

Mixture models & EM algorithm

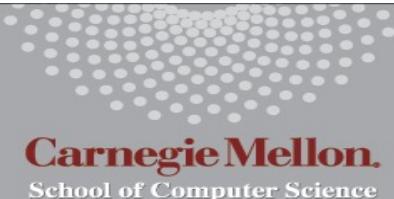
Aarti Singh

Machine Learning 10-315
Apr 11, 2022

Some slides courtesy of Eric Xing, Carlos Guestrin



MACHINE LEARNING DEPARTMENT



Partitioning Algorithms

- K-means
 - **hard assignment**: each object belongs to only one cluster
- Mixture modeling
 - **soft assignment**: probability that an object belongs to a cluster

Generative approach

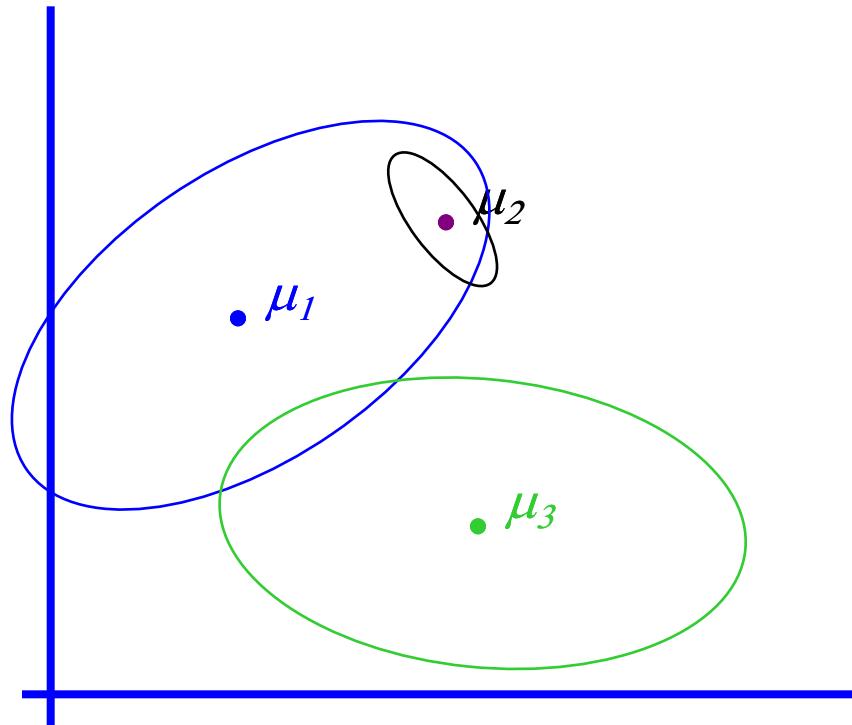
Mixture models

GMM – Gaussian Mixture Model (Multi-modal distribution)

$$p(x|y=i) \sim N(\mu_i, \Sigma_i)$$

$$p(x) = \sum_i p(x|y=i) P(y=i)$$

↓ ↓
Mixture Mixture
component proportion



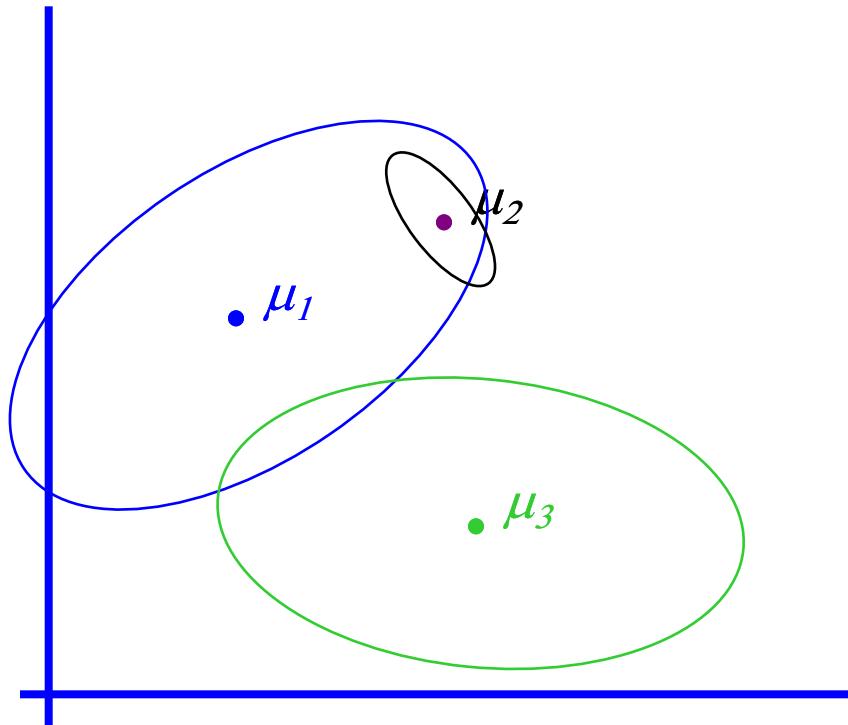
Mixture models

GMM – Gaussian Mixture Model (Multi-modal distribution)

- There are k components
- Component i has a probability of getting picked $p_i = P(y=i)$
- Each component generates data from a Gaussian with mean μ_i and covariance matrix Σ_i

Each data point is generated according to the following recipe:

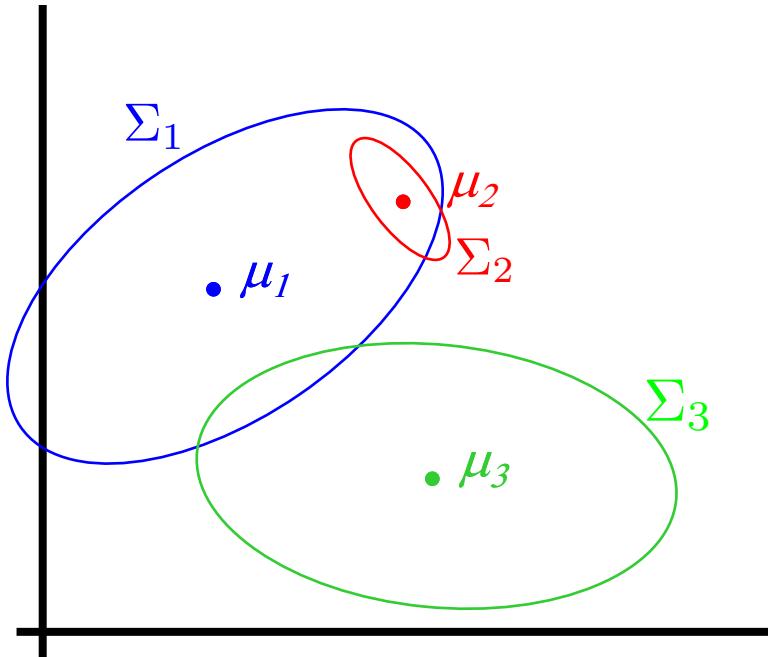
- 1) Pick a component at random:
Choose component i with probability $p_i = P(y=i)$
- 2) Datapoint $x \sim N(\mu_i, \Sigma_i)$



Mixture models (Gaussian)

$$x_1, \dots, x_m \sim p(x) = \sum_{i=1}^k p(x|Y=i)P(Y=i)$$

Mixture component **Mixture proportion, p_i**



Gaussian mixture model

$$p(x|Y=i) \sim \mathcal{N}(\mu_i, \Sigma_i)$$

Parameters: $\{p_i, \mu_i, \Sigma_i\}_{i=1}^K$

- How to estimate parameters? Max Likelihood
But don't know labels Y (recall Gaussian Bayes classifier)

Expectation-Maximization (EM)

A general algorithm to deal with hidden data, but we will study it in the context of unsupervised learning (hidden labels)

- No need to choose step size as in Gradient methods.
- EM is an Iterative algorithm with two linked steps:
 - E-step: fill-in hidden data (Y) using inference
 - M-step: apply standard MLE/MAP method to estimate parameters
$$\{p_i, \mu_i, \Sigma_i\}_{i=1}^k$$
- Not guaranteed to converge to global optimum. BUT...
- This procedure monotonically improves the marginal likelihood of observed data (or leaves it unchanged). Thus it always converges to a local optimum of the likelihood.

