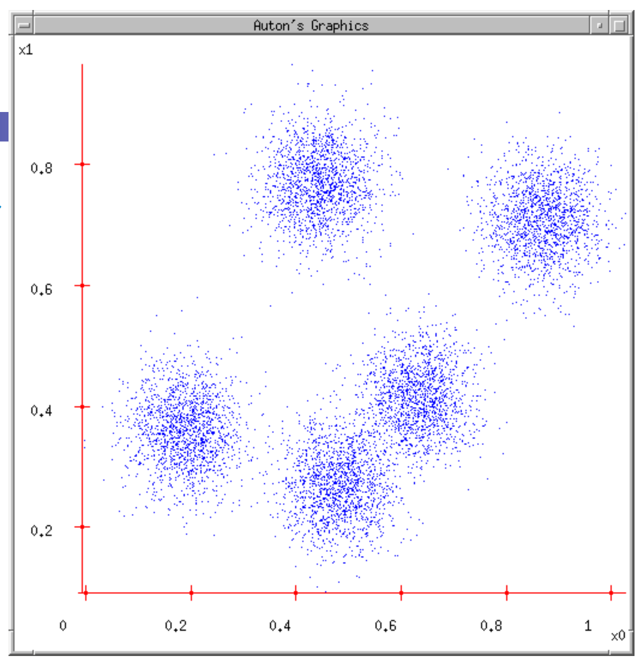
Case Study 2: Document Retrieval

Review:
Mixtures of Gaussians

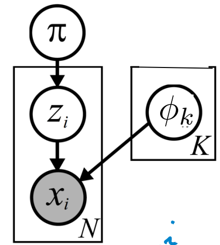
Machine Learning for Big Data
CSE547/STAT548, University of Washington
Emily Fox
January 28th, 2014

Some Data

want to cluster
- unsupervised
- generative approach

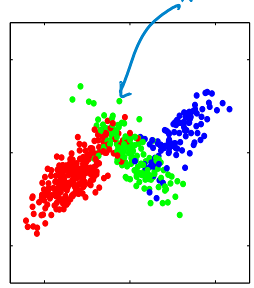


Gaussian Mixture Model



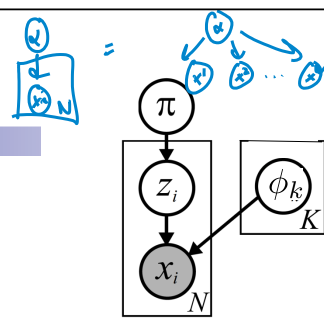
- Most commonly used mixture model
- Observations: $x^1, \dots, x^N$ $\quad x^i \in \mathbb{R}^d$
- Parameters: mix weights
  $\pi = [\pi_1, \dots, \pi_K]$ $\quad$ K # of clusters
  $\phi = \{\phi_k\} = \{\mu_k, \Sigma_k\}$ $\quad$ params for cluster k
- Cluster indicator:
  $z^i \in \{1, \dots, K\}$ $\quad Pr(z^i = k) = \pi_k$
- Per-cluster likelihood:
  $N(x^i \mid \mu_k, \Sigma_k, z^i = k)$

x^i

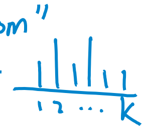


- Ex. z^i = country of origin, x^i = height of i^th person
  - k^{th} mixture component = distribution of heights in country k

Generative Model

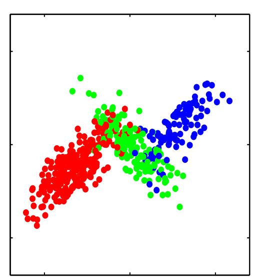


- We can think of *sampling* observations from the model
- For each observation *i*,
  - Sample a cluster assignment
    $z^i \sim \pi$ "drawn from"



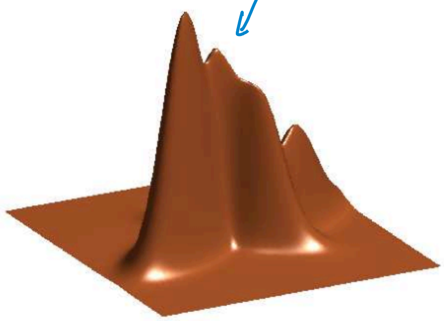
  - Sample the observation from the selected Gaussian
    $x^i \mid z^i \sim N(x^i \mid \mu_{z^i}, \Sigma_{z^i})$

can "generate" obs.

