Machine Learning

Lecture 19c: Unsupervised Clustering (III): Gaussian Mixture Model

Dr. Beilun Wang

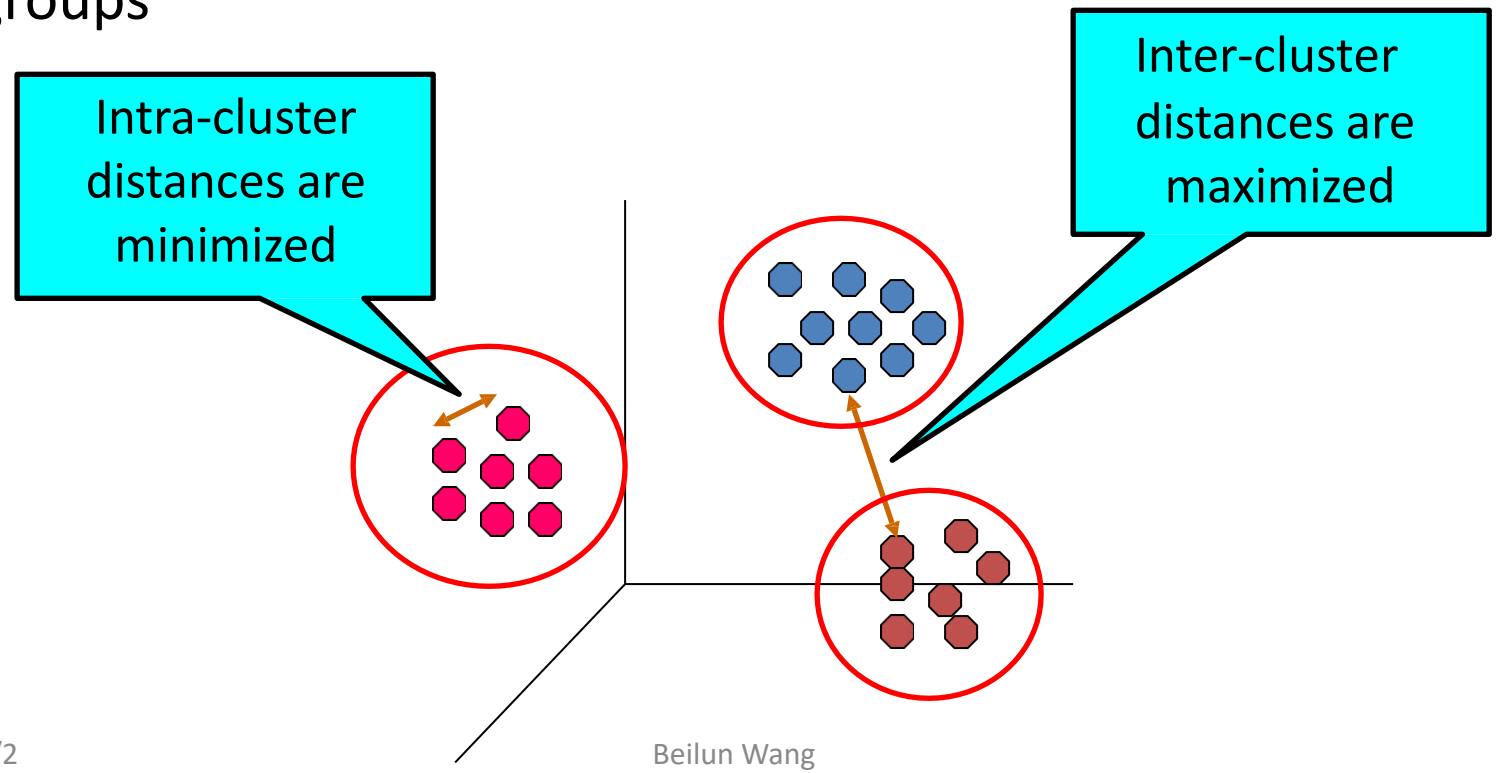
Southeast University
School of Computer Science
and Engineering

Course Content Plan

- Regression (supervised) — Y is a continuous
- Classification (supervised) — Y is a discrete
- Unsupervised models — NO Y
  - Dimension Reduction (PCA)
  - Clustering (K-means, GMM/EM, Hierarchical)
- Learning theory — About f()
- Graphical models — About interactions among X1,… Xp
- Reinforcement Learning — Learn program to Interact with its environment

What is clustering?

- Find groups (clusters) of data points such that data points in a group will be similar (or related) to one another and different from (or unrelated to) the data points in other groups

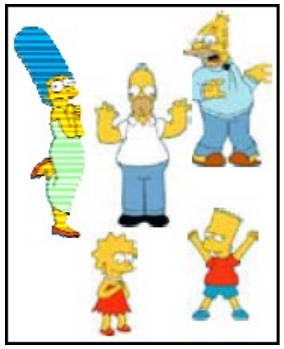


Roadmap: Clustering

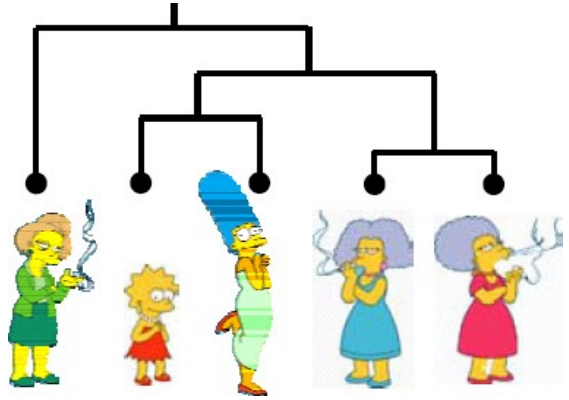
- Definition of "groupness"
- Definition of "similarity/distance"
- Clustering Algorithms
  - Hierarchical algorithms
  - Partitional algorithms
- Formal foundation and convergence
- How many clusters?

Clustering Algorithms

- Partitional algorithms
  - Usually start with a random (partial) partitioning
  - Refine it iteratively
    - K means clustering
    - Mixture-Model based clustering

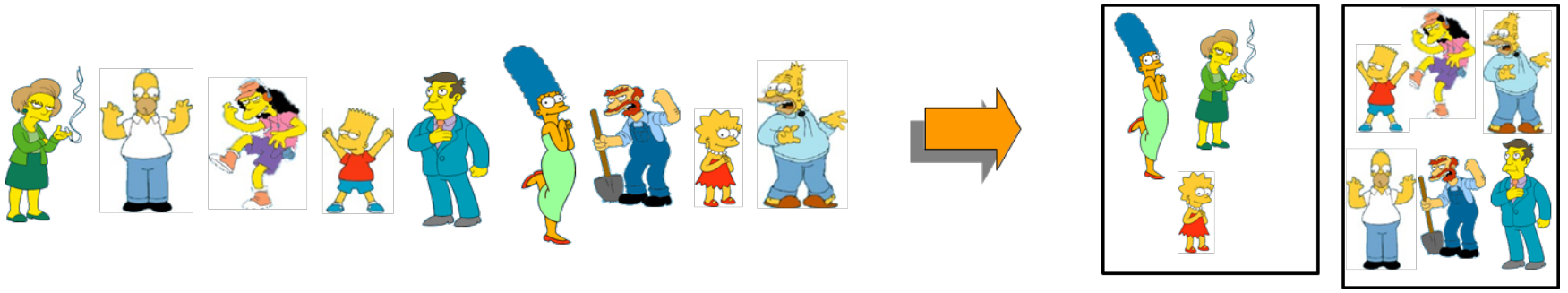


- Hierarchical algorithms
  - Bottom-up, agglomerative
  - Top-down, divisive



Partitional Clustering

- Nonhierarchical
- Construct a partition of n objects into a set of K clusters
- User has to specify the desired number of clusters K.



Other partitioning Methods

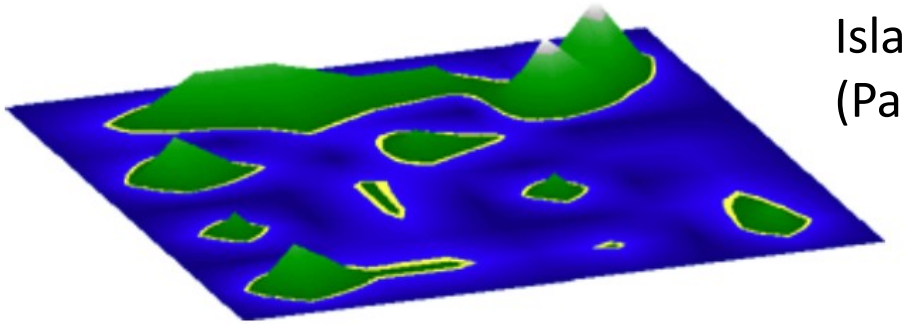
- Partitioning around medoids (PAM): instead of averages, use multidim medians as centroids (cluster “prototypes”). Dudoit and Freedland (2002).

- Self-organizing maps (SOM): add an underlying “topology” (neighboring structure on a lattice) that relates cluster centroids to one another. Kohonen (1997), Tamayo et al. (1999).

E.g. SOM Used for Visualization

Islands of Music

Analysis, Organization, and Visualization of Music Archives



Islands of music
(Pampalk et al., KDD' 03)

piece of music: member of a *music collection* and inhabitant of *islands of music*. Groups of similar pieces of music (also known as *genres*) like to gather around large mountains or small hills depending on the size of the group. Groups which are similar to each other like to live close together. Individuals which are not members of specific groups usually live near the beach and some very individualistic pieces might be found swimming in deep water.

islands of music: serve as graphical *user interface* to a music collection and are intended to help the user explore vast amounts of music in an efficient way. Islands of music are generated automatically based on *psychoacoustics models* and *self-organizing maps*.