EM for spherical, same variance GMMs same mixture proportions

Initialize: $\mu_1, \mu_2, \dots, \mu_K$ randomly

E-step

Compute “expected” classes of all datapoints for each class

$$P(y = i | x_j, \mu_1, \dots, \mu_k) \propto \exp\left(-\frac{1}{2\sigma^2} \|x_j - \mu_i\|^2\right) P(y = i)$$

In K-means “E-step”
we do hard assignment

EM does soft assignment

M-step

Compute Max. like μ given our data’s class membership distributions (weights)

Iterate.

EM for spherical, same variance GMMs same mixture proportions

Initialize: $\mu_1, \mu_2, \dots, \mu_K$ randomly

E-step

Compute “expected” classes of all datapoints for each class

$$P(y = i|x_j, \mu_1, \dots, \mu_k) \propto \exp\left(-\frac{1}{2\sigma^2} \|x_j - \mu_i\|^2\right) P(y = i)$$

In K-means “E-step”
we do hard assignment

EM does soft assignment

M-step

Compute Max. like μ given our data’s class membership distributions (weights)

$$\mu_i = \frac{\sum_{j=1}^m P(y = i|x_j) x_j}{\sum_{j=1}^m P(y = i|x_j)}$$

Exactly same as MLE with
weighted data

Iterate.

EM for general GMMs

Iterate. On iteration t let our estimates be

$$\lambda_t = \{ \mu_1^{(t)}, \mu_2^{(t)} \dots \mu_k^{(t)}, \Sigma_1^{(t)}, \Sigma_2^{(t)} \dots \Sigma_k^{(t)}, p_1^{(t)}, p_2^{(t)} \dots p_k^{(t)} \}$$

$p_i^{(t)}$ is shorthand for estimate of $P(y=i)$ on t 'th iteration

E-step

Compute “expected” classes of all datapoints for each class

$$P(y = i | x_j, \lambda_t) \propto p_i^{(t)} p(x_j | \mu_i^{(t)}, \Sigma_i^{(t)})$$

Just evaluate a Gaussian at x_j

M-step

Compute MLEs given our data's class membership distributions (weights)

EM for general GMMs

Iterate. On iteration t let our estimates be

$$\lambda_t = \{ \mu_1^{(t)}, \mu_2^{(t)} \dots \mu_k^{(t)}, \Sigma_1^{(t)}, \Sigma_2^{(t)} \dots \Sigma_k^{(t)}, p_1^{(t)}, p_2^{(t)} \dots p_k^{(t)} \}$$

$p_i^{(t)}$ is shorthand for estimate of $P(y=i)$ on t 'th iteration

E-step

Compute “expected” classes of all datapoints for each class

$$P(y = i | x_j, \lambda_t) \propto p_i^{(t)} p(x_j | \mu_i^{(t)}, \Sigma_i^{(t)})$$

Just evaluate a Gaussian at x_j

M-step

Compute MLEs given our data's class membership distributions (weights)

$$\mu_i^{(t+1)} = \frac{\sum_j P(y = i | x_j, \lambda_t) x_j}{\sum_j P(y = i | x_j, \lambda_t)}$$

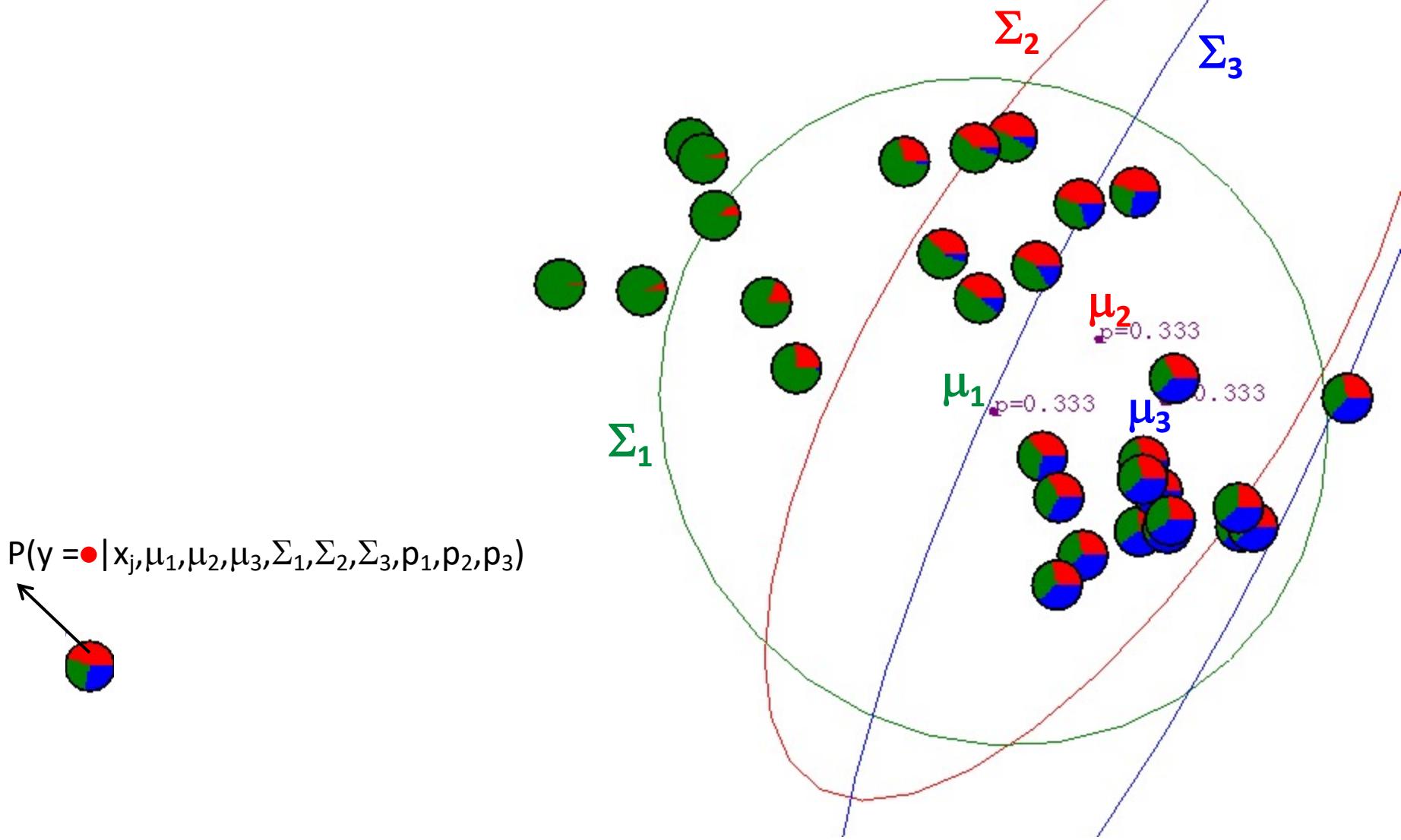
$$\Sigma_i^{(t+1)} = \frac{\sum_j P(y = i | x_j, \lambda_t) (x_j - \mu_i^{(t+1)}) (x_j - \mu_i^{(t+1)})^T}{\sum_j P(y = i | x_j, \lambda_t)}$$

