Density Estimation

Unsupervised learning: describing data

1

Dimensionality Reduction

Developing new data representations

- Feature subset Selection
- Feature projections
- Supervised approaches

2

Density Estimation

Quantifying data distributions

- Histograms
- Nonparametric density estimation
- Parametric models

3

Clustering

Grouping similar data

- Hierarchical
- Centroid-based
- Distribution-based
- Density-based

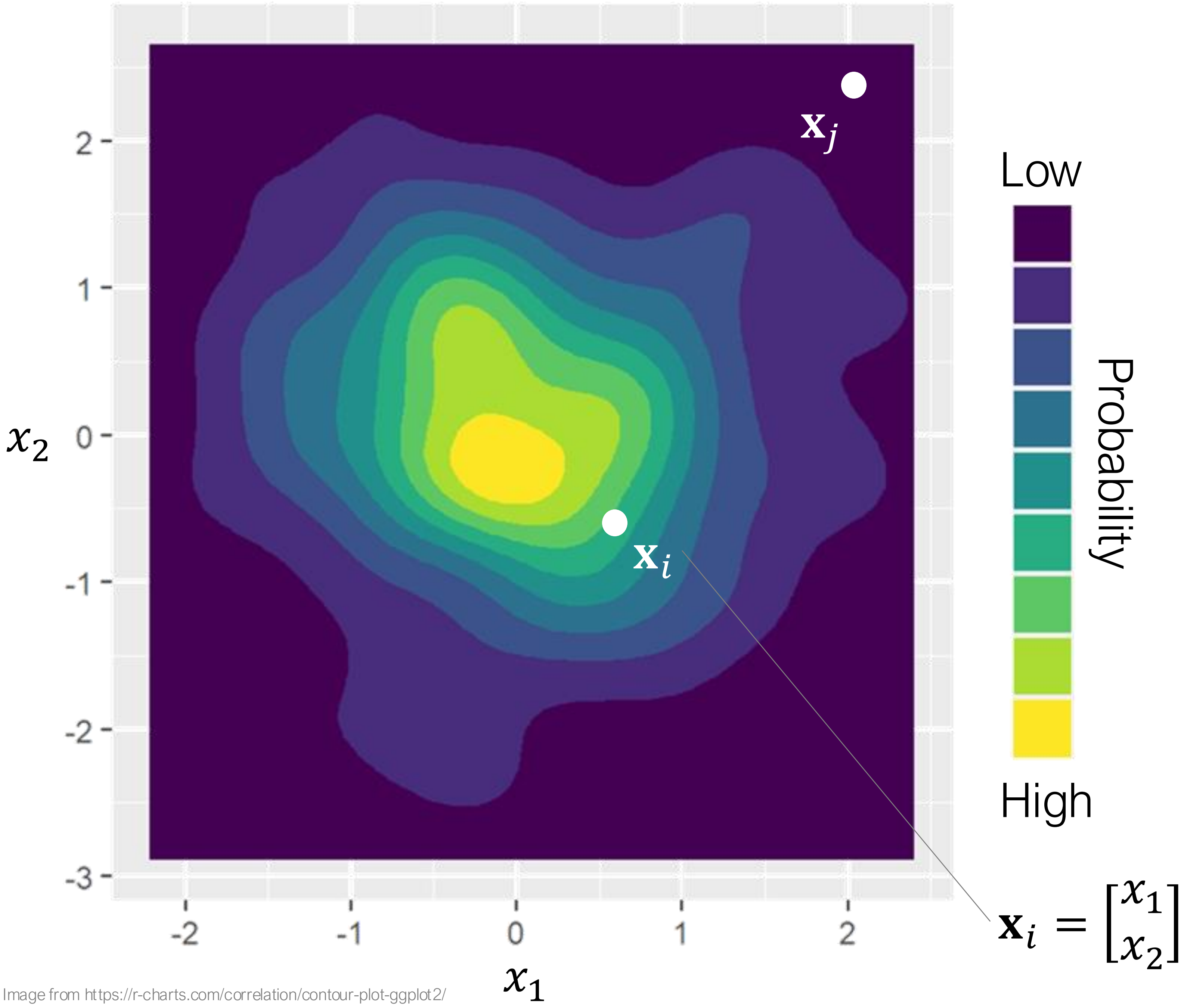
4

Anomaly detection

Identifying anomalies in data

- Probabilistic approaches
- Cluster-based
- Supervised approaches

Density Estimation



$$\mathbf{x}_i = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

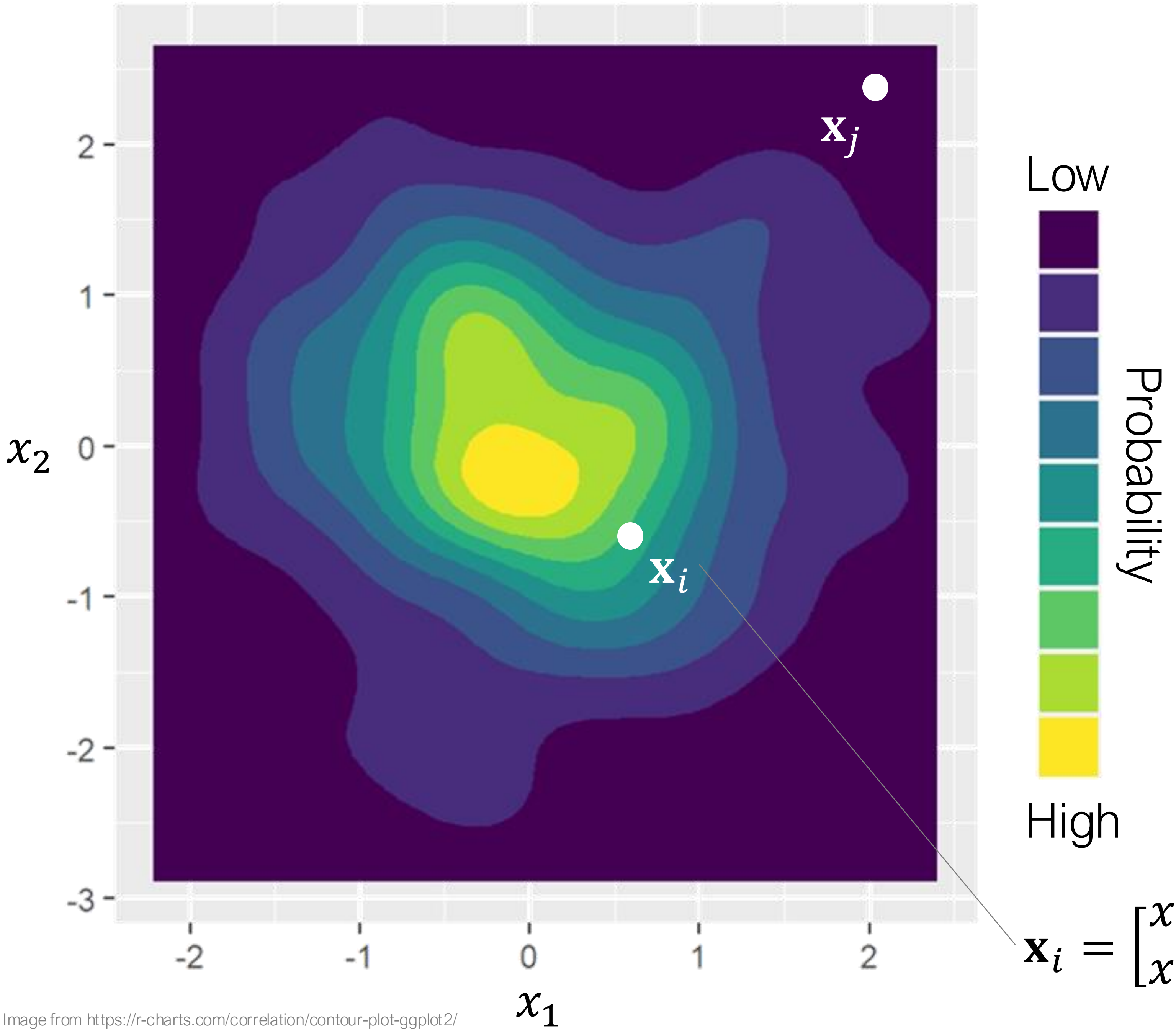
Image from https://r-charts.com/correlation/contour-plot-ggplot2/

Describes data

Image:

Text: "two parrots kiss on a branch"

Tabular: x_1 and x_2 are measurements of bird feathers



$$\mathbf{x}_i = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Image from https://r-charts.com/correlation/contour-plot-ggplot2/

Anomaly detection

$\mathbf{x}_j$ is unlikely,

$\mathbf{x}_i$ is likely

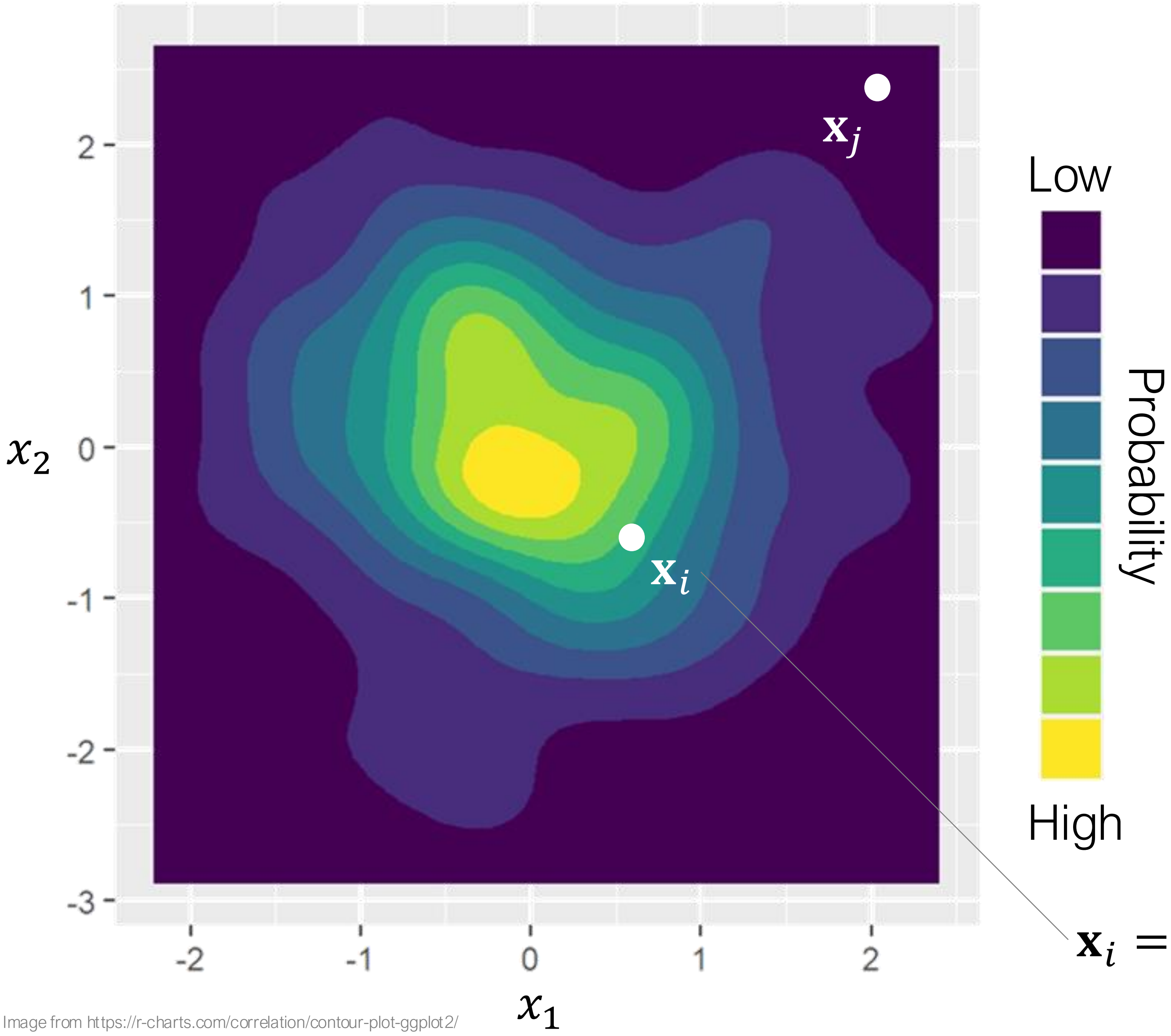


Image from https://r-charts.com/correlation/contour-plot-ggplot2/

$$\mathbf{x}_i = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Data synthesis

