CMSC 726 Lecture 17:Probabilistic Clustering and EM

Lise Getoor

November 2, 2010

ACKNOWLEDGEMENTS: The material in this course is a synthesis of materials from many sources, including: Hal Daume III, Mark Drezde, Carlos Guestrin, **Andrew Moore**, Andrew Ng, **Ben Taskar**, Eric Xing, and others. I am very grateful for their generous sharing of insights and materials.

Families of Clustering Algorithms

- Partition-based methods
  - e.g., K-means
- Hierarchical clustering
  - e.g., hierarchical agglomerative clustering
- Probabilistic model-based clustering
  - e.g., mixture models, Gaussian Mixture Models
  - expectation maximization
- Spectral Clustering

K-means

1. Initialize cluster centroids $\mu_1, \ldots, \mu_k \in R^n$ randomly
2. Repeat until convergence: {
   1. For i= 1 to m

$$c_i = \arg\min_j \left\| x_i - \mu_j \right\|^2$$

   2. For j = 1 to k

$$\mu_j = \frac{\sum_{i=1}^{m} 1\{c_i = j\} x_i}{\sum_{i=1}^{m} 1\{c_i = j\}}$$

K-Means convergence

- K-means can be viewed as optimizing the *distortion*

$$J(c,\mu) = \sum_{i=1}^{m} \left\| x_i - \mu_j \right\|^2$$

- Which is the sum of squared distances between each training example and its cluster centroid
- K-means is coordinate descent on J; inner loop minimizes J wrt c while holding μ fixed; then minimizes J wrt μ while holding c fixed.
- J must monotonically decrease, and value will converge; Usually c and μ will converge too (in theory could oscillate between different c and μ with same J value; this is very uncommon in practice).

K-means caveats

- Converges to *local* optima
- Common approach: run multiple times

Families of Clustering Algorithms

- Partition-based methods
  - e.g., K-means
- Hierarchical clustering
  - e.g., hierarchical agglomerative clustering
- Probabilistic model-based clustering
  - e.g., mixture models, Gaussian Mixture Models
  - expectation maximization
- Spectral Clustering

Density Estimation

- 1D intuition….

Probabilistic Model-based Clustering

- Assume a probability model for each component cluster
- Mixture Model:

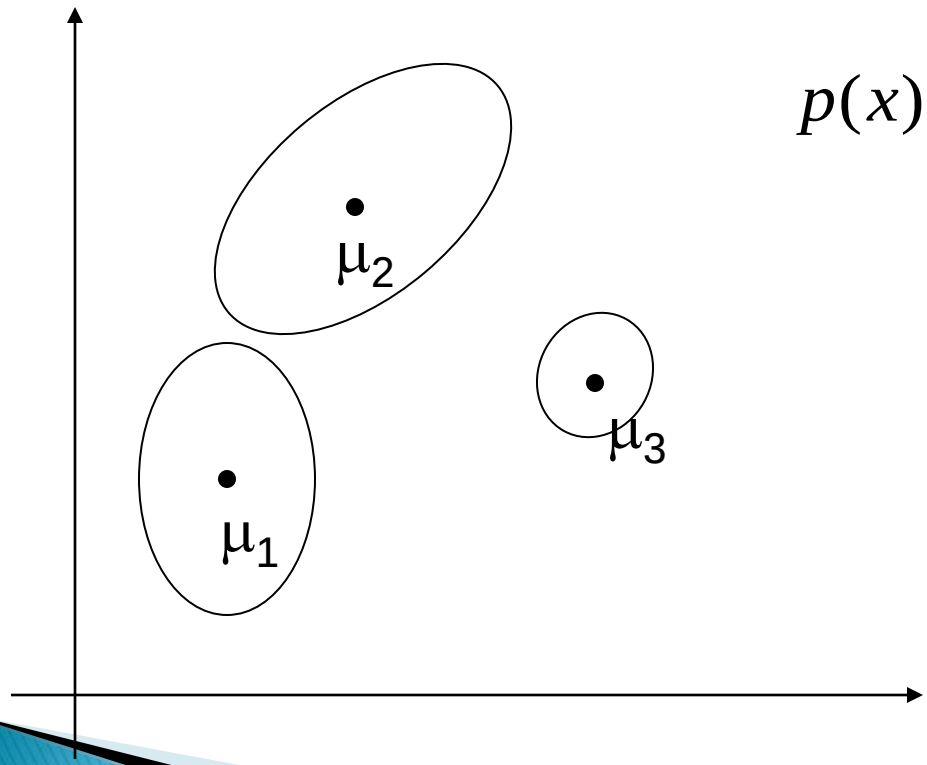
$$p(x) = \sum_{k=1}^{K} \phi_k f_k(x; \theta_k)$$