$$p_i^{(t+1)} = \frac{\sum_j P(y = i | x_j, \lambda_t)}{m}$$

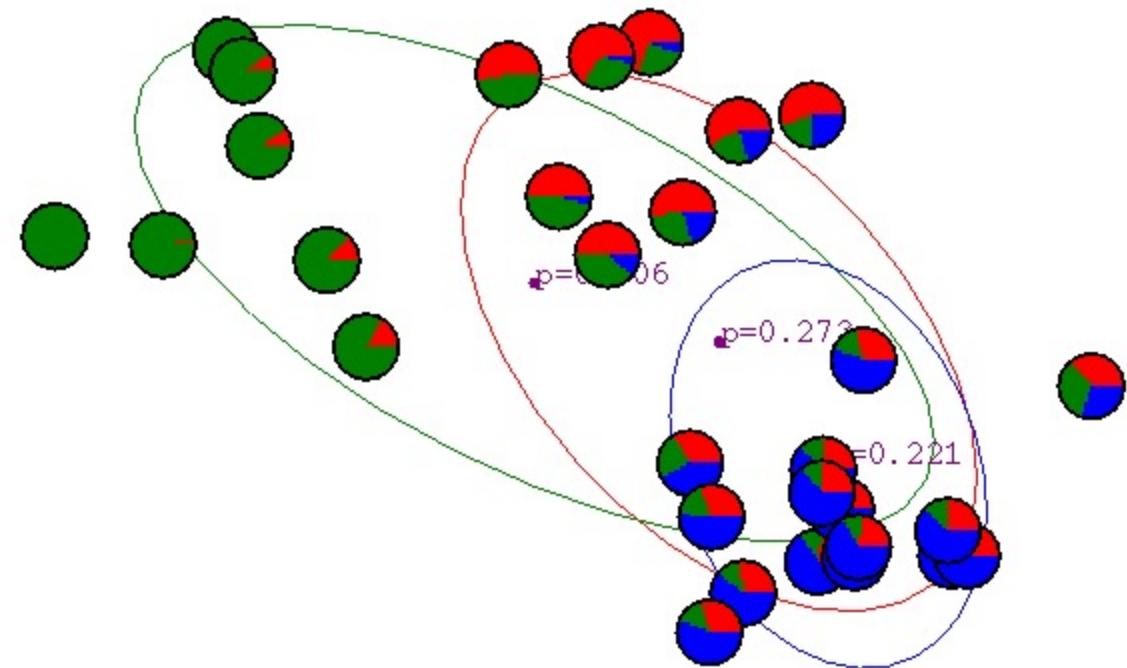
$m = \#$ data points

Iterate.

