

# Density Estimation

## Data Science I

Cristóbal Donoso O.  
September 26, 2018

We have been working...

- Methods for exploring and quantifying structure in a multivariate distribution of points (Exploratory Data Analysis)
- Statistical description of the structure observed to fit models
- The space in the sample can be real physical or space spanned by the measured quantities (attributes)

# Density Estimation

- To infer the Probabilistic Density Function (PDF) from a sample of data
- Given a PDF estimated from a point data we can generate simulated distributions of a data and compare them against observations
- We can estimate the pdf for each subsample in a set of different labeled samples (Classification)

We've been estimating the underlying density of the data using parametric models from **density functions** and estimations of their parameters from **Frequentist and Bayesian perspectives**

# Density Estimation

- To infer the Probabilistic Density Function (PDF) from a sample of data
- Given a PDF estimated from a point data we can generate simulated distributions of a data and compare them against observations
- We can estimate the pdf for each subsample in a set of different labeled samples (Classification)

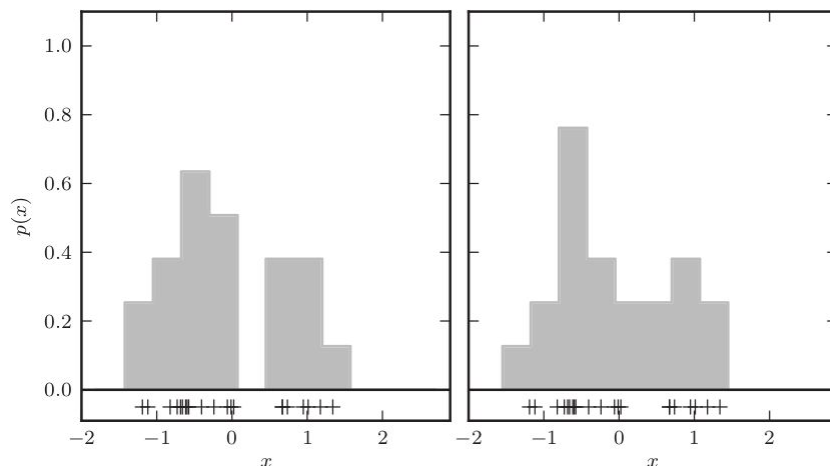
We've been using  
parametric perspectives

Now we'll estimate a **density nonparametrically**, that is, without specifying a specific functional model.

parametric perspectives

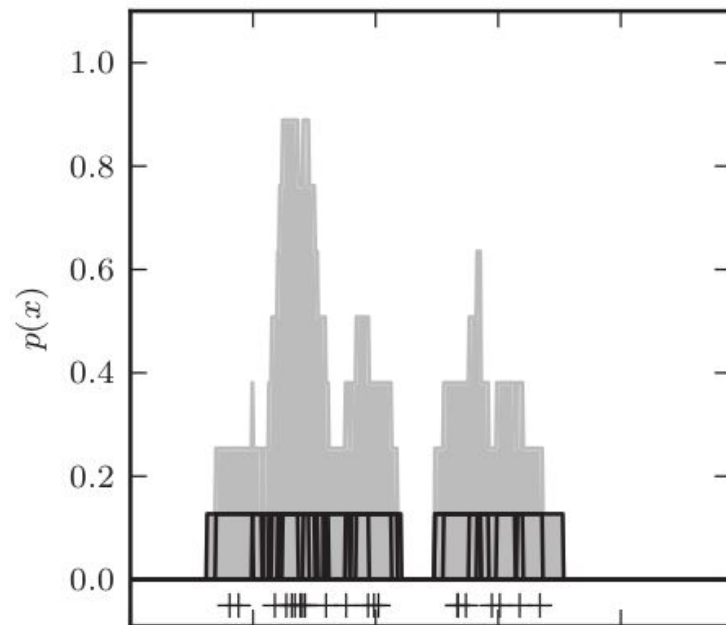
# Histograms

- A straightforward way to estimate densities is to use Histograms.
- **Each point contributes one unit to the height** of the histogram at the position of it's bin
- The exact locations of the bins **can make a difference**



# Improving the basic histogram

- Each point could have it's own bin
- Allow the bins to overlap
- Each point is **replaced by a box** of unit **height** and some predefined **width**
- The box is usually named **kernel**