Image:

Text: "two parrots kiss on a branch"

Tabular: x_1 and x_2 are measurements of bird feathers

Properties of probability distributions

- Always greater than zero
- Integrates to 1

Common approaches to density estimation

- Parametric density estimation
  - Distribution fitting (e.g. normal, exponential, etc.)
  - Mixture models (e.g. Gaussian mixture models)
- Histograms
- Kernel density estimation

Parametric Density Estimation

If we have knowledge of a possible parametric form, we can estimate the parameters of the model

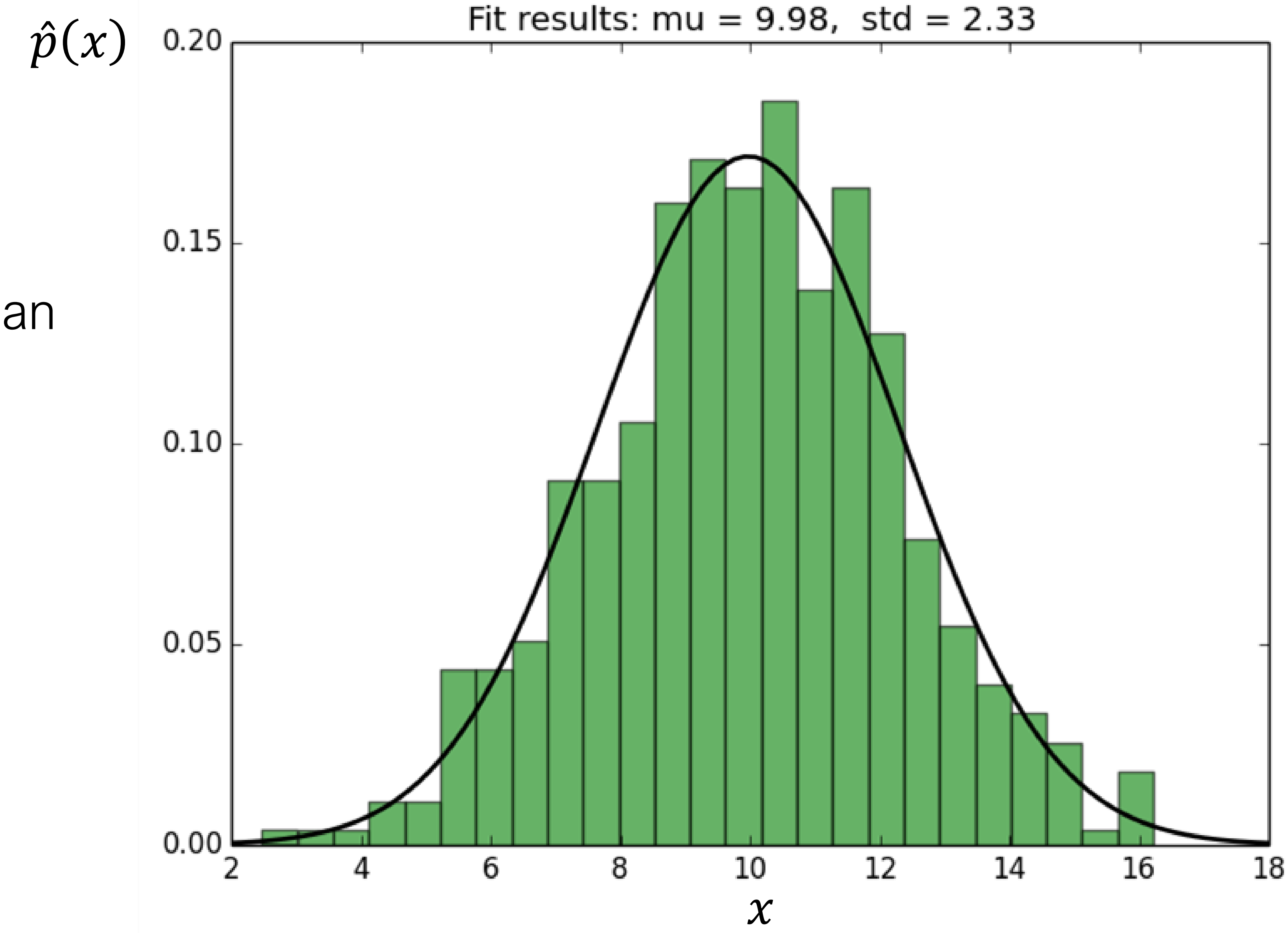
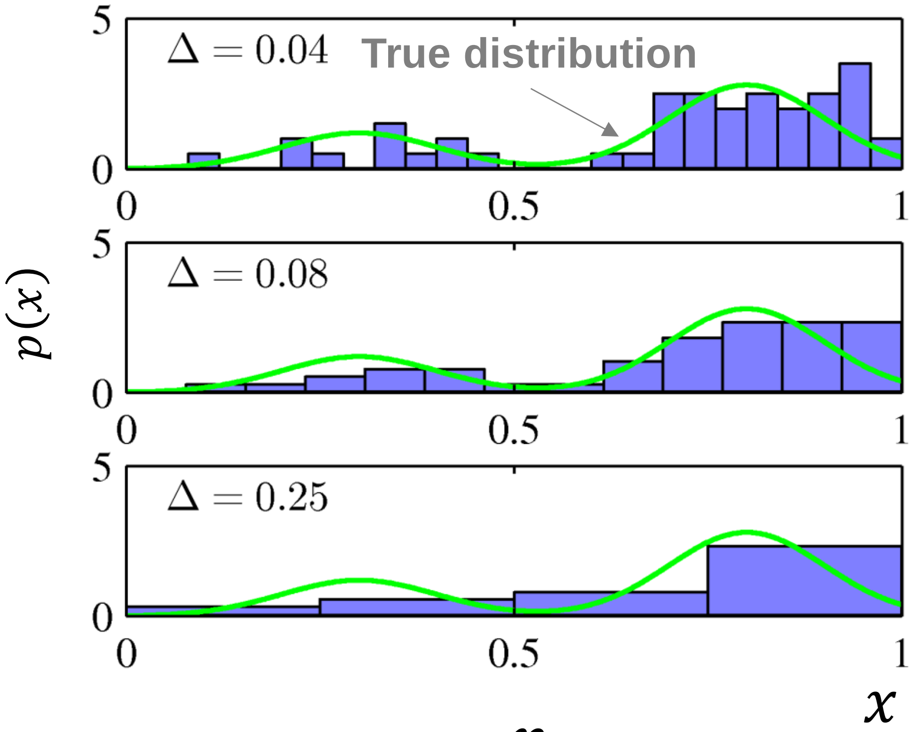


Image from: https://stackoverflow.com/questions/20011122/fitting-a-normal-distribution-to-1d-data

Histogram Density Estimation

Histogram



$$p(x) = \frac{n_i}{N\Delta_i}$$

n_i = # observations of x falling in bin i
N = total # observations
Δ_i = width of bin i

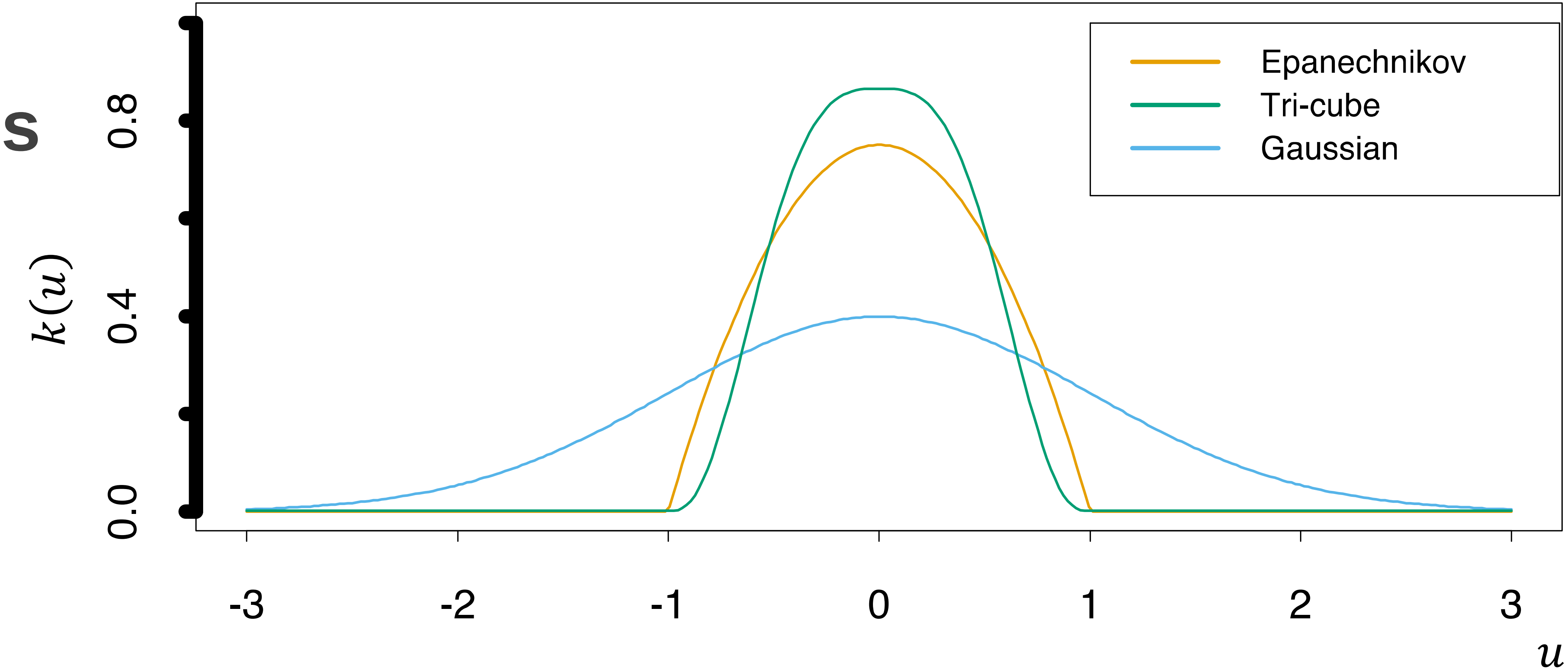
Highly dependent on the choice of bin width, Δ_i

Has discontinuities at the bin edges

Local neighborhoods do appear to be helpful

Bishop, Pattern Recognition and Machine Learning, 2006

Kernel Functions (window kernels)



