Music Machine Learning

VIII – Gaussian Mixture Models

Master ATIAM - Informatique

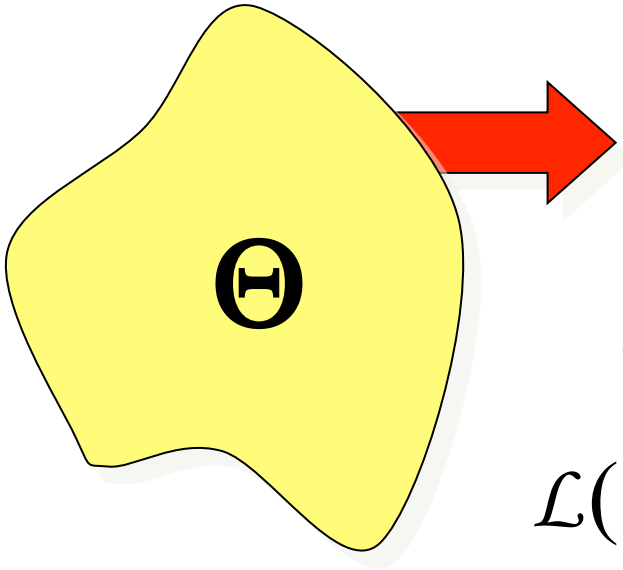
Philippe Esling (esling@ircam.fr)

Maître de conférences – UPMC

Equipe représentations musicales (IRCAM, Paris)

UPMC SORBONNE UNIVERSITÉS master ATIAM ircam Centre Pompidou

Maximum Likelihood



Set of parameters
(for a model)

The data we observe

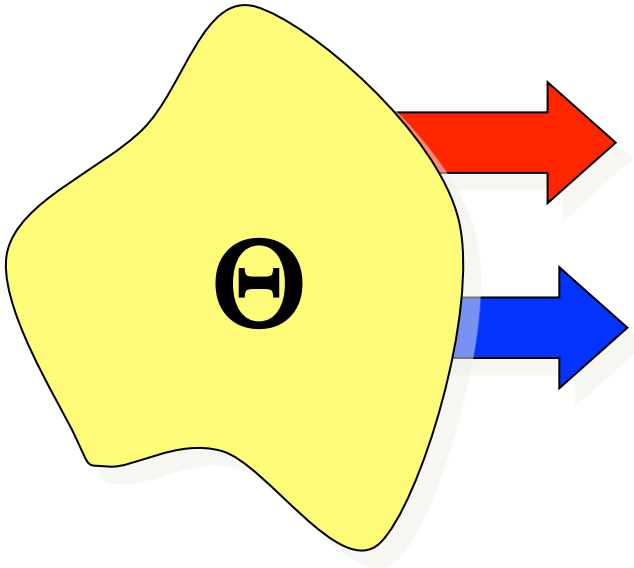
$$\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$$

« How **likely** are the parameters »

$$\mathcal{L}(\Theta \mid \mathbf{X}) = p(\mathbf{X}|\Theta) = \prod_{i=1}^{N} p(\mathbf{x}_i|\Theta)$$

$$\Theta^* = \arg\max_{\Theta} \mathcal{L}(\Theta \mid \mathbf{X})$$

Latent variables



$$\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$$

$$\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N\}$$

Data is incomplete

Complete Data $\mathbf{Z} = (\mathbf{X}, \mathbf{Y})$

$$\mathcal{L}(\Theta \mid \mathbf{Z}) = p(\mathbf{Z}|\Theta) = p(\mathbf{X}, \mathbf{Y}|\Theta)$$

$$= p(\mathbf{Y} \mid \mathbf{X}, \Theta)\, p(\mathbf{X}|\Theta)$$

Expectation-Maximization

$$\mathcal{L}(\Theta \mid \mathbf{Z}) = p(\mathbf{Y} \mid \mathbf{X}, \Theta)\, p(\mathbf{X} \mid \Theta)$$

A function of random variable **Y**.

If we are given **Θ**,

A function of latent variable **Y** and parameter **Θ**

The result is in term of random variable **Y**.