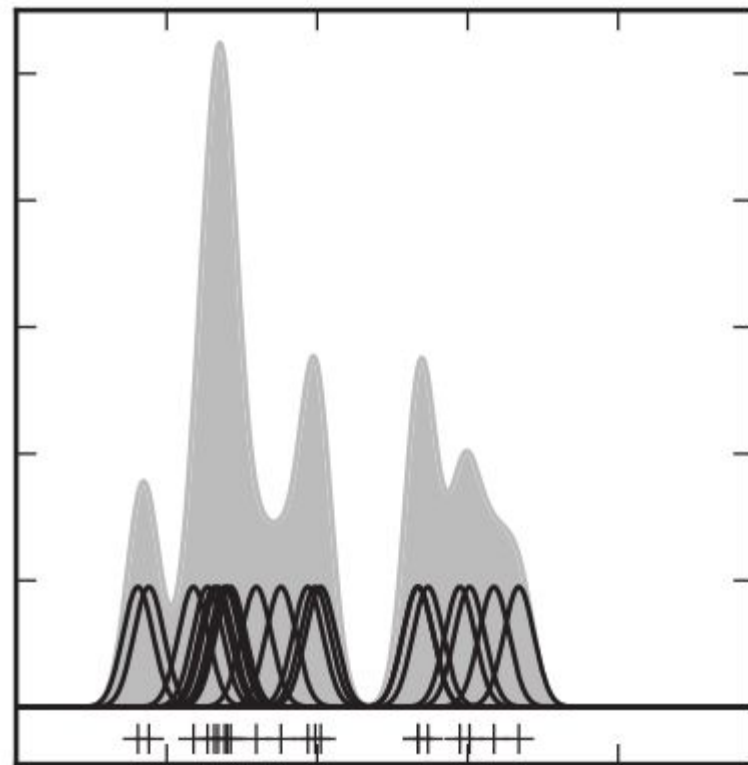


# Kernel Density Estimation

- The **rectangular kernel** does not lead to a **smooth distribution** and can display **suspicious spikes**
- We can use a **Gaussian Kernel** among others

$$\hat{f}_N(x) = \frac{1}{Nh^D} \sum_{i=1}^N K\left(\frac{d(x, x_i)}{h}\right)$$

- The function is estimated as a weighted mean of all points, where the weights are specified via  $K(u)$

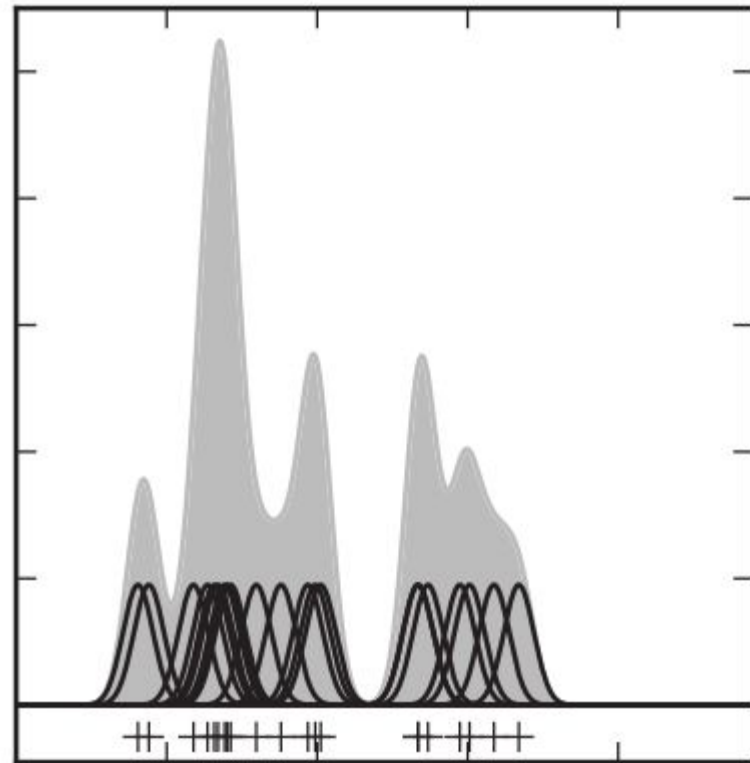


# Kernel Density Estimation

- We can use a **Gaussian Kernel** among others

$$\hat{f}_N(x) = \frac{1}{Nh^D} \sum_{i=1}^N K\left(\frac{d(x, x_i)}{h}\right)$$

- The function is estimated as a weighted mean of all points, where the weights are specified via  $K(u)$
- $K(u)$  is constrained by
  - $(K(u) \geq 0)$ , (positives values)
  - $\int K(u) du = 1$  (normalized)
  - $\int uK(u) du = 0$  (mean)
  - $\sigma_K^2 = \int u^2 K(u) du \geq 0$ , (variance)