  - where ϕ_k are component distributions
  - components: Gaussian, poisson, exponential

  - Most common: Gaussian mixture model (GMM)

Gaussian Mixture Models (GMM)

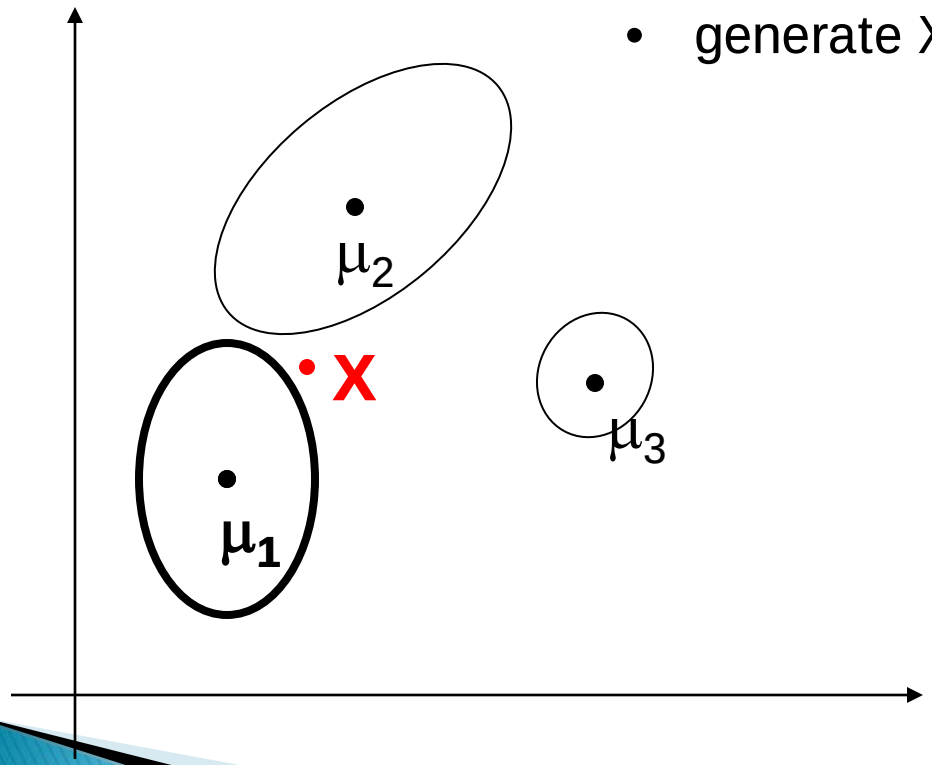
- K components,
- model for each component cluster $N(\mu_k,\sigma_k)$



$$p(x) = \sum_{k=1}^{K} \phi_k f(x; \mu_k, \sigma_k)$$

GMM cont.

- Generative Model
  - choose component with probability ϕ_k
  - generate $X \sim N(\mu_k, \sigma_k)$



Another way to think about it

- Introduce a hidden variable Z in {1,…, k} (also called a latent variable, unobserved variable}

$$p(x) = \sum_{j=1}^{k} p(z = j)\, p(x \mid z = j)$$

- Where

$$p(z) \sim multinomial(\phi_1, \ldots, \phi_k)$$

$$p(x \mid z_j) \sim N(\mu_j, \Sigma_j)$$

Yaye! This looks familiar…

- Parameters ϕ, μ, Π
- To estimate them, write down likelihood of data:

$$\ell(\phi, \mu, \Sigma) = \sum_{i=1}^{m} \log p(x_i; \phi, \mu, \Sigma)$$

$$= \sum_{i=1}^{m} \log \sum_{j=1}^{k} p(x_i \mid z_j; \mu, \Sigma) p(z_j; \phi)$$

IF… if only

- We *knew* what the z_i were, then

$$\ell(\phi, \mu, \Sigma) = \sum_{i=1}^{m} \log p(x_i \mid z_i; \mu, \Sigma) + \log p(z_i; \phi)$$

- we could just choose MLE params for φ, μ, Π:

$$\phi_j = \frac{1}{m} \sum_{i=1}^{m} 1\{z_i = j\}$$

$$\mu_j = \frac{\sum_{i=1}^{m} 1\{z_i = j\} x_i}{\sum_{i=1}^{m} 1\{z_i = j\}}$$

$$\Sigma_j = \frac{\sum_{i=1}^{m} 1\{z_i = j\}(x_i - \mu_j)(x_i - \mu_j)^T}{\sum_{i=1}^{m} 1\{z_i = j\}}$$

look familiar??!?