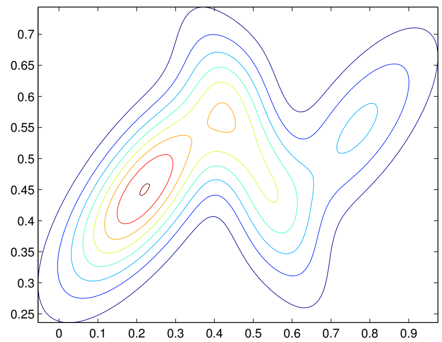


Also Useful for Density Estimation

$x^i \in \mathbb{R}^2$ $p(x^i|\theta)$



Contour Plot of Joint Density

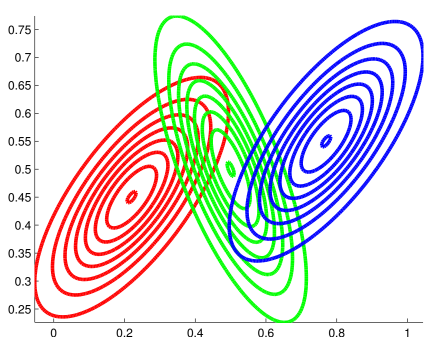


bird's eye view

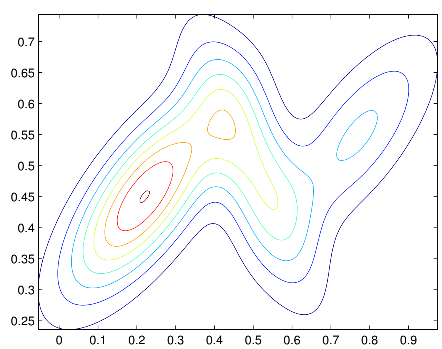
Density as Mixture of Gaussians

- Approximate density with a mixture of Gaussians

Mixture of $K=$ *3 Gaussians*



Contour Plot of Joint Density

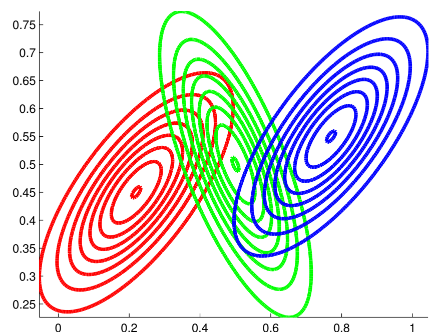


Each Gauss. has weight π_k $(\sum \pi_k = 1)$
and shape params $\{\mu_k, \Sigma_k\}$

Density as Mixture of Gaussians

- Approximate density with a mixture of Gaussians

Mixture of 3 Gaussians



$\{\pi_1, \ldots, \pi_K\}$ $\{\mu_k, \Sigma_k\}$

$$p(x^i \mid \underbrace{\pi, \mu, \Sigma}_{\theta}) = \sum_{k=1}^{K} \pi_k N(x^i \mid \mu_k, \Sigma_k)$$

$p(z^i = k)$ $p(x^i \mid z^i = k, \theta)$

In 1D:

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 :

$$p(x_i \mid \pi, \mu, \Sigma) = \sum_{z_i=1}^{K} \pi_{z_i} \mathcal{N}(x_i \mid \mu_{z_i}, \Sigma_{z_i})$$

$$p(x_i \mid z_i, \pi, \mu, \Sigma) = \mathcal{N}(x_i \mid \mu_{z_i}, \Sigma_{z_i})$$

Case Study 2: Document Retrieval

Application to Document Modeling

Machine Learning for Big Data
CSE547/STAT548, University of Washington
Emily Fox
January 28th, 2014

Task 2: Cluster Documents

- **Now:**
  - Cluster documents based on topic

"sports"

"world news"

Document Representation

- Bag of words model

document *d*

previously: $x^d = [\quad]$ ← vector fcn of word counts (e.g. tf-idf)

performed operations on this vector

now: $x^d = \{w_1^d, \dots, w_{N_d}^d\}$

unordered set of N_d word indices with $w_i^d \in V$ vocab.