Satisfy two properties:

$$k(u) \geq 0$$

$$\int k(u) du = 1$$

Epanechnikov

$$k(u) = \frac{3}{4}(1 - u^2)$$

$$|u| \leq 1$$

Tri-cube

$$k(u) = \frac{70}{81}(1 - |u^3|)^3$$

$$|u| \leq 1$$

Gaussian

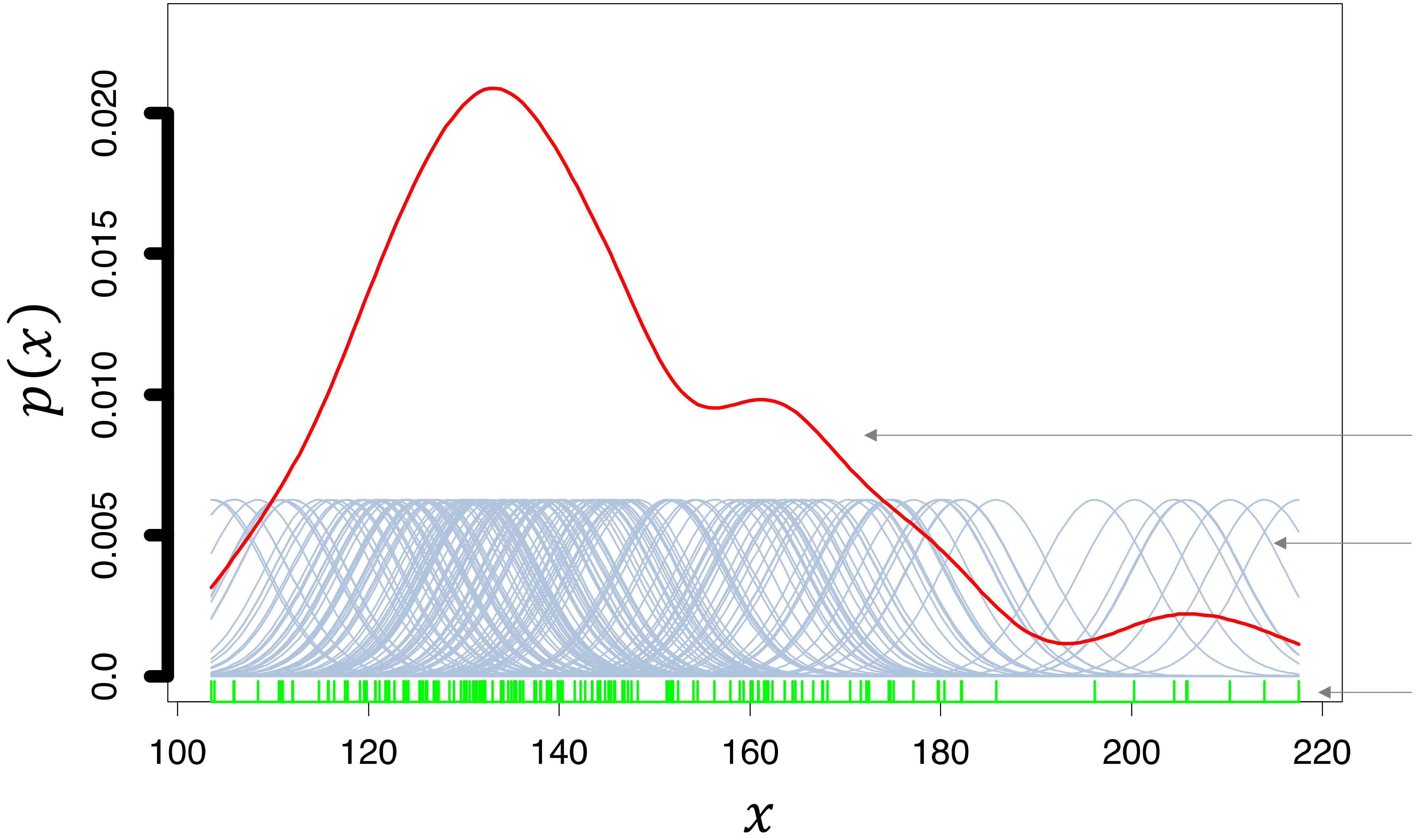
$$k(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}u^2}$$

$$-\infty < u < \infty$$

Hastie, Tibshirani, and Friedman, The Elements of Statistical learning, 2001

Kernel Density Estimation

a. k. a. Parzen Window Density Estimation



Density estimate

Kernel functions
(Gaussian kernel shown)

Observations (x-values)

1. Center a kernel function at each observation (x-value)

2. Estimate $p(x)$ by averaging across all the kernels at x

Hastie, Tibshirani, and Friedman, The Elements of Statistical learning, 2001

Kernel Density Estimation

Center the kernel function at each x-value in the dataset:

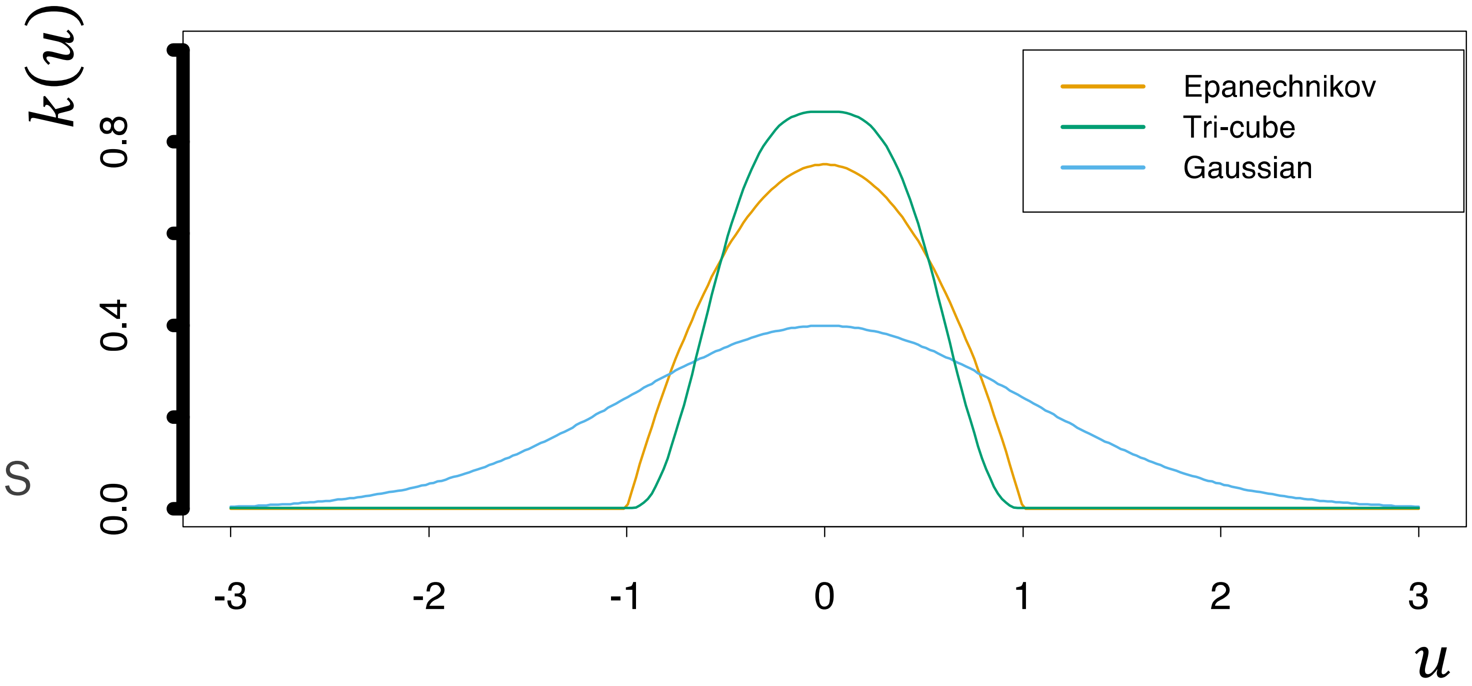
$$k(x - x_n) \qquad n = 1,2,\ldots,N$$

Average over all of the kernel functions to get the density estimate:

$$p(x) = \frac{1}{N}\sum_{n=1}^{N} k(x - x_n)$$

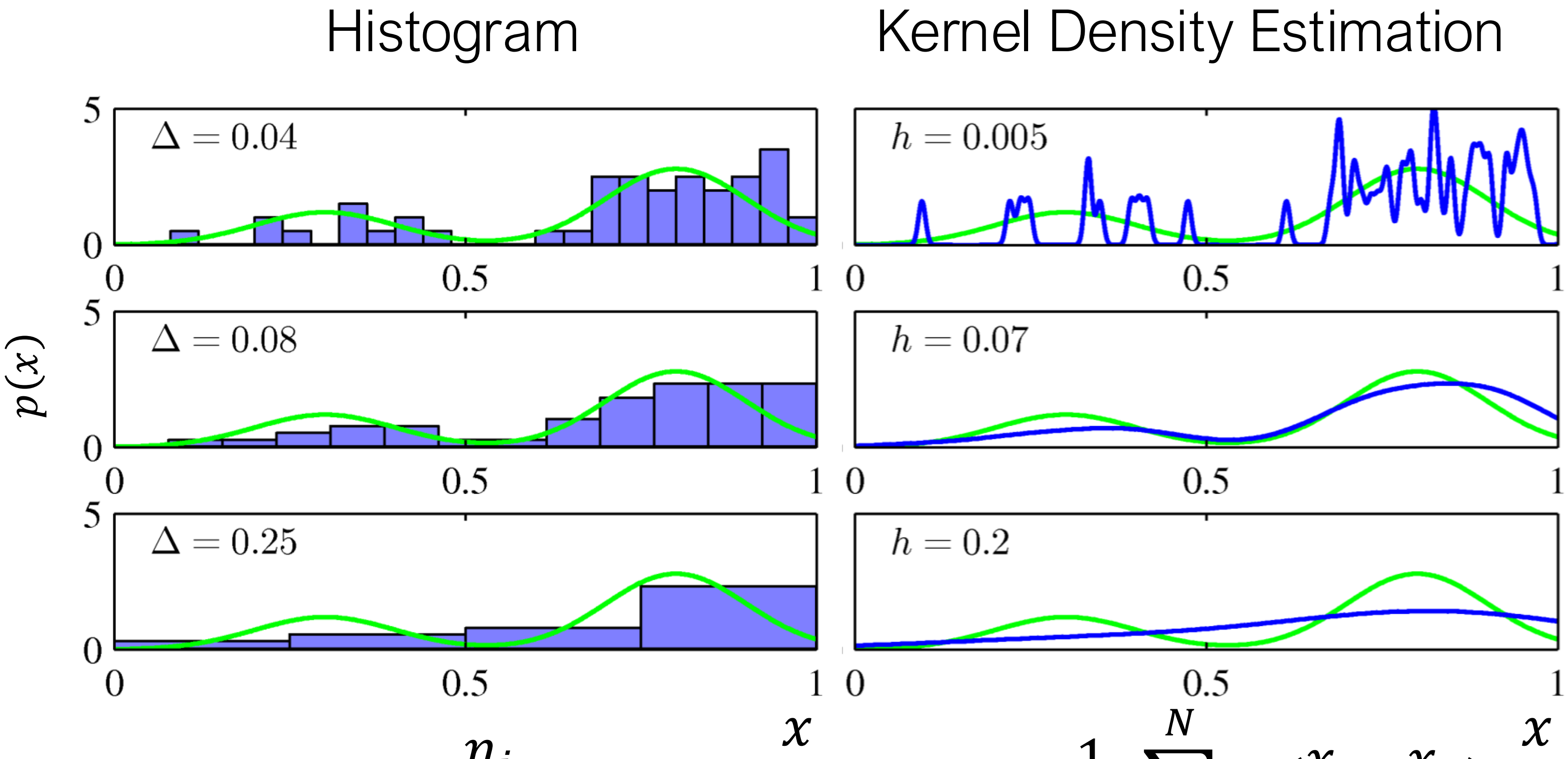
Note: we can scale the width of the kernel function with a scale factor, h:

$$k\left(\frac{x - x_n}{h}\right)$$



For kernel functions with **finite domains**, this means that each observation, x, will only affect the density estimate in a **neighborhood** close to the center of the kernel