A function of parameter **Θ**

Computable

Expectation-Maximization

- Start by devising your model
- Initially **guess** the parameters of the model!
  - Educated guess is best, but random can work

- **Expectation step:** Use current parameters (and observations) to reconstruct hidden structure
- **Maximization step:** Use that hidden structure (and observations) to reestimate parameters

Repeat until convergence!

Expectation step

Let $\boldsymbol{\Theta}^{(i-1)}$ be the parameter vector obtained at the $(i\text{-}1)^{th}$ step.

Define the Q-function =
Conditional Expectation of log likelihood of complete data

$$Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(i-1)}) = E[\log \mathcal{L}(\boldsymbol{\Theta} \mid \mathbf{Z}) \mid \mathbf{X}, \boldsymbol{\Theta}^{(i-1)}]$$

$$= \begin{cases} \int_{\mathbf{y} \in \mathbf{Y}} \log p(\mathbf{X}, \mathbf{y} \mid \boldsymbol{\Theta}) \cdot p(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\Theta}^{(i-1)}) \mathbf{dy} & \text{continuous} \\ \sum_{\mathbf{y} \in \mathbf{Y}} \log p(\mathbf{X}, \mathbf{y} \mid \boldsymbol{\Theta}) \cdot p(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\Theta}^{(i-1)}) & \text{discrete} \end{cases}$$

Maximization step

$$\boldsymbol{\Theta}^{(i)} = \arg\max_{\boldsymbol{\Theta}} Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(i-1)})$$

$$Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(i-1)}) = E[\log \mathcal{L}(\boldsymbol{\Theta} \mid \mathbf{Z}) \mid \mathbf{X}, \boldsymbol{\Theta}^{(i-1)}]$$

$$= \begin{cases} \int_{\mathbf{y} \in \mathbf{Y}} \log p(\mathbf{X}, \mathbf{y} \mid \boldsymbol{\Theta}) \cdot p(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\Theta}^{(i-1)}) \mathbf{dy} & \text{continuous} \\ \sum_{\mathbf{y} \in \mathbf{Y}} \log p(\mathbf{X}, \mathbf{y} \mid \boldsymbol{\Theta}) \cdot p(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\Theta}^{(i-1)}) & \text{discrete} \end{cases}$$

Expectation-Maximization

- as usual, we start from an iid sample $D = \{x_1, \ldots, x_N\}$
- goal is to find parameters Ψ^* that maximize likelihood with respect to D

$$\begin{aligned} \Psi^\star &= \arg\max_{\Psi} P_{\mathbf{X}}(\mathcal{D}; \Psi) \\ &= \arg\max_{\Psi} \int P_{\mathbf{X}|Z}(\mathcal{D}|z; \Psi) P_Z(z; \Psi) dz \end{aligned}$$