Other partitioning Methods

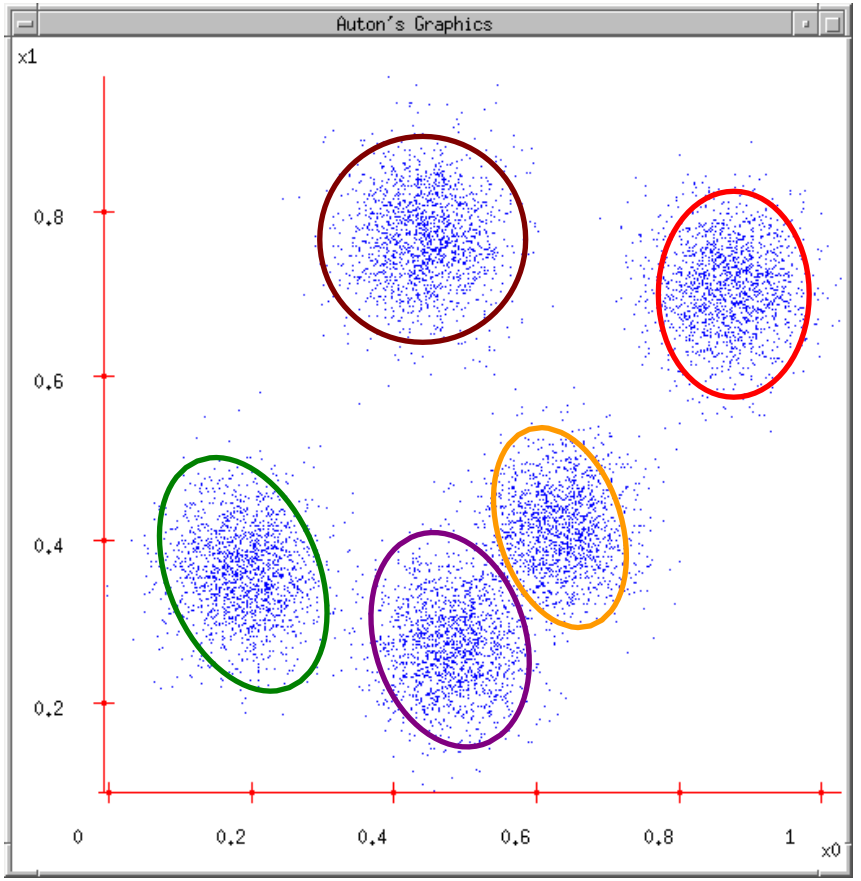
- Partitioning around medoids (PAM): instead of averages, use multidim medians as centroids (cluster "prototypes"). Dudoit and Freedland (2002).
- Self-organizing maps (SOM): add an underlying "topology" (neighboring structure on a lattice) that relates cluster centroids to one another. Kohonen (1997), Tamayo et al. (1999).
- Fuzzy k-means: allow for a "gradation" of points between clusters; soft partitions. Gash and Eisen (2002).
- Mixture-based clustering: implemented through an EM (Expectation-Maximization)algorithm. This provides soft partitioning, and allows for modeling of cluster centroids and shapes. (Yeung et al. (2001), McLachlan et al. (2002))

Today: Gaussian Mixture Model

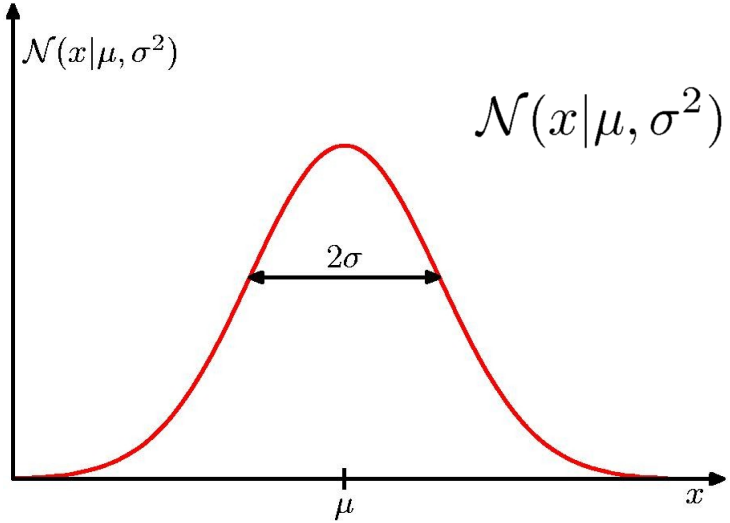
- Review of Gaussian Distribution
- GMM for clustering: basic algorithm
- GMM connecting to K-means
- Problems of GMM and K-means

A Gaussian Mixture Model for Clustering

- Assume that data are generated from a mixture of Gaussian distributions
- For each Gaussian distribution：
  - Center: μ_j
  - covariance: Σ_j
- For each data point：
  - Determine membership:
  - z_{ij}: if x_i belongs to j-th cluster

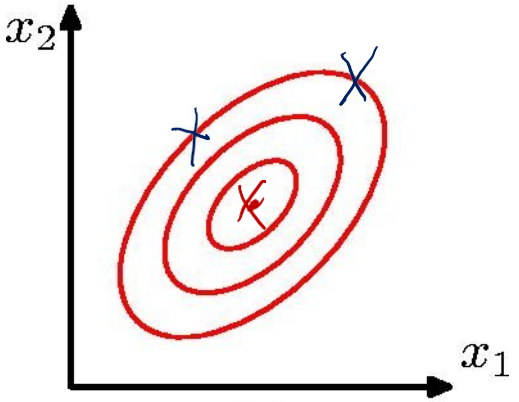


Review: Gaussian Distribution



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

$$X \sim N\left(\mu,\sigma^2\right)$$



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

Mean

Covariance Matrix

Review: the Bivariate Normal distribution

$$f\left(x_1, x_2\right) = \frac{1}{\left(2\pi\right)\left|\Sigma\right|^{1/2}} \mathrm{e}^{-\frac{1}{2}\left(\vec{x}-\vec{\mu}\right)^T \Sigma^{-1}\left(\vec{x}-\vec{\mu}\right)}$$

$$\text{with} \quad \vec{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} \quad \text{and}$$

$$\underset{2\times 2}{\Sigma} = \begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{12} & \sigma_{22} \end{bmatrix} = \begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix}_{2\times 2}$$

$$\left|\Sigma\right| = \sigma_{11}\sigma_{22} - \sigma_{12}^2 = \sigma_1^2\sigma_2^2\left(1-\rho^2\right)$$

Review: use MLE to estimate p-D Gaussian

$$< X_1, X_2, \ldots, X_p > \sim N(\vec{\mu}, \Sigma)$$

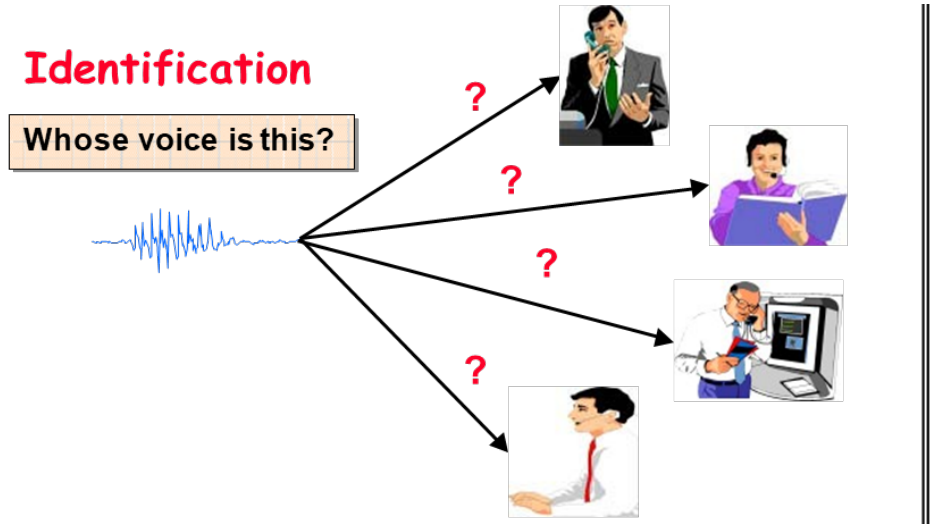
$$\vec{\mu} = \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_p \end{bmatrix} \qquad \Sigma_{p \times p} = \begin{bmatrix} var(X_1) & \ldots & cov(X_1, X_p) \\ \vdots & \ddots & \vdots \\ cov(X_p, X_1) & \ldots & var(X_p) \end{bmatrix}$$

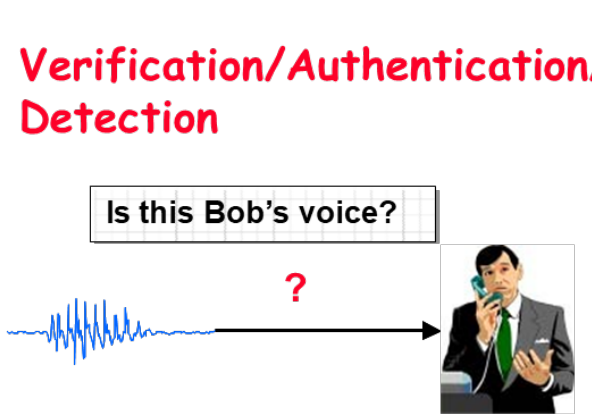
$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} x_i \qquad \hat{\Sigma} = \frac{1}{n} \sum_{i=1}^{n} (x_i - \hat{\mu}) (x_i - \hat{\mu})^T$$

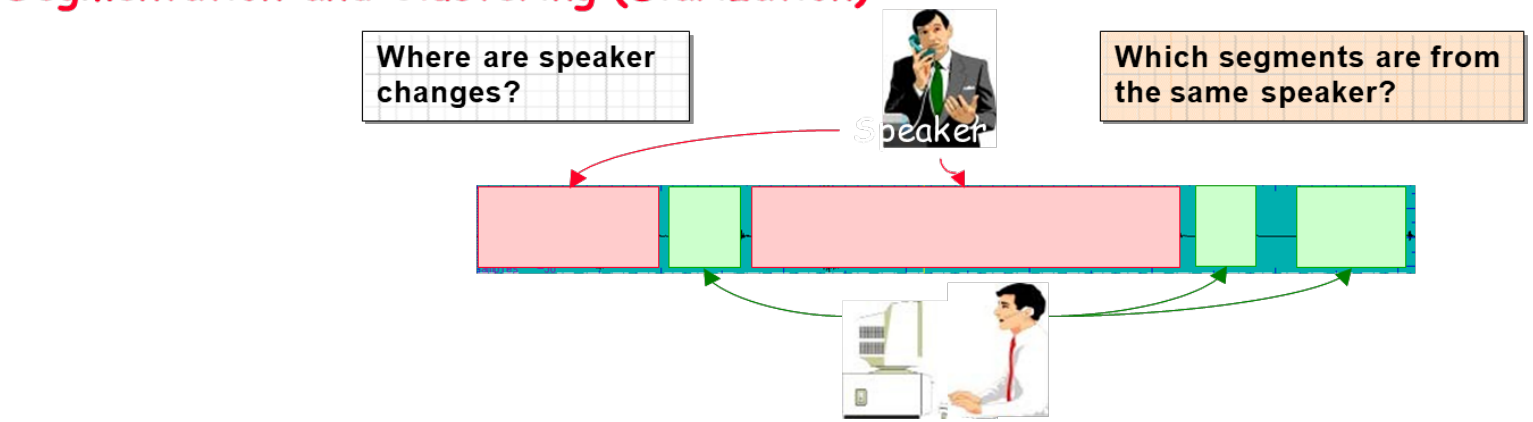
Today: Gaussian Mixture Model

- Review of Gaussian Distribution
- GMM for clustering: basic algorithm
- GMM connecting to K-means
- Problems of GMM and K-means