A Generative Model

- Documents: $x^1, \dots, x^D$ with $x^d = \{w_1^d, \dots, w_{N_d}^d\}$
- Associated topics: $z^1, \dots, z^D$ with $z^d \in \{1, \dots, K\}$ ↑ # topics
- Parameters: $\theta = \{\pi, \beta\}$

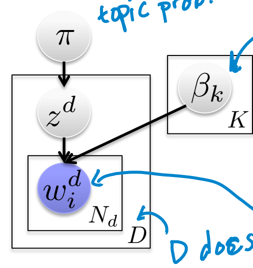
as before: $\pi = [\pi_1, \dots, \pi_K]$ topic probabilities

$\Pr(z^d = k) = \pi_k$

$$\beta = \begin{array}{c|c} & 1 \quad 2 \quad \cdots \quad V \\ \hline \begin{matrix} 1 \\ 2 \\ \vdots \\ K \end{matrix} & \begin{bmatrix} \text{---} \ \beta_1 \ \text{---} \\ \vdots \\ \text{---} \ \beta_K \ \text{---} \end{bmatrix} \end{array}$$

A Generative Model

- Documents: $x^1, \ldots, x^D$
- Associated topics: $z^1, \ldots, z^D$
- Parameters: $\theta = \{\pi, \beta\}$
- Generative model:



topic prob.

for topic k, prob. of words in that topic

N_d words in doc d

D docs

$z^d \sim \pi$ generate topic

$w_i^d \mid z^d \sim \beta_{z^d}$ $i = 1, \ldots, N_d$

Given topic $z^d = k$ for doc d, draw each word from β_k

Form of Likelihood

- Conditioned on topic...

$$p(x^d \mid z^d, \beta) = \prod_{i=1}^{N_d} p(w_i^d \mid z^d, \beta) = \prod_{i=1}^{N_d} \beta_{z^d, w_i^d}$$

$\{w_1^d, \ldots, w_{N_d}^d\}$

- Marginalizing latent topic assignment:

$$p(x^d \mid \beta, \pi) = \sum_{k=1}^{K} \pi_k \, p(x^d \mid z^d = k, \beta_k)$$

$P(z^d = k)$

Case Study 2: Document Retrieval

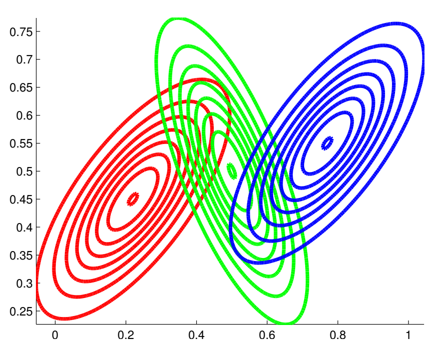
Review:
EM Algorithm

Machine Learning for Big Data
CSE547/STAT548, University of Washington
Emily Fox
January 28th, 2014

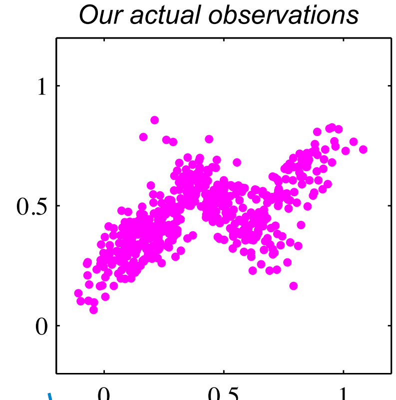
Learning Model Parameters

- Want to learn model parameters

Mixture of 3 Gaussians



Our actual observations



How???...

from obs., estimate model params

C. Bishop, Pattern Recognition & Machine Learning

ML Estimate of Mixture Model Params

- Log likelihood

assume x^i iid

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

introduce cluster ind. + marg.