- the set

$$D_c = \{(x_1, z_1), \ldots, (x_N, z_N)\}$$

  is called the complete data

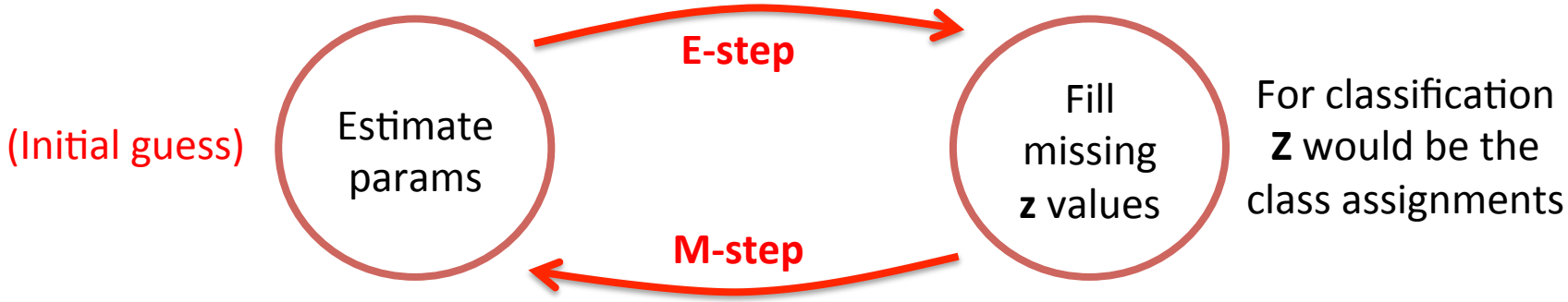
- the set

$$D = \{x_1, \ldots, x_N\}$$

  is called the incomplete data

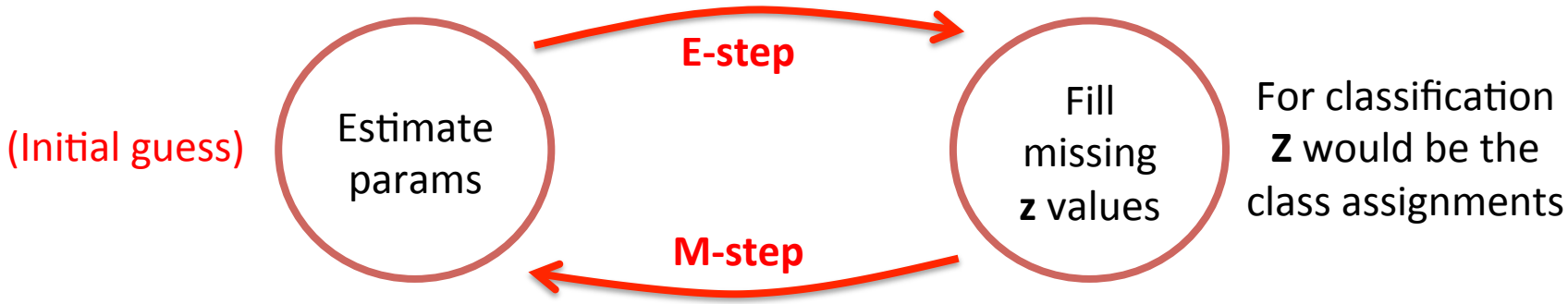
Basics of EM

- the basic idea is quite simple
  1. start with an initial parameter estimate $\Psi^{(0)}$
  2. **E-step:** given current parameters $\Psi^{(i)}$ and observations in D, "guess" what the values of the z_j are
  3. **M-step:** with the new z_j, we have a complete data problem, solve this problem for the parameters, i.e. compute $\Psi^{(i+1)}$
  4. go to 2.
- this can be summarized as



Basics of EM

- the basic idea is quite simple
  1. start with an initial parameter estimate $\Psi^{(0)}$
  2. **E-step:** given current parameters $\Psi^{(i)}$ and observations in D, "guess" what the values of the z_j are
  3. **M-step:** with the new z_j, we have a complete data problem, solve this problem for the parameters, i.e. compute $\Psi^{(i+1)}$
  4. go to 2.
- this can be summarized as **Ring any bell ?**



K-Means

- when covariances are identity and priors uniform
- C-step:
  - $z_l = \arg\min_c ||\mathbf{x}_l - \mu_c^{(i)}||^2, \quad l \in \{1, \ldots, n\}$
  - split the training set according to the labels z_i

    $D^1 = \{x_i | z_i = 1\}, \quad D^2 = \{x_i | z_i = 2\}, \quad \ldots \quad , D^C = \{x_i | z_i = C\}$
- M-step:
  - $\mu_c^{(i+1)} = \frac{1}{|\{\mathbf{x}_i \in \mathcal{D}^c\}|} \sum_{i|\mathbf{x}_i \in \mathcal{D}^c} \mathbf{x}_i$
- this is the K-means algorithm, aka generalized Loyd algorithm, aka LBG algorithm in the vector quantization literature:
  - "assign points to the closest mean; recompute the means"

Expectation-Maximization (summary)

- EM is typically used to compute maximum likelihood estimates given incomplete samples.

- The EM algorithm estimates the parameters of a model iteratively.
  - Starting from some initial guess, each iteration consists of
    - an E step (Expectation step)
    - an M step (Maximization step)

Expectation-Maximization

- E-step:
  - given estimates $\Psi^{(n)} = \{\Psi^{(n)}_1, \ldots, \Psi^{(n)}_C\}$
  - compute expected log-likelihood of complete data

$$Q(\Psi; \Psi^{(n)}) = E_{Z|\mathbf{X};\Psi^{(n)}} \left[\log P_{\mathbf{X},Z}(\mathcal{D}, \{z_1, \ldots, z_N\}; \Psi) | \mathcal{D} \right]$$