# Types of Kernels

## Gaussian Kernel

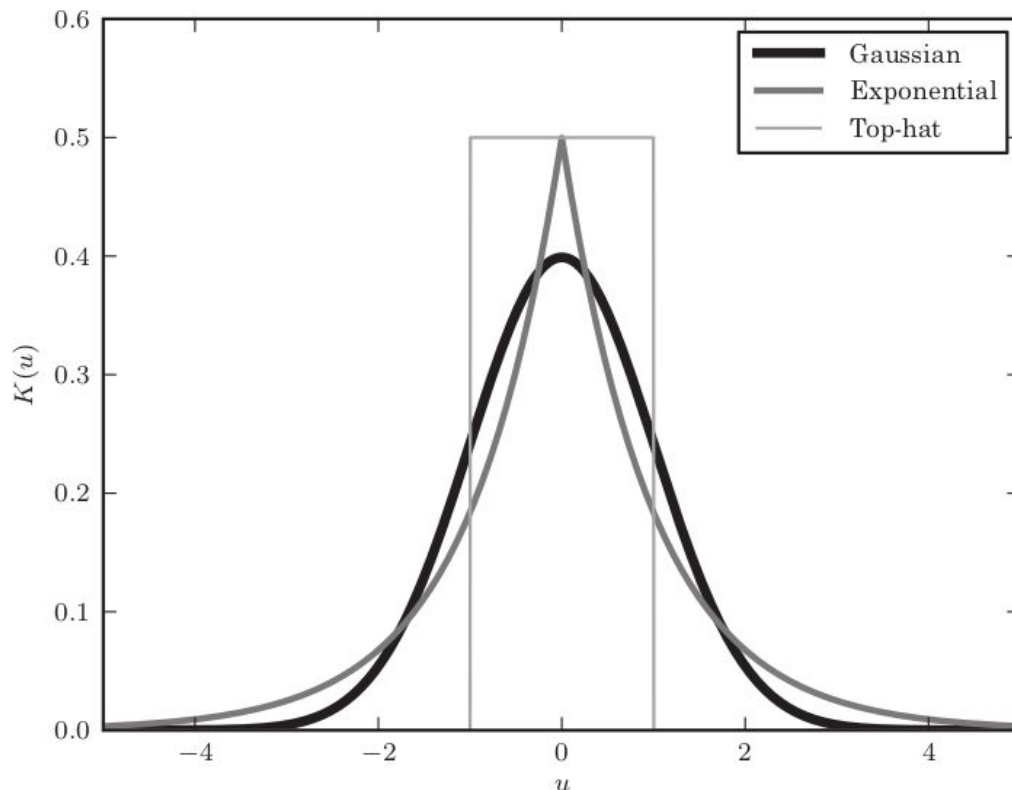
$$K(u) = \frac{1}{(2\pi)^{D/2}} e^{-u^2/2}$$

## Top-hat Kernel

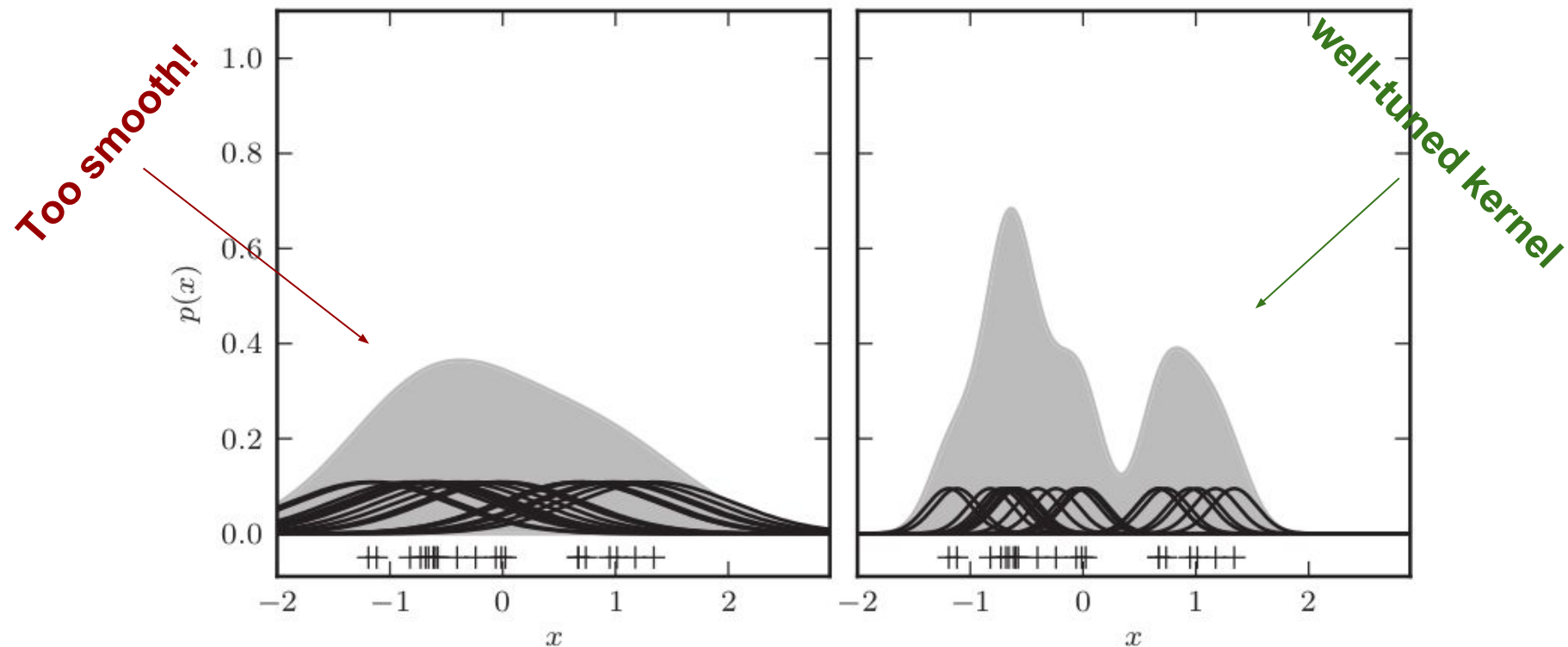
$$K(u) = \begin{cases} \frac{1}{V_D(1)} & \text{if } u \leq 1, \\ 0 & \text{if } u > 1, \end{cases}$$