- Want ML estimate

$$\hat{\theta}^{ML} = \arg\max_{\theta} L_x(\theta)$$

- Assume exponential family $p(x, z \mid \theta) = \frac{1}{Z(\theta)} e^{\theta' \phi(x,z)}$

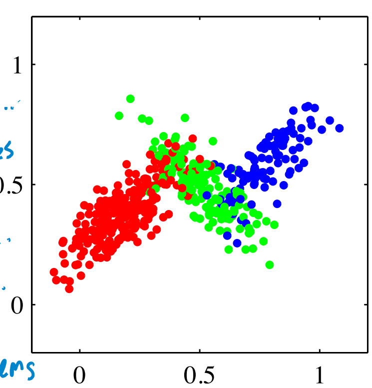
$$L_x(\theta) = \sum_i \log\left(\sum_{z^i} e^{\theta^T \phi(z^i, x^i)}\right) - N \log Z(\theta)$$

- Neither convex nor concave and local optima

Complete Data

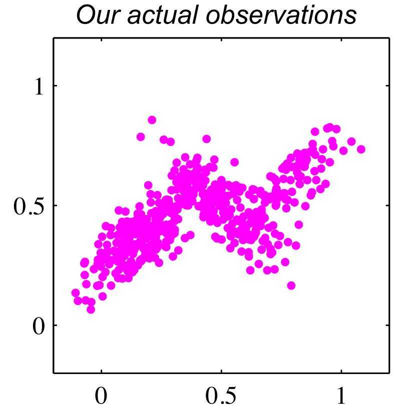
- Imagine we have an assignment of each x^i to a cluster

life would be easier... Decouples into K ind. param. est. problems



Complete data labeled by true cluster assignments

Our actual observations



"incomplete data"

C. Bishop, Pattern Recognition & Machine Learning

If "complete" data were observed…

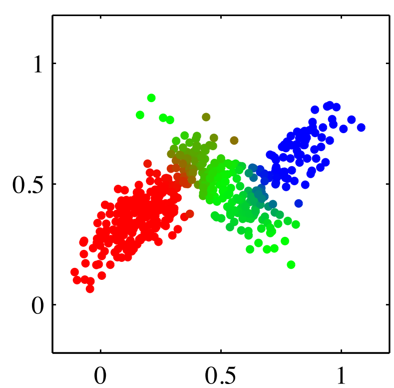
- Assume class labels z^i were observed in addition to x^i

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

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

Cluster Responsibilities

- We must infer the cluster assignments from the observations



Soft assignments to clusters

☆ motivates iterative algorithm ☆

responsibility cluster k takes for obs i.

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

$$r_{ik} = p(z^i = k \mid x^i, \pi, \phi) =$$

$$= \frac{\pi_k \, p(x^i \mid \phi_k)}{\sum \pi_j \, p(x^i \mid \phi_j)}$$

e.g. $N(x^i \mid \mu_j, \Sigma_j)$

C. Bishop, *Pattern Recognition & Machine Learning*

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

$$z^i \underset{\text{max}}{\overset{\text{est.}}{\rightleftarrows}} \theta$$

- Example: MoG
  1. Infer "responsibilities"

$$r_{ik}^{(t)} = p(z^i = k \mid x^i, \hat{\theta}^{(t-1)}) = \frac{\pi_k^{(t-1)} p(x^i \mid \phi_k^{(t-1)})}{\sum_j \pi_j^{(t-1)} p(x^i \mid \phi_j^{(t-1)})}$$

  2. Optimize parameters

$$\text{max w.r.t. } \pi_k : \quad \pi_k^{(t)} = \frac{1}{N} \sum r_{ik}^{(t)} = \frac{r_k^{(t)}}{N} \leftarrow \text{soft counts!}$$