Application: Three Speaker Recognition Tasks

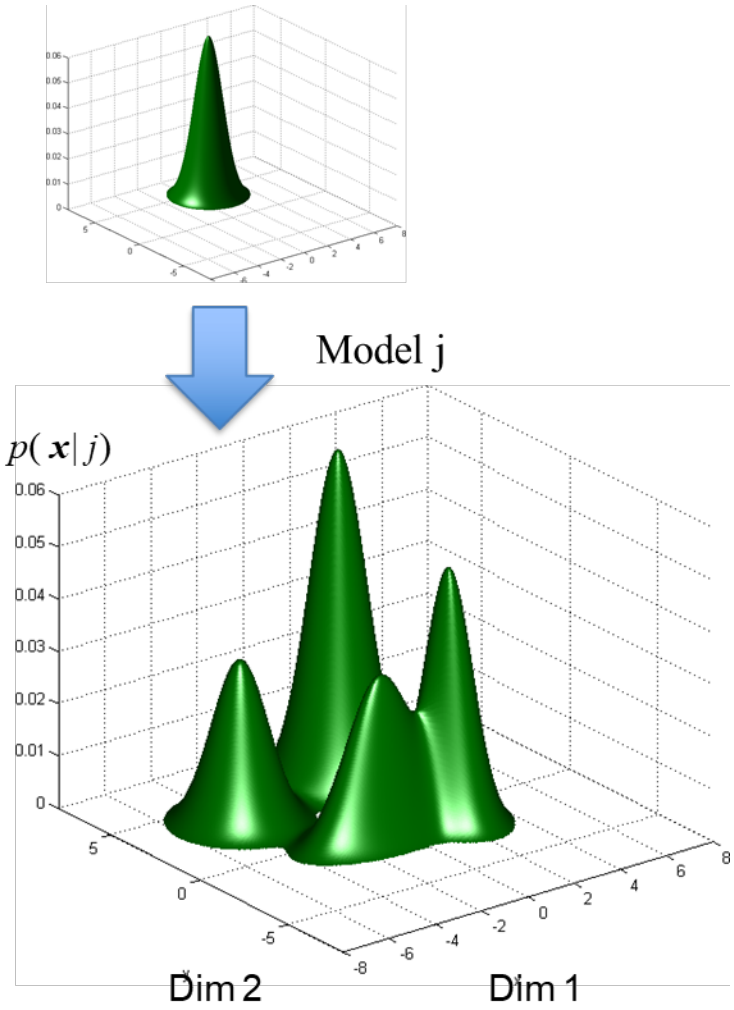




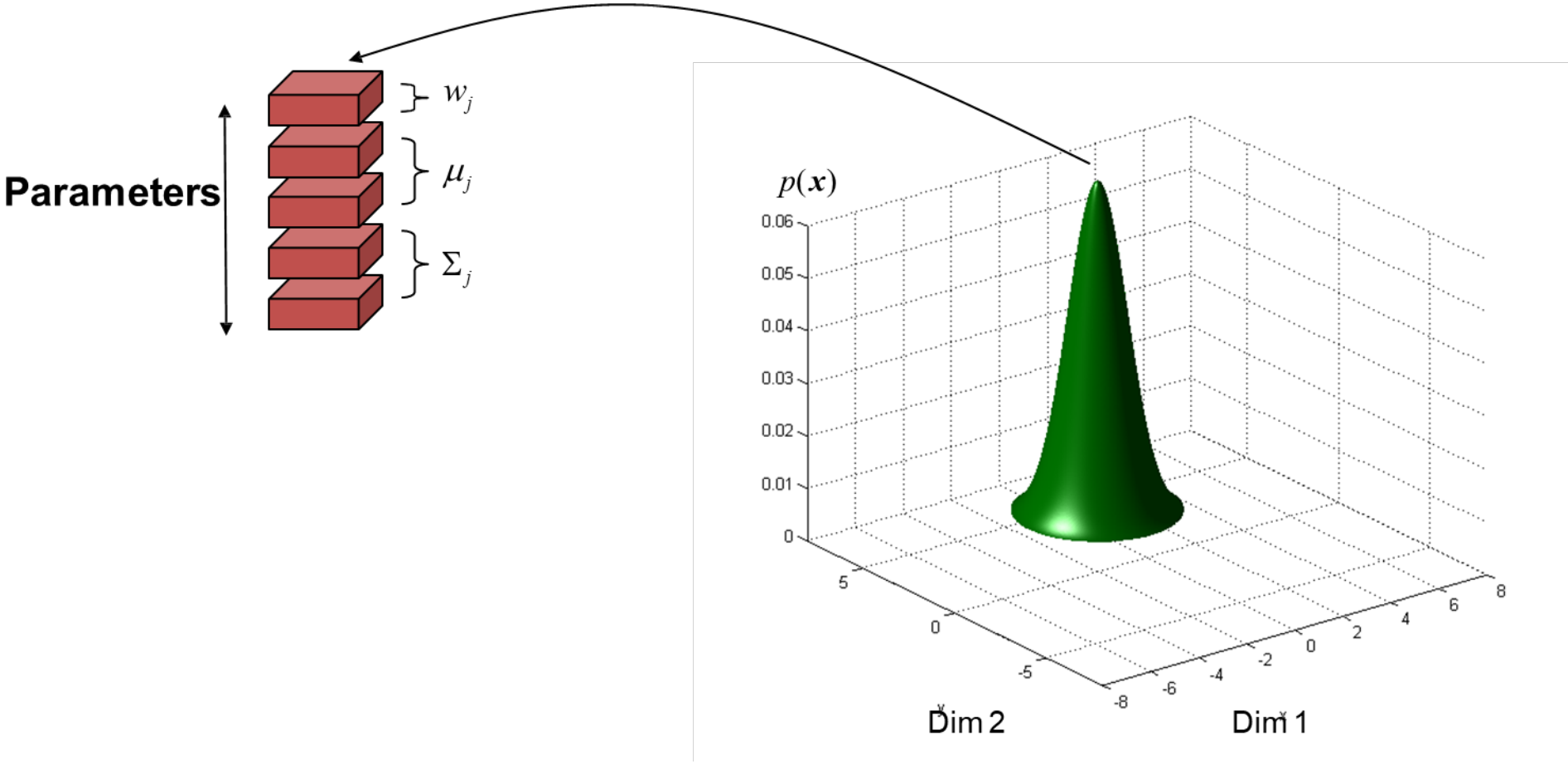


Application: GMMs for speaker recognition

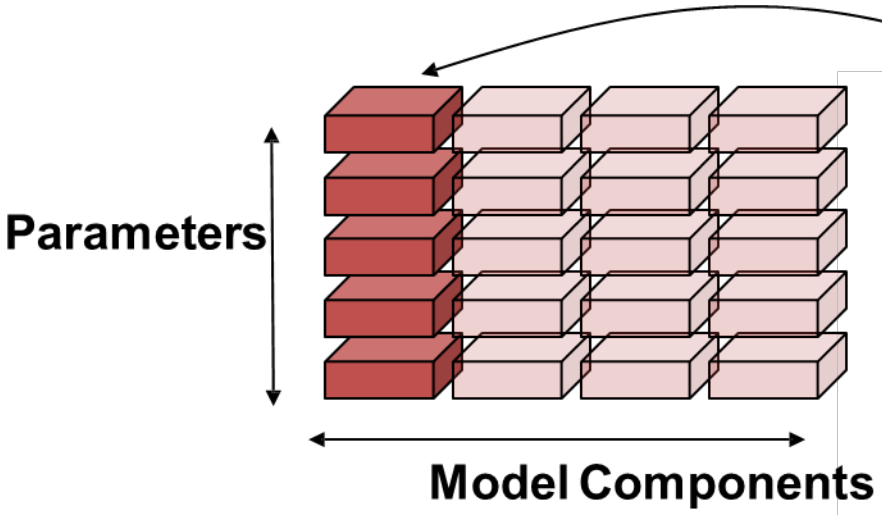
- A Gaussian mixture model (GMM) represents as the weighted sum of multiple Gaussian distributions
- Each Gaussian state j has a
  - Mean: μ_j
  - covariance: Σ_j
  - Weight: w_j



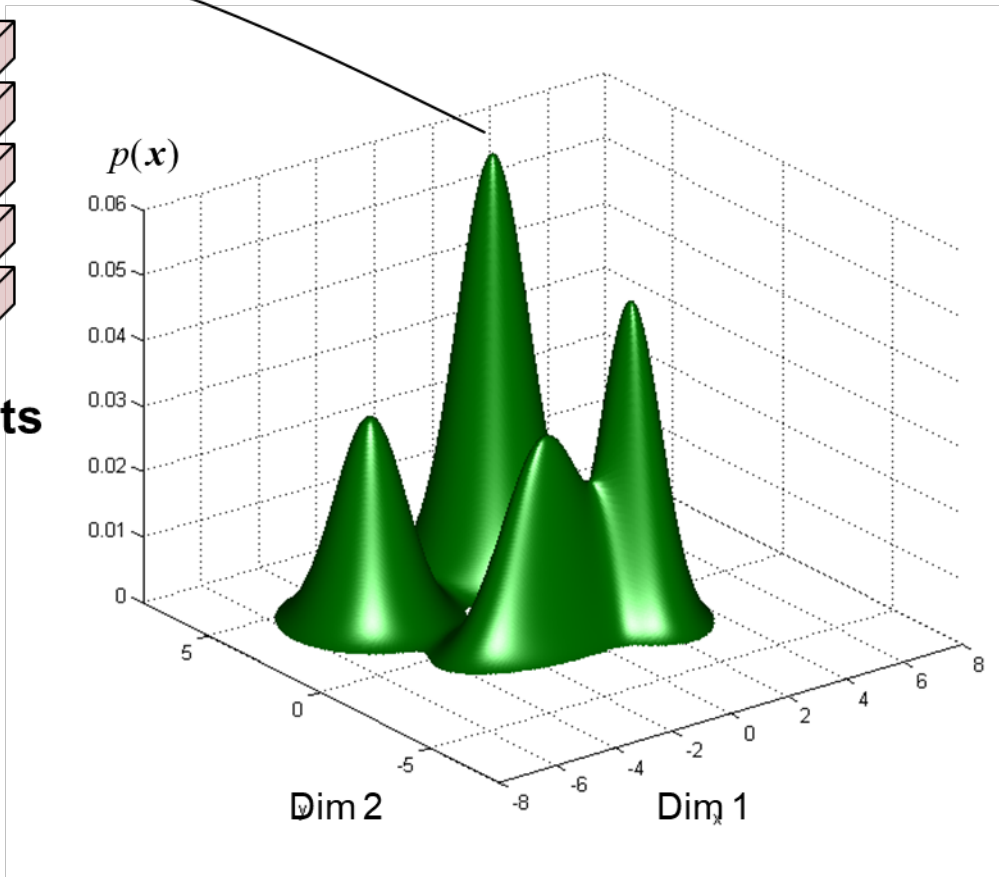
Recognition Systems Gaussian Mixture Models



