Only covered supervised learning

$X \to \mathbb{R}$ regression

$X \to \{0,1,\dots,k\}$ classification

Training data included labels

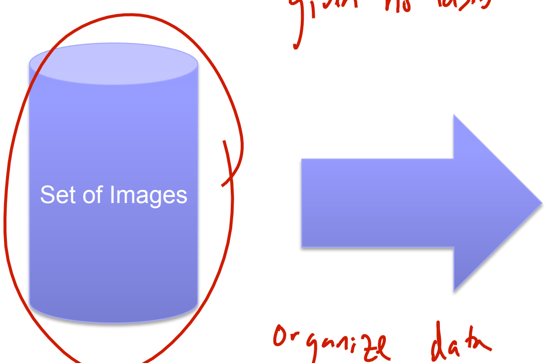
Clustering
K-means

Machine Learning – CSE546
Carlos Guestrin
University of Washington

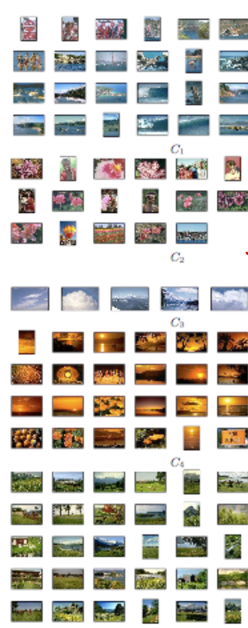
November 4, 2014

Clustering images

given no labels



organize data into themes



beaches

flowers

[Goldberger et al.]

K-means

d-dim vectors

- Randomly initialize k centers or "smartly"
  - $\mu^{(0)} = \mu_1^{(0)}, \ldots, \mu_k^{(0)}$ iteration

Repeat until convergence: no point changes cluster membership

- **Classify**: Assign each point $j \in \{1, \ldots N\}$ to nearest center:
  - $C^{(t)}(j) \leftarrow \arg\min_i ||\mu_i^{(t)} - x_j||^2$

fix μ, OPT C

- **Recenter**: $\mu_i^{(t+1)}$ becomes centroid of its point:
  - $\mu_i^{(t+1)} \leftarrow \arg\min_\mu \sum_{j:C^{(t)}(j)=i} ||\mu - x_j||^2$ sum of points in cluster i
  - Equivalent to $\mu_i \leftarrow$ average of its points!

fix C, OPT μ

$$\mu_i^{(t+1)} = \frac{\sum_{j:C(j)=i} x_j}{|\{j : C(j)=i\}|}$$

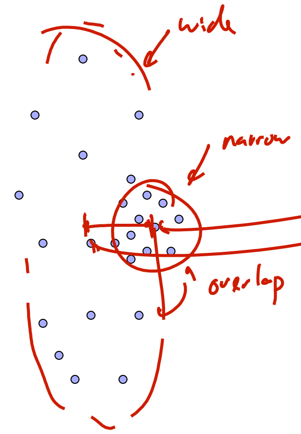
Mixtures of Gaussians

Machine Learning – CSE546
Carlos Guestrin
University of Washington

November 4, 2014

(One) bad case for k-means

- Clusters may overlap
- Some clusters may be "wider" than others



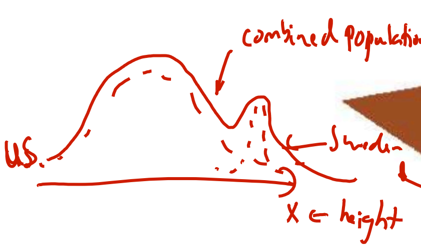
centers not enough to characterize membership