Kernel Density Estimation



Requires tuning h, the kernel width parameter

Computational cost of evaluating this density grows linearly with the size of the data

$$p(x) = \frac{n_i}{N\Delta_i}$$

$$p(x) = \frac{1}{Nh}\sum_{n=1}^{N} k\left(\frac{x - x_n}{h}\right)$$

n_i = # observations of x falling in bin i
N = total # observations
Δ_i = width of bin i

x_n = The nth observation of x
k = kernel function
h = width of the kernel

Bishop, Pattern Recognition and Machine Learning, 2006

Density estimation uses

Describing the distribution of data and its characteristics

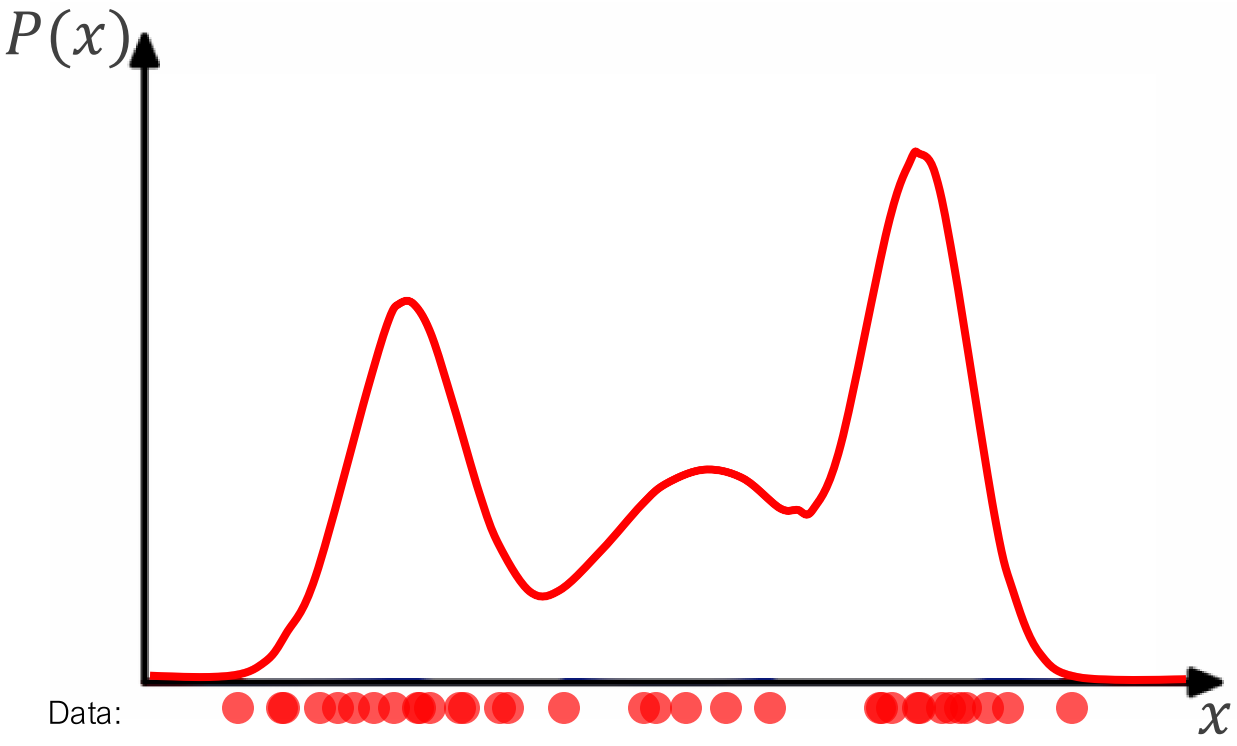
Can be used for anomaly/outlier detection

- If a new sample has a low “probability” given the distribution of the data, then it may be anomalous

Gaussian Mixture Models

For **clustering** and **density estimation**

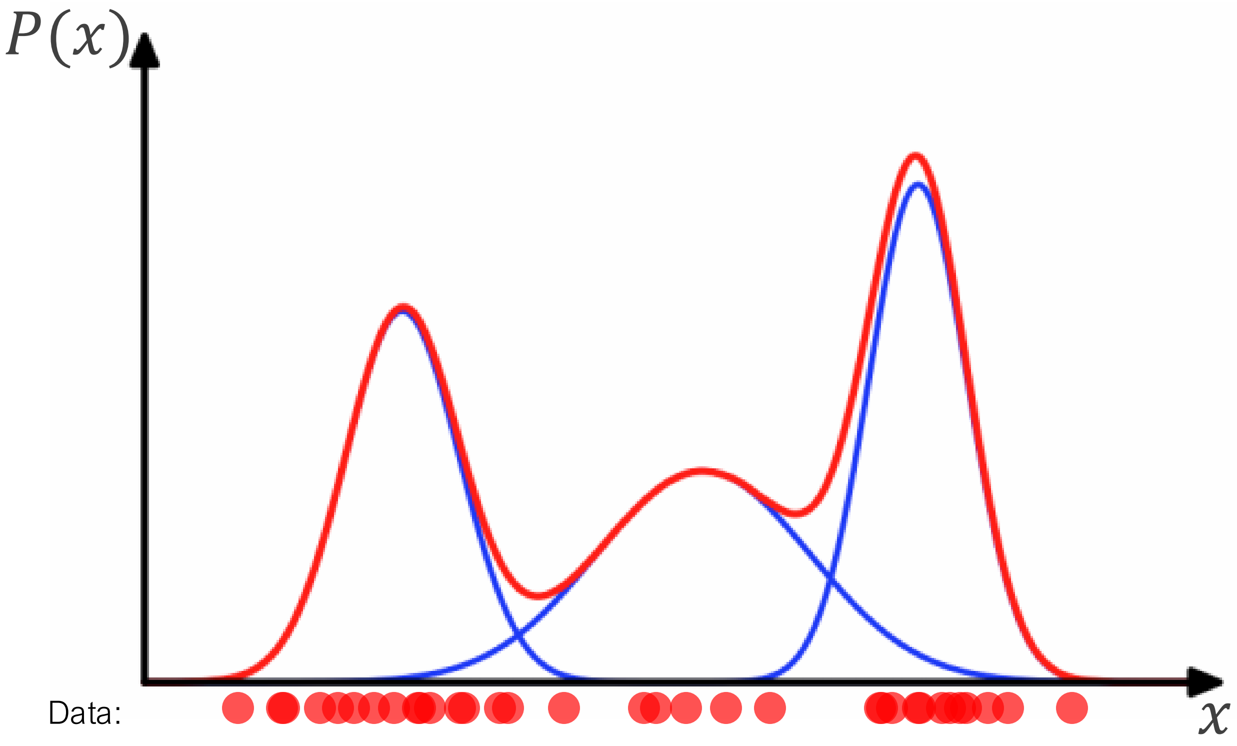
Mixture model



We can estimate the distribution density of our data…

Image from Shaun Dowling

Mixture model

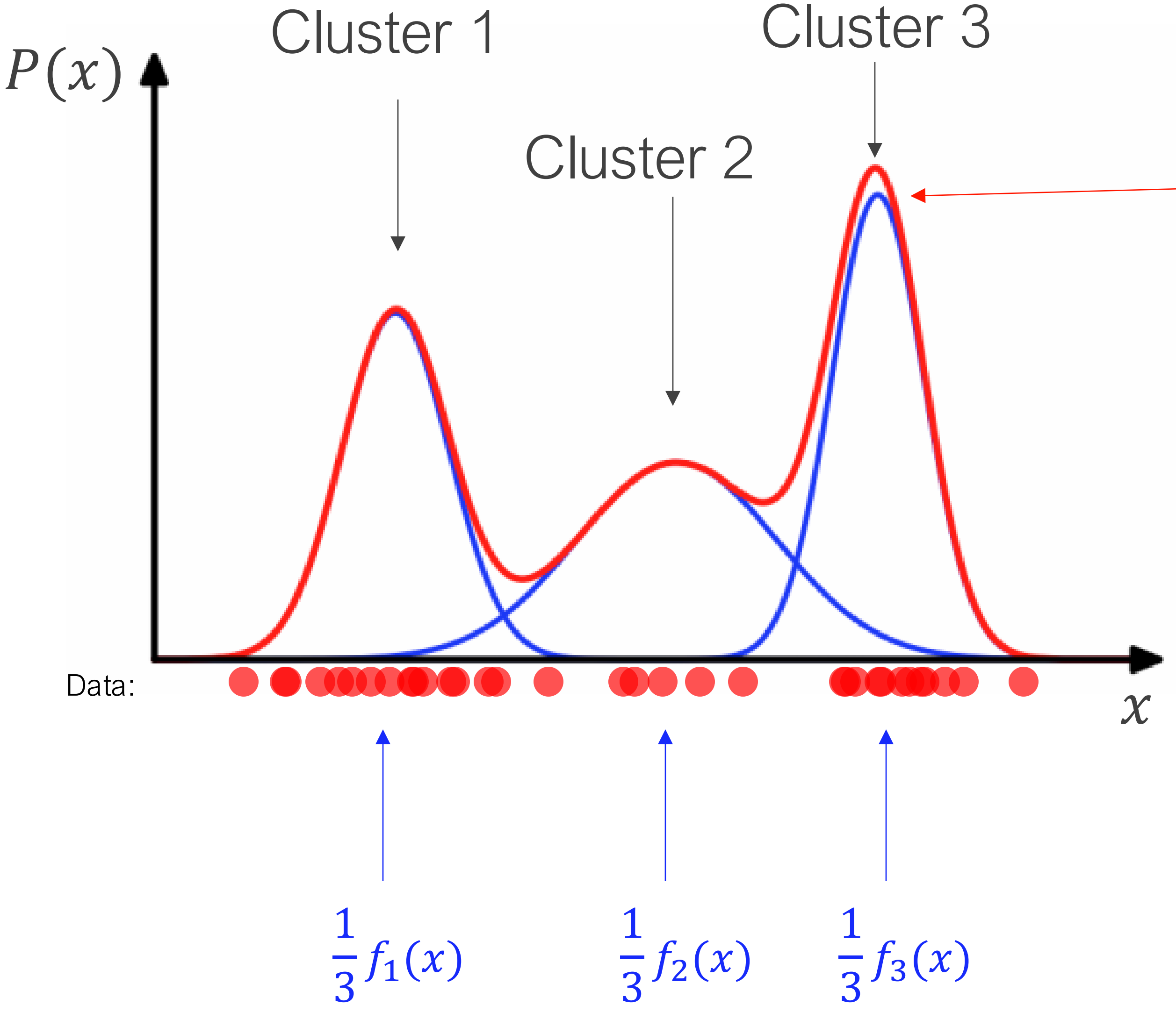


We can estimate the distribution density of our data…

…using a mixture of distributions

Image from Shaun Dowling

Mixture model



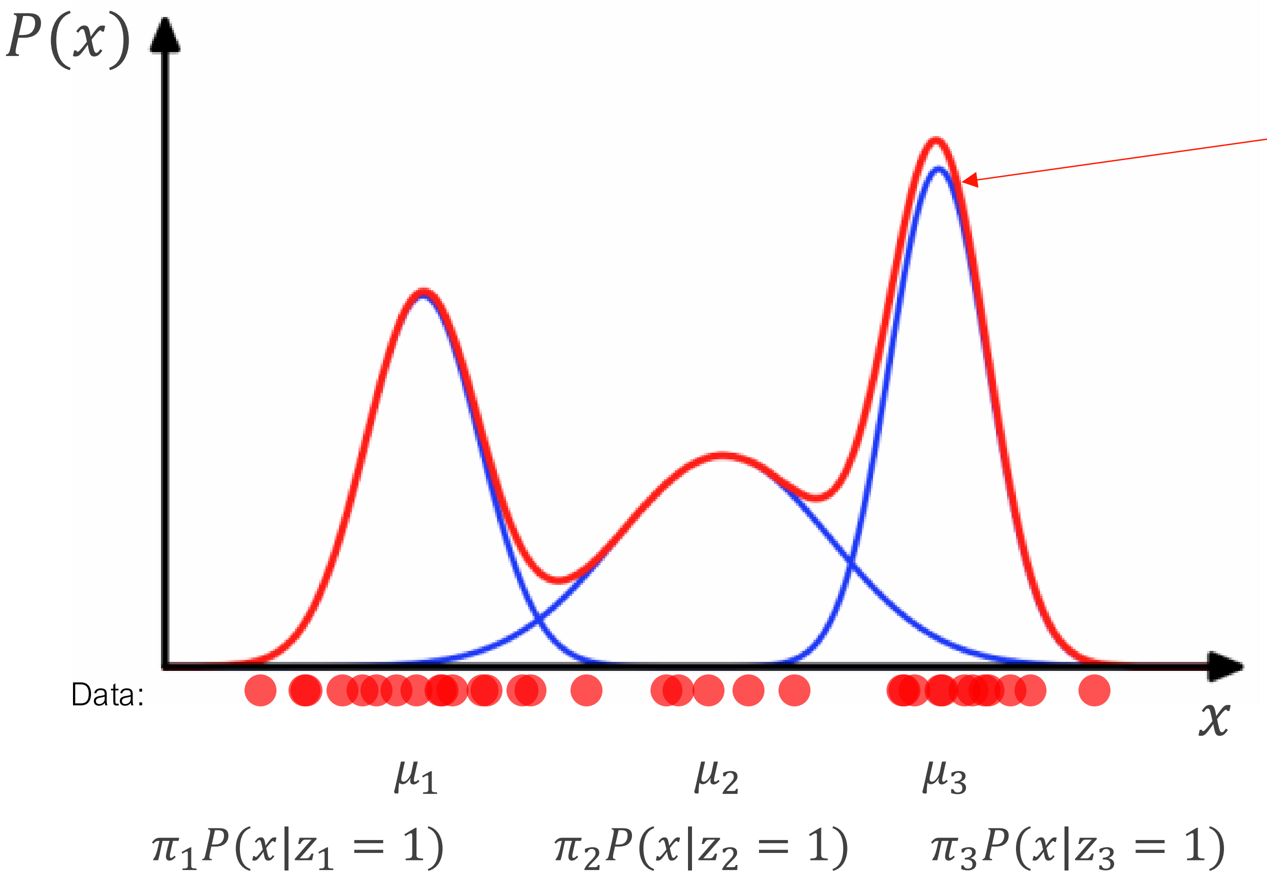
A weighted average of density functions

$$P(x) = \frac{1}{3}f_1(x) + \frac{1}{3}f_2(x) + \frac{1}{3}f_3(x)$$

1. Fit the model to the data
2. Use the model to assign clusters

Image from Shaun Dowling

Gaussian mixture model



A mixture model is represented as:

$$P(x) = \sum_{k=1}^{K} P(z_k = 1) P(x|z_k = 1)$$

If we assume this is Gaussian, it becomes a Gaussian mixture model (GMM)