Recognition Systems Gaussian Mixture Models



j=1, 2, …, K



Learning a Gaussian Mixture

- Probability Model

A Gaussian mixture model (GMM) represents as the weighted sum of multiple Gaussian distributions

$$p(\vec{x} = \vec{x}_i)$$

$$= \sum_j p(\vec{x} = \vec{x}_i, \vec{\mu} = \vec{\mu}_j)$$

Total low of probability

$$= \sum_i p(\vec{\mu} = \vec{\mu}_j) p(\vec{x} = \vec{x}_i \mid \vec{\mu} = \vec{\mu}_j)$$

Chain rule

$$= \sum_j p(\vec{\mu} = \vec{\mu}_j) \frac{1}{(2\pi)^{p/2} |\Sigma_j|^{1/2}} e^{-\frac{1}{2}(\vec{x} - \vec{\mu}_j)^T \Sigma_j^{-1} (\vec{x} - \vec{\mu}_j)}$$

Max Log-likelihood of Observed Data Samples

Log-likelihood of data $logp(x_1, x_2, x_3, \ldots, x_n) =$

$$\log \prod_{i=1..n} \sum_{j=1..K} p(\vec{\mu} = \vec{\mu}_j) \frac{1}{(2\pi)^{p/2} \left|\Sigma_j\right|^{1/2}} \mathrm{e}^{-\frac{1}{2}\left(\vec{x}_i - \vec{\mu}_j\right)^T \Sigma_j^{-1} \left(\vec{x}_i - \vec{\mu}_j\right)}$$

Apply MLE to find optimal Gaussian parameters

$$\left\{\{p(\vec{\mu} = \mu_j)\}, j = 1 \ldots K\right\}$$

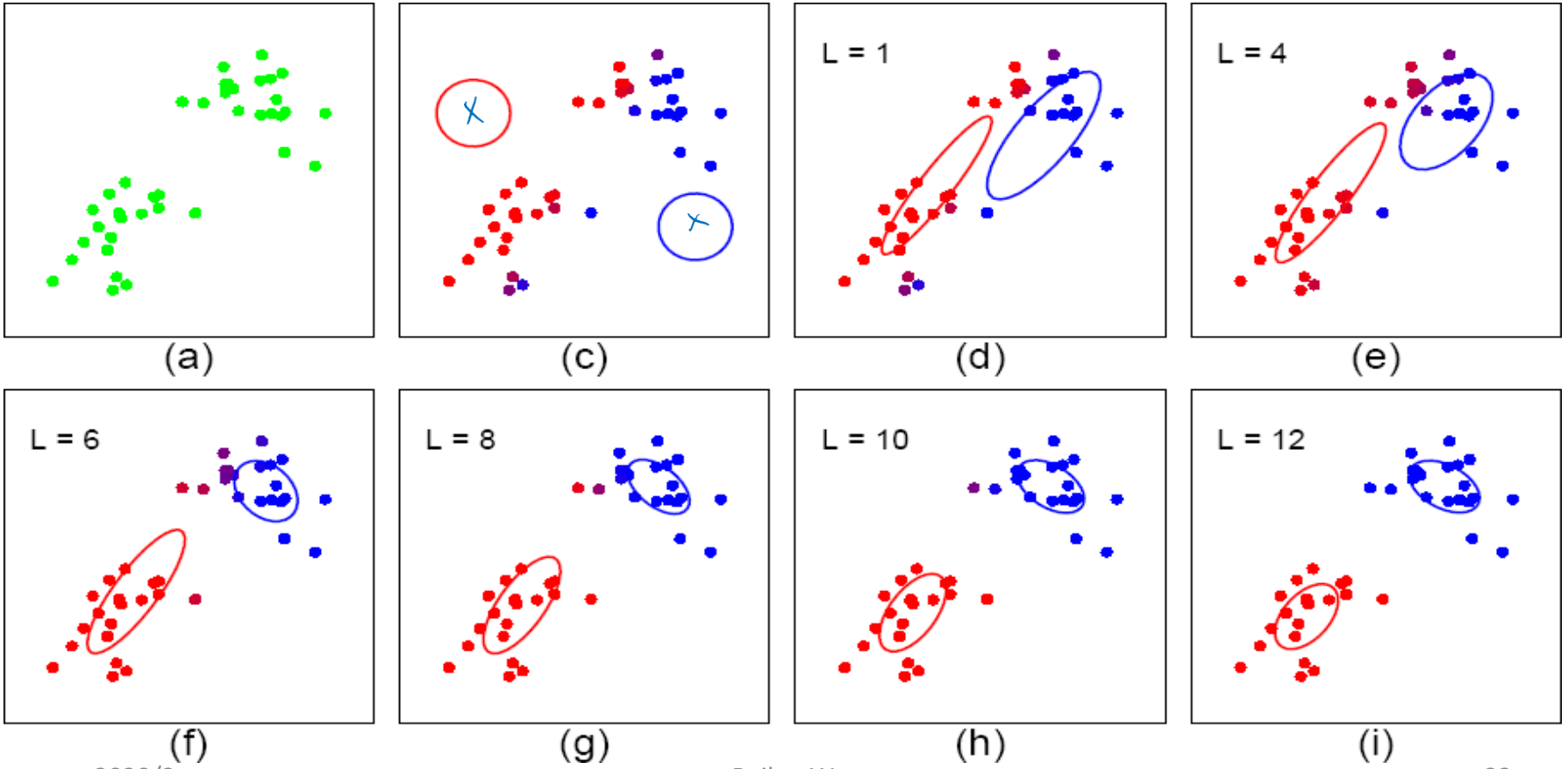
$$\{\vec{\mu}_j, \Sigma_j, j = 1 \ldots K\}$$

Expectation-Maximization for training GMM

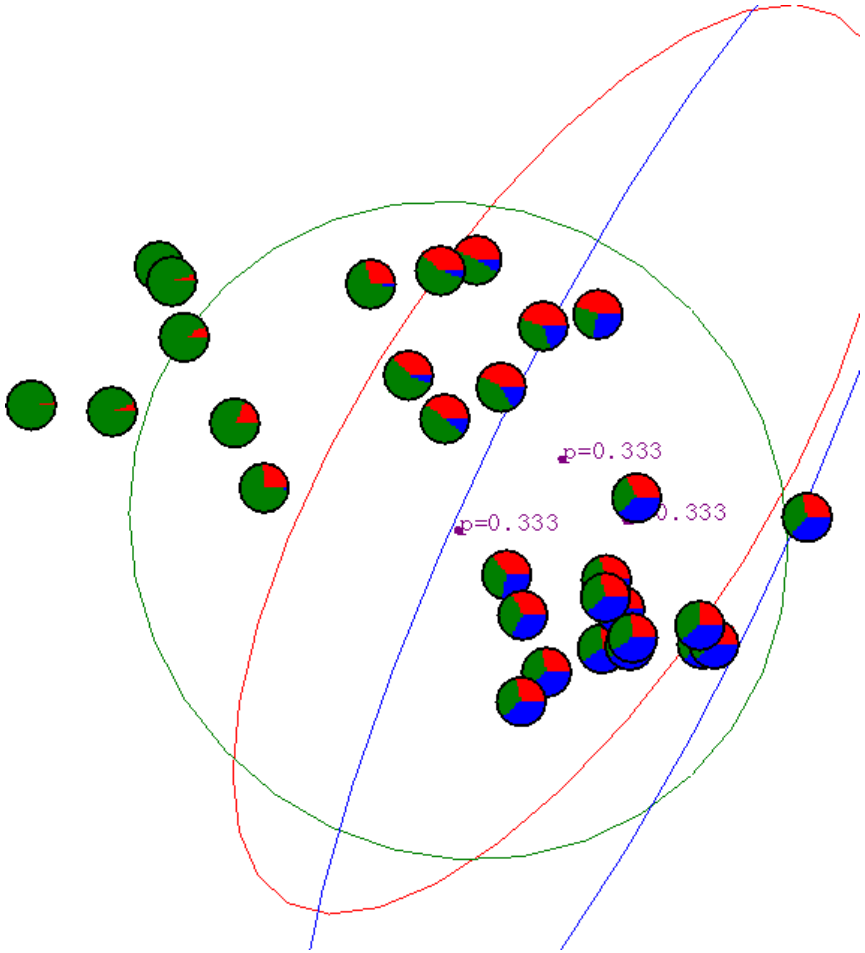
- Start:
  - "Guess" the centroid and covariance for each of the K clusters
  - “Guess” the proportion of clusters, e.g., uniform prob 1/K
- Loop
  - For each point, revising its proportions belonging to each of the K clusters
  - For each cluster, revising both the mean (centroid position) and covariance (shape)

An example

- each cluster, revising both the mean (centroid position) and covariance (shape)