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

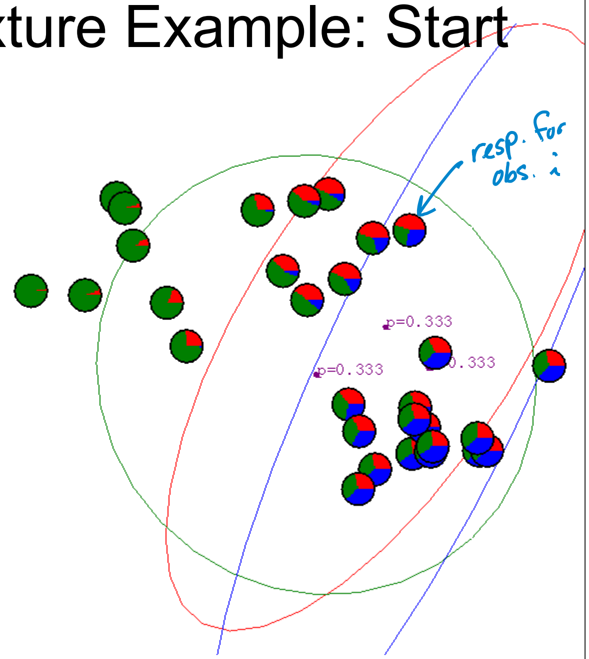
$$\mu_k^{(t)} = \frac{\sum r_{ik}^{(t)} x^i}{r_k^{(t)}} \leftarrow \text{weighted mean} \qquad \Sigma_k^{(t)} = \frac{1}{r_k^{(t)}} \sum r_{ik}^{(t)} x_i x_i^T - \mu_k^{(t)} \mu_k^{(t)T}$$