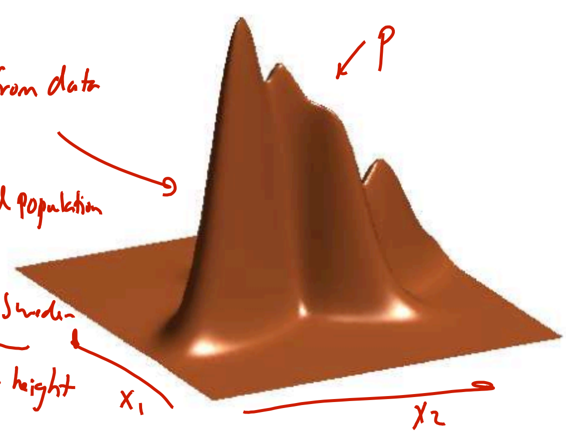
Density Estimation

- Estimate a density based on $x^1,\ldots,x^N$

$x^1,\ldots,x^N \sim p$

learn p from data

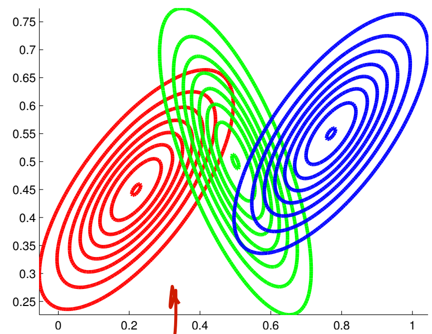




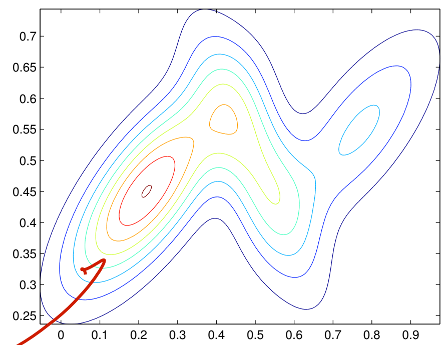
Density as Mixture of Gaussians

- Approximate density with a mixture of Gaussians

Mixture of 3 Gaussians



Contour Plot of Joint Density



original

p sum with weights π_i of each Gaussian

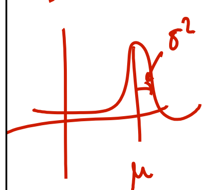
Gaussians in *d* Dimensions

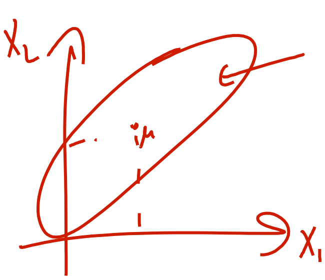
mean vector

covariance matrix

$$P(\mathbf{x}) = \frac{1}{(2\pi)^{d/2} \|\Sigma\|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\mu)^T \Sigma^{-1} (\mathbf{x}-\mu)\right]$$

$\boxed{1D \; P(x) \propto e^{\frac{-(\mu-x)^2}{2\sigma^2}}}$





$$\Sigma = \begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{21} & \sigma_2^2 \end{bmatrix}$$

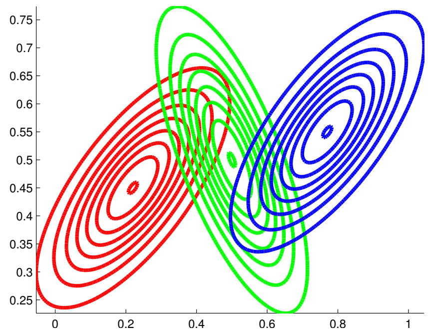
$\sigma_{12} = \sigma_{21}$

Density as Mixture of Gaussians

- Approximate density with a mixture of Gaussians

$\pi_1, \pi_2 \cdots \pi_k$ weights $\sum_{i=1}^{k} \pi_i = 1$

Mixture of 3 Gaussians



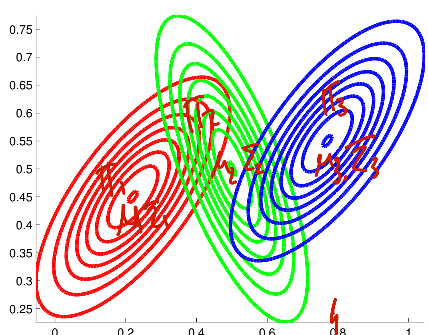
$$p: \quad p(x^j|\pi, \mu, \Sigma) = \;= \sum_{i=1}^{k} \pi_i \, N(x^j | \mu_i, \Sigma_i)$$