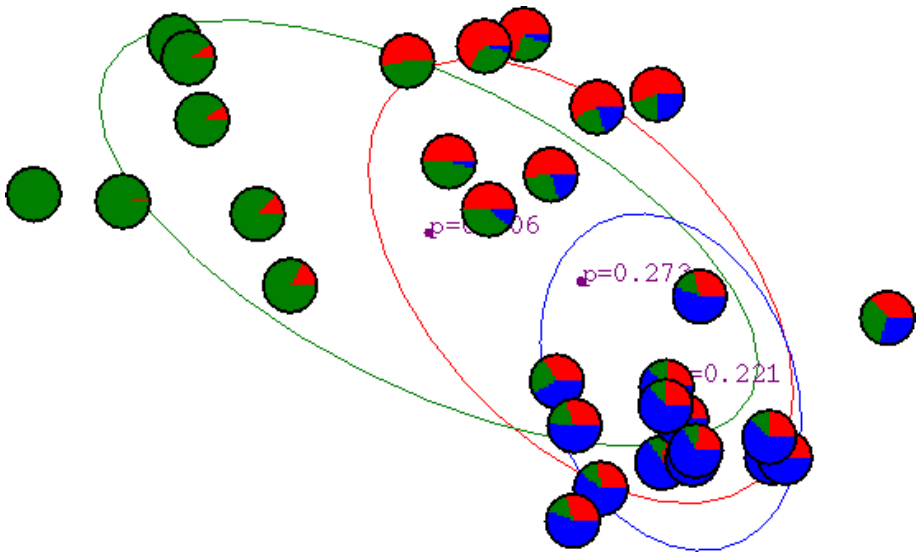


Another Example: Start



Another Example: After First Iteration

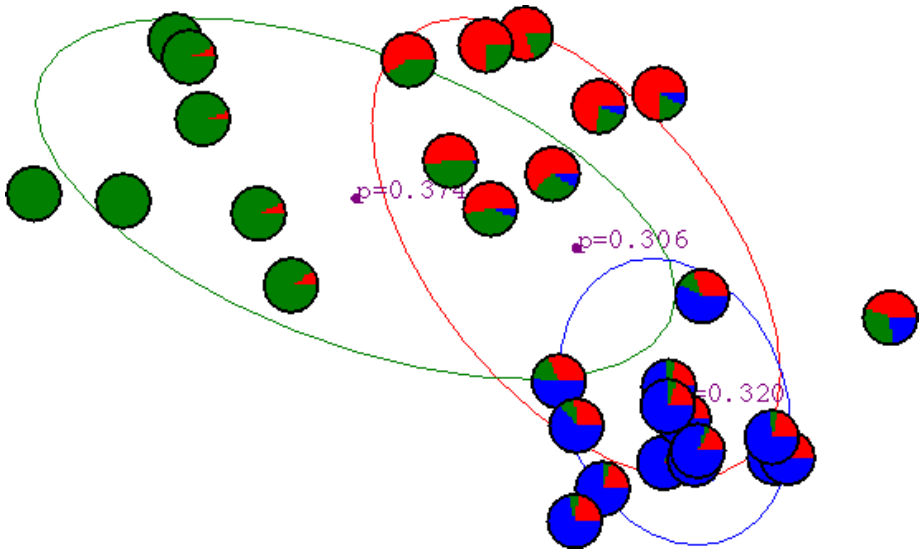
For each point, revising its proportions belonging to each of the K clusters



For each cluster, revising its mean (centroid position), covariance (shape) and proportion in the mixture

Another Example: After 2nd Iteration

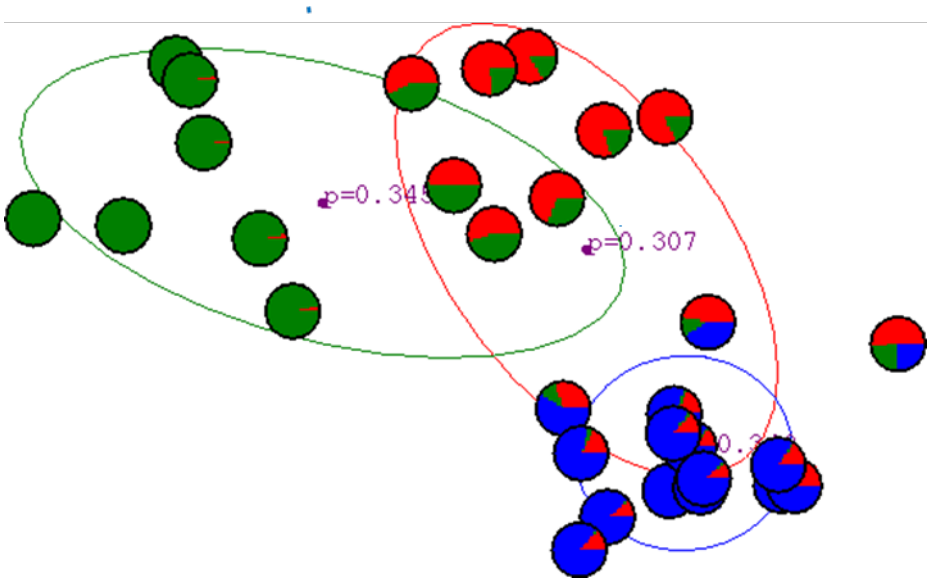
For each point, revising its proportions belonging to each of the K clusters



For each cluster, revising its mean (centroid position), covariance (shape) and proportion in the mixture

Another Example: After 3rd Iteration

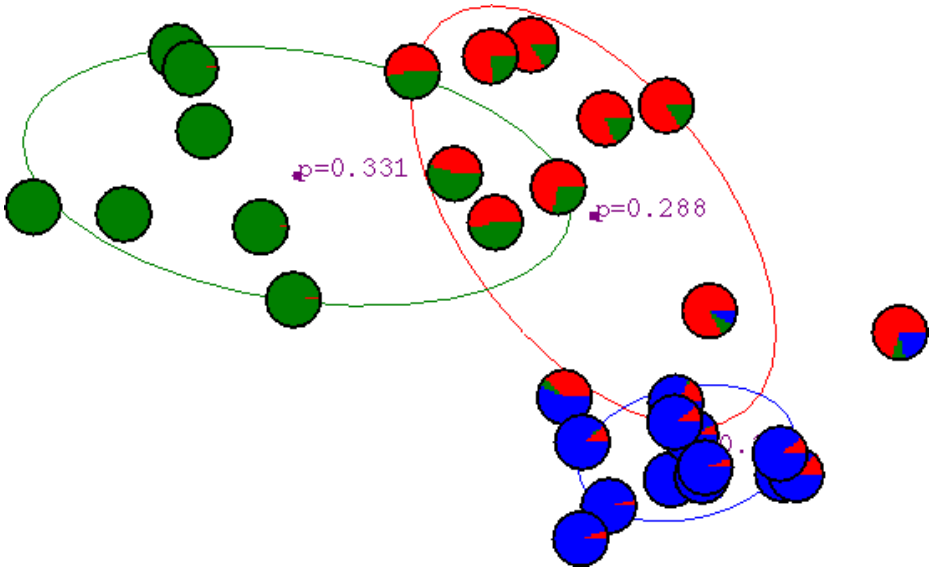
For each point, revising its proportions belonging to each of the K clusters



For each cluster, revising its mean (centroid position), covariance (shape) and proportion in the mixture

Another Example: After 4th Iteration

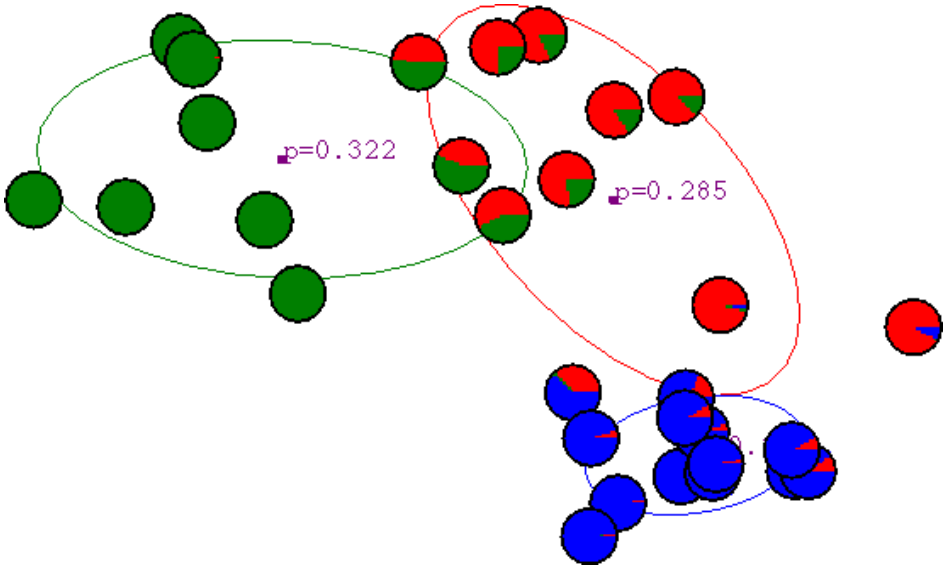
For each point, revising its proportions belonging to each of the K clusters



For each cluster, revising its mean (centroid position), covariance (shape) and proportion in the mixture

Another Example: After 5th Iteration

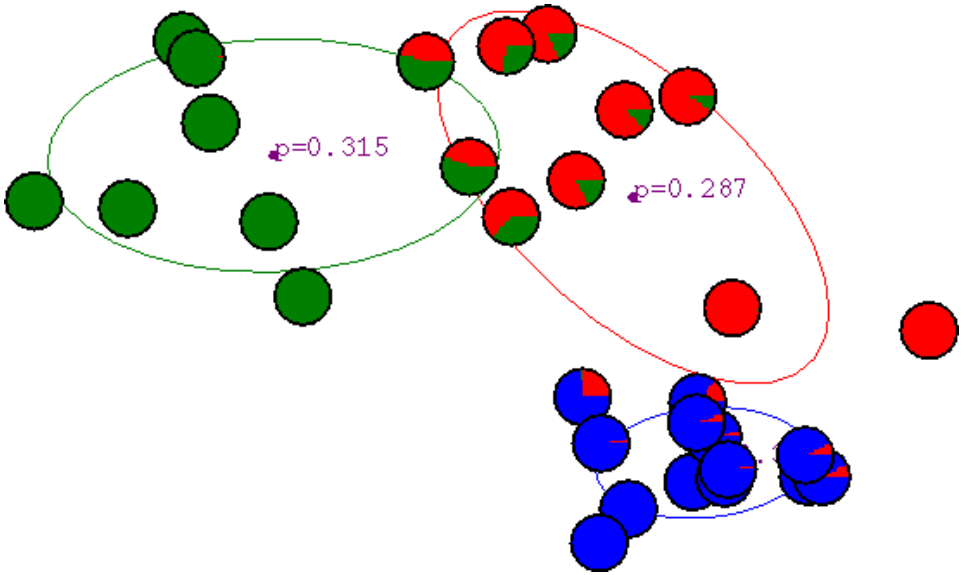
For each point, revising its proportions belonging to each of the K clusters



For each cluster, revising its mean (centroid position), covariance (shape) and proportion in the mixture