## Exponential Kernel

$$K(u) = \frac{1}{D! V_D(1)} e^{-|u|},$$



The amplitude (or bandwidth) of the kernel could generate densities that are too smooth and deleting important information from peaks



# Selecting the bandwidth

1. Assuming an underlying Gaussian distribution: Scott's rule:

$$h = 3.5\sigma/N^{1/3}$$

2. Non-Gaussian distributions:

$$h = 2(q_{75} - q_{25})/N^{1/3}$$

3. Cross-validation: Given **n** data points in the original sample then, **n-p samples are used to train the model** and **p points are used as the validation set**. This is repeated for all combinations in which original sample can be separated this way, and then the error is averaged for all trials, to give overall effectiveness (\*).

# Cross-validation to select the bandwidth

An alternative to likelihood cross-validation is to use the mean integrated square error

$$\int (\hat{f}_h - f)^2 = \int \hat{f}_h^2 - 2 \int \hat{f}_h f + \int f^2.$$

As before, the first term can be obtained analytically, and the last term does not depend on  $h$ . For the second term we have expectation value

$$\mathbb{E} \left[ \int \hat{f}_h(x) f(x) dx \right] = \mathbb{E} \left[ \frac{1}{N} \sum_{i=1}^N \hat{f}_{h,-i}(x) \right].$$

This motivates the L2 cross-validation score:

$$CV_{L_2}(h) = \int \hat{f}_h^2 - 2 \frac{1}{N} \sum_{i=1}^N \hat{f}_{h,-i}(x_i)$$

# Cross-validation to select the bandwidth

An alternative to likelihood cross-validation is to use the mean integrated square error

$$\int (\hat{f}_h - f)^2 = \int \hat{f}_h^2 - 2 \int \hat{f}_h f + \int f^2.$$

As before, the first term can be obtained analytically, and the last term does not depend on  $h$ . For the second term we have expectation value

$$\mathbb{E} \left[ \int \hat{f}_h(x) f(x) dx \right] = \mathbb{E} \left[ \frac{1}{N} \sum_{i=1}^N \hat{f}_{h,-i}(x_i) \right].$$

This motivates the L2 cross-validation score:

$$CV_{L_2}(h) = \int \hat{f}_h^2 - 2 \frac{1}{N} \sum_{i=1}^N \hat{f}_{h,-i}(x_i) \xrightarrow{\text{Optimal } h} \operatorname{argmax}_h CV_{L_2}(h)$$

# Cross-validation to select the bandwidth

This motivates the L 2 cross-validation score:

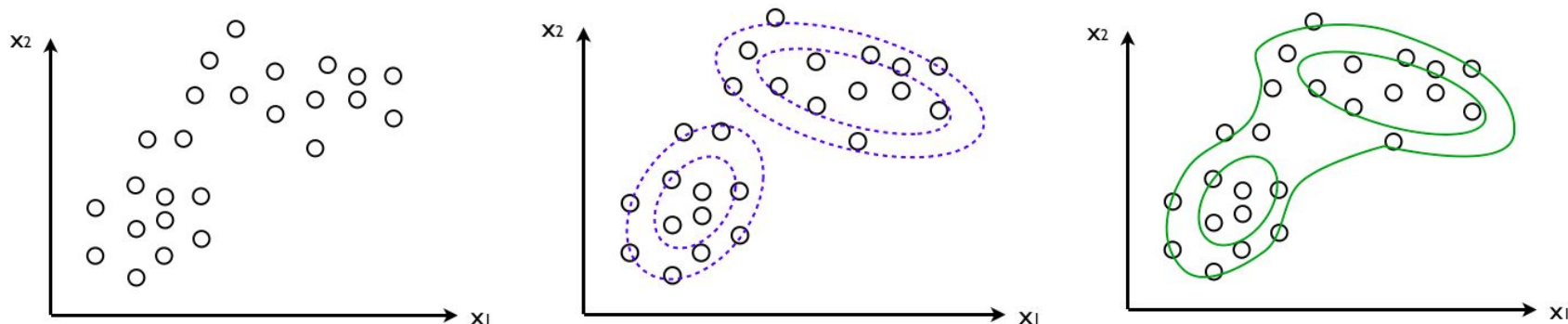
$$CV_{L_2}(h) = \int \hat{f}_h^2 - 2 \frac{1}{N} \sum_{i=1}^N \hat{f}_{h,-i}(x_i)$$

When doing density estimation we are actually estimating the pdf of our parameters. In that sense, we can use our test set (leave-one-out) to directly estimate the likelihood of that set of points.

$$CV_l(h) = \frac{1}{N} \sum_{i=1}^N \log \hat{f}_{h,-i}(x_i), \quad \xrightarrow{\text{Optimal } h} \operatorname{argmax}_h CV_l(h)$$

# Parametric Density Estimation

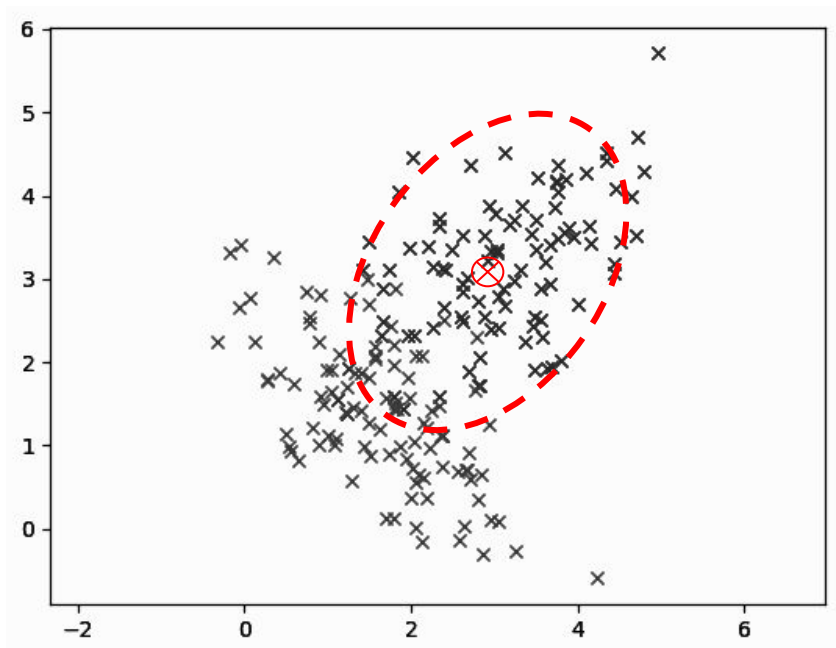
- We can use fewer kernels, and fit for the kernel locations as well as the widths



- The contribution of the full set of clusters at any given point
- The location and size of each component is assumed to reflect some underlying property of the data

