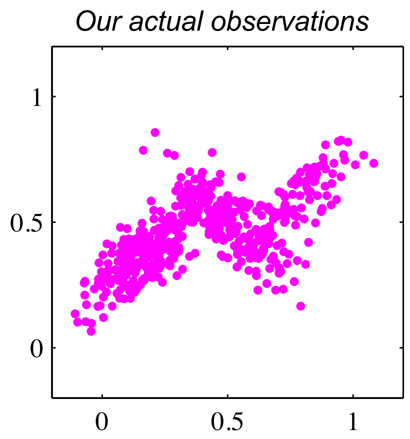
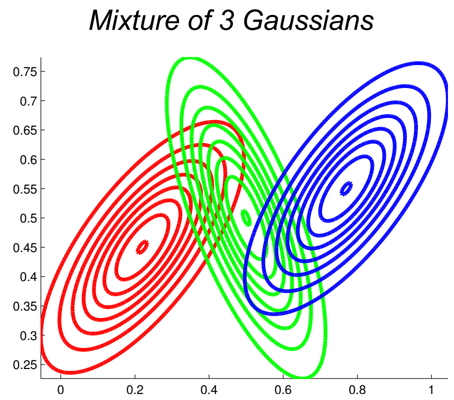
Expectation Maximization for Mixtures of Gaussians

CS229: Machine Learning
Carlos Guestrin
Stanford University

Learning a Mixture of Gaussians





Summary of GMM Components

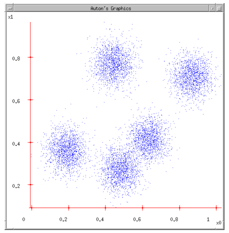
- Observations $x^i \in \mathbb{R}^d, \quad i = 1, 2, \ldots, N$
- Hidden cluster labels $z_i \in \{1, 2, \ldots, K\}, \quad i = 1, 2, \ldots, N$
- Hidden mixture means $\mu_k \in \mathbb{R}^d, \quad k = 1, 2, \ldots, K$
- Hidden mixture covariances $\Sigma_k \in \mathbb{R}^{d \times d}, \quad k = 1,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

Gaussian mixture marginal and conditional likelihood **:**

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

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

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 $\prod_i \sum_k P(z^i=k,x^i)$

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:

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

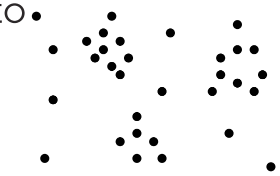
- If each x^i belongs to one class C(i) (hard assignment), marginal likelihood:

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

- Same as K-means!!!

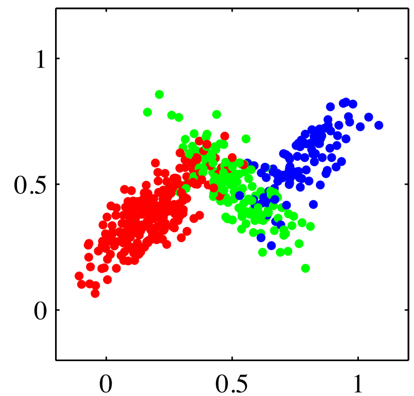
EM: "Reducing" Unsupervised Learning to Supervised Learning

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



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

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



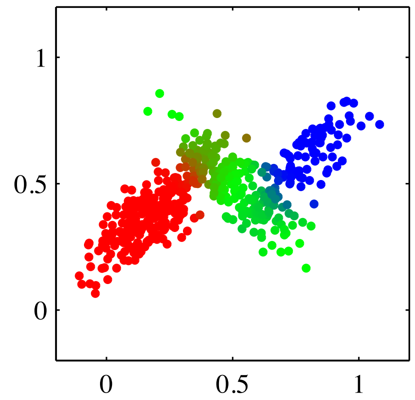
Complete data labeled by true cluster assignments

- Introduce latent cluster indicator variable z^i

- Then we have

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

Expectation: infer cluster assignments from observations



Soft assignments to clusters

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

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

ML Estimate of Mixture Model Params

- Log likelihood

$$L_x(\theta) \triangleq \log p(\{x^i\} \mid \theta) = \sum_{i=1}^{N} \log \sum_{j=1}^{K} p(x^i, z = j \mid \theta)$$