- M-step:
  - find parameter set that maximizes this expected log-likelihood

$$\Psi^{(n+1)} = \arg\max_{\Psi} Q(\Psi; \Psi^{(n)})$$

- let's make this more concrete by looking at the mixture case

Expectation-Maximization

How to **derive an EM algorithm**

1. Write down the **likelihood of the complete data**
2. E-step: **write down the Q function**
   (its expectation given the observed data)
3. M-step: solve the maximization

(**deriving a closed-form solution if there is one**)

Q function

- is defined as

$$Q(\Psi; \Psi^{(n)}) = E_{Z|\mathbf{X};\Psi^{(n)}} \left[\log P_{\mathbf{X},Z}(\mathcal{D}, \{z_1, \ldots, z_N\}; \Psi) | \mathcal{D} \right]$$

- and is a bit tricky:
  - it is the expected value of likelihood with respect to complete data (joint X and Z)
  - given that we observed incomplete data (X=$\mathcal{D}$)
  - note that the likelihood is a function of Ψ (the parameters that we want to determine)
  - but to compute the expected value we need to use the parameter values from the previous iteration (because we need a distribution for Z|X)

Expectation-Maximization

How to **derive an EM algorithm**

1. Write down the **likelihood of the complete data**
2. E-step: **write down the Q function**
   (its expectation given the observed data)
3. M-step: solve the maximization

(**deriving a closed-form solution if there is one**)

Important E-step advice

- **Do not compute terms that you do not need**
- **In the end we only care about the parameters**
- Terms of Q that do not depend on the parameters are **useless**

Expectation-Maximization

- to derive an EM algorithm you need to do the following
  1. write down the likelihood of the COMPLETE data

$$\log P_{\mathbf{X},Z}(\mathcal{D}, \{\mathbf{z}_1, \ldots, \mathbf{z}_N\}; \Psi) = \sum_{i,j} z_{ij} \log \left[P_{\mathbf{X}|\mathbf{Z}}(\mathbf{x}_i|\mathbf{e}_j, \Psi)\pi_j \right]$$

  2. E-step: write down the Q function, i.e. its expectation given the observed data

$$h_{ij} = P_{\mathbf{Z}|\mathbf{X}}(\mathbf{e}_j|\mathbf{x}_i; \Psi^{(n)})$$

$$Q(\Psi; \Psi^{(n)}) = \sum_{i,j} h_{ij} \log \left[P_{\mathbf{X}|\mathbf{Z}}(\mathbf{x}_i|\mathbf{e}_j, \Psi)\pi_j \right]$$

  3. M-step: solve the maximization, deriving a closed-form solution if there is one

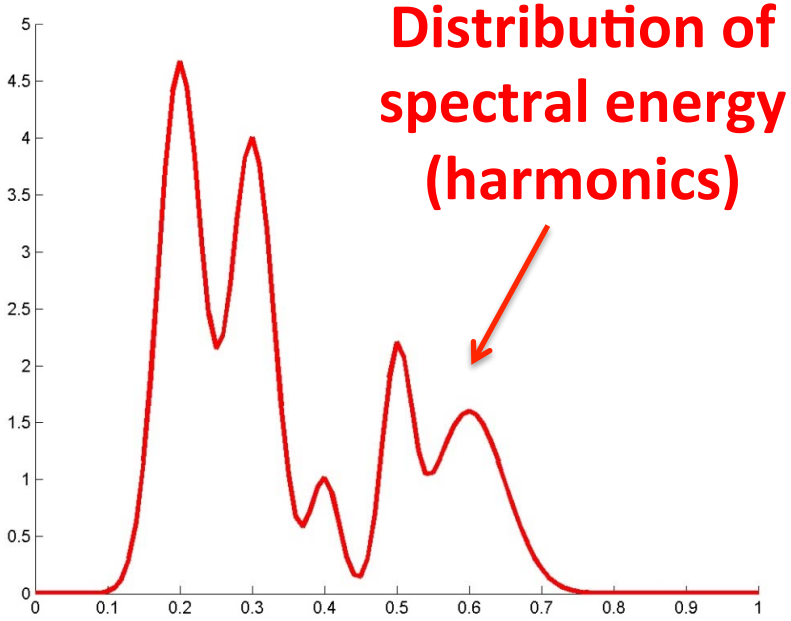
$$\Psi^{(n+1)} = \arg\max_{\Psi} \sum_{ij} h_{ij} \log \left[P_{\mathbf{X}|\mathbf{Z}}(\mathbf{x}_i|\mathbf{e}_j, \Psi)\pi_j \right]$$