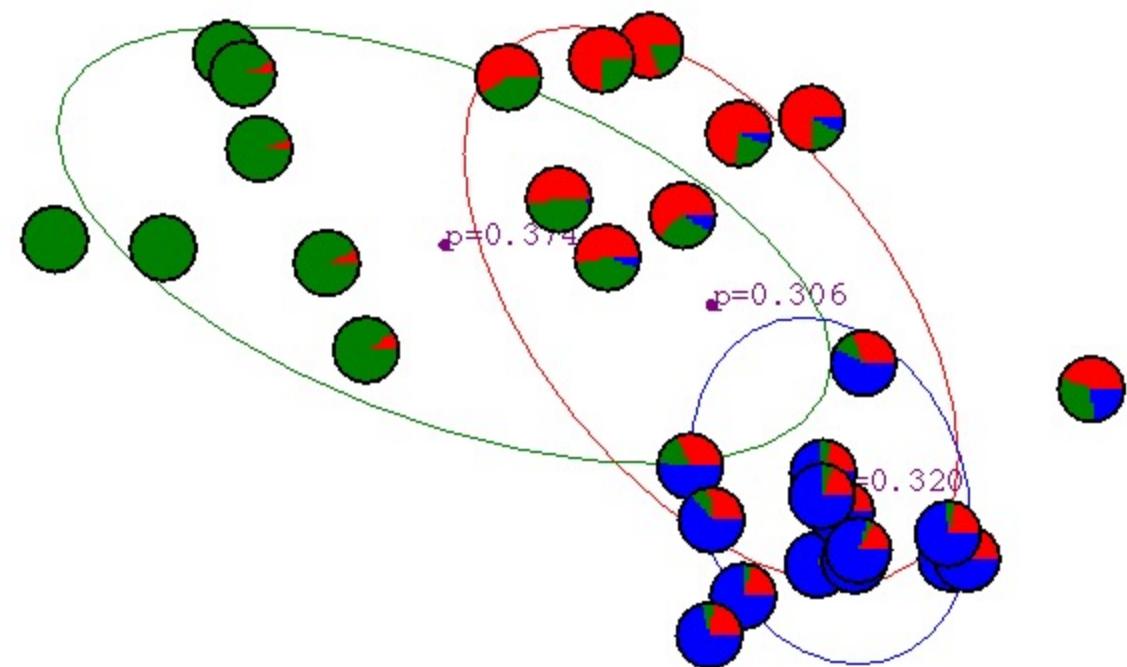
EM for general GMMs: Example



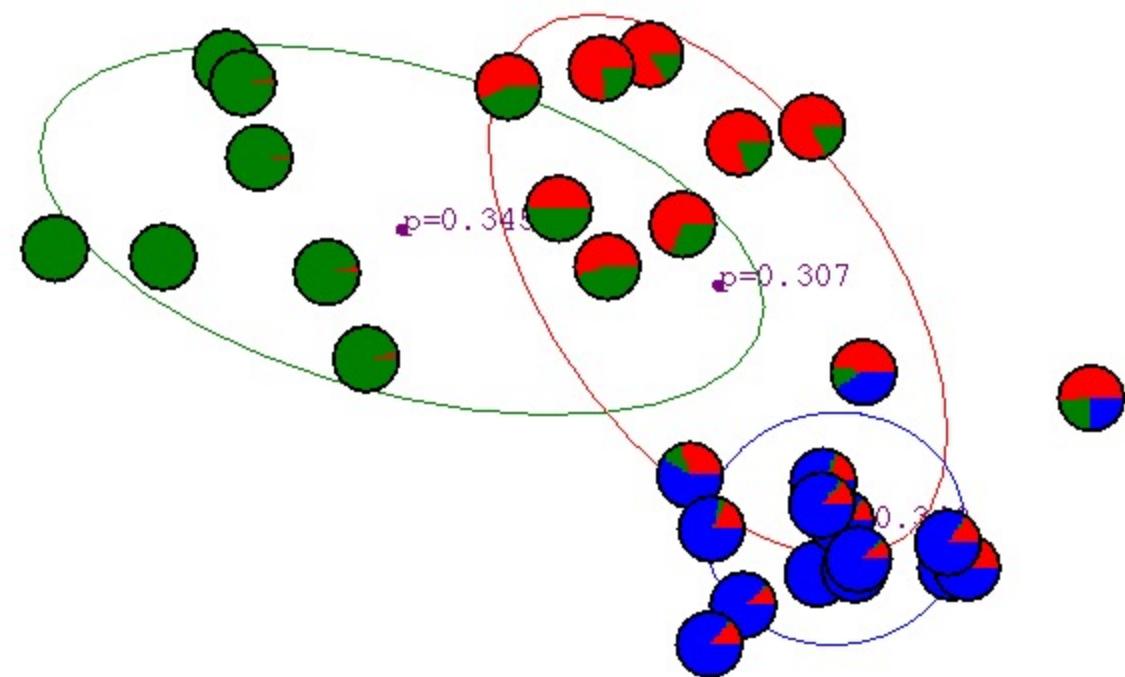
After 1st iteration



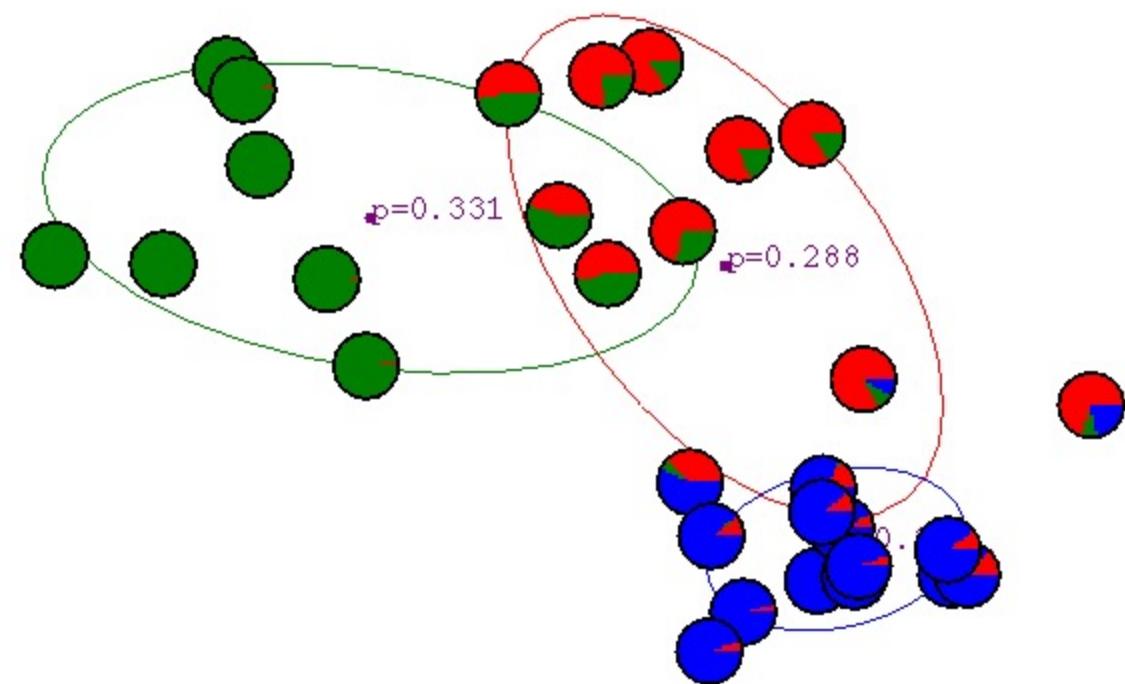
After 2nd iteration



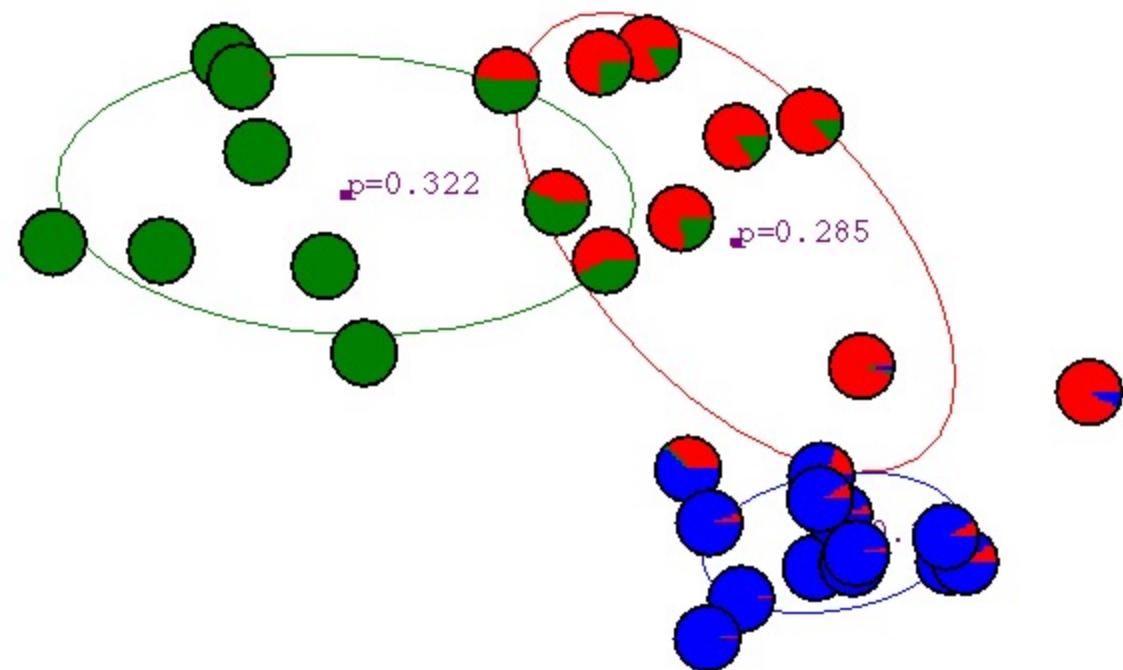
After 3rd iteration



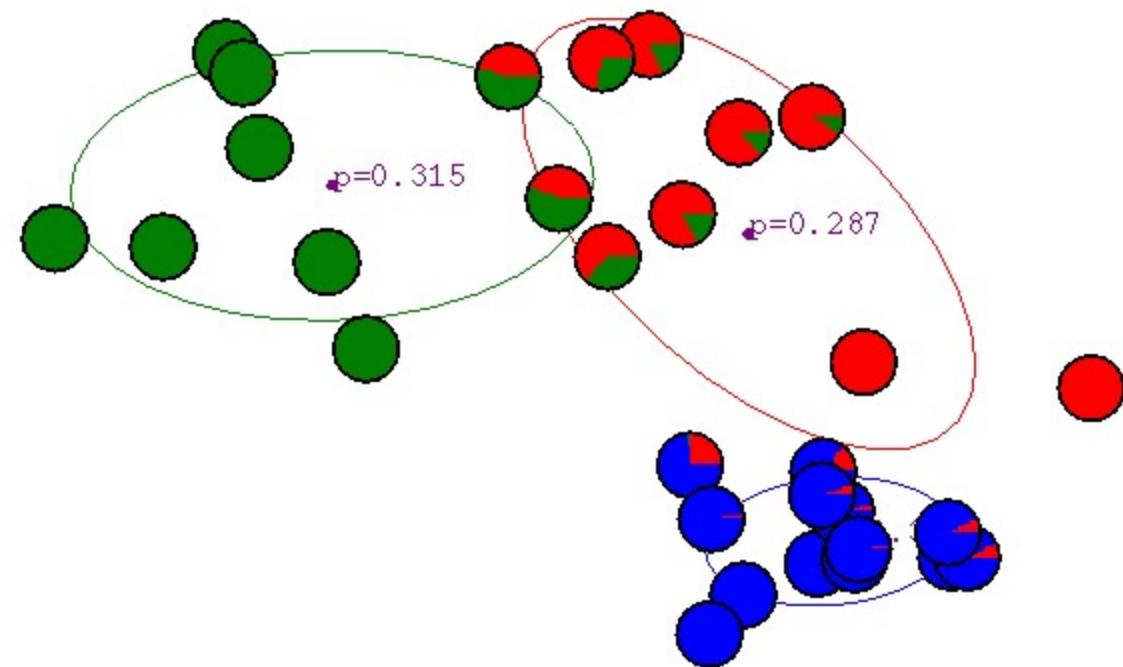