- Want ML estimate

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

- Neither convex nor concave and local optima

Maximization: If "complete" data were observed…

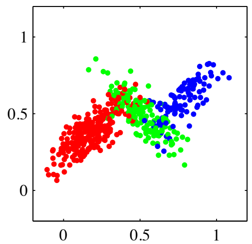
- Assume class labels z^i were observed in addition to x^i

$$L_{x,z}(\theta) = \sum_{i=1}^{N} \log p(x^i, z^i \mid \theta)$$

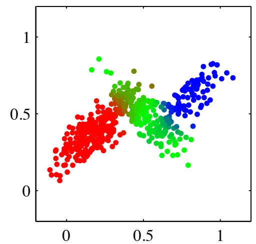
- Compute ML estimates
  - Separates over clusters *k*!

- Example: mixture of Gaussians (MoG) $\theta = \{\pi_k, \mu_k, \Sigma_k\}_{k=1}^{K}$

Maximization: if inferred cluster assignments from observations



Complete data labeled by true cluster assignments



Soft assignments to clusters

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

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

Expectation-Maximization 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
  1. Infer "responsibilities"

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

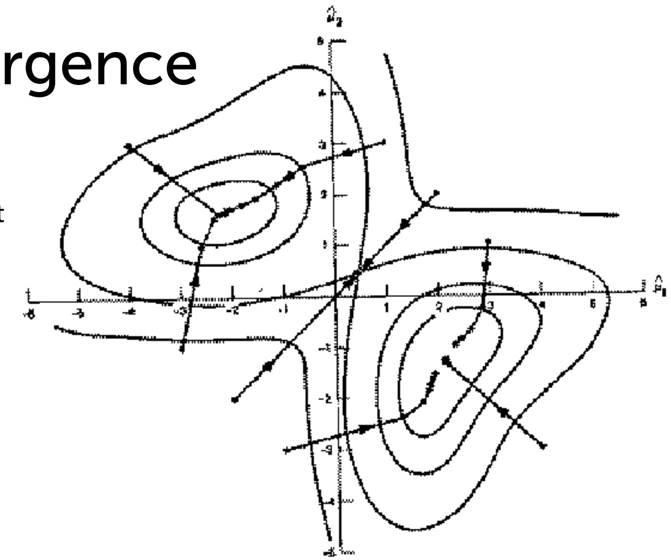
  2. Optimize parameters

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

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

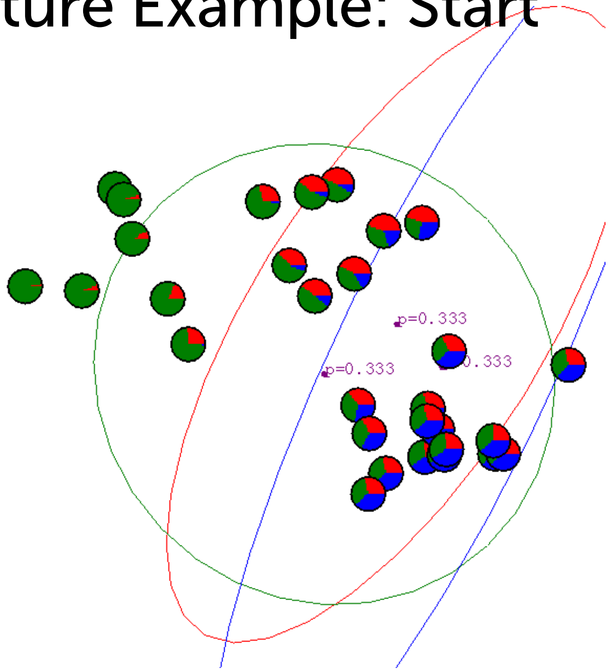
E.M. Convergence

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

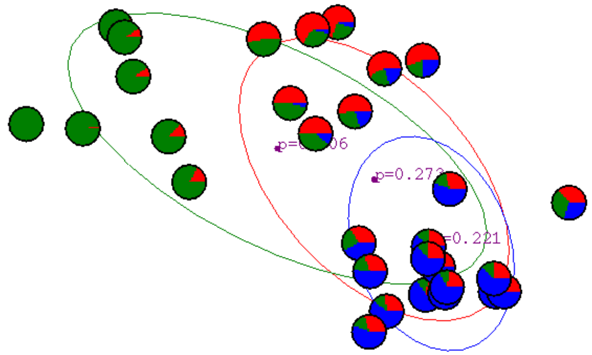


- This algorithm is REALLY USED. And in high dimensional state spaces, too.