In 1D

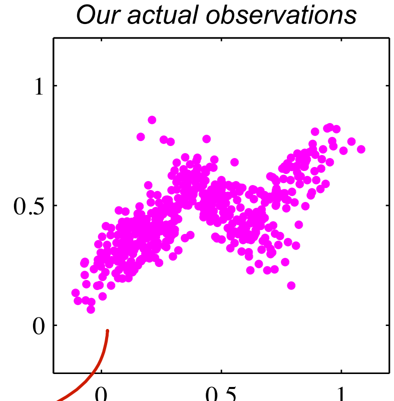
Density as Mixture of Gaussians

- Approximate with density with a mixture of Gaussians

Mixture of 3 Gaussians



Our actual observations



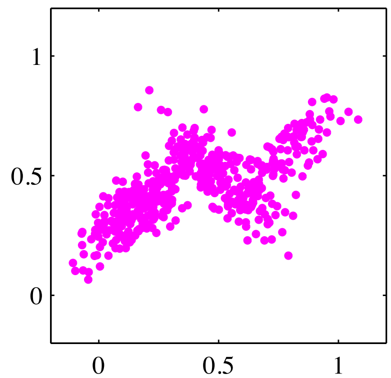
recover original densities How??

C. Bishop, Pattern Recognition & Machine Learning

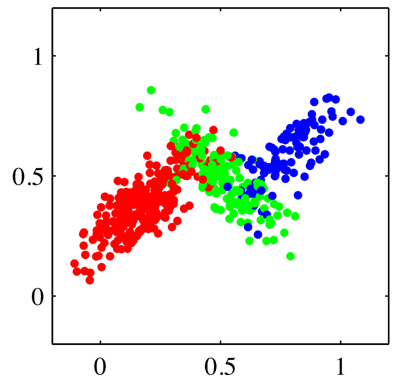
Clustering our Observations

- Imagine we have an assignment of each x^i to a Gaussian

Our actual observations



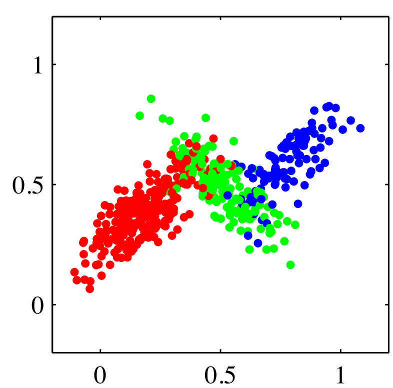
Complete data labeled by true cluster assignments



C. Bishop, Pattern Recognition & Machine Learning

Clustering our Observations

- Imagine we have an assignment of each x^i to a Gaussian



Complete data labeled by true cluster assignments

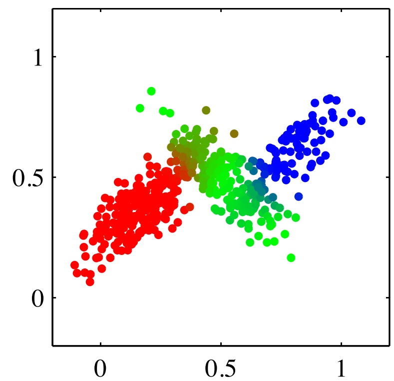
- Introduce latent cluster indicator variable z^i
- Then we have

$$p(x^i|z^i,\pi,\mu,\Sigma) =$$

C. Bishop, Pattern Recognition & Machine Learning

Clustering our Observations

- We must infer the cluster assignments from the observations



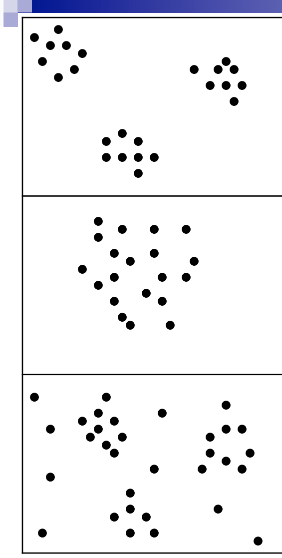
Soft assignments to clusters

- Posterior probabilities of assignments to each cluster *given* model parameters:

$$r_{ik} = p(z^i = k | x^i, \pi, \mu, \Sigma) =$$

C. Bishop, Pattern Recognition & Machine Learning

Unsupervised Learning: not as hard as it looks



Sometimes easy

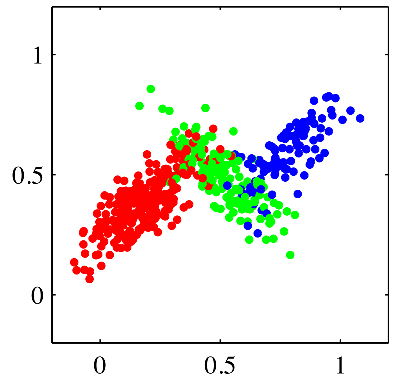
Sometimes impossible

and sometimes in between

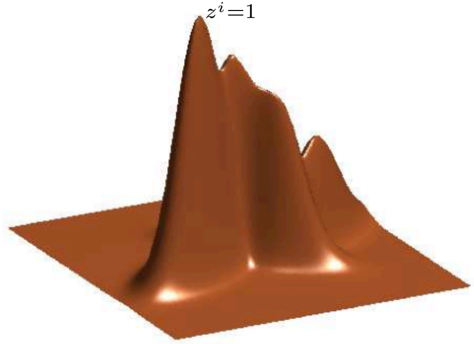
Summary of GMM Concept

- Estimate a density based on $x^1,\ldots,x^N$

$$p(x^i|\pi,\mu,\Sigma) = \sum_{z^i=1}^{K} \pi_{z^i} \mathcal{N}(x^i|\mu_{z^i},\Sigma_{z^i})$$



Complete data labeled by true cluster assignments



Surface Plot of Joint Density, Marginalizing Cluster Assignments

Summary of GMM Components

- Observations $x^i \in \mathbb{R}^d, \quad i = 1,2,\ldots,N$
- Hidden cluster labels $z_i \in \{1,2,\ldots,K\}, \quad i = 1,2,\ldots,N$
- Hidden mixture means $\mu_k \in \mathbb{R}^d, \quad k = 1,2,\ldots,K$
- Hidden mixture covariances $\Sigma_k \in \mathbb{R}^{d\times d}, \quad k = 1,2,\ldots,K$
- Hidden mixture probabilities $\pi_k, \quad \sum_{k=1}^{K} \pi_k = 1$

Gaussian mixture marginal and conditional likelihood :

$$p(x^i|\pi,\mu,\Sigma) = \sum_{z^i=1}^{K} \pi_{z^i}\; p(x^i|z^i,\mu,\Sigma)$$

$$p(x^i|z^i,\mu,\Sigma) = \mathcal{N}(x^i|\mu_{z^i},\Sigma_{z^i})$$

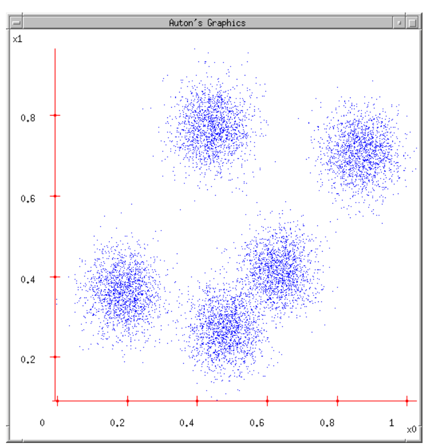
Expectation Maximization

Machine Learning – CSE546
Carlos Guestrin
University of Washington

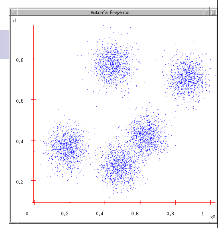
November 6, 2014

Next… back to Density Estimation

What if we want to do density estimation with multimodal or clumpy data?



But we don't see class labels!!!



