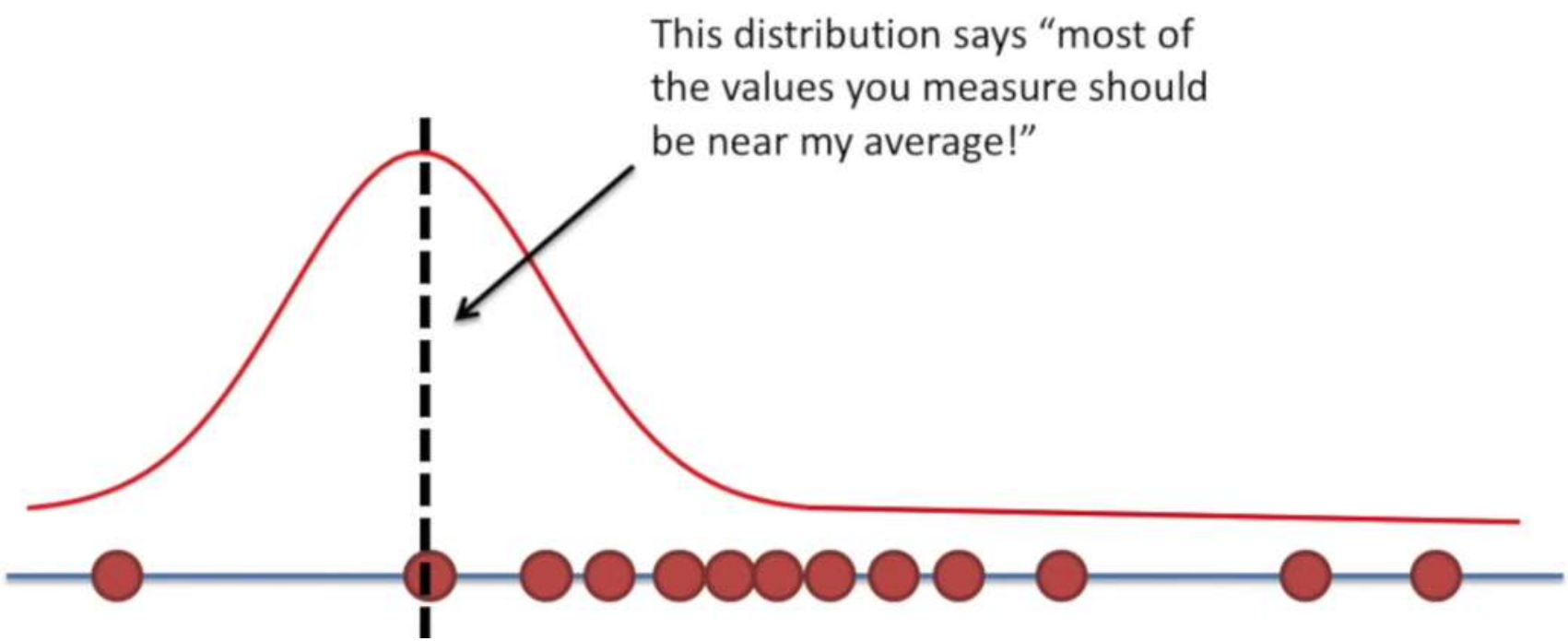
Maximum Likelihood

"Normally distributed" means a number of things:

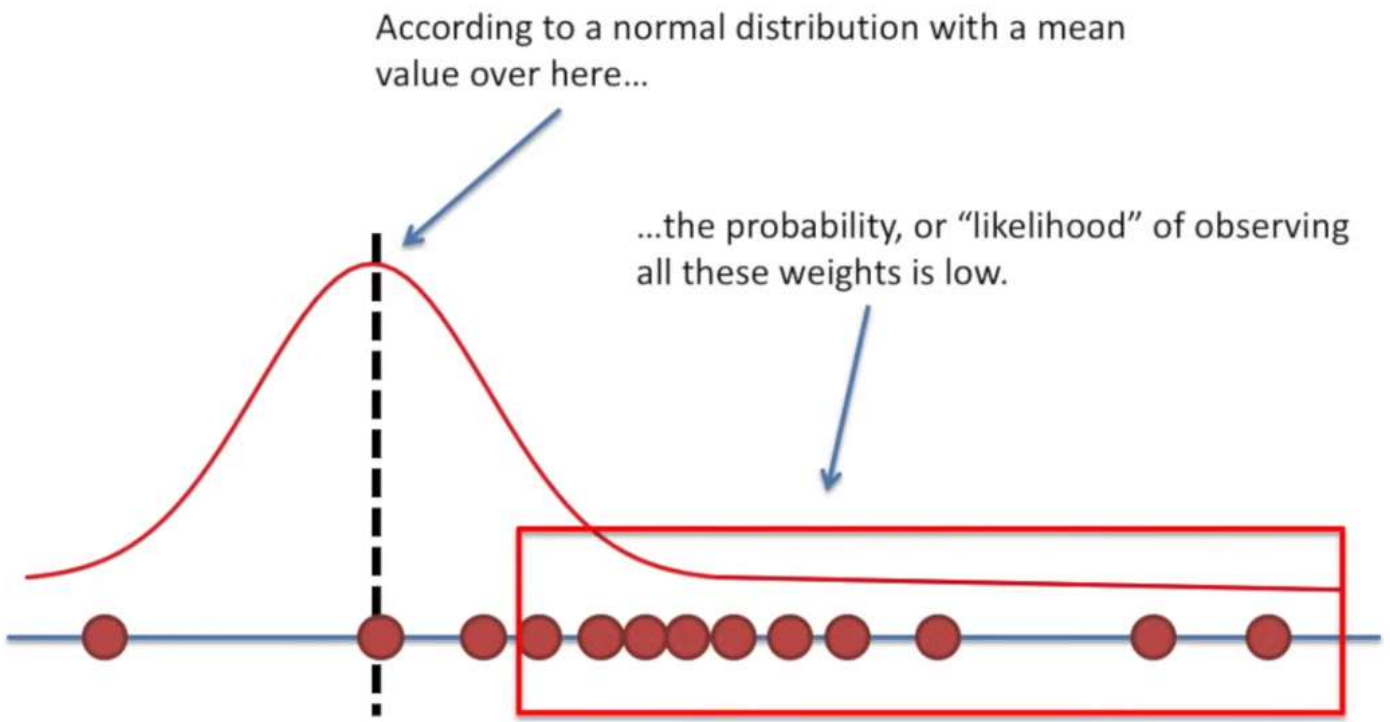
1) We expect most of the measurements (mouse weights) to be close to the mean (average).