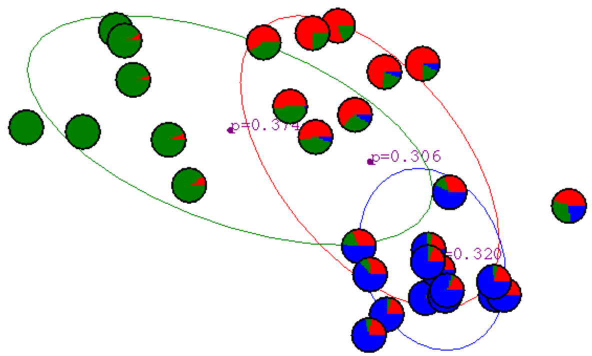
Gaussian Mixture Example: Start



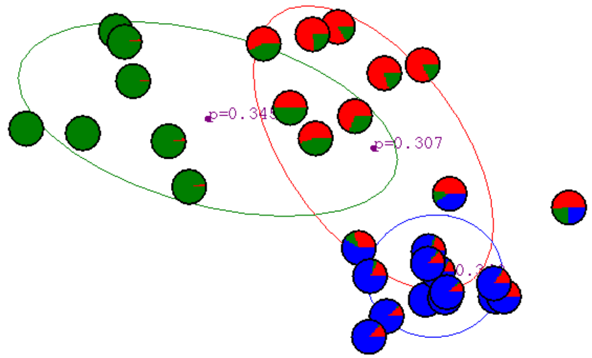
After first iteration



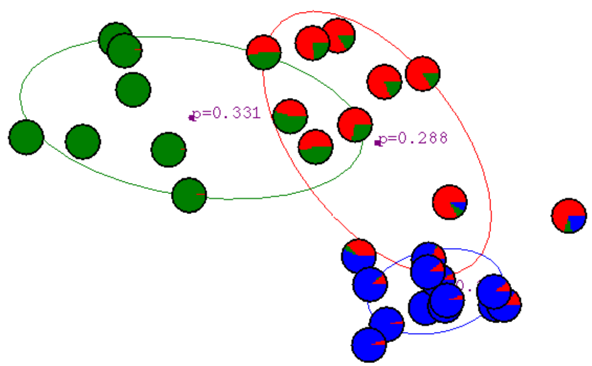
After 2nd iteration



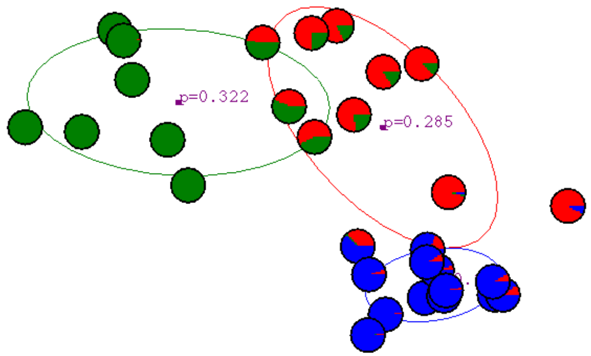
After 3rd iteration



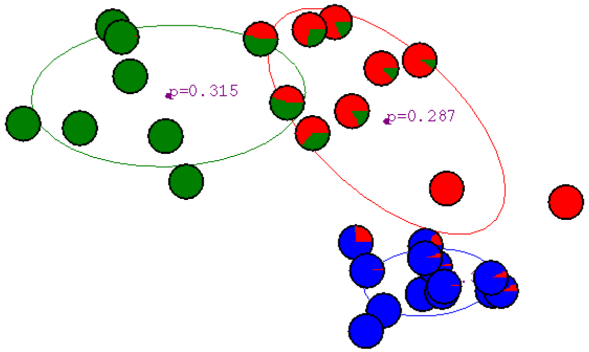