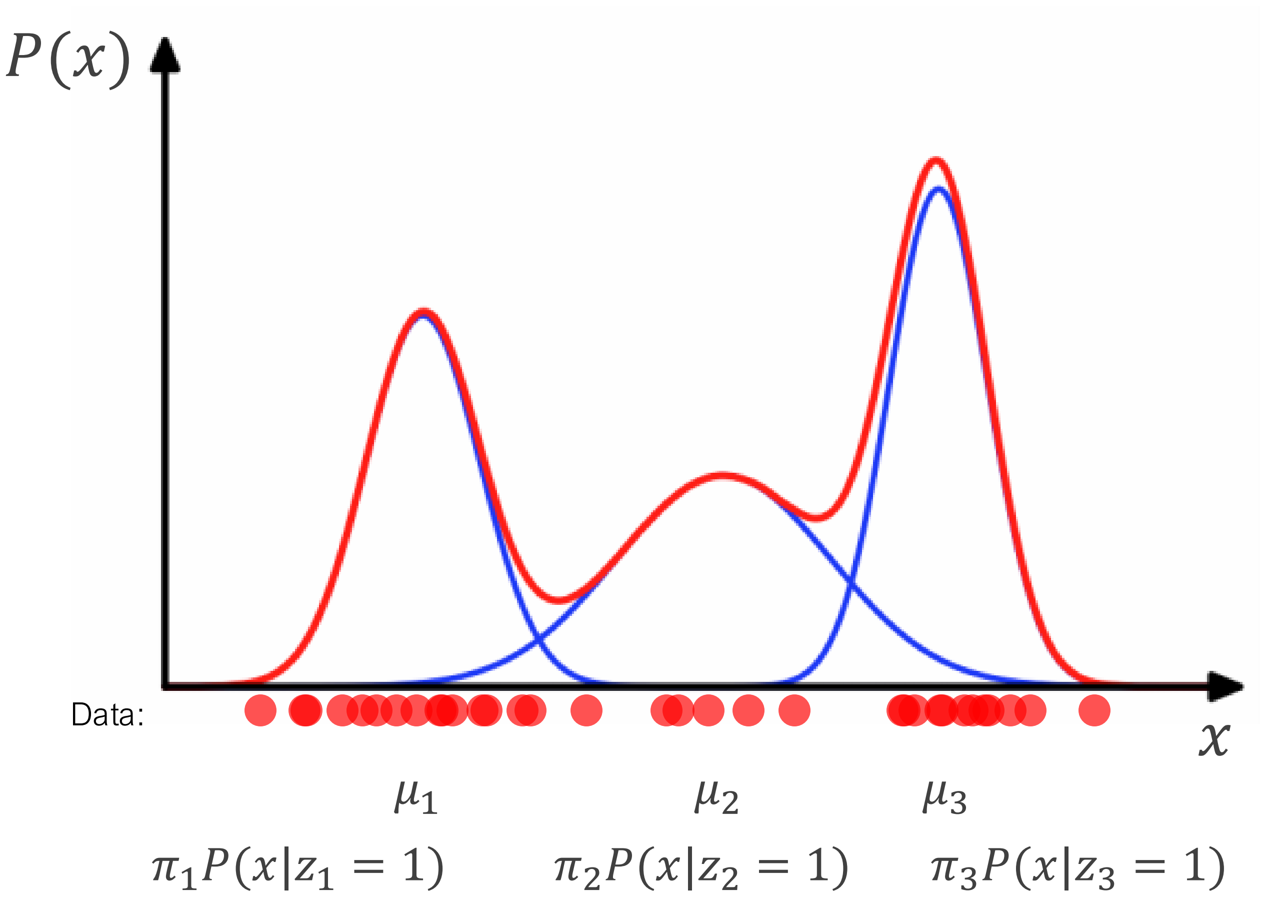
The mixing coefficients $\pi_k = P(z_k = 1)$ need to sum to 1 for a valid distribution

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

z_k = binary variable that represents cluster membership

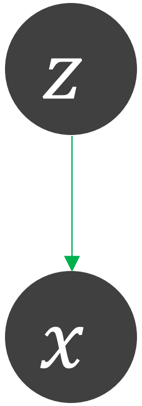
Image from Shaun Dowling

Gaussian mixture model



$$P(x) = \sum_{k=1}^{K} P(z_k = 1) P(x|z_k = 1)$$

Here we assume z is a **latent** (hidden / unobservable) variable



Hidden

This variable controls which of the k mixture components a sample is drawn from. We don't DIRECTLY see this.

Observable

Given z, we assume a sample is drawn from $P(x|z_k = 1)$

Note: We can use these terms to compute the posterior probability $P(z_k|x)$

Image from Shaun Dowling

Gaussian Mixture Model Latent Variables

Complete data with latent variable “labels” z

Incomplete data without latent variable labels

Posterior probabilities, a.k.a. responsibilities

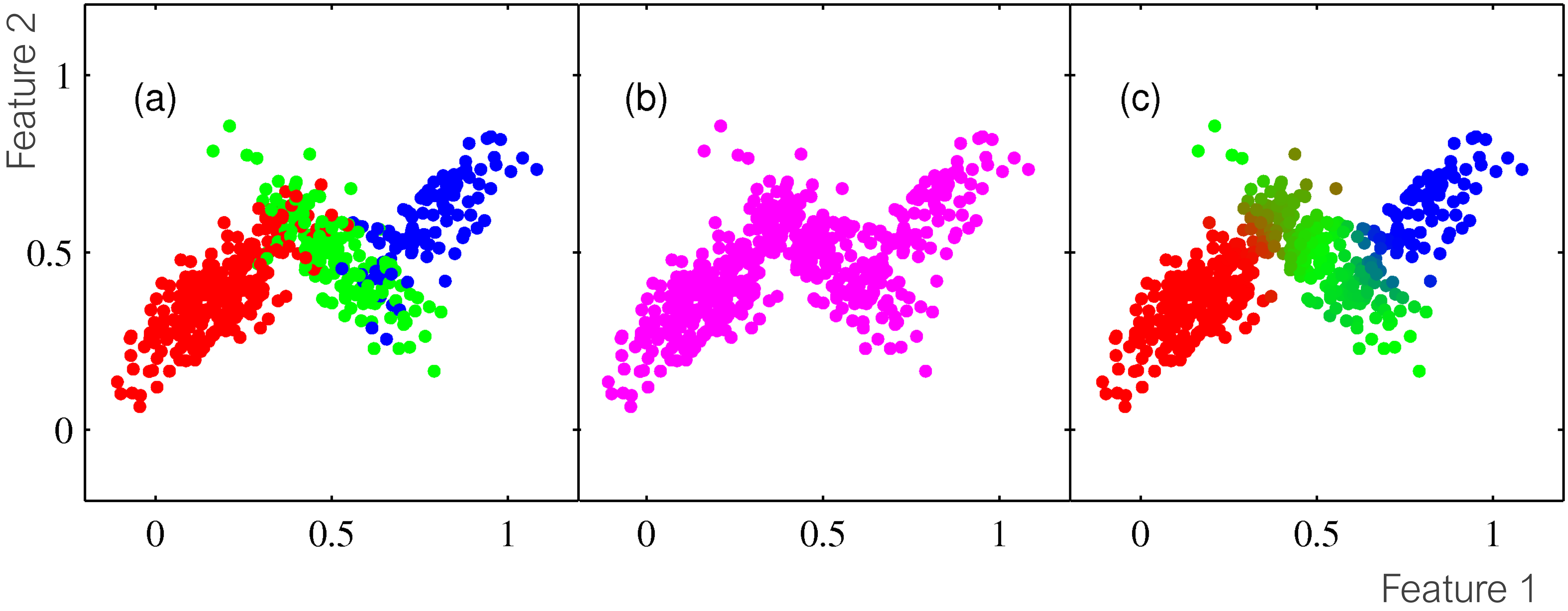
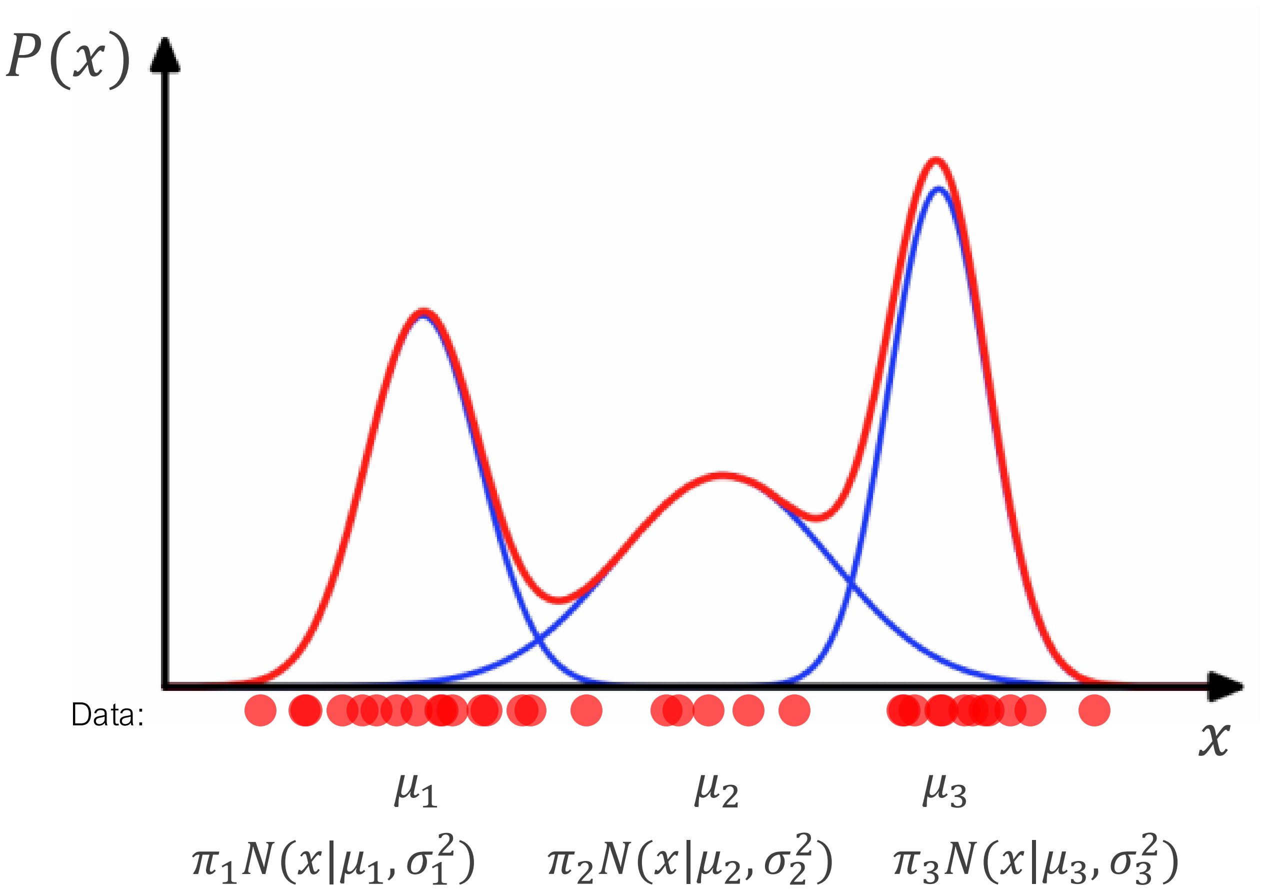