# Parametric Density Estimation: Gaussian Approach

- We can use Gaussians to model the behavior of the data

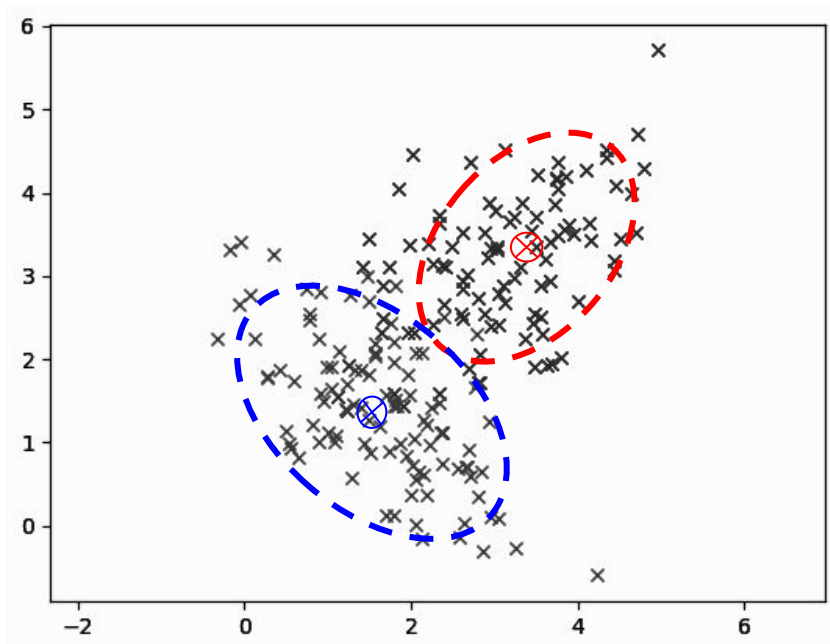


- We assume that exists some **latent variable** which generate the data
- This **latent variable** follows:  
$$p(\mathbf{x}|z) = \mathcal{N}(\mathbf{x}|\mu, C)$$
- We could not capture distributions of minority data groups



# Parametric Density Estimation: Gaussian Approach

- If one gaussian doesn't work, we could use several of them



- We assume that exists a group of latent variables  $\mathbf{z} \in \{e_1, \dots, e_K\}$ , where  $e$  is a categorical vector which is used to select a particular variable  $z$
- The density of each data point
$$p(\mathbf{x} | \mathbf{z} = e_k) = \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$
- The marginal distribution of  $\mathbf{x}$  is then obtained by summing the joint distribution over all possible states of  $\mathbf{z}$

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

# Gaussian Mixture Model

GMM models the underlying density (pdf) of points as a sum of Gaussians.

The density of the points is given by

$$\rho(\mathbf{x}) = Np(\mathbf{x}) = N \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \xrightarrow{\text{log-likelihood}} \sum_{n=1}^N \ln \left\{ \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}.$$

where

- $K$  is the number of Gaussian
- $\boldsymbol{\mu}_k$  means or locations of the Gaussians
- $\boldsymbol{\Sigma}_k$  covariances of the Gaussians
- $\boldsymbol{\pi}$  mixing coefficient for each Gaussian, restricted  $\sum_{k=1}^K \pi_k = 1.$

# Gaussian Mixture Model: Optimizing parameters

- **MLE** is more complicated for optimization in multiple dimensions
- Alternatively, we can use the **expectation maximization** method, where the model depends on unobserved latent variables:
- The goal of this technique is to assume some initial mean  $\mu$ , standard deviation  $\Sigma$  and normalization factors  $\alpha$  and **iteratively improve the estimate**
  - **E-step:** calculate the expected joint (data + latent variable) log-likelihood at iteration  $t$ .
  - **M-step:** maximize such likelihood in terms of the model parameters.  $\theta = (\alpha, \{\mu_k\}, \{\Sigma_k\})$

# EM for Gaussian Mixture Model: Expectation step

Making the derivatives of  $\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$  w.r.t each means  $\boldsymbol{\mu}_k$  to zero

$$0 = - \sum_{n=1}^N \underbrace{\frac{\pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_j \pi_j \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}}_{\gamma(z_{nk})} \boldsymbol{\Sigma}_k (\mathbf{x}_n - \boldsymbol{\mu}_k)$$

where  $\gamma(z_{nk})$  is the **responsibility**. It can be viewed as the posterior probability once we have observed  $\mathbf{x}$ .

In this step **we evaluate the responsibilities using the current parameter values**