Aside… Variance variations

1. Full covariance: Σ_k is arbitrary for each class (clusters are ellipsoids), $O(Kn^2)$ parameters
2. Shared full covariance; Σ_k is arbitrary for same for each class, $O(Kn^2)$ parameters
3. Diagonal; Σ_k is a diagonal matrix (all clusters are axis aligned ellipsoids), $O(Kn)$ parameters
4. Shared Diagonal; Σ_k is a diagonal matrix, same for each class (same axis aligned ellipsoid), $O(n)$ parameters
5. Spherical: Σ_k is $\sigma_k I$ (clusters have spherical shape), $O(k)$ parameters
6. Shared Spherical: Σ_k is σI (all clusters have same radius), $O(1)$ parameters

7. And for 'mixture' parameters, ϕ_k, they can be all the same $(1/K)$, or different.

Back to the Problem

- Problem: we have a bunch on non-linear non-analytically-solvable equations
- One solution: gradient descent.... slow
- instead....

Expectation Maximization (EM)

- Dempster, Laird, and Rubin, 1977
- **extremely** popular
- applicable in a wide range of problems
- many uses besides clustering: hidden markov models, Bayesian networks
- basic idea is quite simple…

EM for GMM

1. Initialize cluster means randomly
2. Repeat until convergence: {
   1. For each i, j

$$w_{ij} = p(z_i = j \mid x_i; \phi, \mu, \Sigma)$$

E-step

   2. Update the parameters

$$\phi_j = \frac{1}{m}\sum_{i=1}^{m} w_{ij}$$

$$\mu_j = \frac{\sum_{i=1}^{m} w_{ij} x_i}{\sum_{i=1}^{m} w_{ij}}$$

$$\Sigma_j = \frac{\sum_{i=1}^{m} w_{ij}(x_i - \mu_j)(x_i - \mu_j)^T}{\sum_{i=1}^{m} w_{ij}}$$