Image from Bishop, Pattern Recognition, 2006

Gaussian mixture model



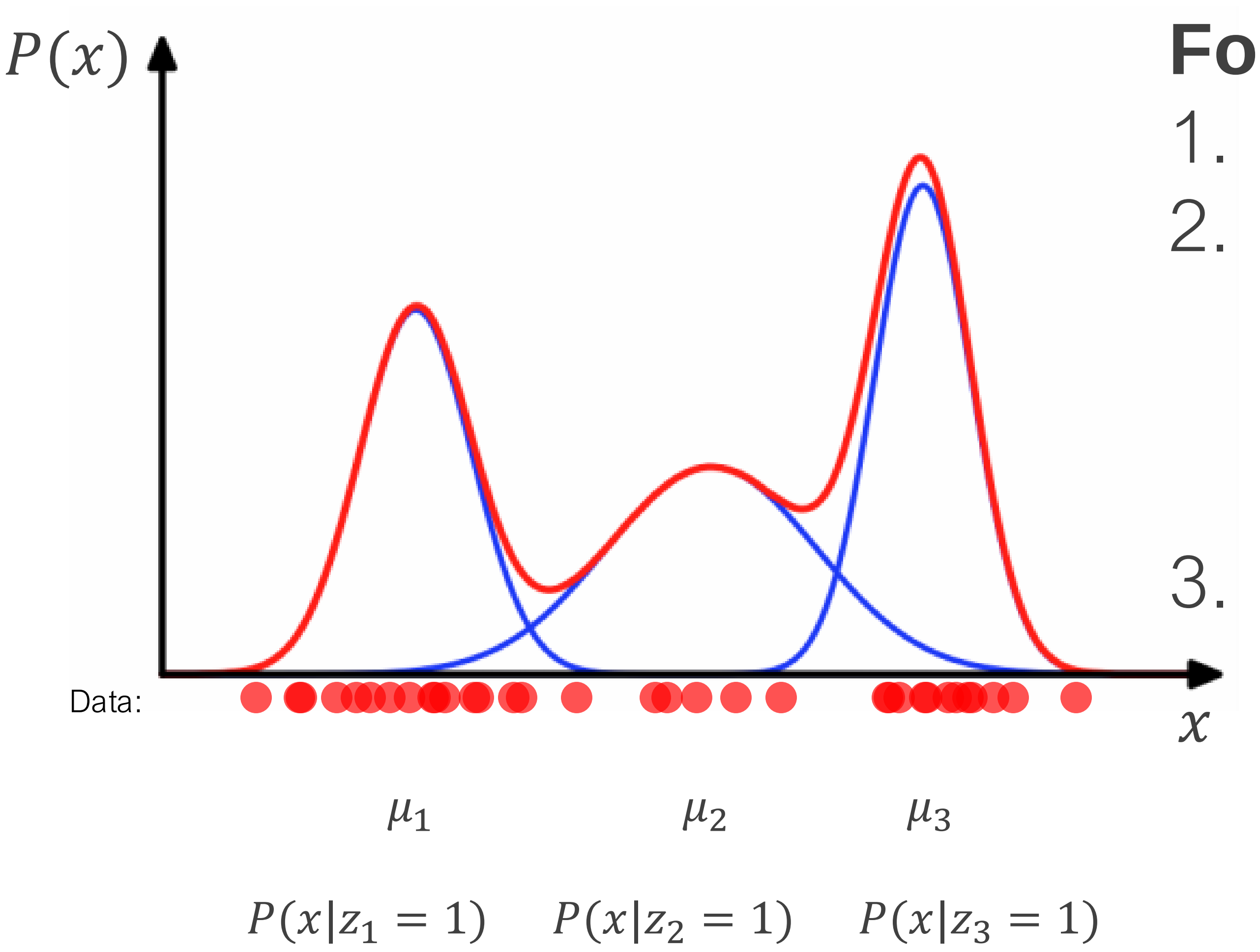
The Gaussian mixture model is represented as:

$$P(x) = \sum_{k=1}^{K} \pi_k N(x|\mu_k, \sigma_k^2)$$

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

Image from Shaun Dowling

Gaussian mixture model



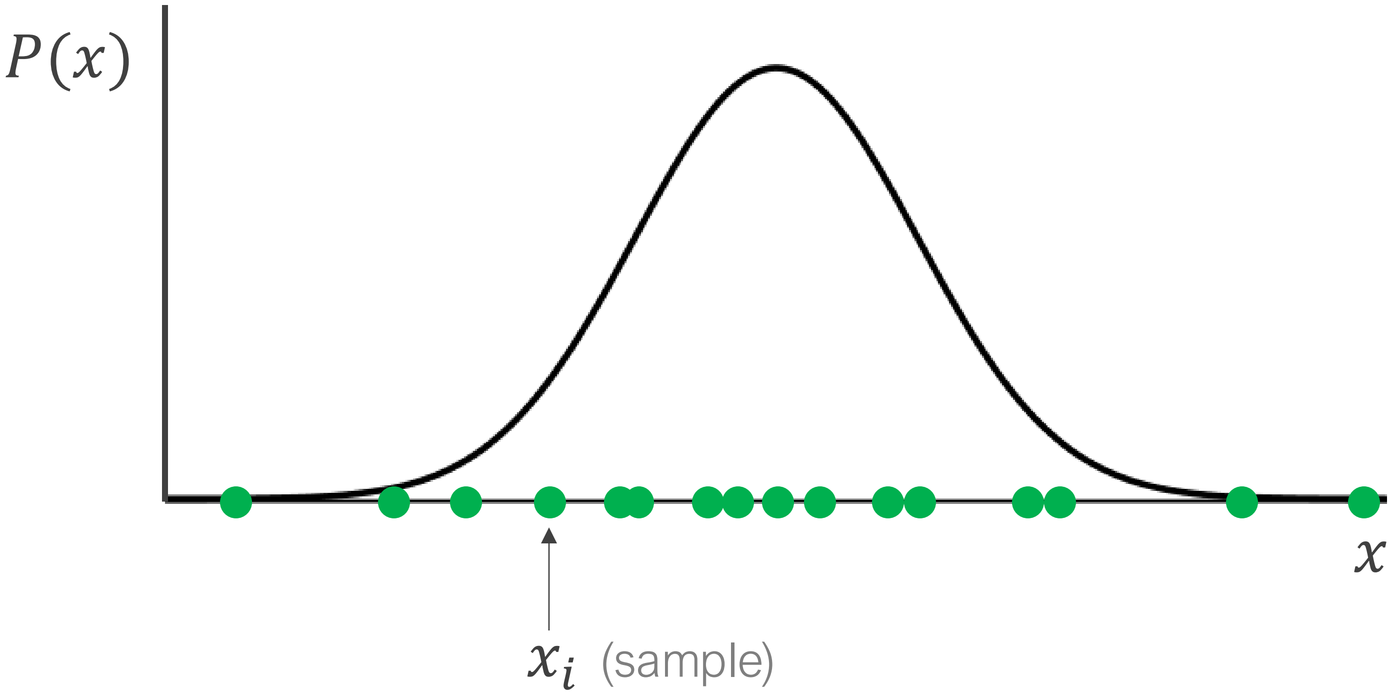
For clustering:

1. Pick a number of clusters, K
2. Fit a GMM to the data (estimate π_k, μ_k, σ_k^2 for $k = 1, \ldots, K$ to maximize the likelihood of the data given the model)
3. Pick the cluster, z_k, that each data point was most likely to come from

Image from Shaun Dowling

Density estimation for a single mixture component

a.k.a. model fitting



Likelihood of one sample given the model

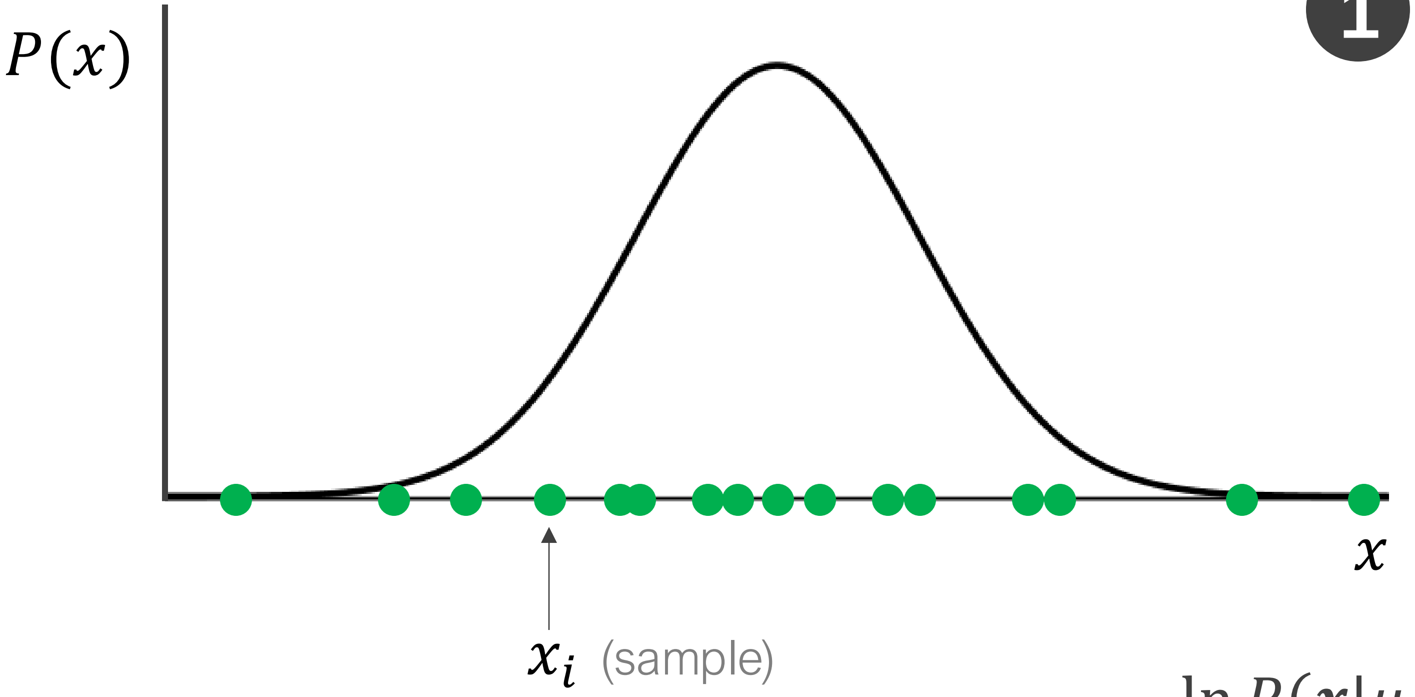
$$P(x_i|\mu,\sigma^2) = N(x_i|\mu,\sigma^2)$$
$$= \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$$

Assuming independent samples, the likelihood of the data given the model is:

$$P(\boldsymbol{x}|\mu,\sigma^2)$$
$$= \prod_{i=1}^{N} P(x_i|\mu,\sigma^2)$$
$$= \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$$

Density estimation for a single mixture component

a.k.a. model fitting



1 We follow our familiar pattern: maximize the likelihood of the data by choosing our model parameters: μ, σ^2

$$P(\boldsymbol{x}|\mu,\sigma^2) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$$

2 Calculate the log likelihood:

$$\ln P(\boldsymbol{x}|\mu,\sigma^2) = -\frac{N}{2}\ln 2\pi\sigma^2 - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(x_i-\mu)^2$$