Another Example: After 6th Iteration

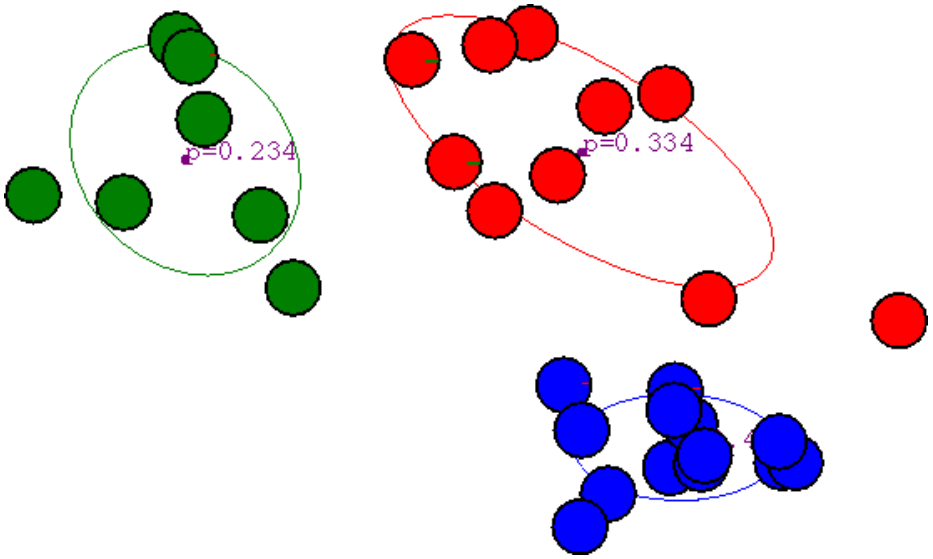
For each point, revising its proportions belonging to each of the K clusters



For each cluster, revising its mean (centroid position), covariance (shape) and proportion in the mixture

Another Example: After 20th Iteration

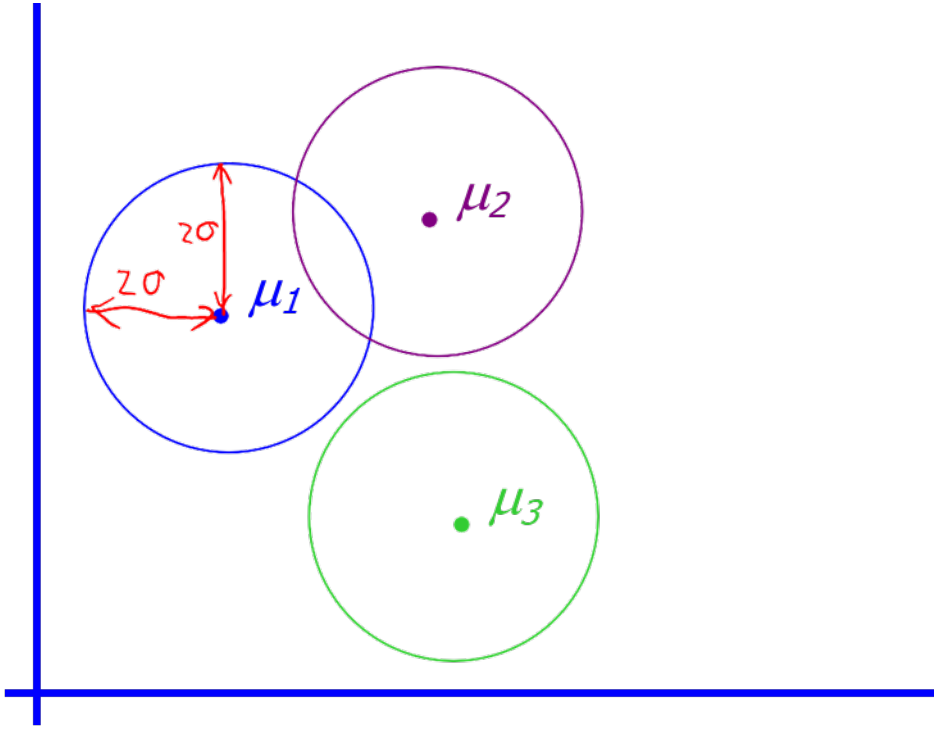
For each point, revising its proportions belonging to each of the K clusters



For each cluster, revising its mean (centroid position), covariance (shape) and proportion in the mixture

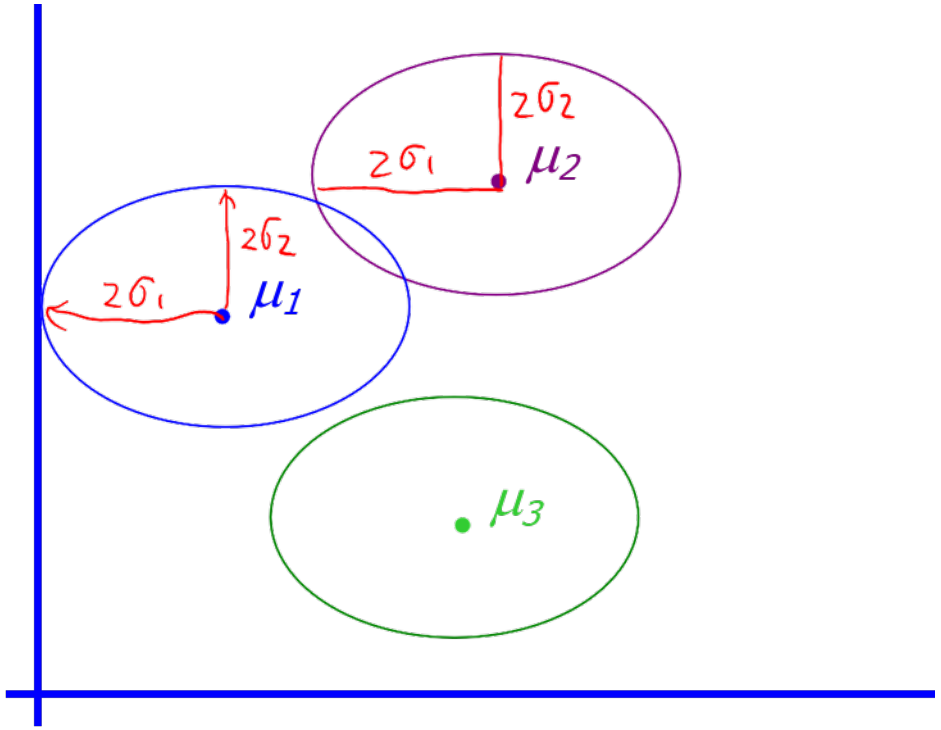
The Simplest GMM assumption

- Each component generates data from a Gaussian with
  - Mean: μ_i
  - Shared diagonal covariance matrix: $\sigma^2 I$



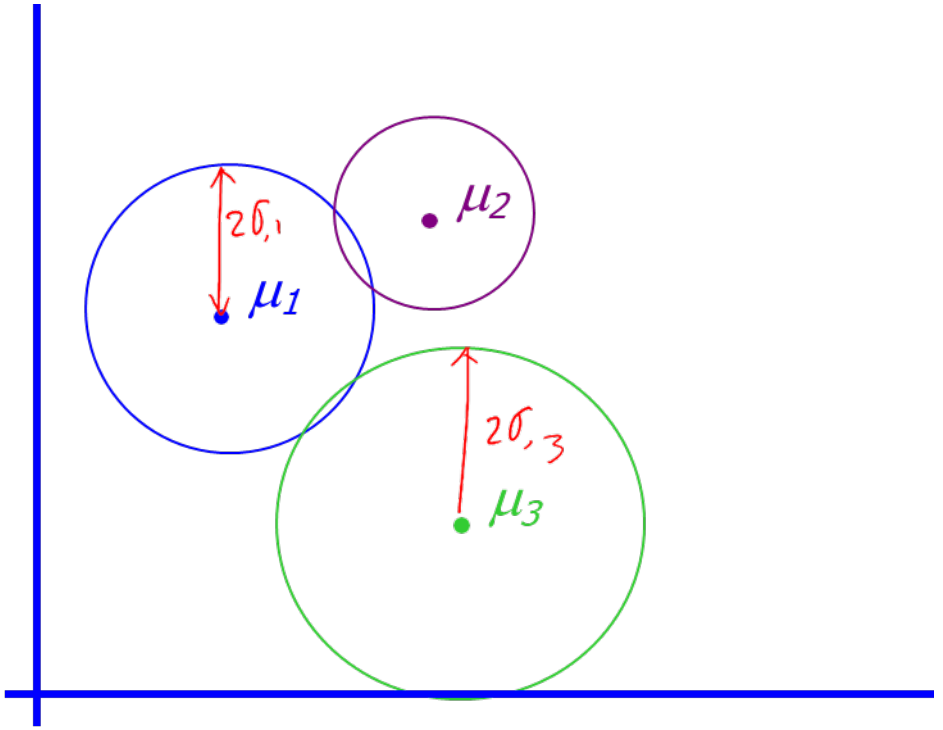
Another Simple GMM assumption

- Each component generates data from a Gaussian with
  - Mean: μ_i
  - Shared covariance matrix as diagonal matrix



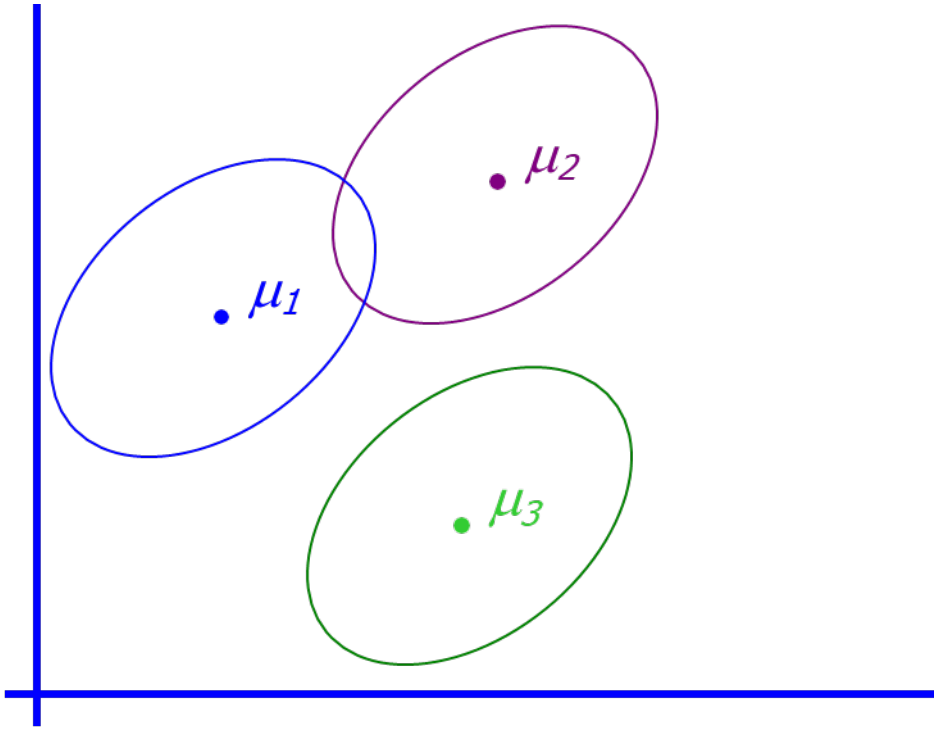
Another Simple GMM assumption

- Each component generates data from a Gaussian with
  - Mean: μ_i
  - Cluster-specific diagonal covariance matrix as $\sigma_\varphi{}^2 I$