- MLE:
  - argmax $\prod_i P(z^i, x^i)$
- But we don't know z^i
- Maximize marginal likelihood:
  - argmax $\prod_i P(x^i)$ = argmax $\prod_i \sum_{k=1}^{K} P(z^i=k, x^i)$

Special case: spherical Gaussians and hard assignments

$$P(z^i = k, \mathbf{x}^i) = \frac{1}{(2\pi)^{m/2} \|\Sigma_k\|^{1/2}} \exp\left[-\frac{1}{2}\left(\mathbf{x}^i - \mu_k\right)^T \Sigma_k^{-1} \left(\mathbf{x}^i - \mu_k\right)\right] P(z^i = k)$$

- If P(X|z=k) is spherical, with same σ for all classes:

$$P(\mathbf{x}^i \mid z^i = k) \propto \exp\left[-\frac{1}{2\sigma^2}\left\|\mathbf{x}^i - \mu_k\right\|^2\right]$$

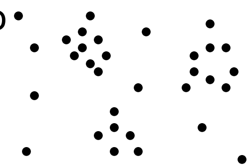
- If each x^i belongs to one class C(i) (hard assignment), marginal likelihood:

$$\prod_{i=1}^{N} \sum_{k=1}^{K} P(\mathbf{x}^i, z^i = k) \propto \prod_{i=1}^{N} \exp\left[-\frac{1}{2\sigma^2}\left\|\mathbf{x}^i - \mu_{C(i)}\right\|^2\right]$$

- Same as K-means!!!

EM: "Reducing" Unsupervised Learning to Supervised Learning

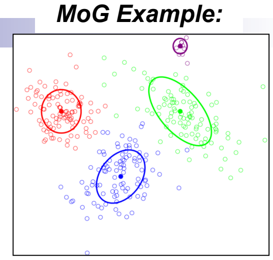
- If we knew assignment of points to classes ➔ Supervised Learning!



- Expectation-Maximization (EM)
  - Guess assignment of points to classes
    - In standard ("soft") EM: each point associated with prob. of being in each class
  - Recompute model parameters
  - Iterate

Generic Mixture Models

MoG Example:



- Observations:

- Parameters:

- Likelihood:

- Ex. z^i = country of origin, x^i = height of i[th] person
  - k^{th} mixture component = distribution of heights in country k

ML Estimate of Mixture Model Params

- Log likelihood

$$L_x(\theta) \triangleq \log p(\{x^i\} \mid \theta) = \sum_i \log \sum_{z^i} p(x^i, z^i \mid \theta)$$

- Want ML estimate

$$\hat{\theta}^{ML} =$$

- Neither convex nor concave and local optima

If "complete" data were observed…

- Assume class labels z^i were observed in addition to x^i

$$L_{x,z}(\theta) = \sum_i \log p(x^i, z^i \mid \theta)$$

- Compute ML estimates
  - Separates over clusters *k*!
- Example: mixture of Gaussians (MoG) $\theta = \{\pi_k, \mu_k, \Sigma_k\}_{k=1}^K$

Iterative Algorithm

- Motivates a coordinate ascent-like algorithm:
  1. Infer missing values z^i given estimate of parameters $\hat{\theta}$
  2. Optimize parameters to produce new $\hat{\theta}$ given "filled in" data z^i
  3. Repeat
- Example: MoG (derivation soon…)
  1. Infer "responsibilities"