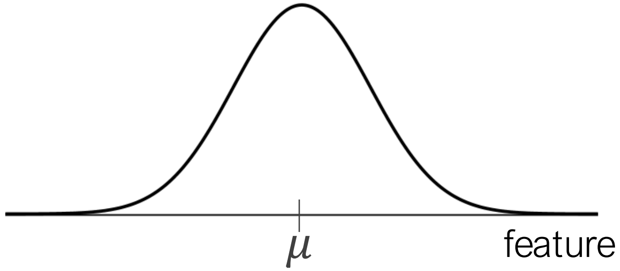
3 Take the derivative of the log likelihood w.r.t. each parameter (μ, σ^2), set equal to zero, solve for μ, σ^2

$$\hat{\mu} = \frac{1}{N}\sum_{i=1}^{N} x_i \qquad \hat{\sigma}^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i-\hat{\mu})^2$$

From a univariate to a multivariate Gaussian

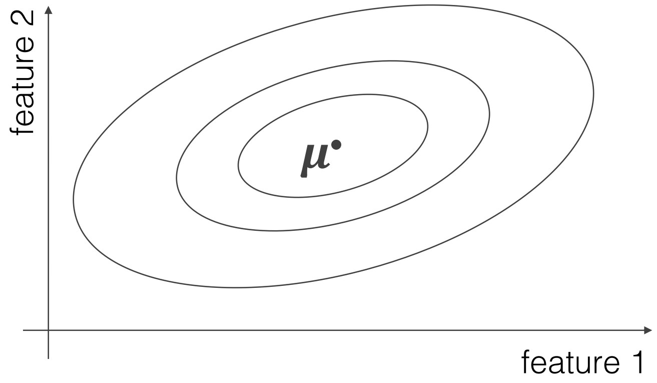
Univariate Normal density

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



Multivariate Normal density

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



From a univariate to a multivariate Gaussian

Univariate Normal MLE parameter estimates:

$$\hat{\mu} = \frac{1}{N}\sum_{i=1}^{N} x_i \qquad \hat{\sigma}^2 = \frac{1}{N}\sum_{i=1}^{N} (x_i - \hat{\mu})^2$$

Multivariate Normal MLE parameter estimates:

$$\hat{\boldsymbol{\mu}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_i \qquad \hat{\boldsymbol{\Sigma}} = \frac{1}{N}\sum_{i=1}^{N} (\boldsymbol{x}_i - \hat{\boldsymbol{\mu}})(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}})^T$$

Moving from a single Gaussian to a mixture of Gaussians

Density estimation for a Gaussian mixture model

0 We define the likelihood of one observation given our model with parameters $\boldsymbol{\pi}_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$ for $k = 1, \dots, K$

$$P(\boldsymbol{x}_i | \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{k=1}^{K} \pi_k N(\boldsymbol{x}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

1 We assume the observations are independent and calculate the likelihood for all our data

$$P(\boldsymbol{X} | \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{i=1}^{N} \sum_{k=1}^{K} \pi_k N(\boldsymbol{x}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

2 Calculate the log likelihood:

$$\ln P(\boldsymbol{X} | \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{i=1}^{N} \ln \left[\sum_{k=1}^{K} \pi_k N(\boldsymbol{x}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right]$$

3 Take the derivative of the log likelihood w.r.t. each parameter ($\boldsymbol{\pi}_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$ for $k = 1, \dots, K$), set equal to zero, solve for the parameters

Density estimation for a Gaussian mixture model

Log likelihood of the data given the model parameters

$$\ln P(\boldsymbol{X}|\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Sigma}) = \sum_{i=1}^{N} \ln\left[\sum_{k=1}^{K} \pi_k N(\boldsymbol{x}_i|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)\right]$$

There is no **closed-form solution** that maximizes this.

We could use gradient descent BUT this approach can suffer from **severe overfitting**

Example: $k = 2$ mixture components

$$\ln P(\boldsymbol{X}|\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Sigma}) = \sum_{i=1}^{N} \ln[\pi_1 N(\boldsymbol{x}_i|\boldsymbol{\mu}_1,\boldsymbol{\Sigma}_1) + \pi_2 N(\boldsymbol{x}_i|\boldsymbol{\mu}_2,\boldsymbol{\Sigma}_2)]$$

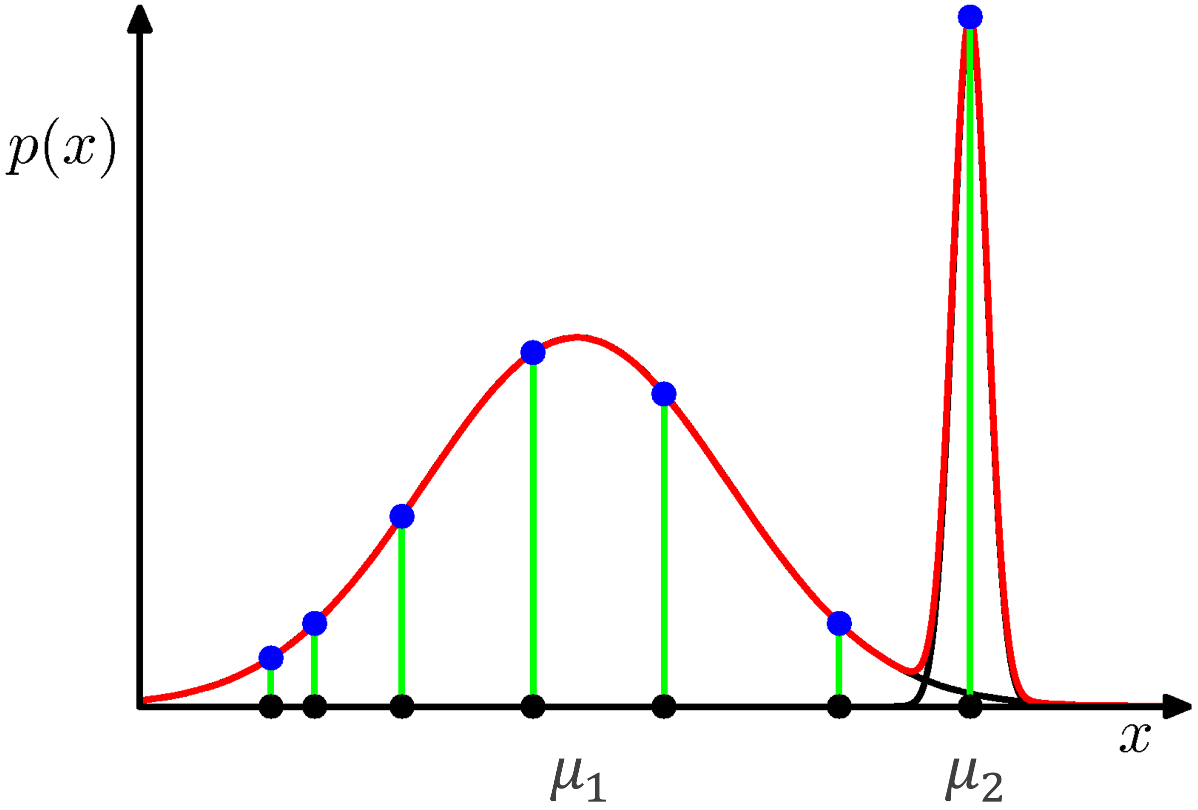
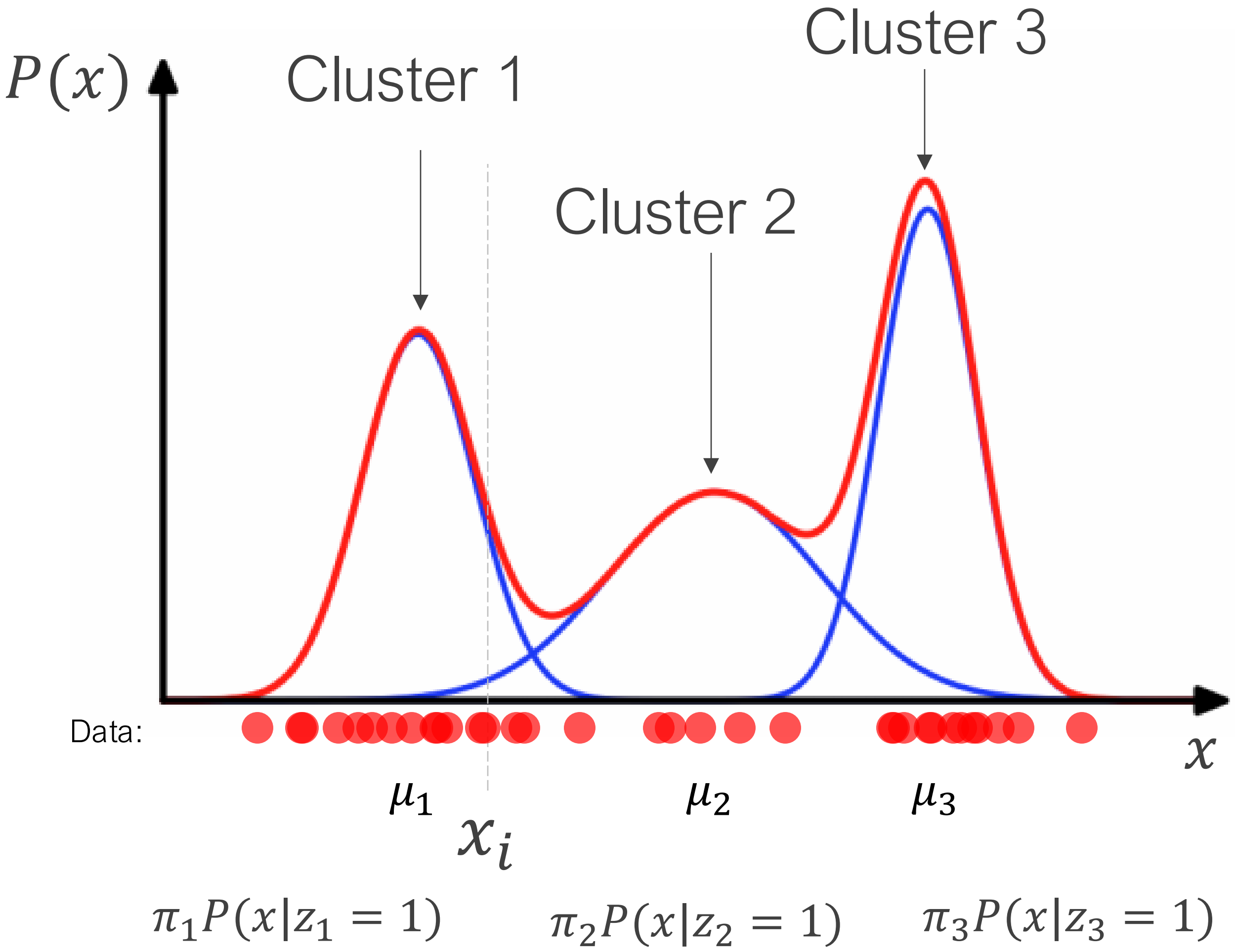


Image from Bishop, Pattern Recognition, 2006

How do we assign a cluster?