After 4th iteration



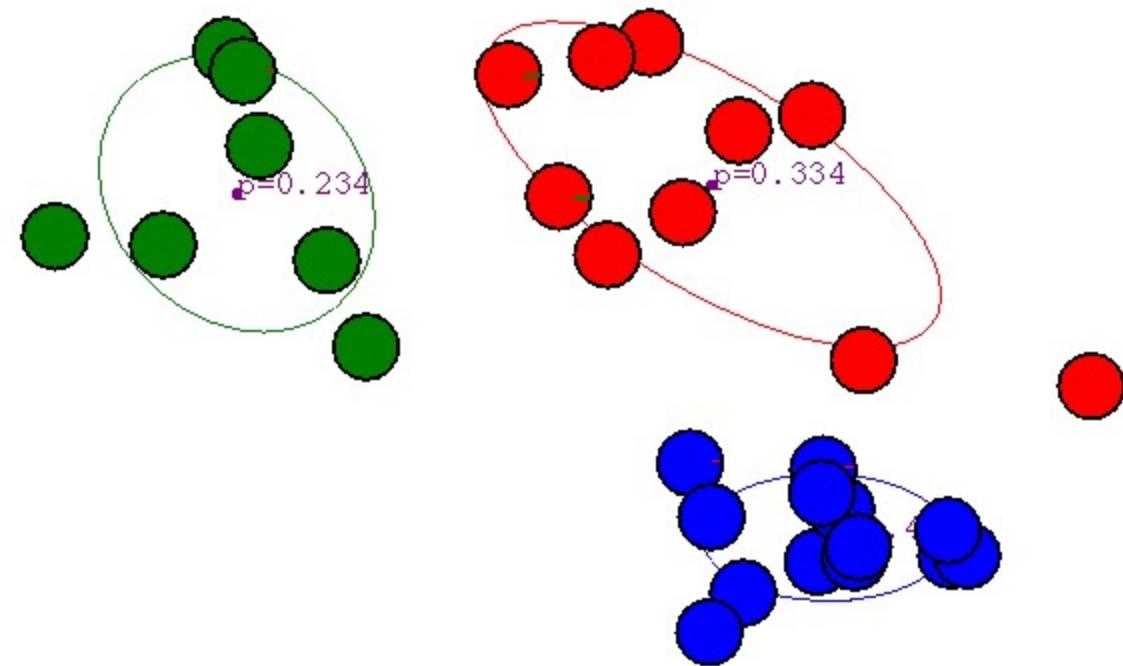
After 5th iteration



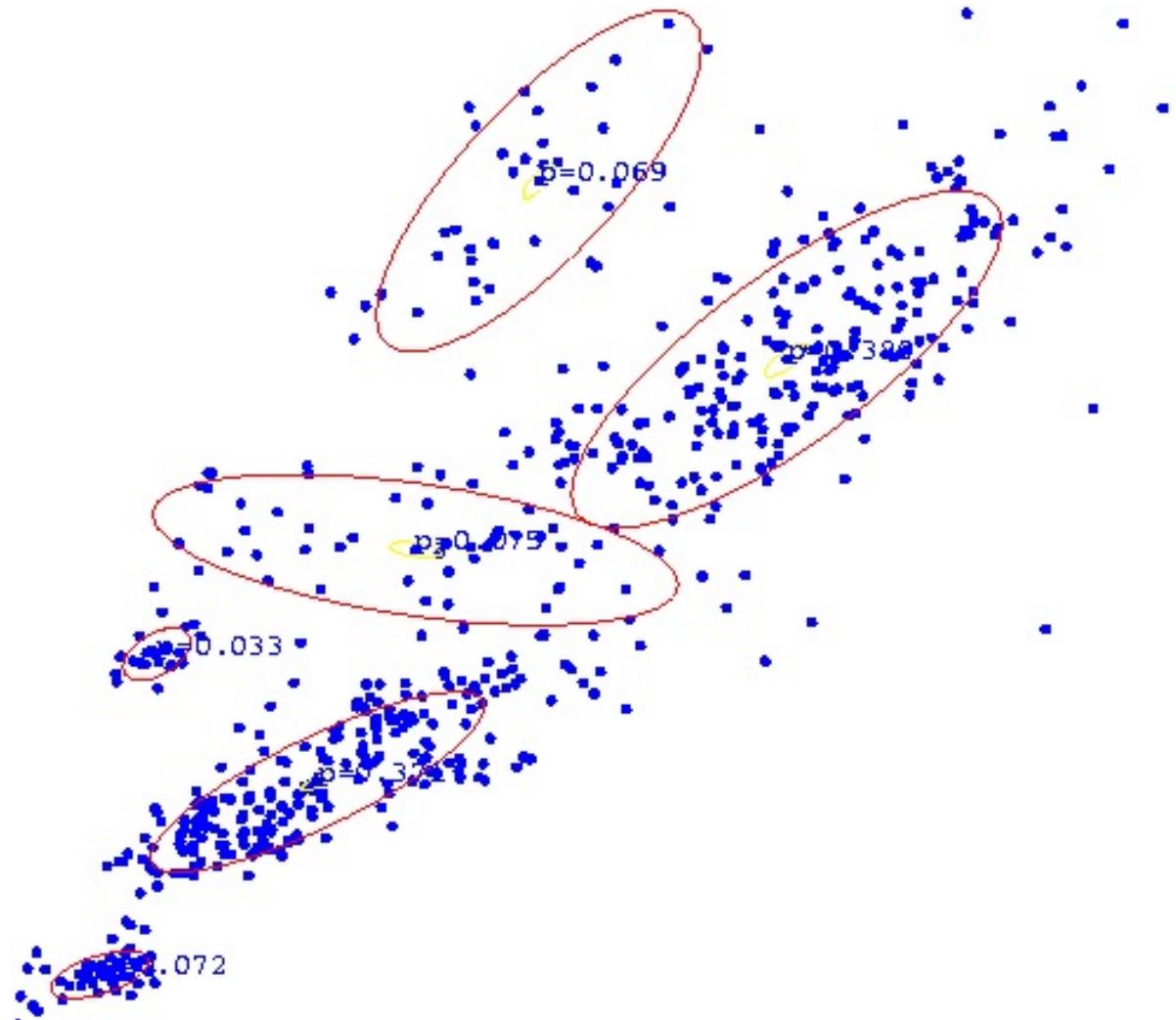
After 6th iteration



After 20th iteration



GMM clustering of assay data



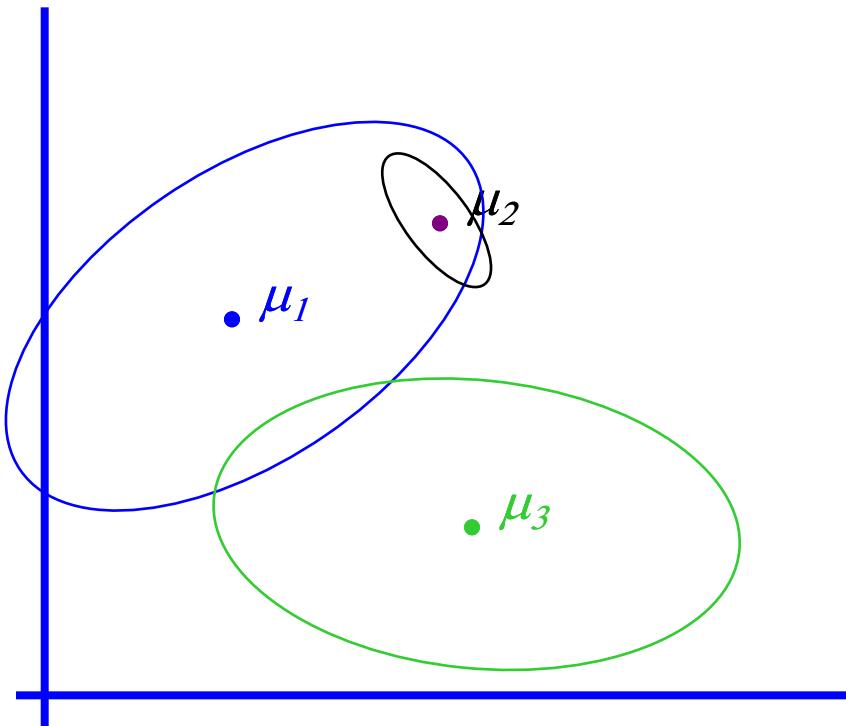
General GMM

GMM – Gaussian Mixture Model (Multi-modal distribution)