M-step

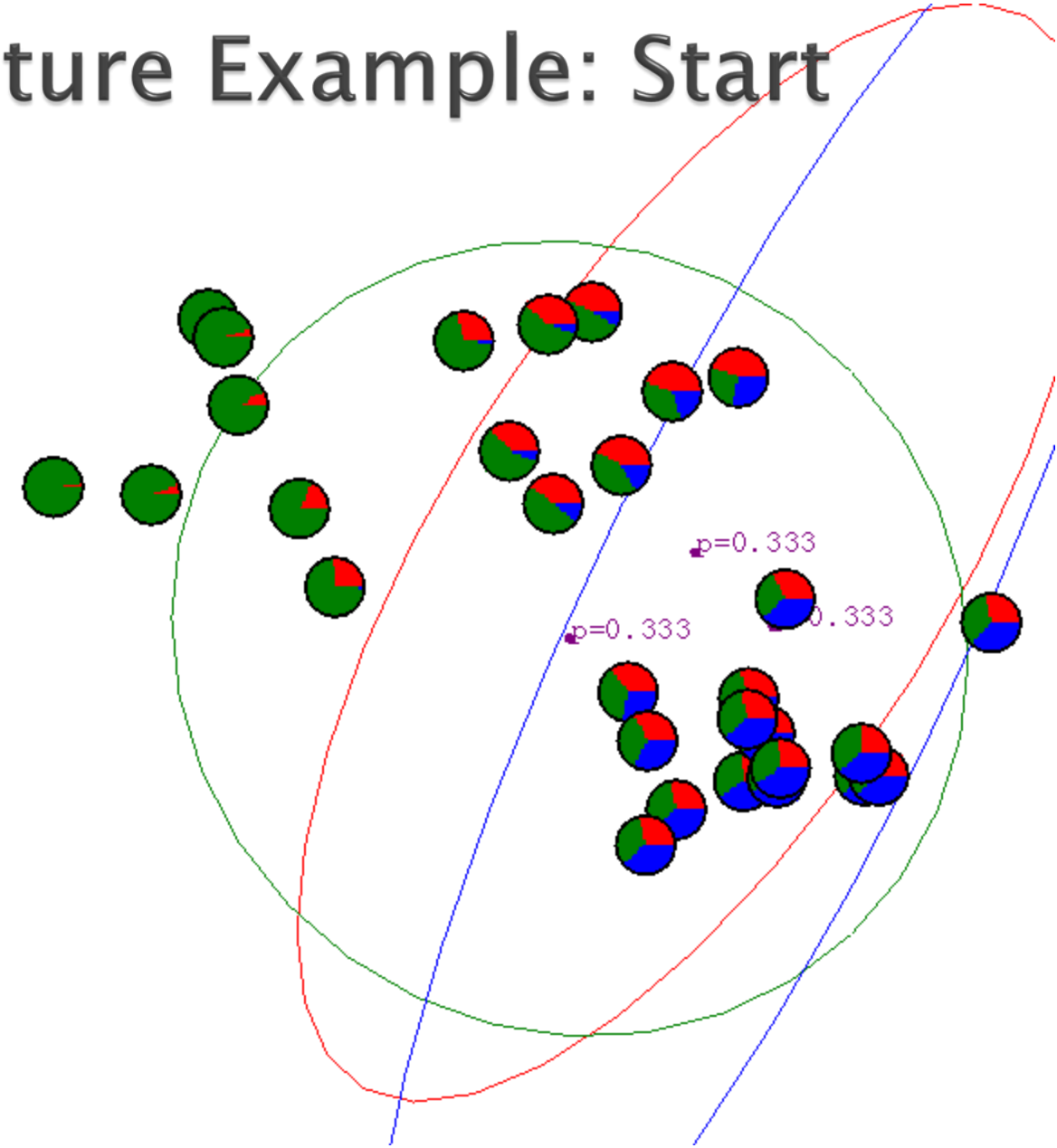
EM for GMM

- In E-step, calculate posterior probability of z_i, given x_i and current setting of parameters.

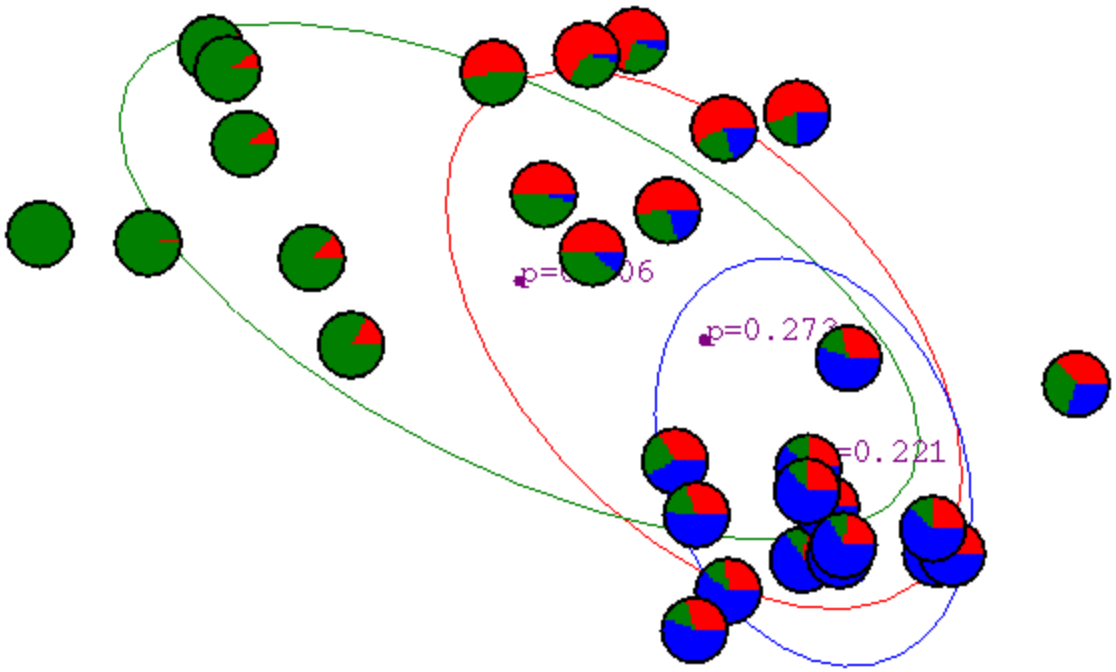
$$p(z_i = j \mid x_i; \phi, \mu, \Sigma) = \frac{p(x_i \mid z_i = j; \mu, \Sigma)\, p(z_i = j; \phi)}{\sum_{j=1}^{k} p(x_i \mid z_i = j; \mu, \Sigma)\, p(z_i = j; \phi)}$$

- The values of the w_{ij} in E-step are our "soft guesses" for the values of z_i