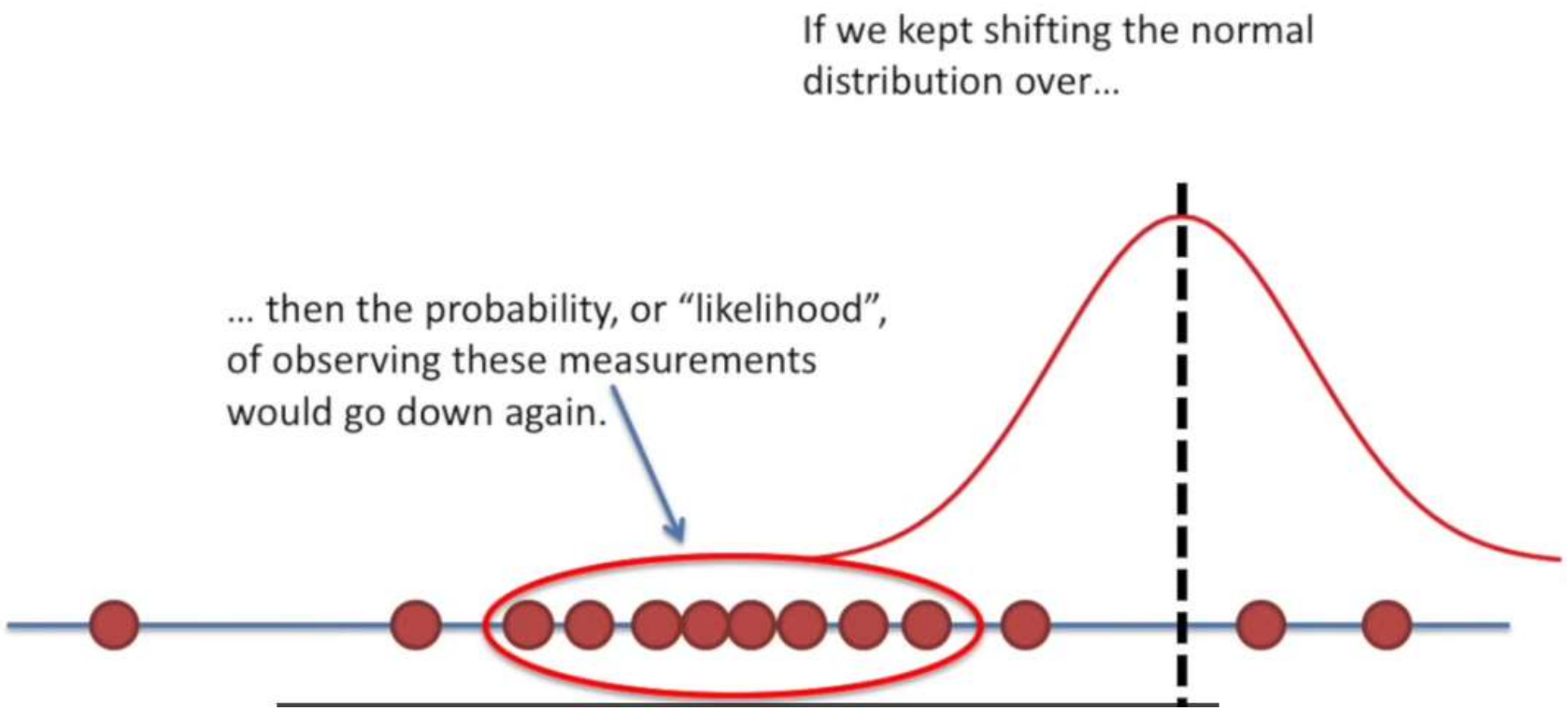
Maximum Likelihood



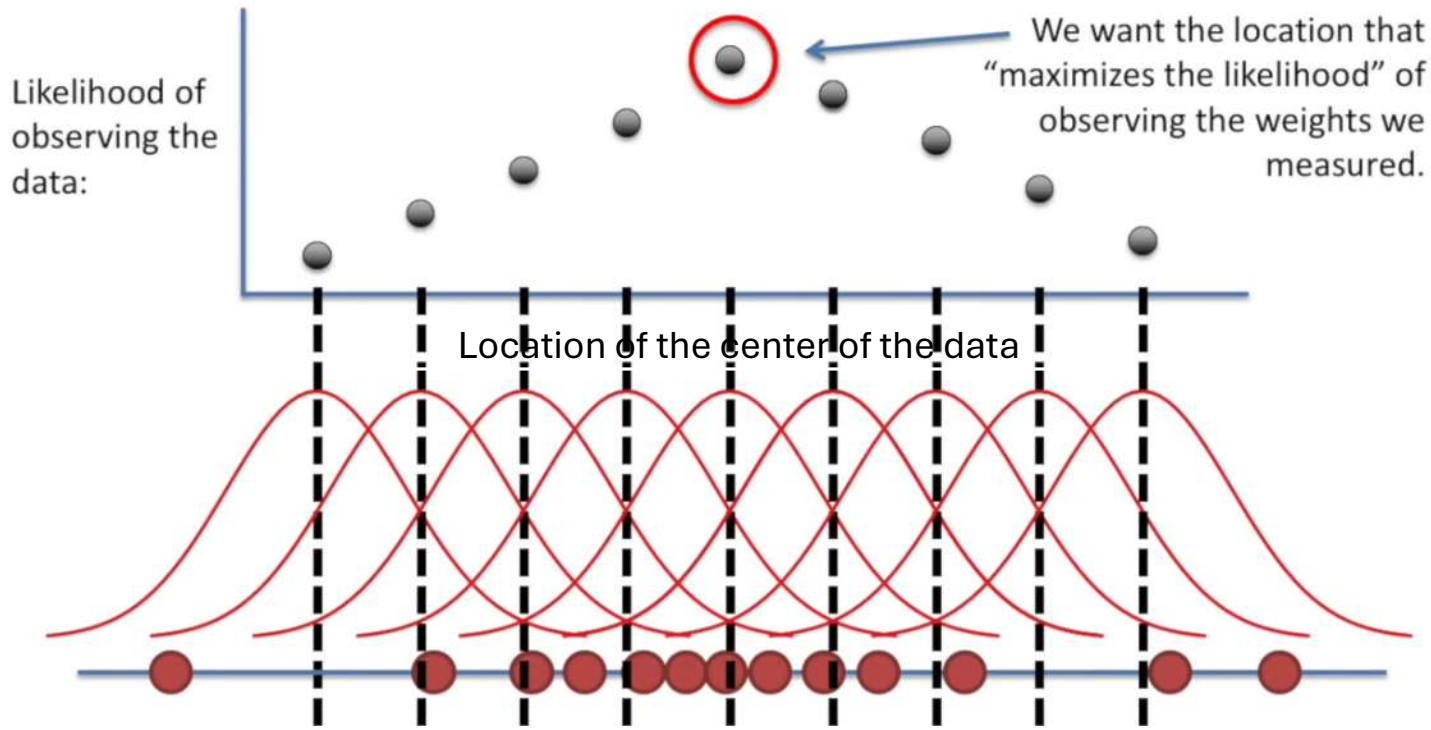
Maximum Likelihood



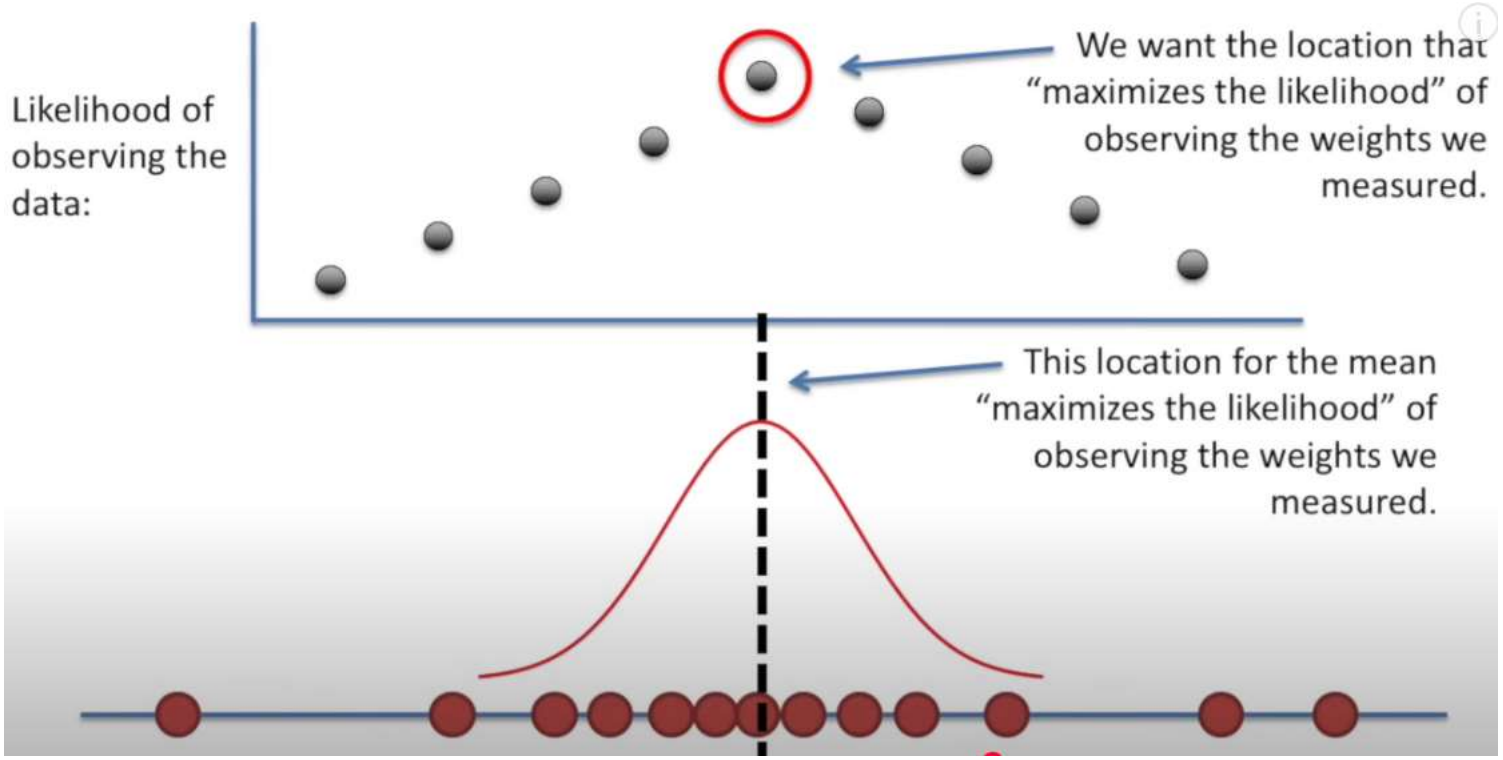
Maximum Likelihood



Maximum Likelihood



Maximum Likelihood



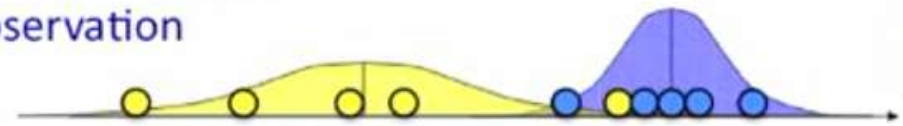
It's the mean of the distribution not the mean of the data

Mixture models

- Observations $x_1 \ldots x_n$
  - K=2 Gaussians with unknown μ, σ^2
  - estimation trivial if we know the source of each observation

$$\mu_b = \frac{x_1 + x_2 + \ldots + x_{n_b}}{n_b}$$

$$\sigma_b^2 = \frac{(x_1 - \mu_1)^2 + \ldots + (x_n - \mu_n)^2}{n_b}$$