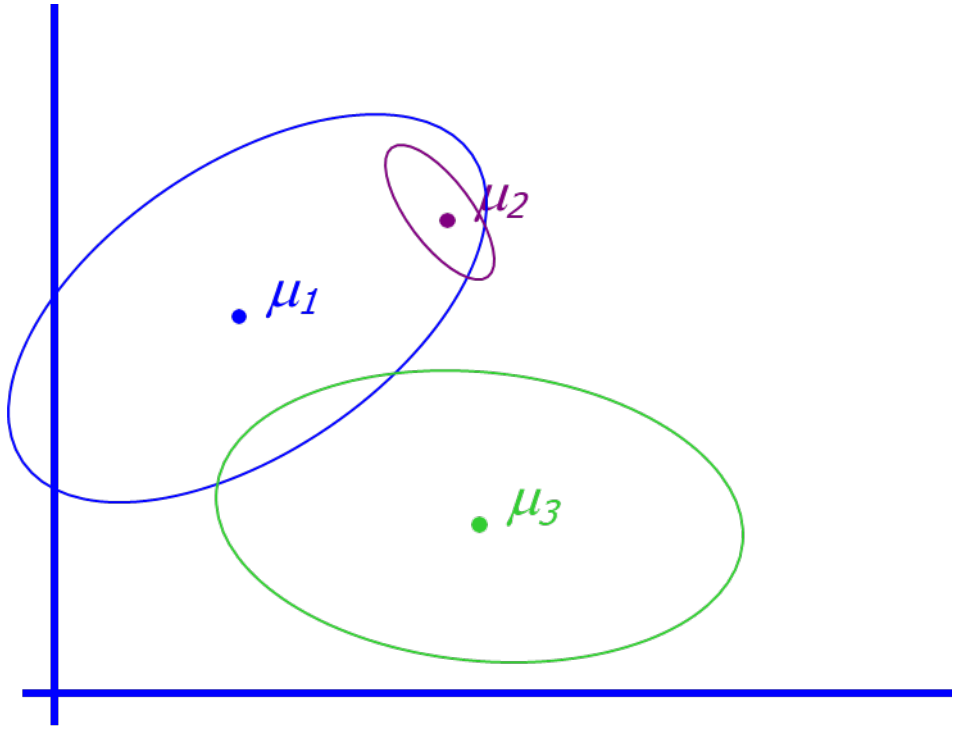
A bit More General GMM assumption

- Each component generates data from a Gaussian with:
  - Mean: μ_i
  - Shared covariance matrix as full matrix



The General GMM assumption

- Each component generates data from a Gaussian with:
  - Mean: μ_i
  - Covariance matrix: Σ_i



Concrete Equations for Learning a Gaussian Mixture (assuming with known shared covariance)

$$
\begin{aligned}
& p(\vec{x} = \vec{x}_i) \\
&= \sum_{\mu_j} p(\vec{x} = \vec{x}_i, \vec{\mu} = \vec{\mu}_j) \\
&= \sum_{j} p(\vec{\mu} = \vec{\mu}_j) p(\vec{x} = \vec{x}_i \mid \vec{\mu} = \vec{\mu}_j) \\
&= \sum_{j} p(\vec{\mu} = \vec{\mu}_j) \frac{1}{(2\pi)^{p/2} |\Sigma|^{1/2}} \mathrm{e}^{-\frac{1}{2}\left(\vec{x}_i - \vec{\mu}_j\right)^T \Sigma_j^{-1} \left(\vec{x}_i - \vec{\mu}_j\right)}
\end{aligned}
$$

Learning a Gaussian Mixture (assuming with known shared covariance)

E-Step

$$E[z_{ij}] = p(\vec{\mu} = \mu_j \mid x = x_i)$$

$$= \frac{p(x = x_i \mid \mu = \mu_j)p(\mu = \mu_j)}{\sum_{s=1}^{k} p(x = x_i \mid \mu = \mu_s)p(\mu = \mu_s)}$$

$$= \frac{\frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(\vec{x}_i - \vec{\mu}_j)^T \Sigma^{-1}(\vec{x}_i - \vec{\mu}_j)} p(\mu = \mu_j)}{\sum_{s=1}^{k} \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(\vec{x}_i - \vec{\mu}_s)^T \Sigma^{-1}(\vec{x}_i - \vec{\mu}_s)} p(\mu = \mu_s)}$$

Learning a Gaussian Mixture (assuming with known shared covariance)

M-Step

$$\mu_j \leftarrow \frac{1}{\sum_{i=1}^{n} E[z_{ij}]} \sum_{i=1}^{n} E[z_{ij}] x_i$$

$$p(\mu = \mu_j) \leftarrow \frac{1}{n} \sum_{i=1}^{n} E[z_{ij}]$$

Covariance: Σ_j (j: 1 to K) can also be derived in the M-step under a full setting

M-step for Estimating (unknown Covariance Matrix)

- more general, details in EM-Extra lecture

$$\sum_j^{(t+1)} = \frac{\sum_{i=1}^{n} E[z_{ij}]^{(t)} (x_i - \mu_j^{(t+1)})(x_i - \mu_j^{(t+1)})^T}{\sum_{i=1}^{n} E[z_{ij}]^{(t)}}$$

Recap: Expectation-Maximization for training GMM

- Start:
  - "Guess" the centroid and covariance for each of the K clusters
  - “Guess” the proportion of clusters, e.g., uniform prob 1/K
- Loop
  - For each point, revising its proportions belonging to each of the K clusters
  - For each cluster, revising both the mean (centroid position) and covariance (shape)

Today: Gaussian Mixture Model

- Review of Gaussian Distribution
- GMM for clustering : basic algorithm
- GMM connecting to K-means
- Problems of GMM and K-means

Recap: K-means iterative learning

$$\arg\min_{\{\vec{C}_j, m_{i,j}\}} \sum_{j=1}^{K} \sum_{i=1}^{n} m_{i,j} \left(\vec{x}_i - \vec{C}_j \right)^2$$

Memberships $\{m_{i,j}\}$ and centers $\{C_j\}$ are correlated.

E-Step Given centers $\{\vec{C}_j\}$, $m_{i,j} = \begin{cases} 1 & j = \arg\min_k (\vec{x}_i - \vec{C}_j)^2 \\ 0 & \text{otherwise} \end{cases}$

M-Step Given memberships $\{m_{i,j}\}$, $\vec{C}_j = \dfrac{\sum_{i=1}^{n} m_{i,j} \vec{x}_i}{\sum_{i=1}^{n} m_{i,j}}$

Compare: K-means

- The EM algorithm for mixtures of Gaussians is like a "soft version" of the K-means algorithm.
- In the K-means “E-step” we do hard assignment:
- In the K-means “M-step” we update the means as the weighted sum of the data, but now the weights are 0 or 1

Compare: K-means & GMM

K-means:

$$\arg\min_{\{\vec{C}_j, m_{i,j}\}} \sum_{j=1}^{K}\sum_{i=1}^{n} m_{i,j}\left(\vec{x}_i - \vec{C}_j\right)^2$$

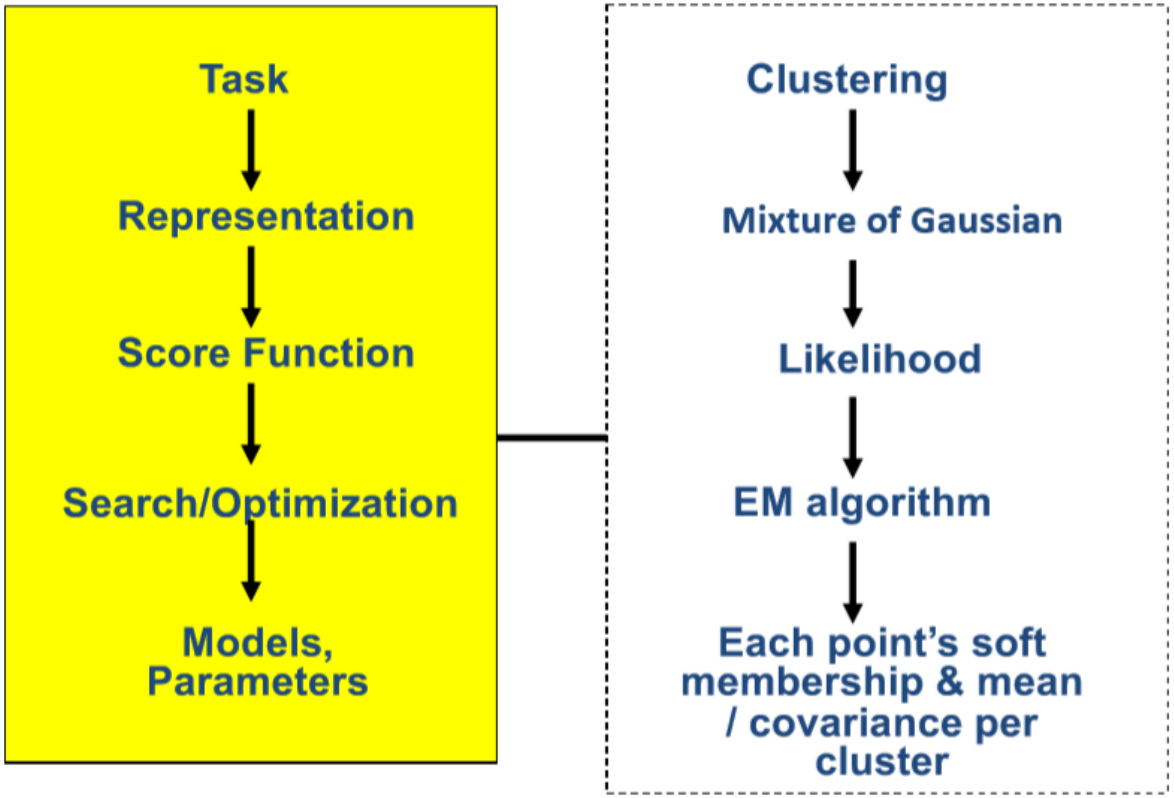
GMM:

$$\sum_i \log \prod_{i=1}^{n} p(x = x_i)$$

$$= \sum_i \log\left(\sum_j p(\vec{\mu} = \vec{\mu}_j) \frac{1}{(2\pi)^{p/2}\left|\Sigma\right|^{1/2}} \mathrm{e}^{-\frac{1}{2}\left(\vec{x}_i - \vec{\mu}_j\right)^T \Sigma_j^{-1}\left(\vec{x}_i - \vec{\mu}_j\right)}\right)$$

- K-Mean only detect spherical clusters.
- GMM can adjust its self to elliptic shape clusters.