$$r_{ik} = p(z^i = k \mid x^i, \hat{\theta}^{(t-1)}) =$$

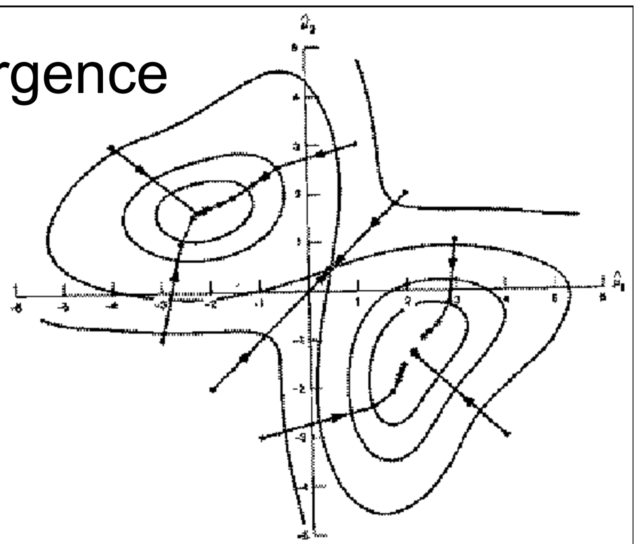
  2. Optimize parameters

$$\max \text{ w.r.t. } \pi_k :$$

$$\max \text{ w.r.t. } \mu_k, \Sigma_k :$$

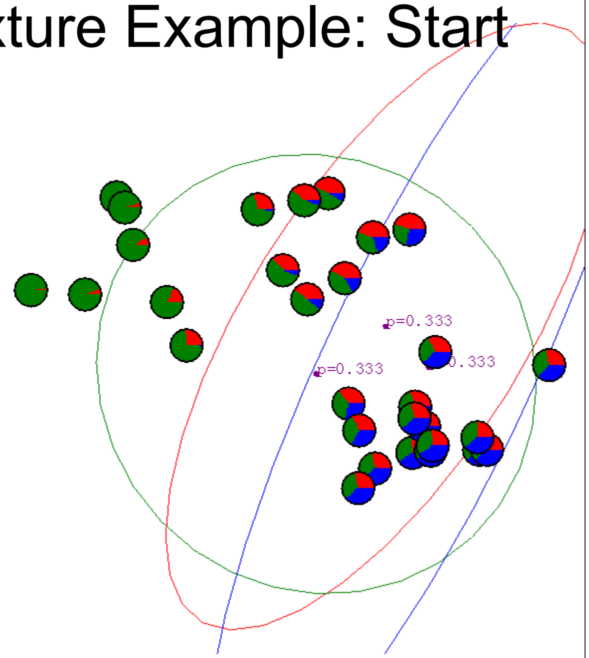
E.M. Convergence

- EM is coordinate ascent on an interesting potential function
- Coord. ascent for bounded pot. func. ➔ convergence to a local optimum guaranteed

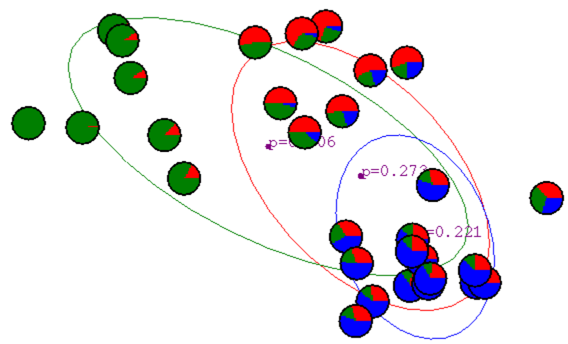


- This algorithm is REALLY USED. And in high dimensional state spaces, too. E.G. Vector Quantization for Speech Data

