So far, supervised learning

$h: X \to \mathbb{R}$ "regression"

$h: X \to \{0,1,\ldots,K\}$ "classification"

Unsupervised

# Clustering
# K-means

Machine Learning – CSE546
Emily Fox
University of Washington

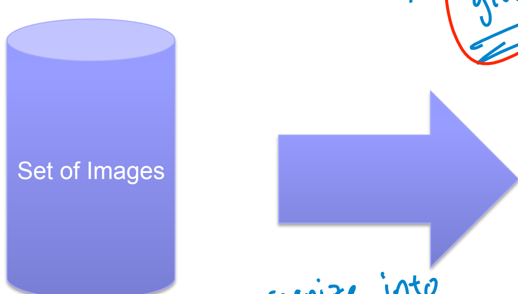
November 4, 2013



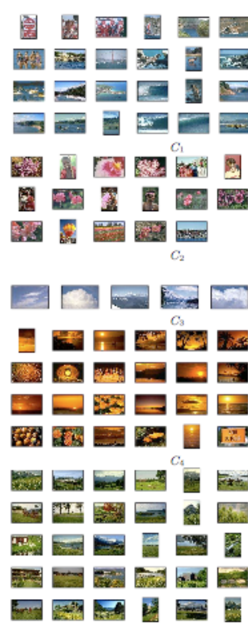


## Clustering images

key: no labels given




organize into coherent "themes"




beaches

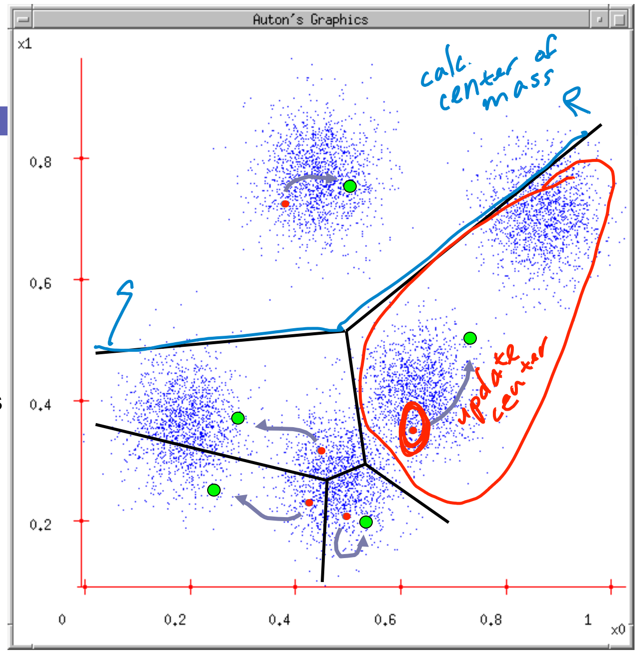
flowers

sky

[Goldberger et al.]

# 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







# K-means

Coord. desc. alg. → converges to local mode

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

converged when nothing moves (no point changes its cluster)

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

i-th cluster center, j-th data pt

$\equiv$ fix $\mu$, opt. C

$C(j)=k \Rightarrow$ j-th obs. is assoc. w/ cluster k

iterate

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

model that can be used for clustering, density est., ⋮

# Mixtures of Gaussians


Machine Learning – CSE546

Emily Fox

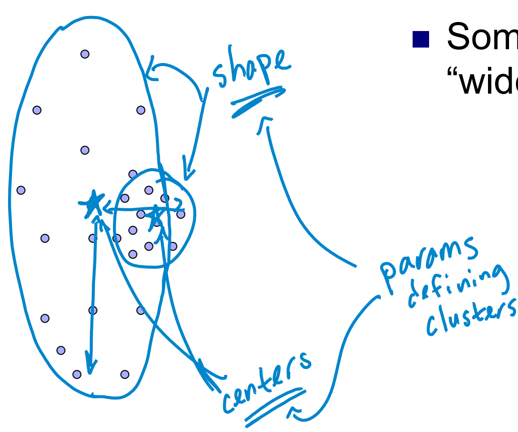
University of Washington


November 4, 2013



## (One) bad case for k-means

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

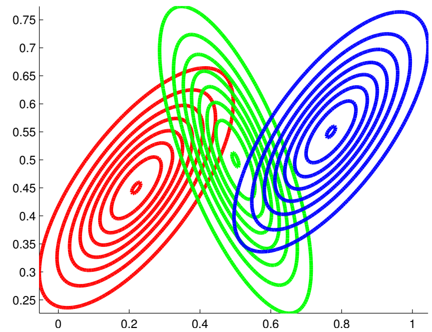



so centers alone don't tell the whole story

# Density as Mixture of Gaussians

- Approximate density with a mixture of Gaussians

*Mixture of 3 Gaussians*

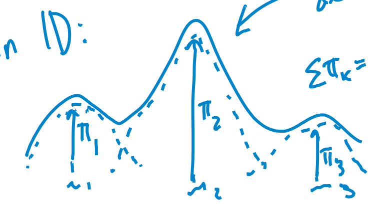



$[\pi_1, \ldots, \pi_K]$ $\{\mu_k, \Sigma_k\}$

$$P = p(x^i | \pi, \mu, \Sigma) = \sum_{k=1}^{K} \pi_k \boxed{\underline{N(x^i | \mu_k, \Sigma_k)}}$$

target density?