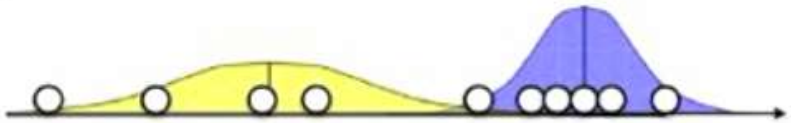


- If we knew parameters of the Gaussians (μ, σ^2)
  - can guess whether point is more likely to be a or b

When we don't know the source

$$P(b \mid x_i) = \frac{P(x_i \mid b)P(b)}{P(x_i \mid b)P(b) + P(x_i \mid a)P(a)}$$

$$P(x_i \mid b) = \frac{1}{\sqrt{2\pi\sigma_b^2}} \exp\left(-\frac{(x_i - \mu_b)^2}{2\sigma_b^2}\right)$$



  - need (μ_a, σ_a^2) and (μ_b, σ_b^2) to guess source of points
  - need to know source to estimate (μ_a, σ_a^2) and (μ_b, σ_b^2)

Expectation Maximization (EM) Algorithms

The Problem

We have observed data points $\{x_1, x_2, \ldots, x_N\}$.

- We **don't know** which Gaussian each point came from.
- We **want to estimate** the parameters π_k, μ_k, σ_k^2.

This is exactly where **EM** comes in.

Expectation Maximization (EM) Algorithms

EM for Normal Mixture Models

E-step (Expectation):

For each data point x_i, compute the probability it belongs to cluster k:

$$\gamma_{ik} = P(z_i = k \mid x_i, \theta) = \frac{\pi_k \mathcal{N}(x_i \mid \mu_k, \sigma_k^2)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_i \mid \mu_j, \sigma_j^2)}$$

This is called the **responsibility** (soft assignment).