# EM for Gaussian Mixture Model: Maximization Step

Re-estimate the parameters using the **current responsibilities**

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

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

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

where

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

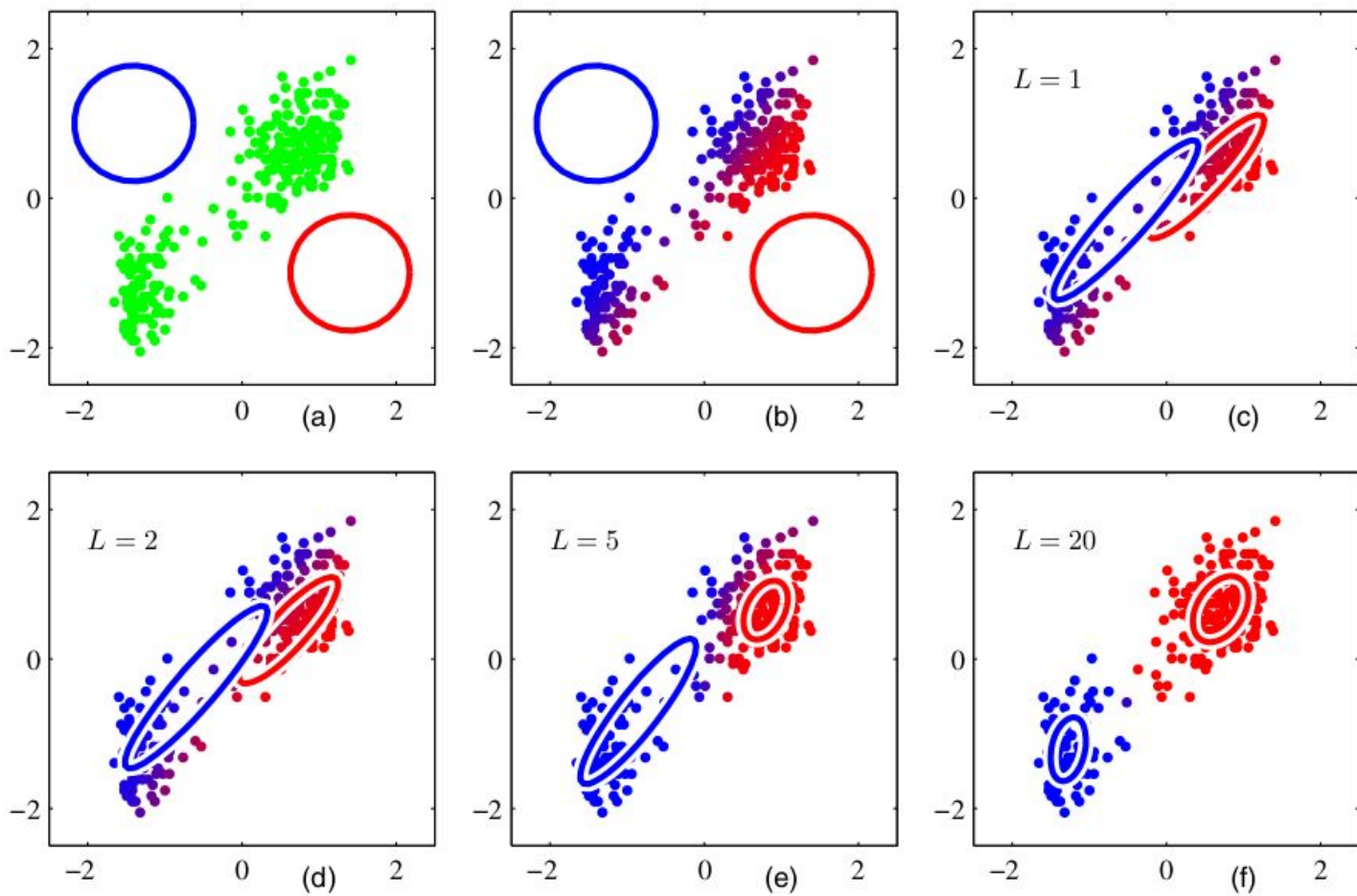
# EM for Gaussian Mixture Model: Evaluating convergence

In order to stop the iterative process we shall define a convergence criterion. We have to:

1. evaluate the log-likelihood, using the maximized parameters and check the convergence.

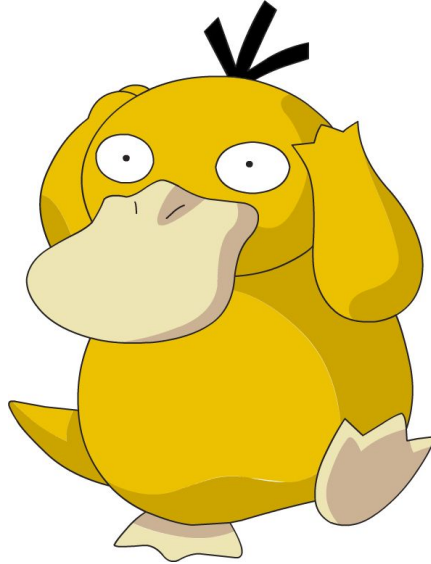
$$\ln p(\mathbf{X}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = \sum_{n=1}^N \ln \left\{ \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$

2. if the statement is not satisfied, go back and repeat the EM steps



Ok, we've obtained a fitted beautiful Gaussian from data, but

**How many Gaussians should we use?**





# Model Selection

As we have seen before, in order to compare different models, the preferred technique is cross-validation. Nevertheless, there are classical methods that are easy to use and often effective for simple models. Two of those are

1. the Akaike Information Criterion (AIC)
2. the Bayesian Information Criterion (BIC).