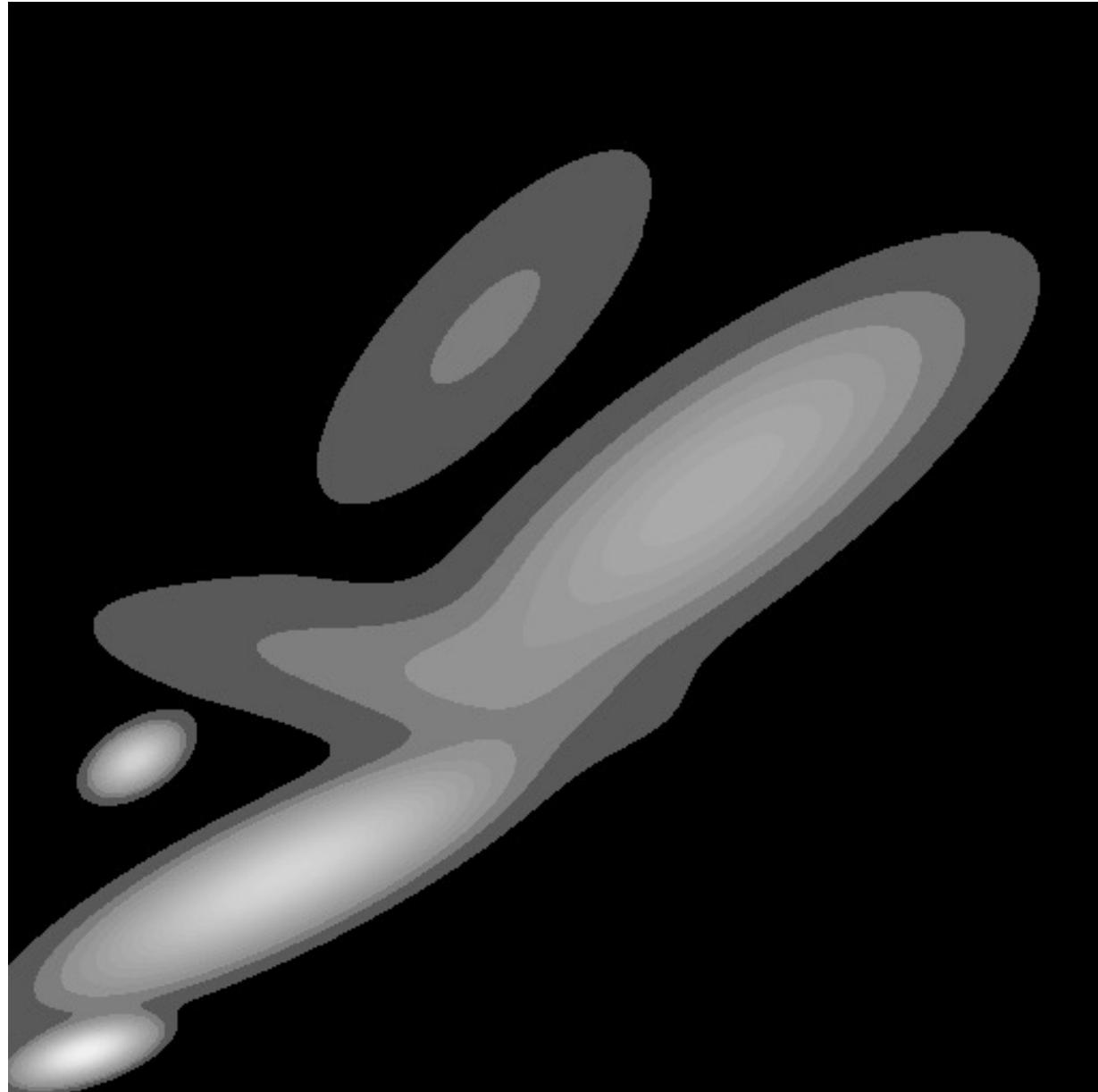
$$p(x) = \sum_i p(x|y=i) P(y=i)$$

↓ ↓
Mixture Mixture
component proportion

$$p(x|y=i) \sim N(\mu_i, \Sigma_i)$$

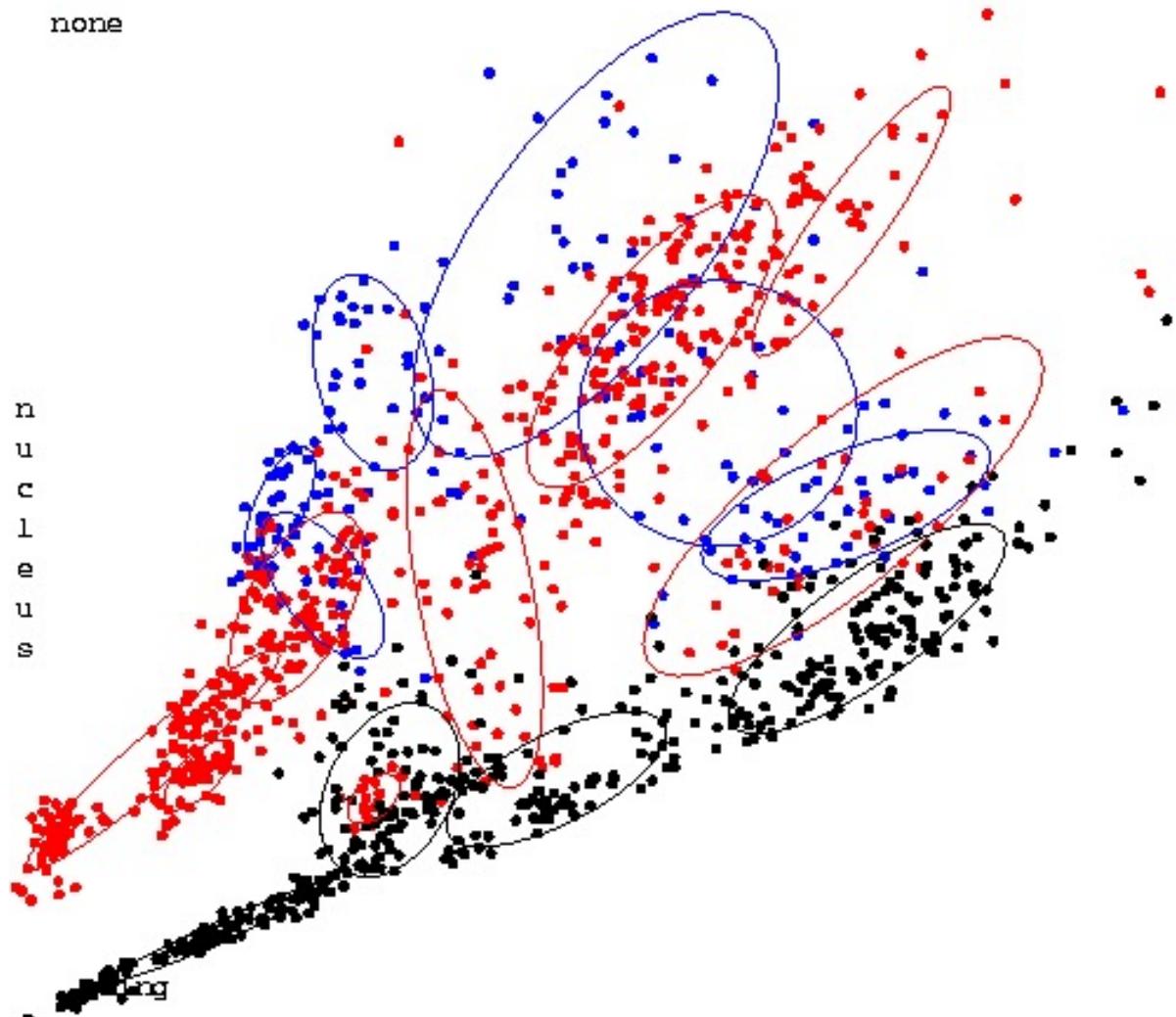


Resulting Density Estimator

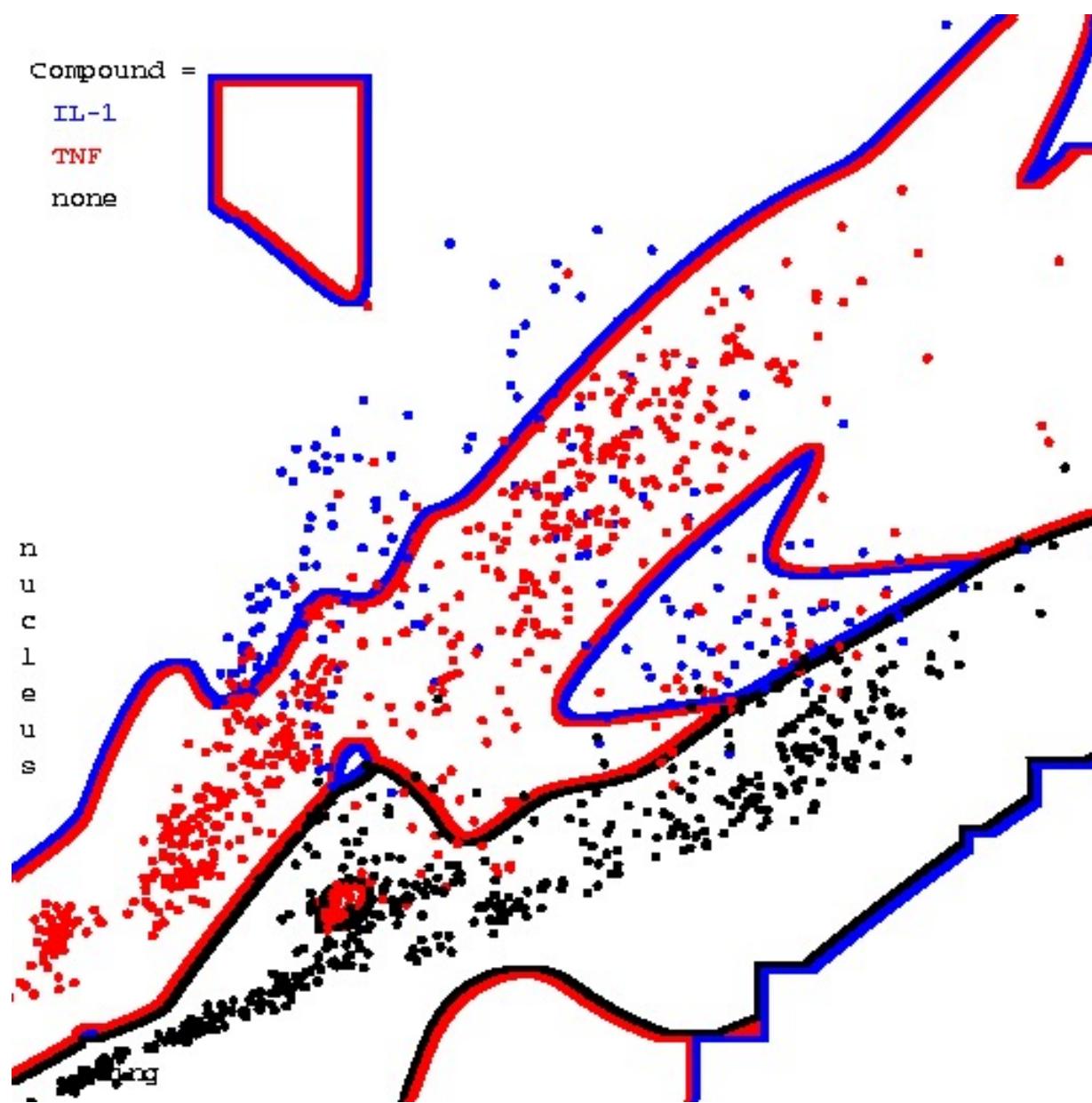


Three classes of assay (each learned with it's own mixture model)

Compound =
IL-1
TNF
none



Resulting Bayes Classifier

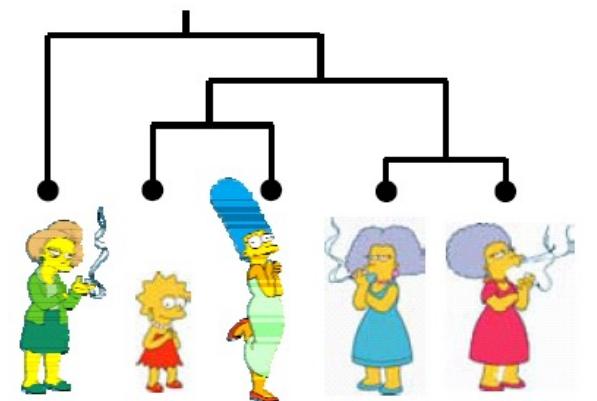
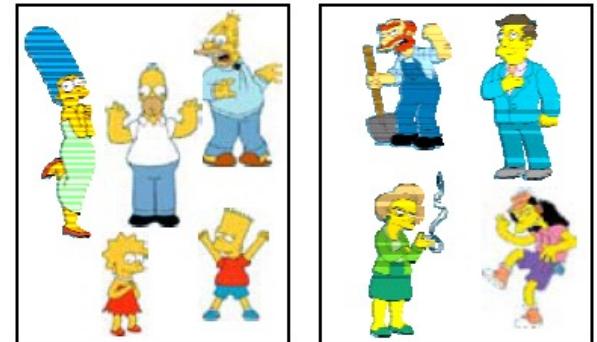


Summary: EM Algorithm

- A way of maximizing likelihood function for hidden variable models. Finds MLE of parameters when the original (hard) problem can be broken up into two (easy) pieces:
 1. Estimate some “missing” or “unobserved” data from observed data and current parameters.
 2. Using this “complete” data, find the maximum likelihood parameter estimates.
- Alternate between filling in the latent variables using the best guess (posterior) and updating the parameters based on this guess:
 1. E-step: soft cluster assignment for each data point
 2. M-step: update parameters of each mixture component
- EM can get stuck in local minima.
- BUT very popular in practice.

Clustering Algorithms

- Partition algorithms
 - K means clustering
 - Mixture-Model based clustering
- Hierarchical algorithms