Mixture density estimates

- back to BDR-based classifiers

  Consider the problem of instrument classification

- summary:
  - Estimate instrument type (brass, string, percu) from audio
  - Measure some audio feature
  - estimate pdf
  - use BDR

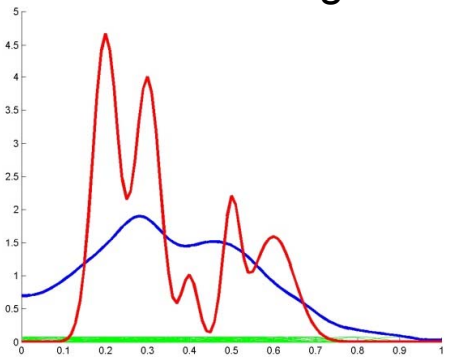


- clearly this is not Gaussian
- possible solution: use a kernel-based model

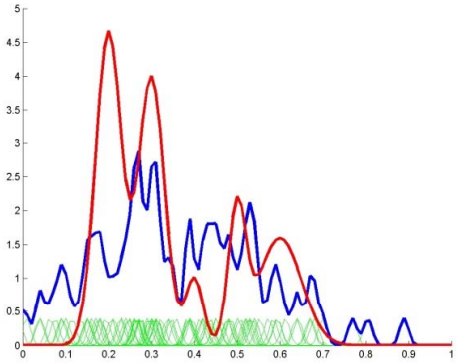
Mixture density estimates

- simple learning procedure
  - measure audio feature X
  - place a Gaussian on top of each measurement
- can be overkill
  - spending all degrees of freedom (# of training points) just to get the Gaussian means
  - cannot use the data to determine variances
- handpicking of bandwidth can lead to too much bias or variance

bandwidth too large: bias

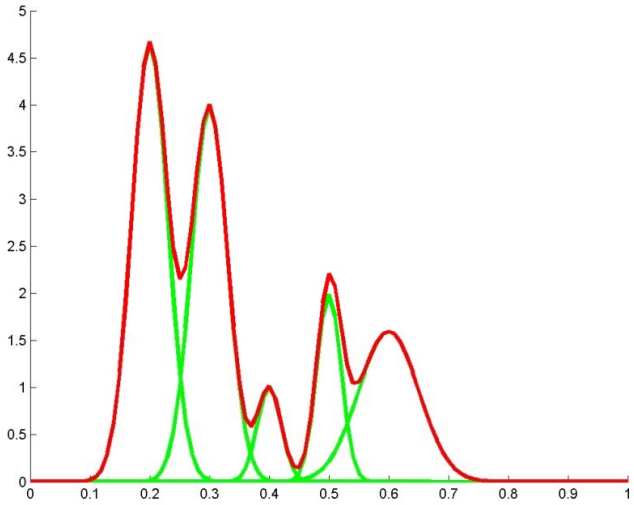


bandwidth too small: variance



Mixture density estimates

- it looks like we could do better by just picking the right # of Gaussians



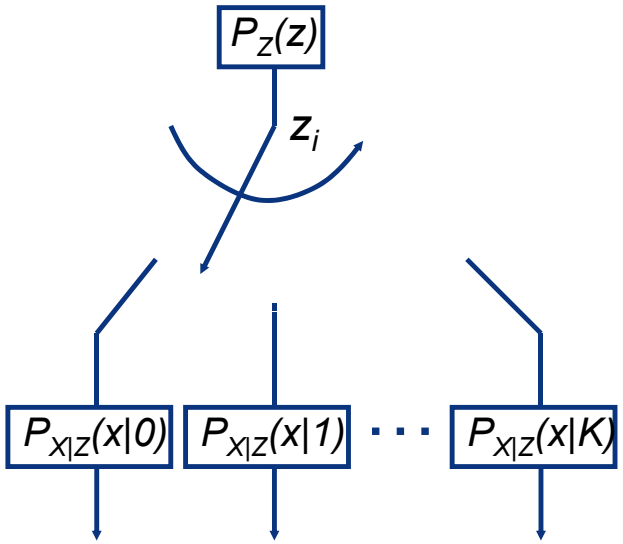
- this is indeed a good model:
  - density is multimodal because there is a hidden variable Z
    - Z can determine the type of intermediate musical instruments (for example)