# Model Selection: Akaike Information Criterion (AIC)

Akaike Information Criterion penalizes the model based on its complexity. It is defined a

$$AIC(\theta) = -2 \log p(X|\theta) + 2k$$

where,

- $\theta$  is the model
- $X$  is the input data
- $\log p(X|\theta)$  is the log likelihood
- $k$  is the number of parameters

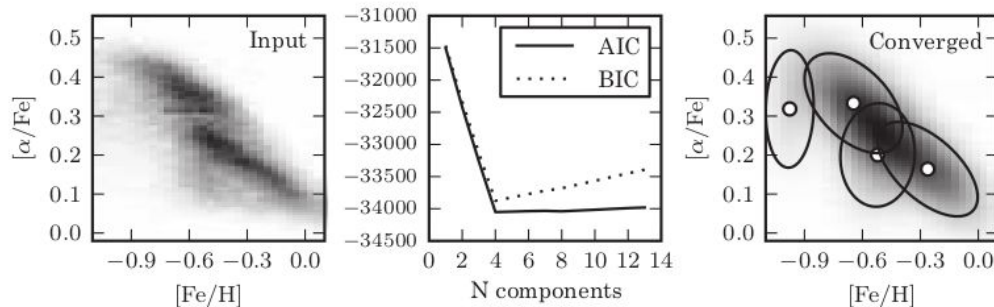


# Model Selection: Bayesian Information Criterion (BIC).

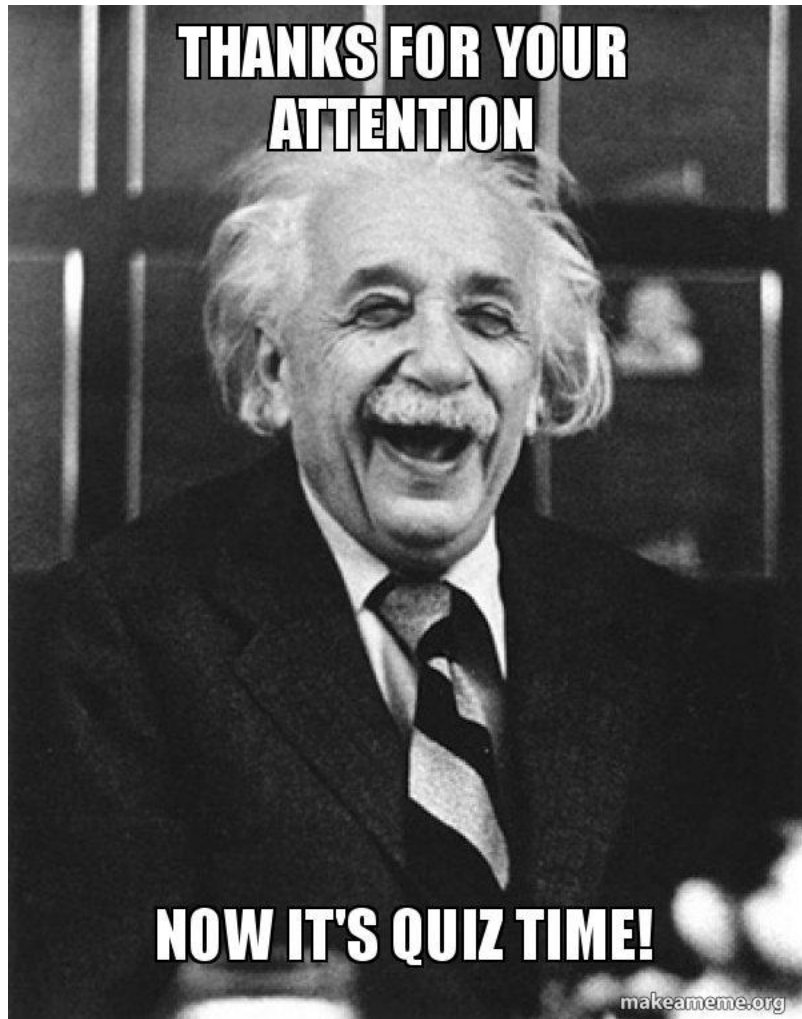
BIC gives us a way to choose between different models with different numbers of parameters by **selecting the one which gives us the lowest BIC score**.

$$BIC(\theta) = -2\log p(X|\theta) + k \log(n)$$

It is only valid for sample size  $n$  much larger than the number  $k$  of parameters in the model.



**THANKS FOR YOUR  
ATTENTION**



**NOW IT'S QUIZ TIME!**