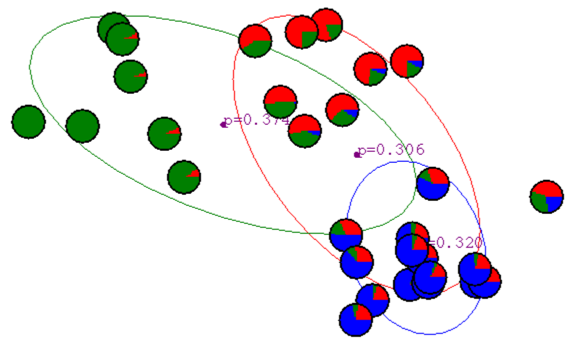
Gaussian Mixture Example: Start



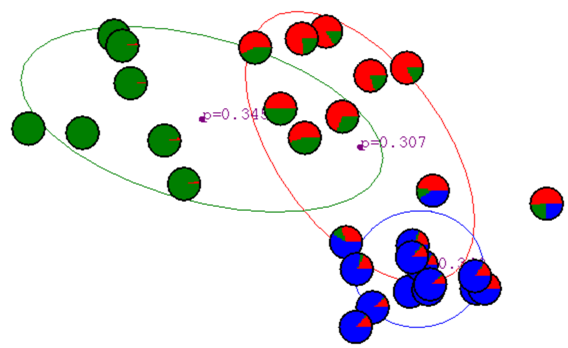
After first iteration



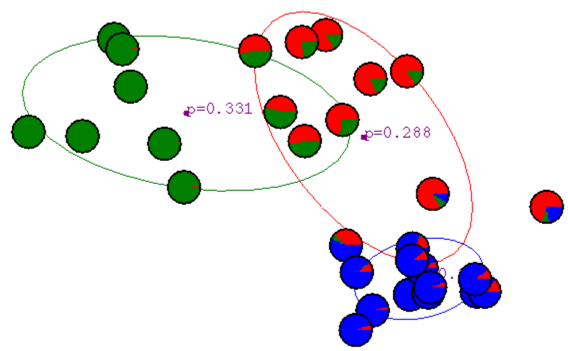
After 2nd iteration



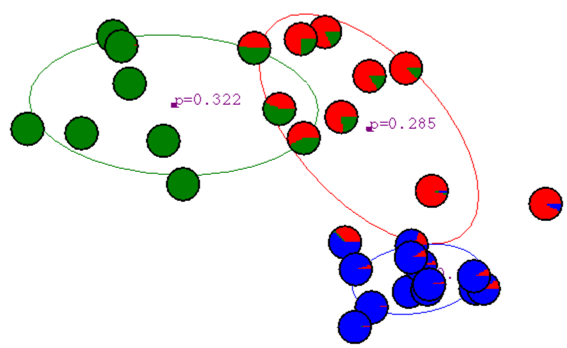
After 3rd iteration



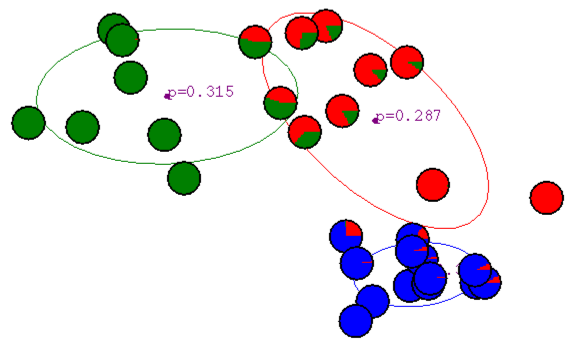
After 4th iteration



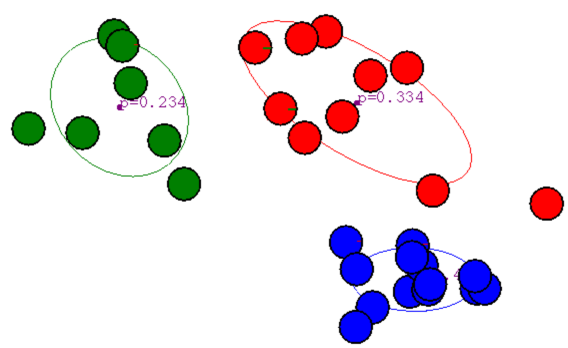
After 5th iteration



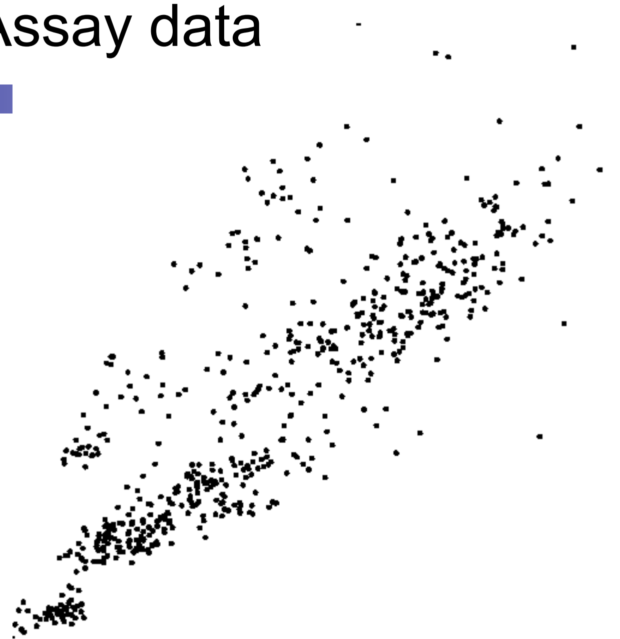
After 6th iteration



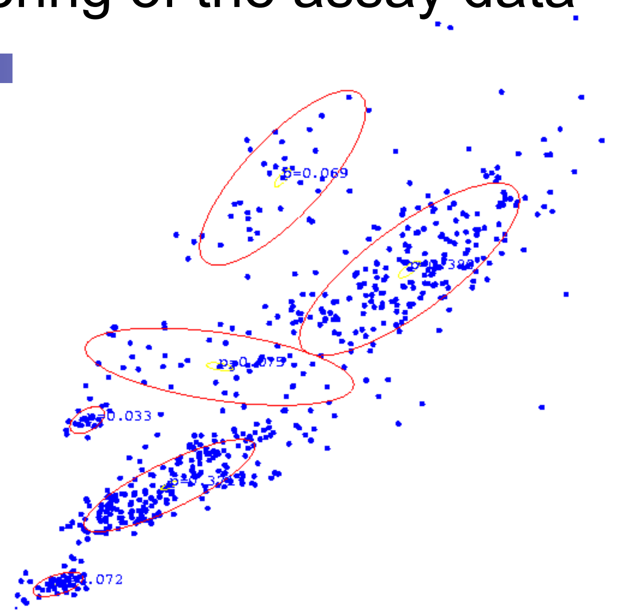
After 20th iteration



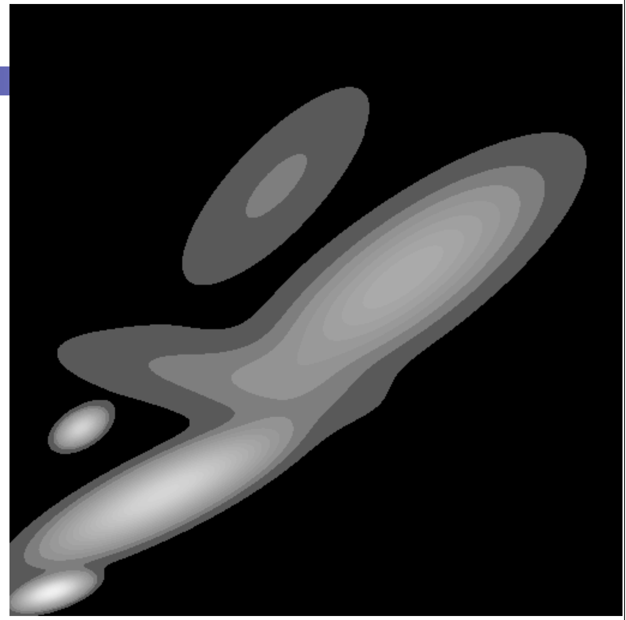
Some Bio Assay data



GMM clustering of the assay data



Resulting Density Estimator



E.M.: The General Case

- E.M. widely used beyond mixtures of Gaussians
  - The recipe is the same…
- Expectation Step: Fill in missing data, given current values of parameters, $\theta^{(t)}$
  - If variable *y* is missing (could be many variables)
  - Compute, for each data point $\mathbf{x}^j$, for each value *i* of *y*:
    - $P(y=i|\mathbf{x}^j,\theta^{(t)})$
- Maximization step: Find maximum likelihood parameters for (weighted) "completed data":
  - For each data point $\mathbf{x}^j$, create *k* weighted data points
    -
  - Set $\theta^{(t+1)}$ as the maximum likelihood parameter estimate for this weighted data
- Repeat

Initialization

- In mixture model case where $y^i = \{z^i, x^i\}$ there are many ways to initialize the EM algorithm
- Examples:
  - Choose K observations at random to define each cluster. Assign other observations to the nearest "centriod" to form initial parameter estimates
  - Pick the centers sequentially to provide good coverage of data
  - Grow mixture model by splitting (and sometimes removing) clusters until K clusters are formed
- Can be quite important to quality of solution in practice

What you should know

- K-means for clustering:
  - algorithm
  - converges because it's coordinate ascent
- EM for mixture of Gaussians:
  - How to "learn" maximum likelihood parameters (locally max. like.) in the case of unlabeled data
- Remember, E.M. can get stuck in local minima, and empirically it <u>DOES</u>
- EM is coordinate ascent