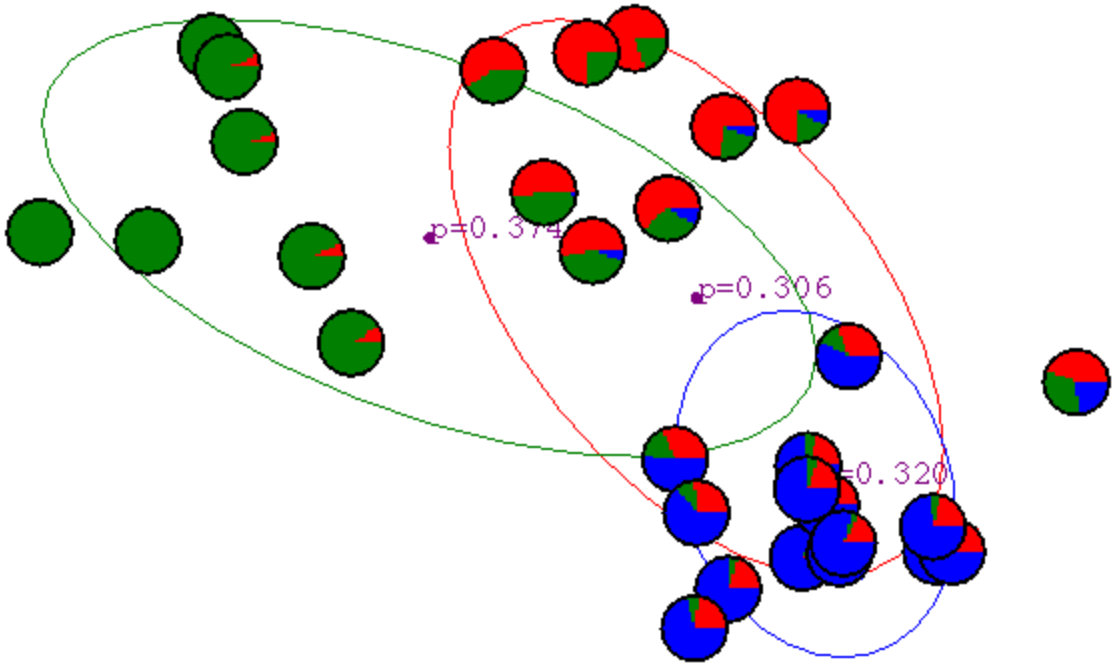
Gaussian Mixture Example: Start



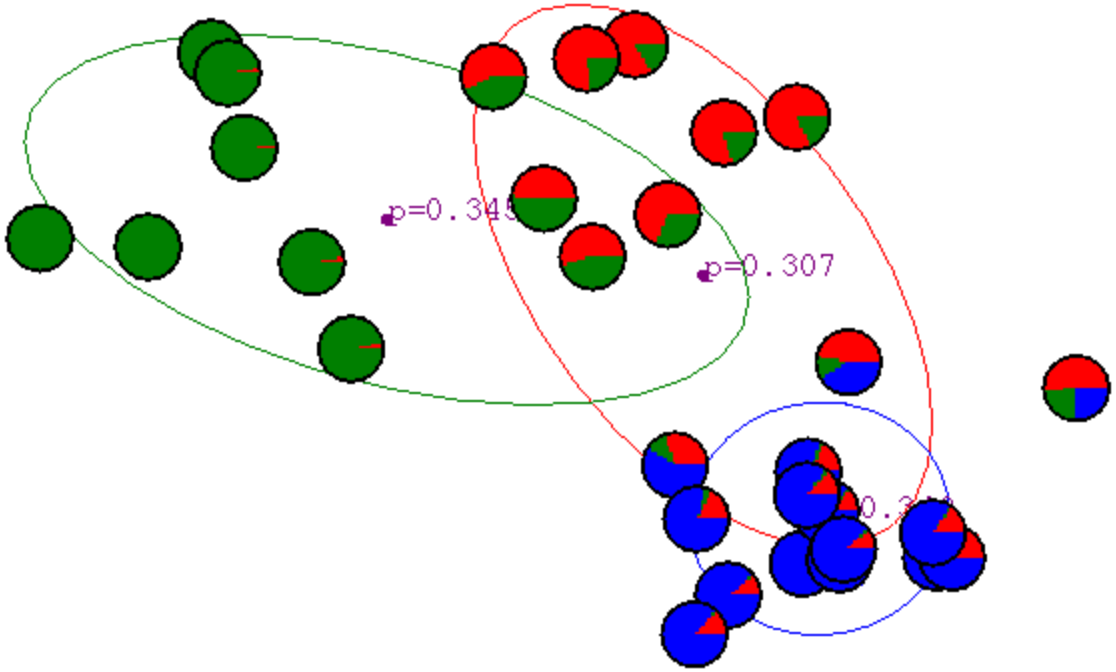
After first iteration



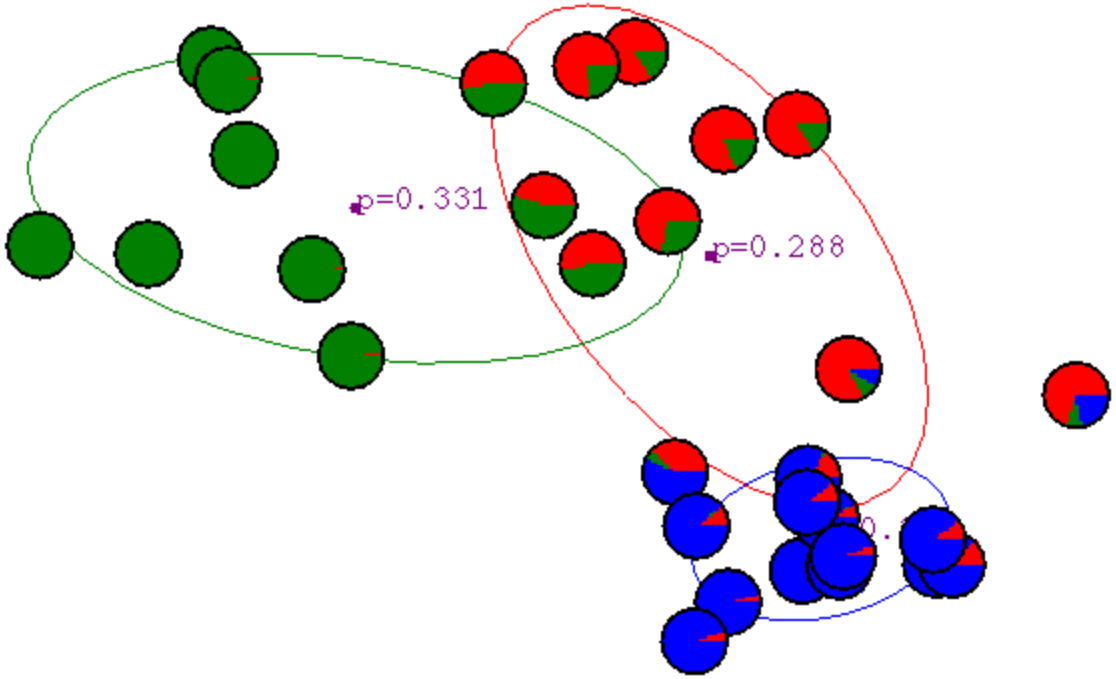
After 2nd iteration