 - Single-linkage
 - Average-linkage
 - Complete-linkage
 - Centroid-based



Hierarchical Clustering

- Bottom-Up Agglomerative Clustering

Starts with each object in a separate cluster, and repeat:

- Joins the most similar pair of clusters,
- Update the similarity of the new cluster to others until there is only one cluster.

Greedy - less accurate but simple to implement



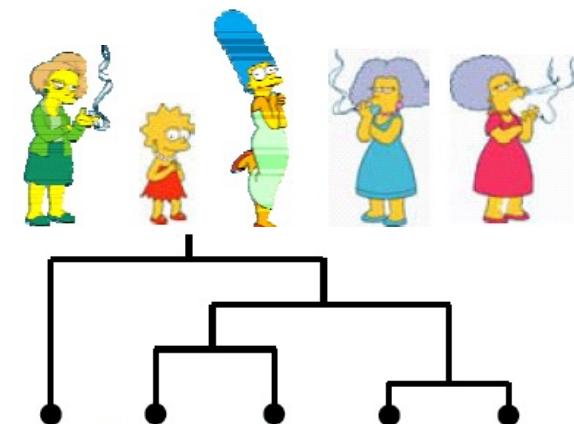
- Top-Down divisive

Starts with all the data in a single cluster, and repeat:

- Split each cluster into two using a partition algorithm

Until each object is a separate cluster.

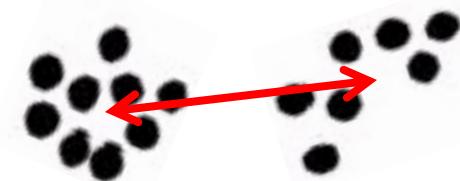
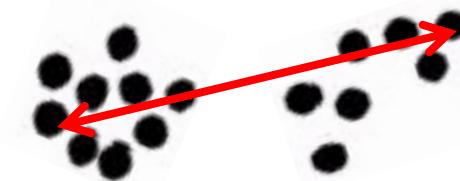
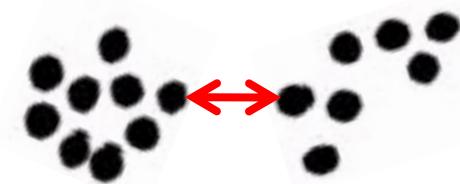
More accurate but complex to implement



Bottom-up Agglomerative clustering

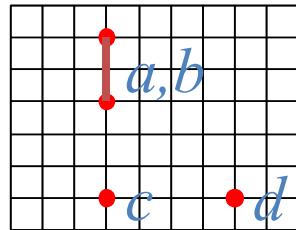
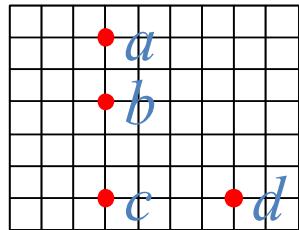
Different algorithms differ in how the similarities are defined (and hence updated) between two clusters

- Single-Linkage
 - Nearest Neighbor: similarity between their closest members.
- Complete-Linkage
 - Furthest Neighbor: similarity between their furthest members.
- Centroid
 - Similarity between the centers of gravity
- Average-Linkage
 - Average similarity of all cross-cluster pairs.