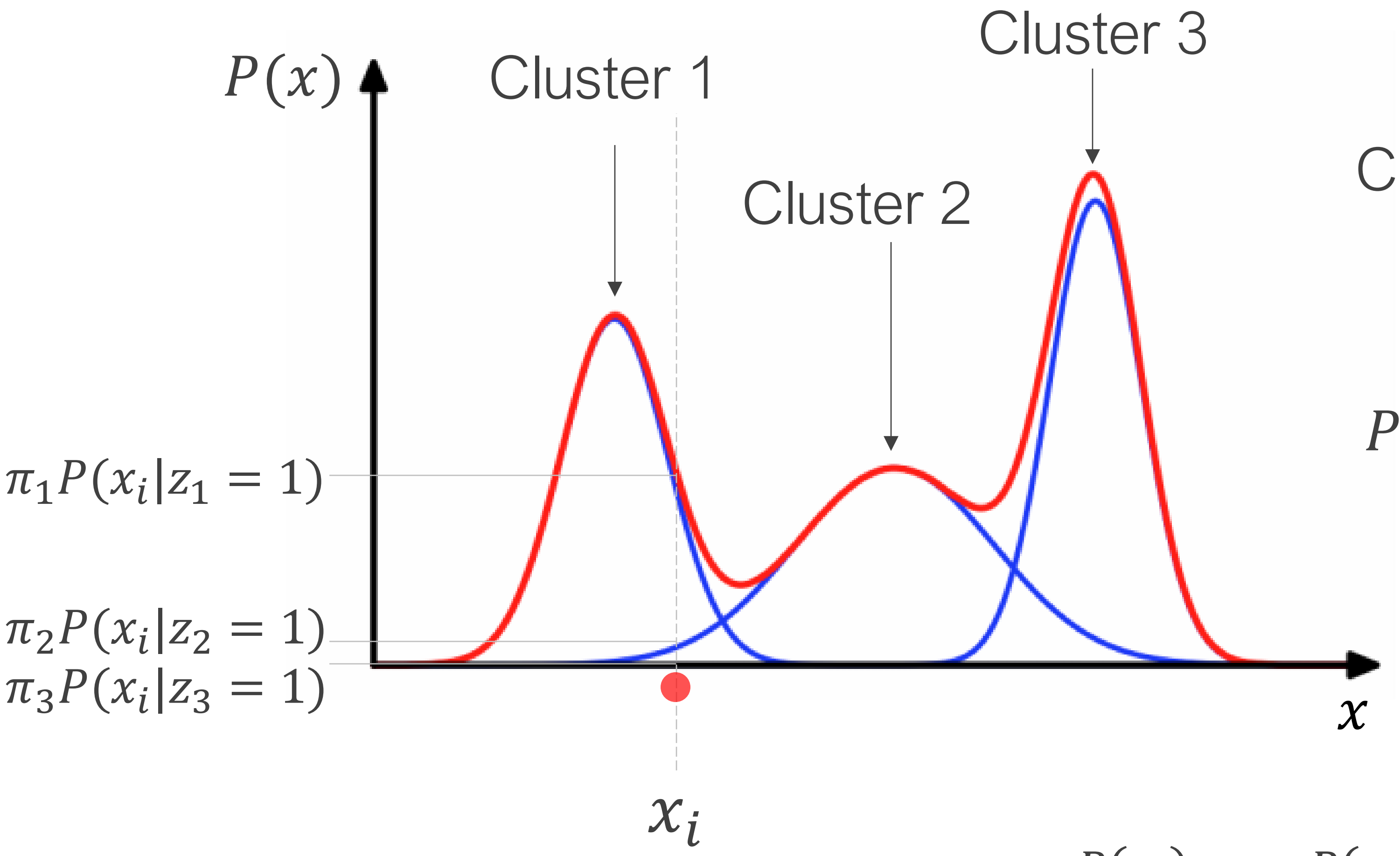


The probability of x_i is "explained" most by cluster 1, a little by cluster 2, and very little by cluster 3

We assign the cluster, z_k so that $P(z_k = 1|x)$ is the largest for all the k's

We need an expression for: $P(z_k = 1|x)$

How do we assign a cluster?



Consider observation x_i

normal distribution
for the kth cluster

π_k

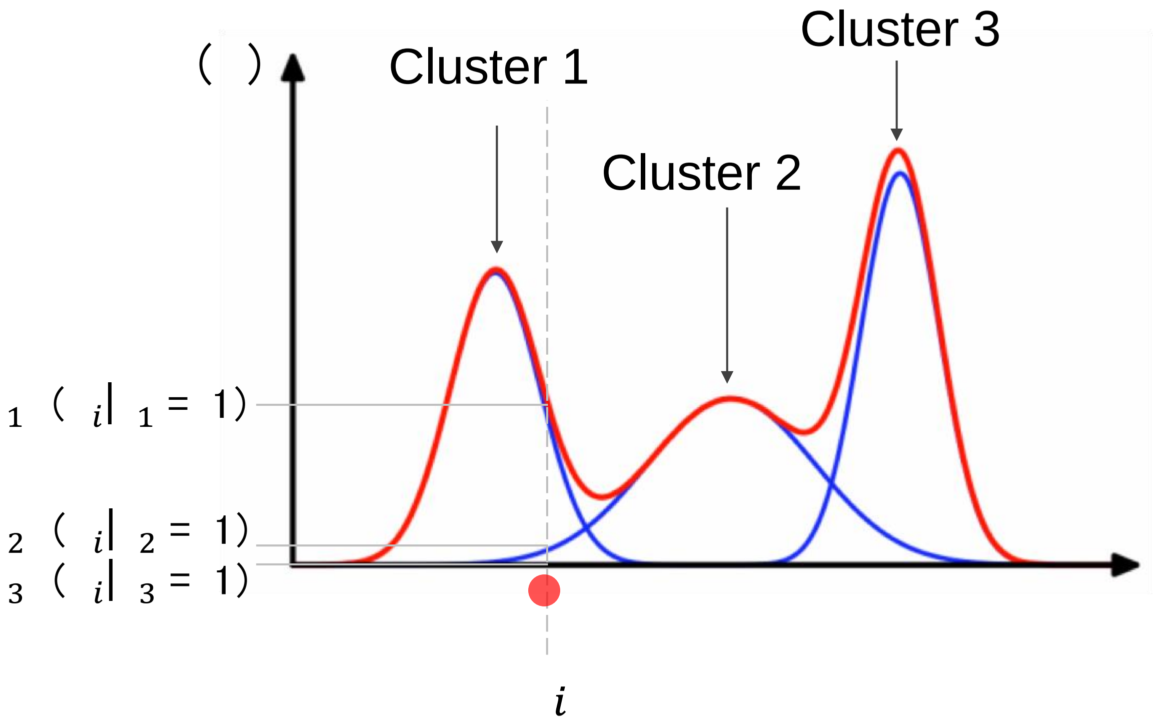
$$P(z_k = 1|x_i) = \frac{P(x_i|z_k = 1)P(z_k = 1)}{P(x_i)}$$

by Bayes' Rule

$$P(x_i) = \pi_1 P(x_i|z_1 = 1) + \pi_2 P(x_i|z_2 = 1) + \pi_3 P(x_i|z_3 = 1)$$

normalizes the probability, $P(z_k = 1|x_i)$, to add to one when summed over k

Posterior probabilities / “responsibilities”



Another interpretation of this quantity is what “fraction” of an observation is assigned to this cluster (“fuzzy” or ”soft” clustering)

$N(\boldsymbol{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ π_k

$$r(z_{ik}) \triangleq P(z_k = 1|x_i) = \frac{P(x_i|z_k = 1)P(z_k = 1)}{\sum_{k=1}^{K} P(x_i|z_k = 1)P(z_k = 1)}$$

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

Define $N_k = \sum_{i=1}^{N} r(z_{ik})$

Expected number of samples per cluster

Expectation Maximization for a GMM

Note: EM is a general technique for finding maximum likelihood solutions for models with latent variables

Goal: maximize the log likelihood of the data given the model parameters:

$$\ln P(\boldsymbol{X}|\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Sigma}) = \sum_{i=1}^{N} \ln\left[\sum_{k=1}^{K} \pi_k N(\boldsymbol{x}_i|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)\right]$$

0. Initialization

Initialize all the parameters
(often K-means is used for this purpose)

1. Expectation-step

Calculate the “responsibilities” based on the model parameters

$$r(z_{ik}) \triangleq P(z_k = 1|x_i)$$

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

2. Maximization-step

Use the “responsibilities” to update the model parameters to maximize the log likelihood

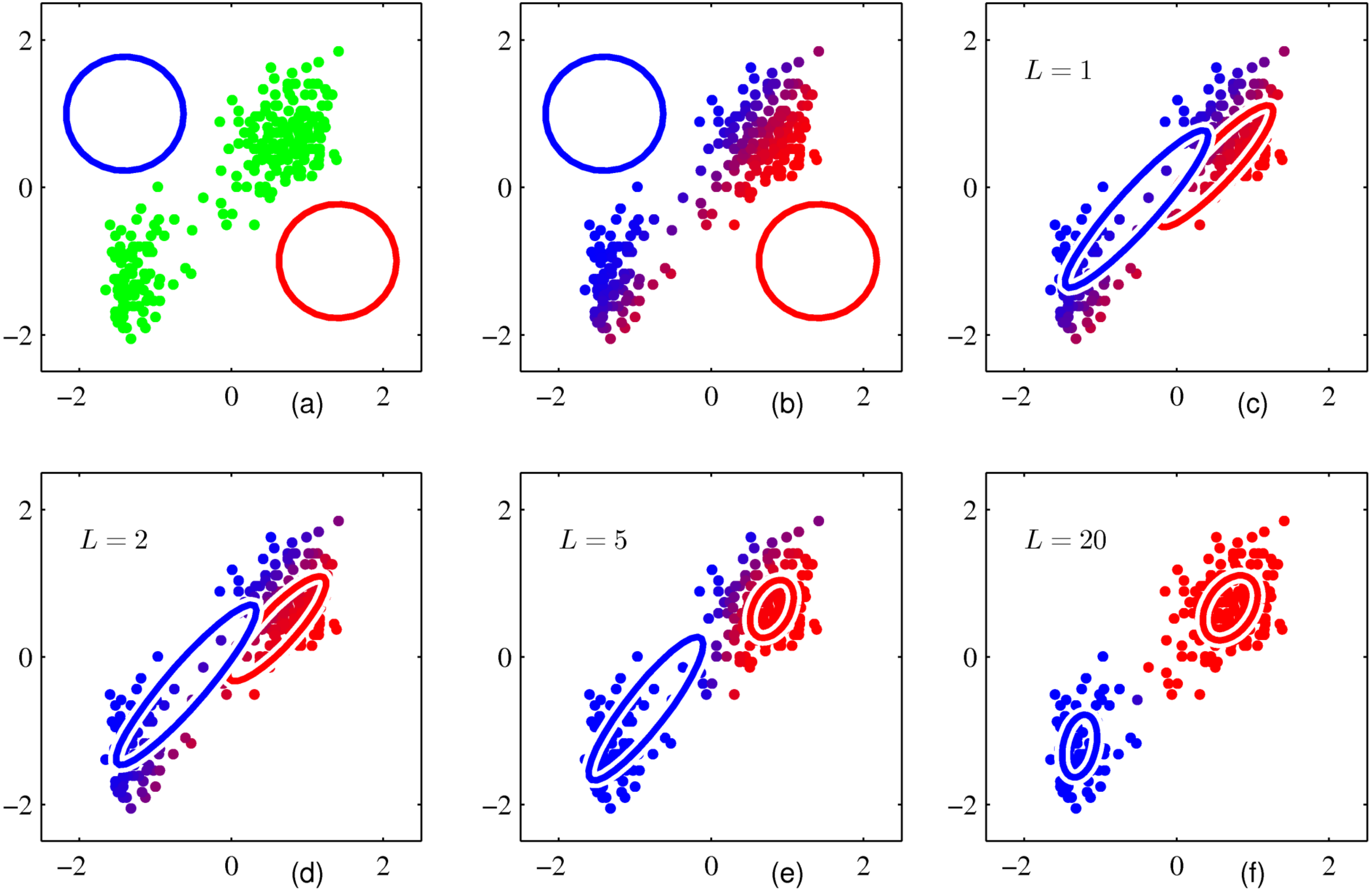
$$\boldsymbol{\mu}_k^{new} = \frac{1}{N_k}\sum_{i=1}^{N} r(z_{ik})\boldsymbol{x}_i$$

$$\boldsymbol{\Sigma}_k^{new} = \frac{1}{N_k}\sum_{i=1}^{N} r(z_{ik})(\boldsymbol{x}_i - \boldsymbol{\mu}_k^{new})(\boldsymbol{x}_i - \boldsymbol{\mu}_k^{new})^T$$

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

Where $N_k = \sum_{i=1}^{N} r(z_{ik})$

Expectation Maximization for GMM Example



L = number of EM cycles

Image from Bishop, Pattern Recognition, 2006

Gaussian Mixture Models

Generative models: model $P(\boldsymbol{X}|\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ are the model parameters

Very useful for density estimation

Produce hard or soft (fuzzy) clustering

When you restrict the covariance matrix to be diagonal and equal for all clusters, the GMM and K-means algorithm become the same