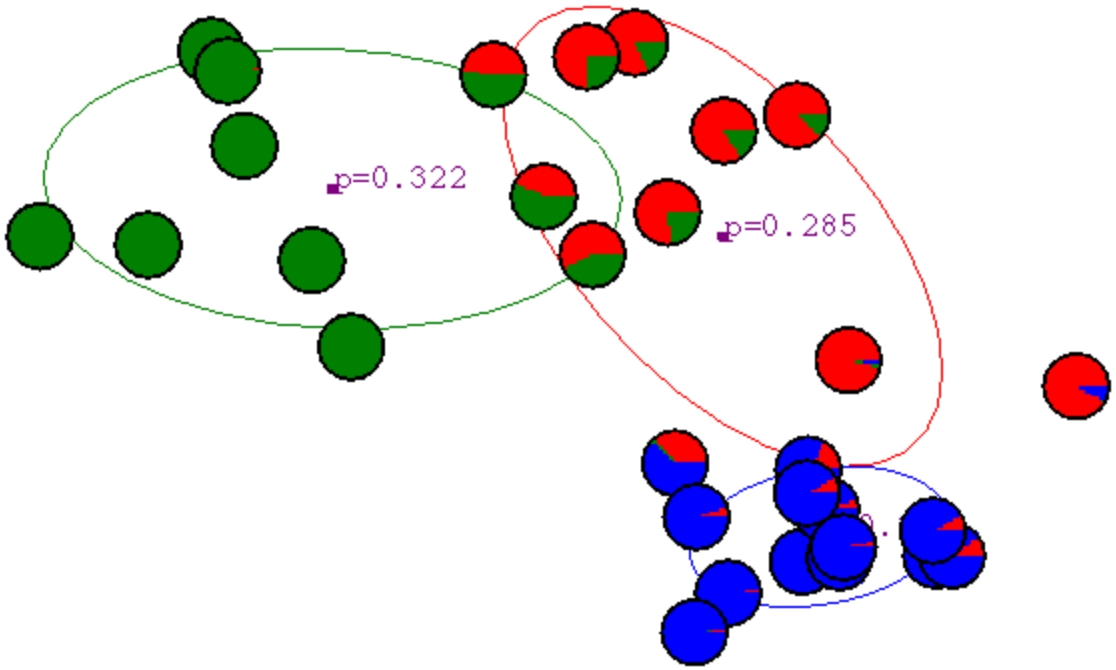
After 3rd iteration



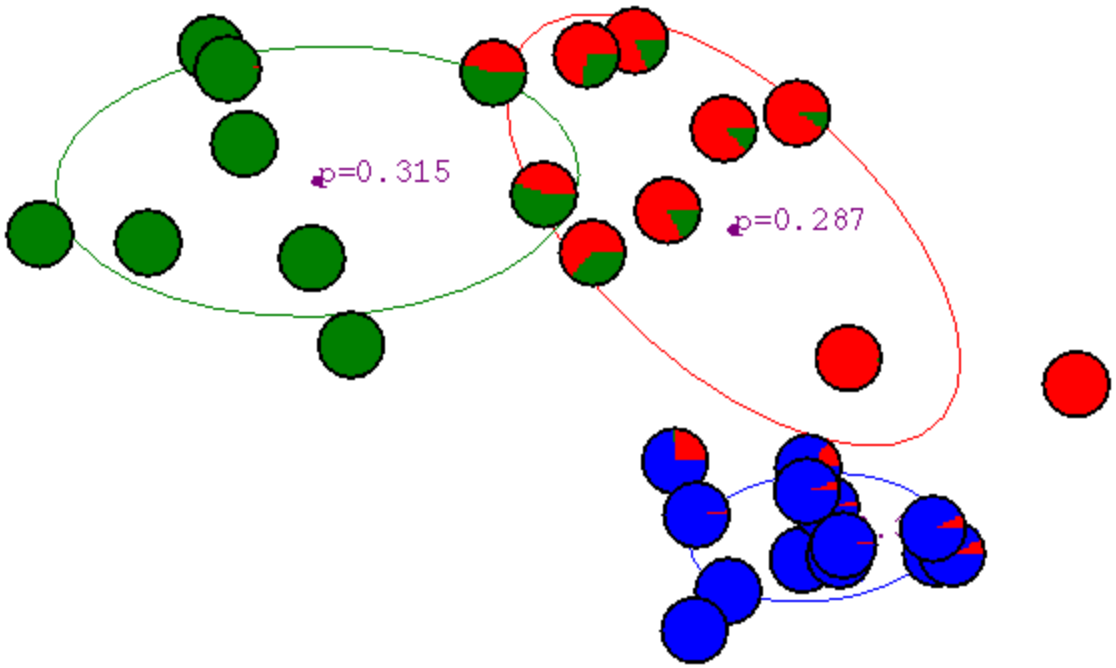
After 4th iteration



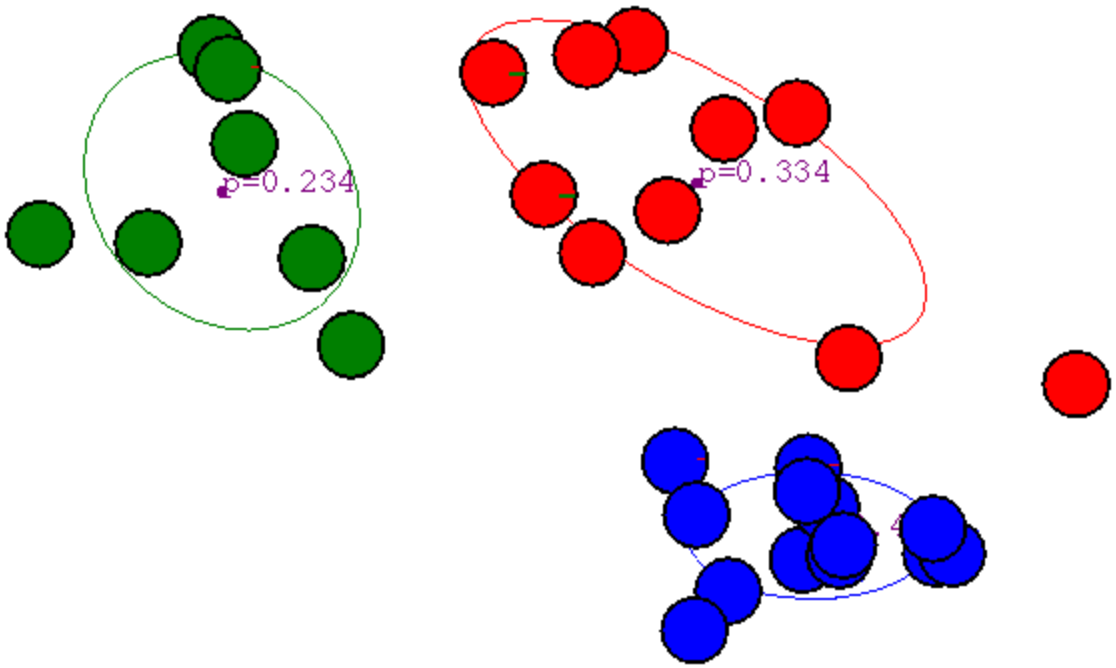
After 5th iteration



After 6th iteration



After 20th iteration



EM: Broader Perspective

- The EM algorithm presented for GMM is just one example; EM can be applied to a broad class of estimation problems involving latent variables.