Gaussian Mixture Example: Start

Initialize

$\pi^{(0)}, \phi^{(0)}$

$\rightarrow$ compute $r_{ik}^{(1)}$



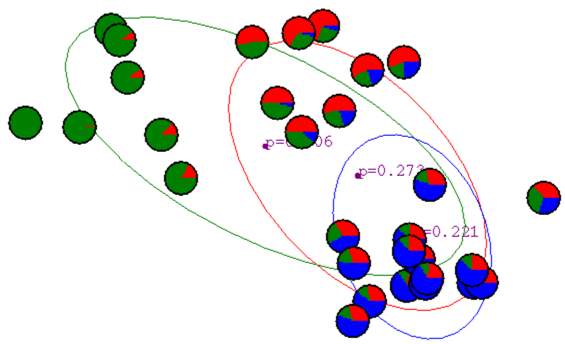
resp. for obs. i

After first iteration

max like. given soft counts

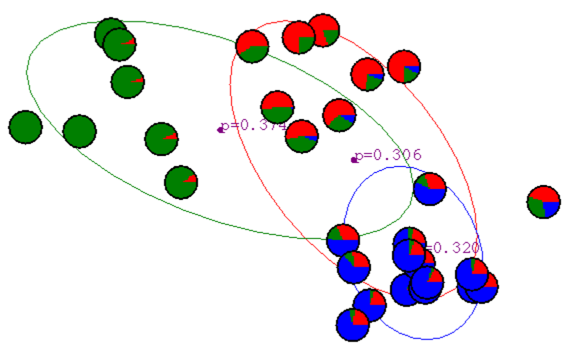
$\rightarrow \pi^{(1)}, \phi^{(1)}$

$\rightarrow$ new $r_{ik}^{(2)}$

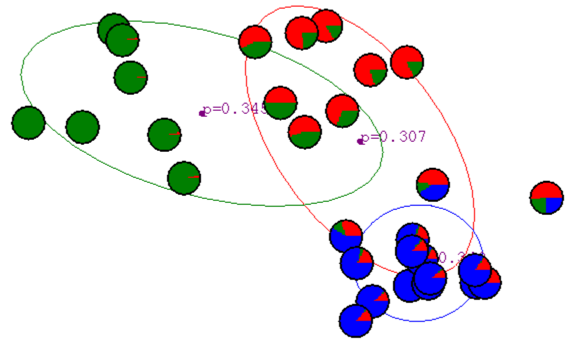


After 2nd iteration

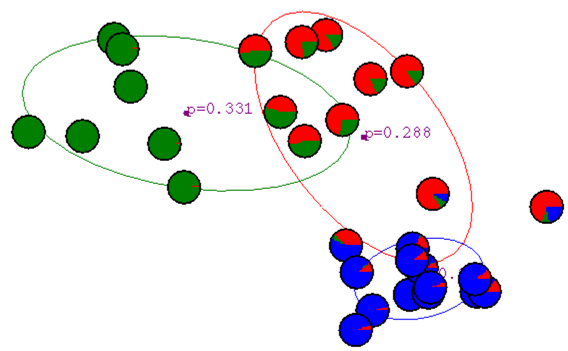
rinse + repeat



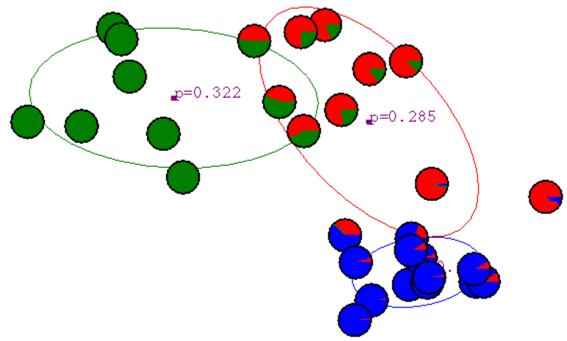
After 3rd iteration



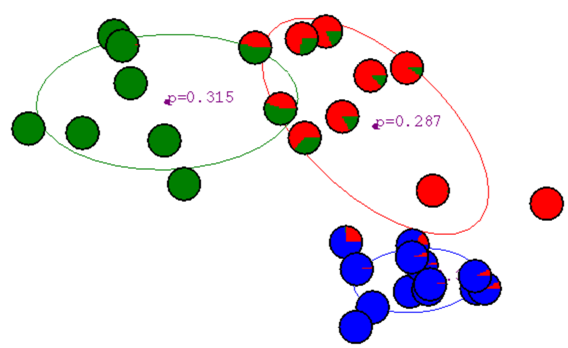
After 4th iteration



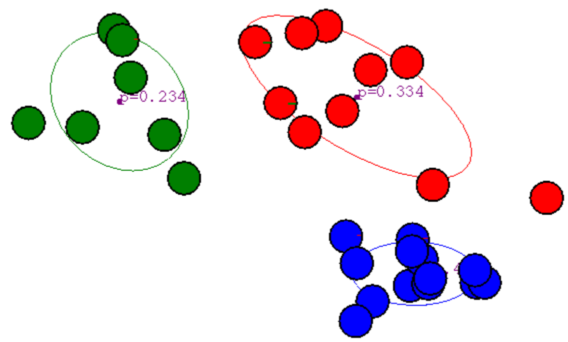
After 5th iteration



After 6th iteration



After 20th iteration



Expectation Maximization (EM) – Setup

- More broadly applicable than just to mixture models considered so far

what we actually have

- Model: x observable – *"incomplete" data*
  y not (fully) observable – *"complete" data*
  θ parameters

what we wish we had

- Interested in maximizing (wrt θ):

$$p(x \mid \theta) = \sum_y p(x, y \mid \theta)$$

introduce complete data + marg.

- Special case:

$$x = g(y)$$

e.g. $y = \begin{bmatrix} z \\ X \end{bmatrix}$ ← class obs. ← obs.

in standard mix model

EM Algorithm

- Initial guess: $\hat{\theta}^{(0)}$
- Estimate at iteration *t*: $\hat{\theta}^{(t)}$
- **E-Step**

  Compute $U(\theta, \hat{\theta}^{(t)}) = E[\log p(\underbrace{y}_{\text{complete data}} \mid \theta) \mid \underbrace{x}_{\text{actual obs.}}, \hat{\theta}^{(t)}]$

- **M-Step**

  Compute $\hat{\theta}^{(t+1)} = \arg\max_{\theta} U(\theta, \hat{\theta}^{(t)})$

$\Rightarrow L_x(\hat{\theta}^{(t+1)}) \geq L_x(\hat{\theta}^{(t)})$

$\Rightarrow \hat{\theta}^{(t)}$ converges to a local mode

Example – Mixture Models

$E_{q_t}[I(z^i=k)] = p(z^i=k \mid x^i, \hat{\theta}^{(t)}) \triangleq r_{ik}$

- **E-Step** Compute $U(\theta, \hat{\theta}^{(t)}) = E[\log p(y \mid \theta) \mid x, \hat{\theta}^{(t)}]$
- **M-Step** Compute $\hat{\theta}^{(t+1)} = \arg\max_{\theta} U(\theta, \hat{\theta}^{(t)})$
- Consider $y^i = \{z^i, x^i\}$ i.i.d.