$$Z \in \{Violin, Piano, Saxophone, Flute, Drum\}$$

    - Note that this is different from Y which is the instrument type (brass, string, percussion)
    - For a given instrument type, the density is approximate Gaussian here.
    - The density is a *mixture of Gaussians*

Mixture density estimates

- two types of random variables
  - Z – hidden state variable
  - X – observed variable
- observations sampled with a two-step procedure
  - a state (class) is sampled from the distribution of the hidden variable



$$P_Z(z) \rightarrow z_i$$

  - an observation is drawn from the class conditional density for the selected state

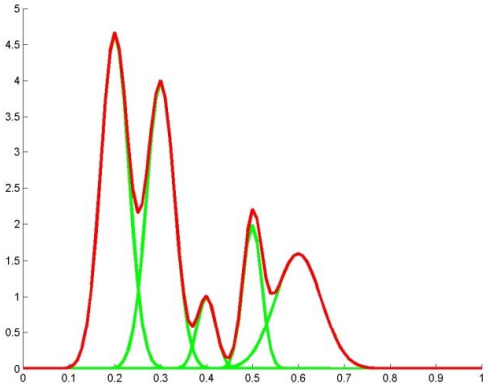
$$P_{X|Z}(x|z_i) \rightarrow x_i$$

Mixture density estimates

- the sample consists of pairs (x_i, z_i)

  $D = \{(x_1, z_1), \ldots, (x_n, z_n)\}$

  but we never get to see the z_i



- the pdf of the observed data is

$$P_{\mathbf{X}}(\mathbf{x}) = \sum_{c=1}^{C} P_{\mathbf{X}|Z}(\mathbf{x}|c) P_Z(c)$$

$$= \sum_{c=1}^{C} P_{\mathbf{X}|Z}(\mathbf{x}|c) \pi_c$$

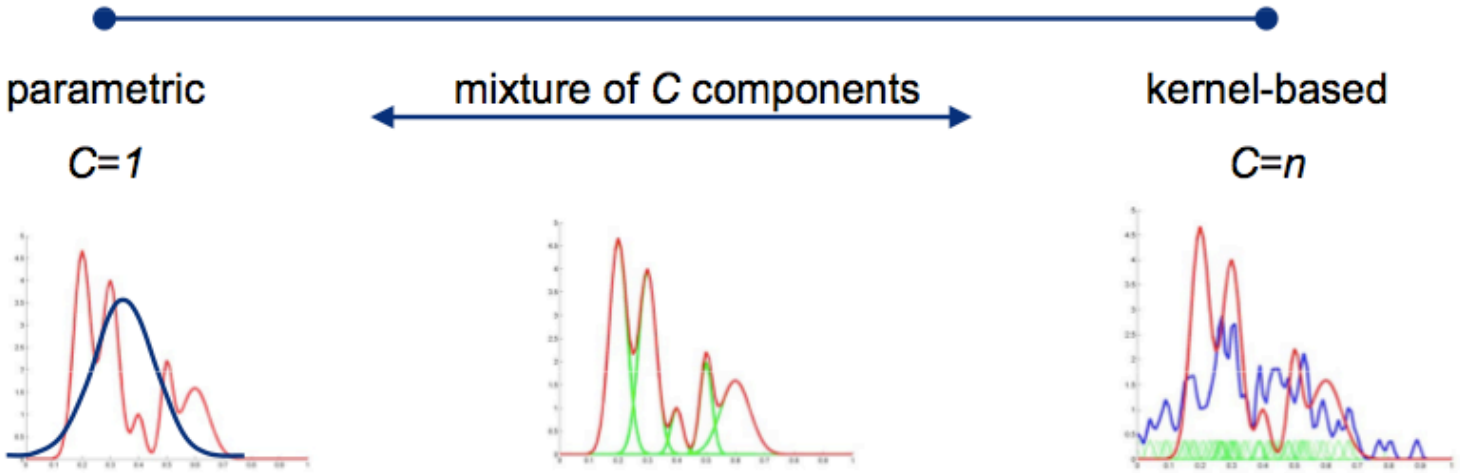
# of mixture components

component "weight"

c^{th} "mixture component"