Jensen's Inequality

- **Theorem:** Let f be a convex function, and let X be a random variable. Then

$$E[f(x)] \geq f(E[x])$$

further, if f is strictly convex (f"(x)>0), then E[f(x)]=f(E[x]) iffX=E[X] with probability 1 (i.e., X is constant)

- If f is concave, holds with inequality direction reversed.
- For more info, http://www.engineering.usu.edu/classes/ece/7680/lecture2/node5.html

Formal EM setup

- Let X = {x(1), …, x(m)} be m observed data vectors
- Let Z = {z(1),…, z(m)} be m values of hidden variable (these might be the cluster labels)
- Then the log-likelihood of the observed data is

$$l(\theta) = \log p(X \mid \theta) = \log \sum_Z p(X, Z \mid \theta)$$

- *both* θ and Z are unknown
- Let Q(Z) be any probability distribution for Z.

$$\begin{aligned} l(\theta) &= \log \sum_Z p(X, Z \mid \theta) \\ &= \log \sum_Z Q(Z) \frac{p(X, Z \mid \theta)}{Q(Z)} \\ &\geq \sum_Z Q(Z) \log \frac{p(X, Z \mid \theta)}{Q(Z)} \\ &= \sum_Z Q(Z) \log p(X, Z \mid \theta) + \sum_Z Q(Z) \log \frac{1}{Q(Z)} \\ &= F(Q, \theta) \end{aligned}$$

lower bound on l(θ)

EM Algorithm

- EM algorithm alternates between
  - maximize F with respect to dist. Q with θ fixed

E-step: $Q^{k+1} = \arg\max_{Q} F(Q^k, \theta^k)$

  - maximize F with respect to θ with Q = p(Z) fixed

M-step: $\theta^{k+1} = \arg\max_{\theta} F(Q^{k+1}, \theta^k)$

- Maximum for E step:

Intuition:
•In the E-step, we estimate the distribution on the hidden variables, conditioned on a particular setting of the parameter vector θ^k
•In the M-step, we choose new set of parameters θ^{k+1} to maximize the expected log-likelihood of observed data

EM

1. Initialize randomly
2. Repeat until convergence: {
   1. For each i

$$Q_i(z_i) = p(z_i \mid x_i; \theta)$$

E-step

   2. Update the parameters

$$\theta = \arg\max_{\theta} \sum_i \sum_{z_i} Q_i(z_i) \log \frac{p(x_i, z_i; \theta)}{Q_i(z_i)}$$

M-step

Notes

- Often both the E and M step can be solved in closed form
- Neither the E step nor the M step can decrease the log-likelihood
- Under relatively general conditions the algorithm is guaranteed to converge to a local maximum of log-likelihood
- We must specify a starting point for the algorithm, for example a random choice of θ
- We must specify stopping criteria, or convergence detection
- Computational complexity: number of iterations, time to compute E and M steps

EM Comments

- complexity of EM for GMM with K components: dominated by calculation of K covariance matrices.
  - With n dimensions, $O(Kn^2)$ covariance parameters to be estimated
  - Each requires summing over m data points and cluster weights, leading to $O(mKn^2)$ per step
- Often times there are large increases in likelihood over first few iteration and then can slowly converge; likelihood as function of iterations not necessarily concave

and finally…

how do we choose K?

How to choose K

- Choose K that maximizes likelihood?
- NOT.
- As K is increased, the value of the likelihood at maximum cannot decrease
- Problem of scoring models with different complexities
  - Model too flexible $\Rightarrow$ overfit the data $\Rightarrow$ high variance
  - Model too restrictive $\Rightarrow$ can't fit the data $\Rightarrow$ high bias
  - Bias-variance tradeoff: compromise
- Solutions:
  - external validation (use k-fold cross validation, LOOCV)
  - scoring function – MDL, BIC, AIC
  - Bayesian model selection
- Still, the choice is often subjective, and is often done by hand, based on knowledge of the problem domain.

Next Time….

- Spectral Clustering