$p(x^i, z^i \mid \theta) = \pi_{z^i} p(x^i \mid \phi_{z^i}) = \prod_{k=1}^{K} \left(\pi_k p(x^i \mid \phi_k)\right)^{I(z^i=k)}$

$E_{q_t}[\log p(y \mid \theta)] = \sum_i E_{q_t}[\log p(x^i, z^i \mid \theta)] =$

$= \sum_i \sum_k r_{ik} \log \pi_k + \sum_i \sum_k r_{ik} \log p(x^i \mid \phi_k)$

E-step: computing r_{ik}'s

M-step: maximize wrt π_k, ϕ_k

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 convergence rates in practice

  + quality of local optima reached

What you need to know

- Mixture model formulation
  - Generative model
  - Likelihood
- Expectation Maximization (EM) Algorithm
  - Derivation ← from prev. ML / Stat course
  - Concept of non-decreasing log likelihood
  - Application to standard mixture models

Case Study 2: Document Retrieval

Review: Connection to k-means

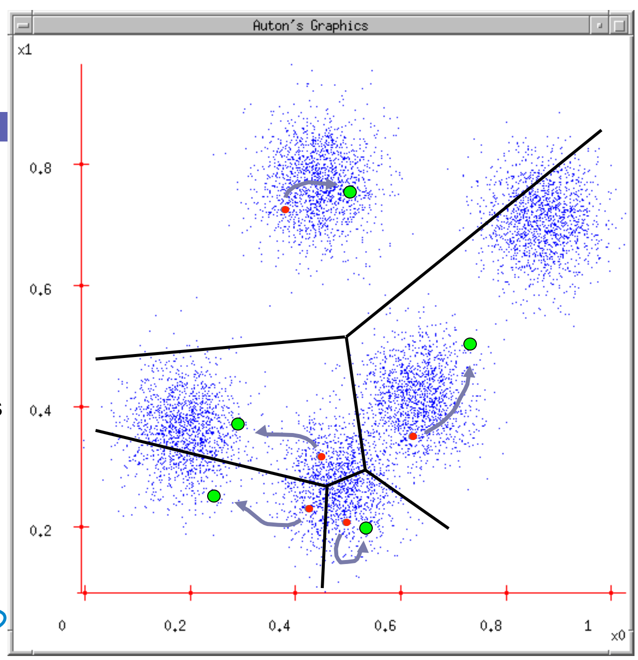
Machine Learning for Big Data
CSE547/STAT548, University of Washington
Emily Fox
January 28th, 2014

Recall:

K-means

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns

iterative alg. making HARD assignments



K-means

- Randomly initialize k centers
  - $\mu^{(0)} = \mu_1^{(0)}, \ldots, \mu_k^{(0)}$
- **Classify**: Assign each point $j \in \{1,\ldots m\}$ to nearest center:
  - $z^j \leftarrow \arg\min_i ||\mu_i - \mathbf{x}^j||_2^2$ ← hard assign.
- **Recenter**: μ_i becomes centroid of its point:
  - $\mu_i^{(t+1)} \leftarrow \arg\min_\mu \sum_{j:z^j=i} ||\mu - \mathbf{x}^j||_2^2$
  - Equivalent to μ_i ← average of its points!

Special case: spherical Gaussians and hard assignments

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

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

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

assume $\Sigma_k = \begin{bmatrix} \sigma^2 & & \\ & \ddots & \\ & & \sigma^2 \end{bmatrix} = \sigma^2 I$

- Then, compare EM objective with k-means:

EM: $\max_\theta \prod_i \sum_{z^i} p(x^i, z^i | \theta)$

maximizing marginal likelihood

K-means: $\max_{\{z^i\}, \theta} \prod_i p(x^i | z^i, \theta)$

OR if $\pi_k = \frac{1}{K} \; \forall k$

$\max_{\{z^i\}, \theta} \prod_i p(x^i, z^i | \theta)$

$= \max_\theta \prod_i \max_{z^i} p(x^i, z^i | \theta)$