Mixture density estimates

- A parametric model is a mixture with *one* component
  - The weight is *one*
  - The mixture density is the parametric density itself!
  - More degrees of freedom in mixture => less bias
- A mixture density is like a kernel density less components
  - less components => less learning parameters, less variance
- Mixture is a compromise between these two extremes:



Mixture problems

- main disadvantage is learning complexity
- non-parametric estimates
  - simple: store the samples (NN); place a kernel on top of each point (kernel-based)
- parametric estimates
  - small amount of work: if ML equations have closed-form
  - substantial amount of work: otherwise (numerical solution)
- mixtures:
  - there is usually no closed-form solution
  - always need to resort to numerical procedures
- standard tool is the expectation-maximization (EM)

EM for Gaussian mixtures

- Initialize **k** cluster centers
- Iterate between two steps
  - **E**xpectation step: assign points to clusters

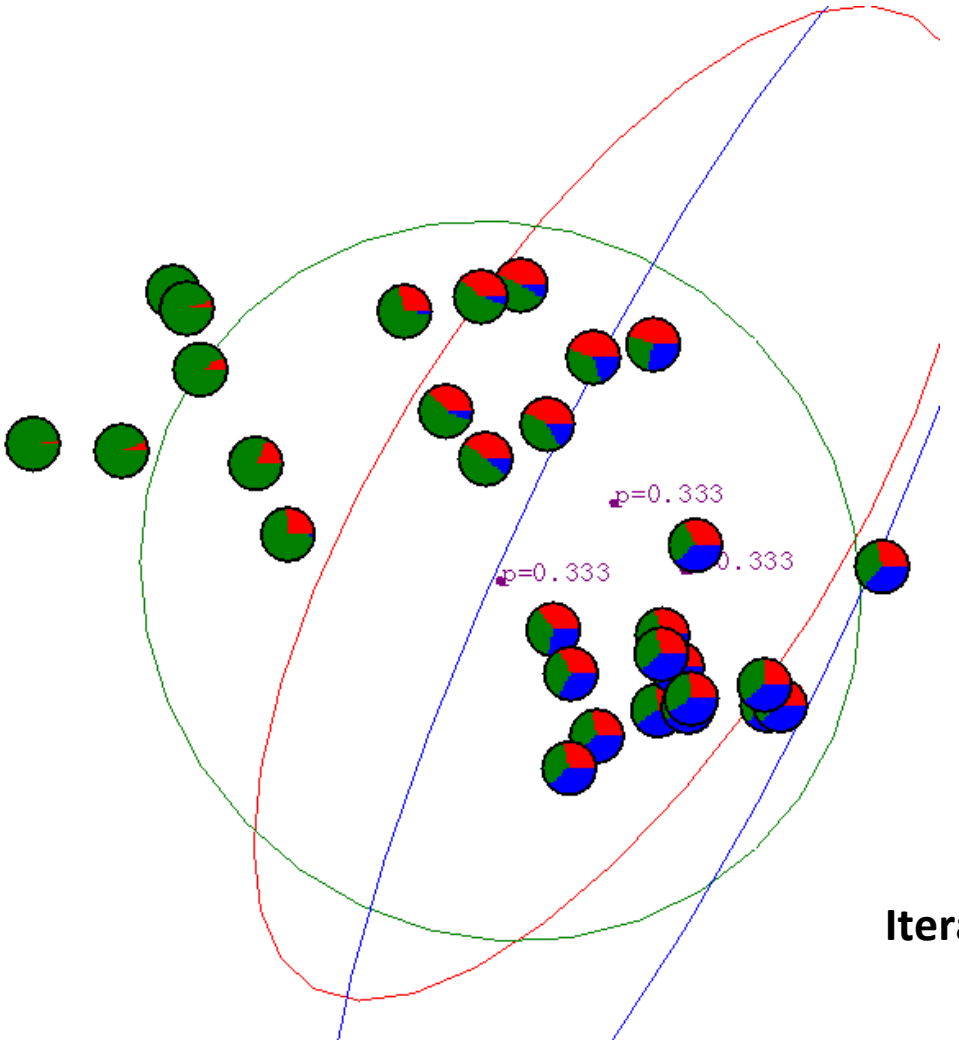
$$\Pr(x_i \in C_k) = \Pr(x_i \mid C_k) \Big/ \sum_j \Pr(x_i \mid C_j)$$