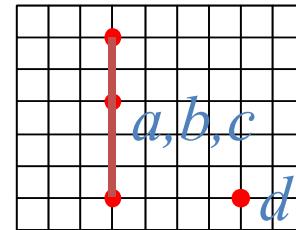


Single-Linkage Method

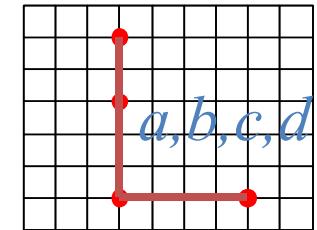
Euclidean Distance



(1)



(2)



(3)

	<i>b</i>	<i>c</i>	<i>d</i>
<i>a</i>	2	5	6
<i>b</i>		3	5
<i>c</i>			4

A distance matrix is shown with points *a, b, c, d* as rows and columns. The diagonal elements are 0. The off-diagonal elements are: *a, b*: 2, *a, c*: 5, *a, d*: 6; *b, c*: 3, *b, d*: 5; *c, d*: 4. The cell for *a, b* is circled in red, and the cells for *b, c* and *b, d* are highlighted with blue boxes.

	<i>b</i>	<i>c</i>	<i>d</i>
<i>a</i>	2	5	6
<i>b</i>		3	5
<i>c</i>			4

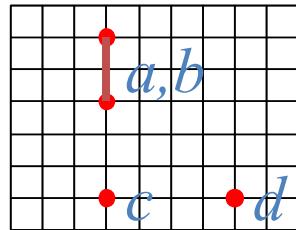
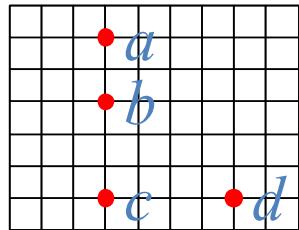
The final merged cluster contains all four points *a, b, c, d*. The distance between the farthest points *a* and *d* is 4.

	<i>d</i>
<i>a, b, c</i>	4

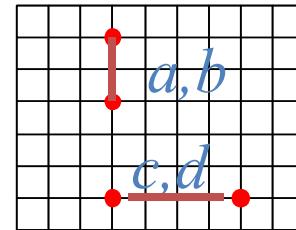
Distance Matrix

Complete-Linkage Method

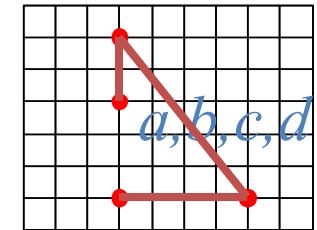
Euclidean Distance



(1)

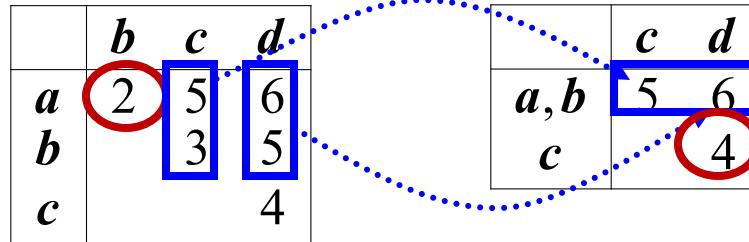


(2)



(3)

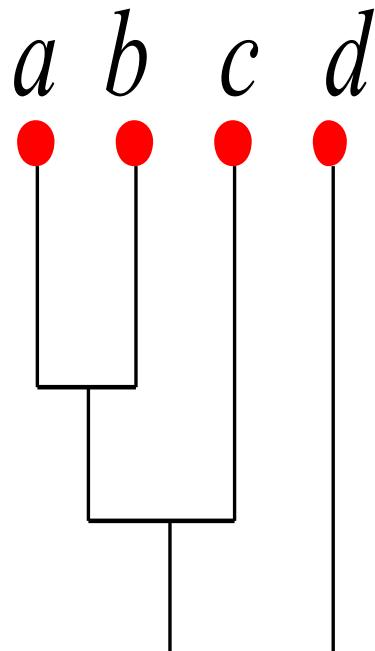
	<i>b</i>	<i>c</i>	<i>d</i>
<i>a</i>	2	5	6
<i>b</i>		3	5
<i>c</i>			4



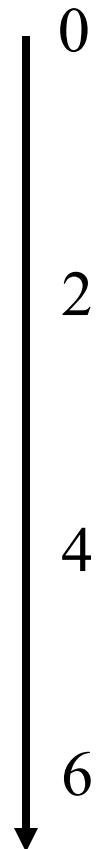
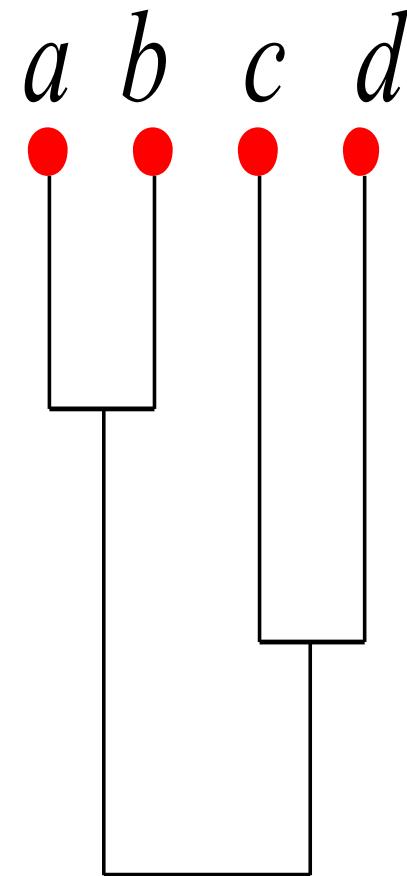
Distance Matrix

Dendograms

Single-Linkage

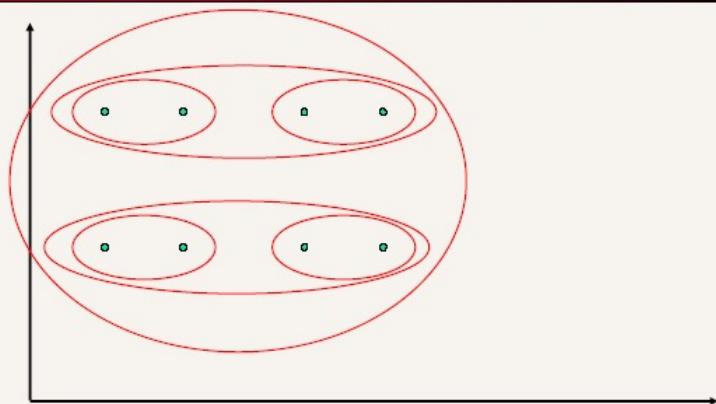


Complete-Linkage

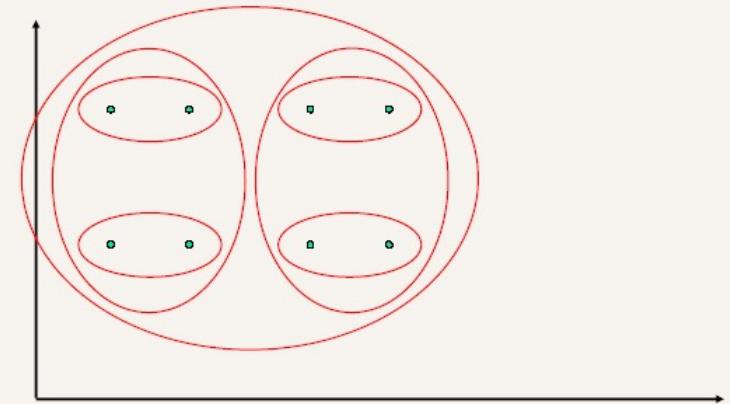


Another Example

Single Link Example



Complete Link Example



Single vs. Complete Linkage

Shape of clusters

Single-linkage

allows anisotropic and
non-convex shapes

Complete-linkage

assumes isotropic, convex
shapes



What you need to know...

- Partition based clustering algorithms
 - K-means
 - Coordinate descent
 - Seeding
 - Choosing K
 - Mixture models
 - EM algorithm
- Hierarchical clustering algorithms
 - Single-linkage
 - Complete-linkage
 - Centroid-linkage
 - Average-linkage