GMM Clustering

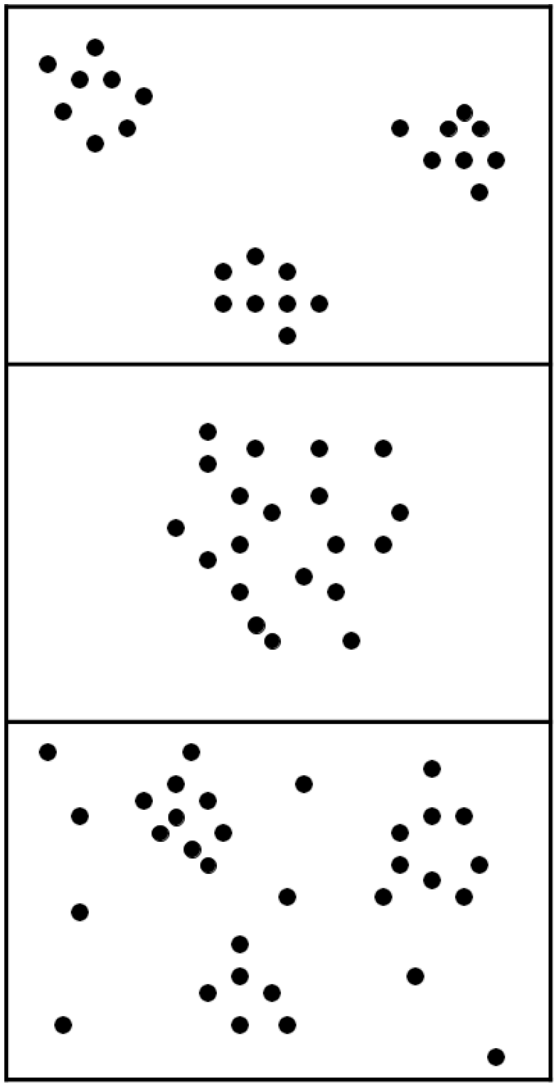


$$\sum_i \log \prod_{i=1}^{n} p(x = x_i) = \sum_i \log \left[\sum_{\mu_j} p(\mu = \mu_j) \frac{1}{(2\pi)\left|\Sigma_j\right|^{1/2}} e^{-\frac{1}{2}\left(\vec{x}-\vec{\mu}_j\right)^T \Sigma_j^{-1}\left(\vec{x}-\vec{\mu}_j\right)} \right]$$

Today: Gaussian Mixture Model

- Review of Gaussian Distribution
- GMM for clustering: basic algorithm
- GMM connecting to K-means
- Problems of GMM and K-means

Unsupervised Learning: not as hard as it looks



Sometimes easy

Sometimes impossible

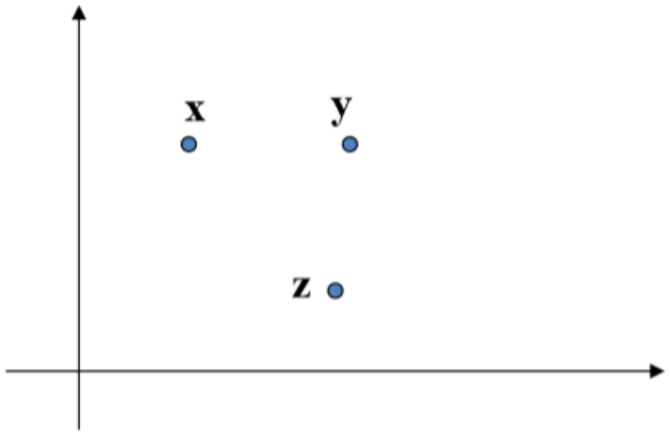
and sometimes
in between

Problems (I)

- Both k-means and mixture models need to compute centers of clusters and explicit distance measurement
  - Given strange distance measurement, the center of clusters can be hard to compute

E.g.,

$$\left\|\vec{x}-\vec{x}'\right\|_\infty = \max\left(\left|x_1-x_1'\right|,\left|x_2-x_2'\right|,\ldots,\left|x_p-x_p'\right|\right)$$



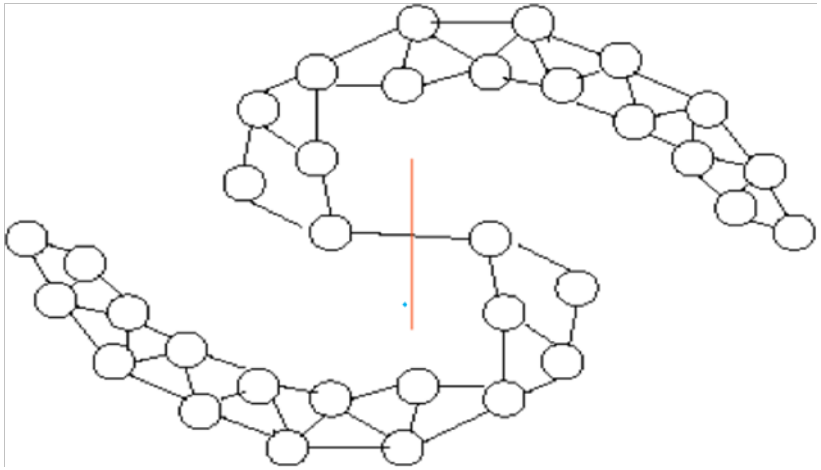
$$\|\mathbf{x}-\mathbf{y}\|_\infty = \|\mathbf{x}-\mathbf{z}\|_\infty$$

Problems (II)

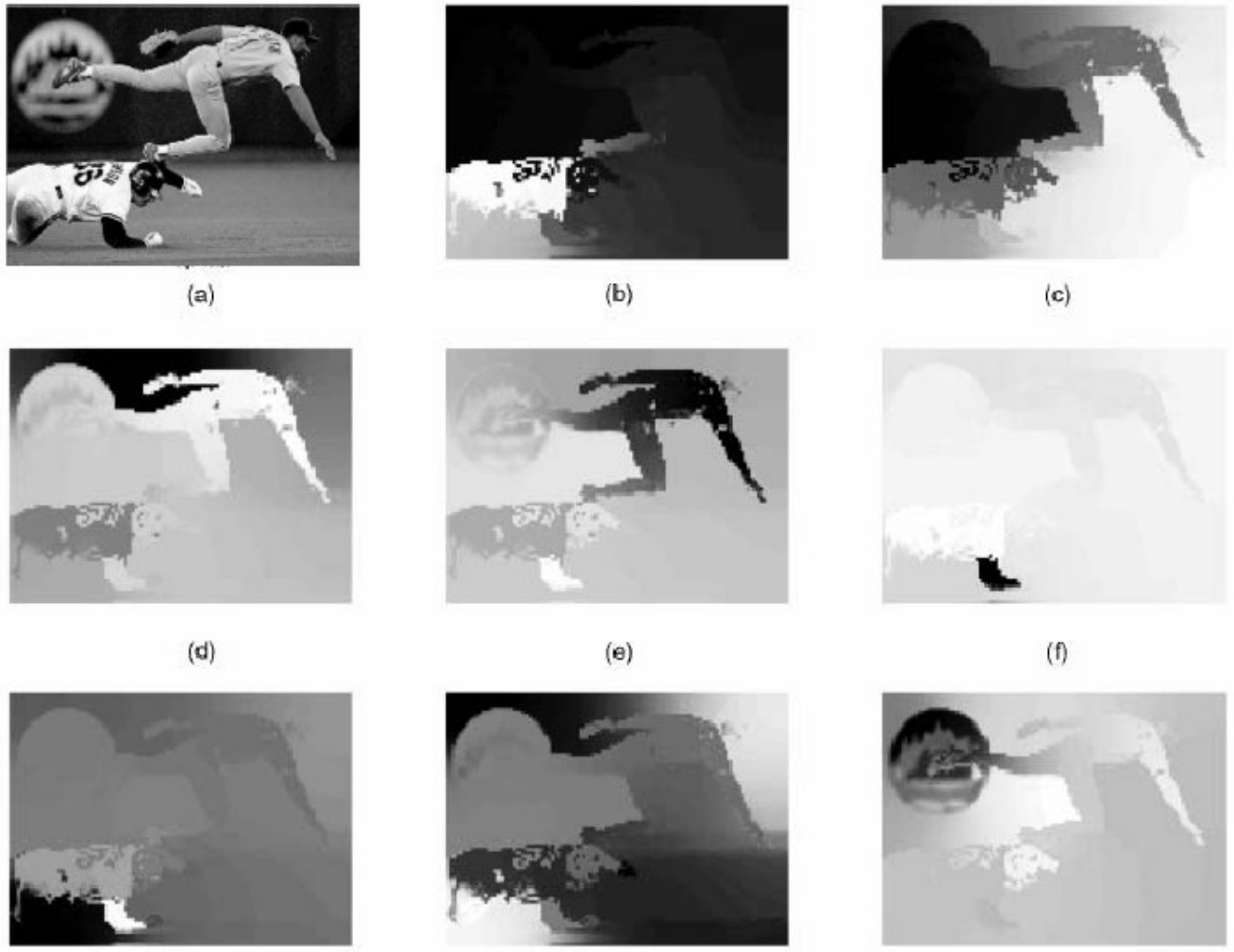
- Both k-means and mixture models look for compact clustering structures
  - In some cases, connected clustering structures are more desirable

Graph based clustering

e.g. MinCut, Spectral clustering



e.g. Image Segmentation through minCut



References

- https://qiyanjun.github.io/2019f-UVA-CS6316-MachineLearning/
- Hastie, Trevor, et al. The elements of statistical learning. Vol. 2. No. 1. New York: Springer, 2009.
- Big thanks to Prof. Eric Xing @ CMU for allowing me to reuse some of his slides
- Big thanks to Prof. Ziv Bar-Joseph @ CMU for allowing me to reuse some of his slides
- clustering slides from Prof. Rong Jin @ MSU

Thanks for listening