Expectation Maximization (EM) Algorithms

EM for Normal Mixture Models

E-step (Expectation):

For each data point x_i, compute the probability it belongs to cluster k:

$$\gamma_{ik} = P(z_i = k \mid x_i, \theta) = \frac{\pi_k \mathcal{N}(x_i \mid \mu_k, \sigma_k^2)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_i \mid \mu_j, \sigma_j^2)}$$

This is called the **responsibility** (soft assignment).

Expectation Maximization (EM) Algorithms

M-step (Maximization):

Update the parameters using these responsibilities:

- New mixing weights:

$$\pi_k = \frac{1}{N} \sum_{i=1}^{N} \gamma_{ik}$$

- New means:

$$\mu_k = \frac{\sum_{i=1}^{N} \gamma_{ik} \boldsymbol{x}_i}{\sum_{i=1}^{N} \gamma_{ik}}$$

Expectation Maximization (EM) Algorithms

M-step (Maximization):

Update the parameters using these responsibilities:

- New mixing weights:

$$\pi_k = \frac{1}{N} \sum_{i=1}^{N} \gamma_{ik}$$

- New means:

$$\mu_k = \frac{\sum_{i=1}^{N} \gamma_{ik} \boldsymbol{x}_i}{\sum_{i=1}^{N} \gamma_{ik}}$$

Expectation Maximization (EM) Algorithms

- New variances:

$$\sigma_k^2 = \frac{\sum_{i=1}^{N} \gamma_{ik}(x_i - \mu_k)^2}{\sum_{i=1}^{N} \gamma_{ik}}$$

Repeat until parameters converge.

After several iterations, the algorithm discovers the “hidden” structure: each data point’s cluster probability and the parameters of each Gaussian.

Expectation Maximization (EM) Algorithms

Data (1-D):

$$x = [1.0,\ 1.5,\ 2.0,\ 5.0,\ 6.0,\ 6.5]$$

Model: mixture of $K = 2$ Gaussians

$$p(x) = \pi_1 \mathcal{N}(x \mid \mu_1, \sigma_1^2) + \pi_2 \mathcal{N}(x \mid \mu_2, \sigma_2^2)$$

Initialization (we pick):

- $\pi_1 = \pi_2 = 0.5$
- $\mu_1 = 1.0,\ \mu_2 = 6.0$
- $\sigma_1^2 = \sigma_2^2 = 1.0$

Expectation Maximization (EM) Algorithms

point x_i	γ_{i1} (cluster 1)	γ_{i2} (cluster 2)
1.0	0.999996	0.000004
1.5	0.999955	0.000045
2.0	0.999447	0.000553
5.0	0.000553	0.999447
6.0	0.0000037	0.9999996
6.5	0.00000031	0.99999969

E-step (responsibilities)

For each point x_i and cluster k we compute

$$\gamma_{ik} = \frac{\pi_k \mathcal{N}(x_i \mid \mu_k, \sigma_k^2)}{\sum_{j=1}^{2} \pi_j \mathcal{N}(x_i \mid \mu_j, \sigma_j^2)}.$$

Expectation Maximization (EM) Algorithms

M-step (update parameters)

Using the responsibilities we compute the effective counts $N_k = \sum_i \gamma_{ik}$ and update:

$$\pi_k = \frac{N_k}{N}, \qquad \mu_k = \frac{\sum_i \gamma_{ik} x_i}{N_k}, \qquad \sigma_k^2 = \frac{\sum_i \gamma_{ik} (x_i - \mu_k)^2}{N_k}.$$

After one full E+M iteration the updated parameters are approximately:

- $\pi_1 \approx 0.49999,\ \pi_2 \approx 0.50001$
- $\mu_1 \approx 1.50056,\ \mu_2 \approx 5.83271$
- $\sigma_1^2 \approx 0.1689,\ \sigma_2^2 \approx 0.3918$

Expectation Maximization (EM) Algorithms

(repeat E then M), the algorithm quickly **converges** to:

- $\pi_1 \approx 0.5,\ \pi_2 \approx 0.5$
- $\mu_1 = 1.5,\ \mu_2 \approx 5.8333333$
- $\sigma_1^2 \approx 0.1666667,\ \sigma_2^2 \approx 0.3888889$