After 4th iteration



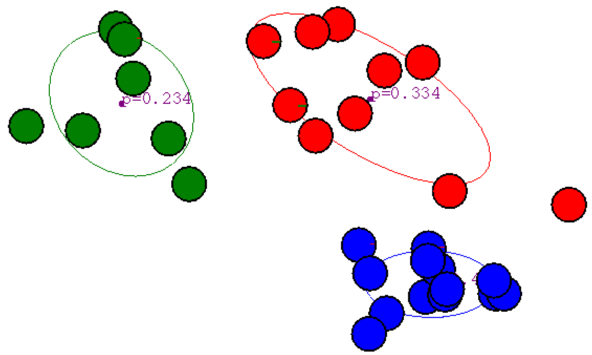
After 5th iteration



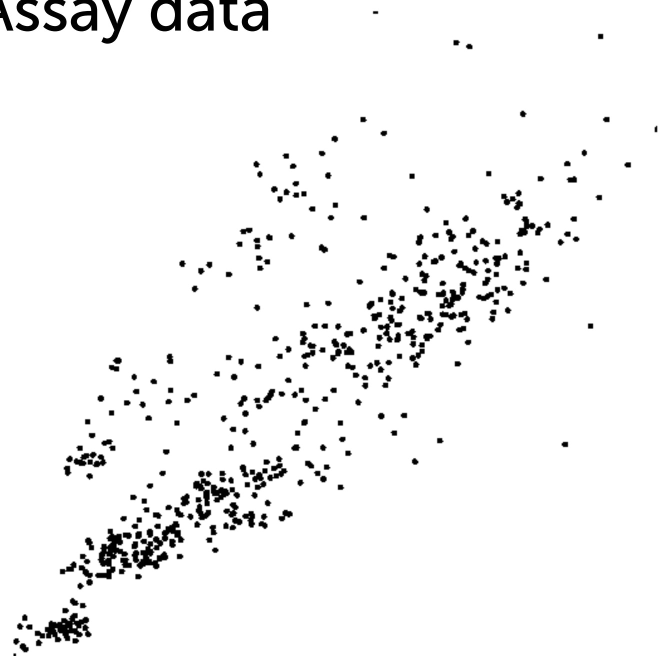
After 6th iteration



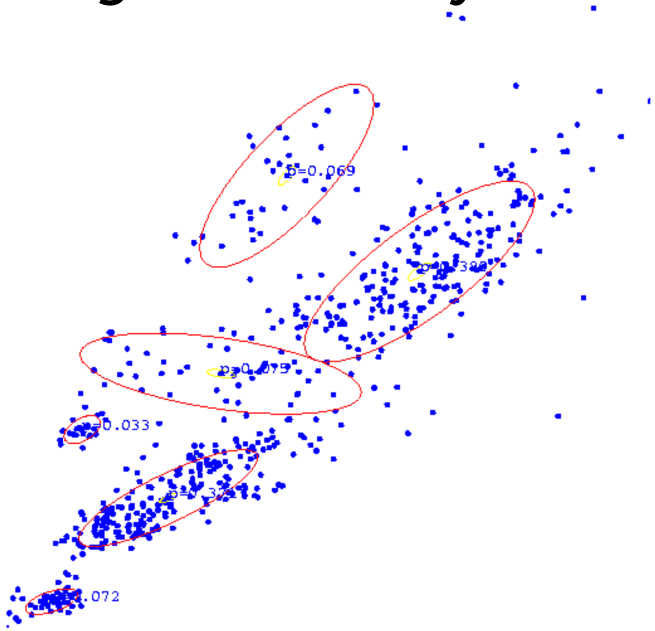
After 20th iteration



Some Bio Assay data



GMM clustering of the assay data



Resulting Density Estimator