$$w_k = \frac{\sum_i \Pr(x_i \in C_k)}{n}$$

  - **M**aximation step: estimate model parameters

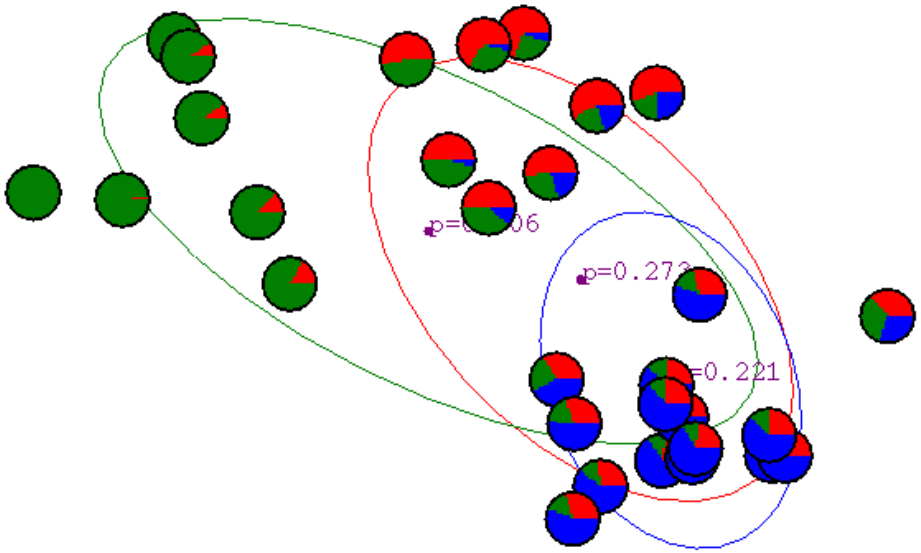
$$r_k = \frac{1}{n} \sum_{i=1}^{n} \frac{\Pr(x_i \in C_k)}{\sum_k \Pr(x_i \in C_j)}$$

Gaussian mixture example



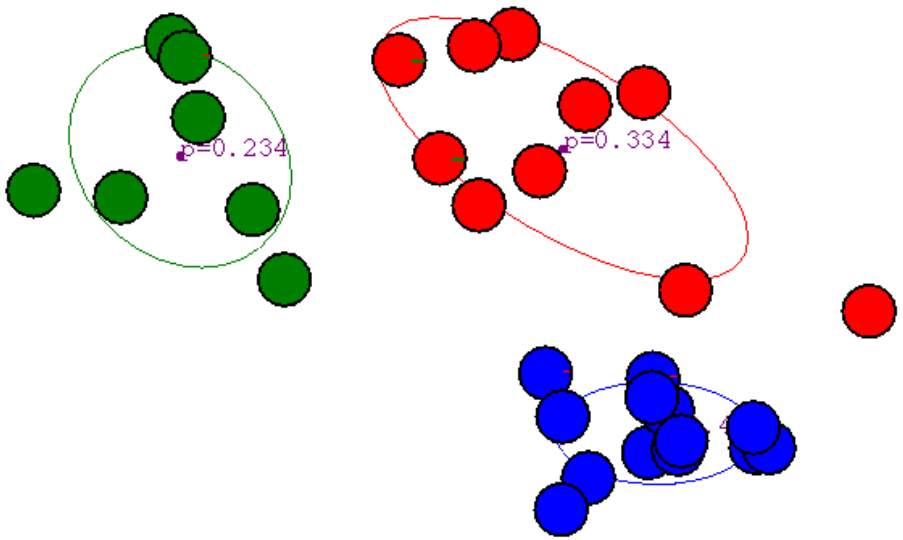
Iteration = 0

Gaussian mixture example



Iteration = 1

Gaussian mixture example



Iteration = 20