In 1D:




$\Sigma \pi_k = 1$

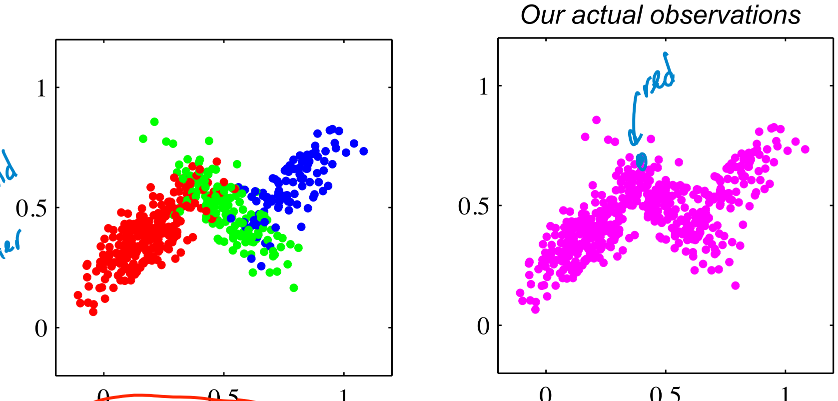


# Clustering our Observations

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

*Our actual observations*

life would be easier



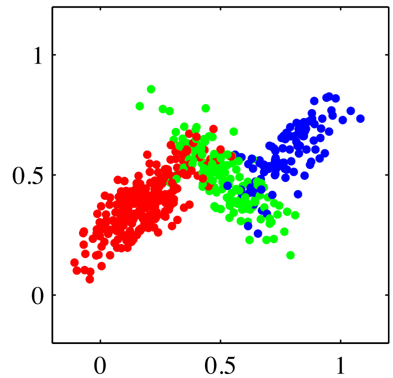

*Complete data labeled by true cluster assignments*

"Incomplete data"

*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 $z^i$ $\quad c(i) \to z^i$

  $z^i \in \{1, \dots, K\}$

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

- Then we have

$$p(x^i | z^i = k, \cancel{\pi}, \mu, \Sigma) = N(x^i | \mu_k, \Sigma_k)$$

param est. is easy if we have $\{z_i\}$

$\Rightarrow$ decouples into $K$ Gauss. est.

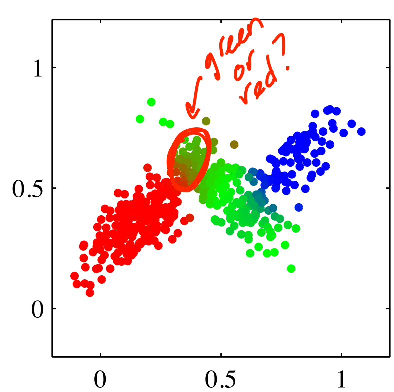
*C. Bishop, Pattern Recognition & Machine Learning*





# Clustering our Observations

- We must infer the cluster assignments from the observations




*Soft assignments to clusters*

"responsibilities"

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

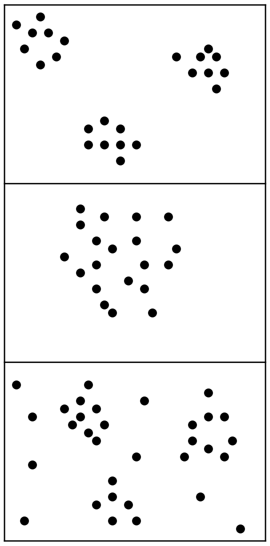
resp.

$$\underset{\text{obs } i \ \ \text{cluster } k}{r_{ik}} = p(z^i = k | x^i, \pi, \mu, \Sigma) = \frac{\pi_k N(x^i | \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j N(x^i | \mu_j, \Sigma_j)}$$

motivates an iterative alg.

*C. Bishop, Pattern Recognition & Machine Learning*

# Unsupervised Learning: not as hard as it looks



Sometimes easy

Sometimes impossible

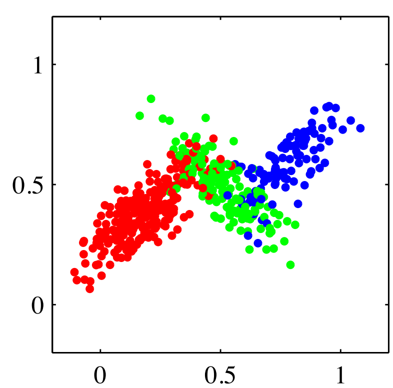
and sometimes in between



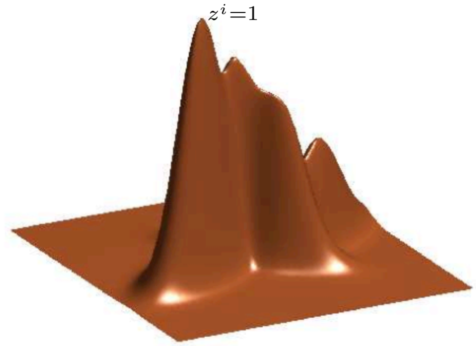
# Summary of GMM Concept

- 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*** **:**

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

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




iterative alg. for MLE

# Expectation Maximization

Machine Learning – CSE546
Emily Fox
University of Washington

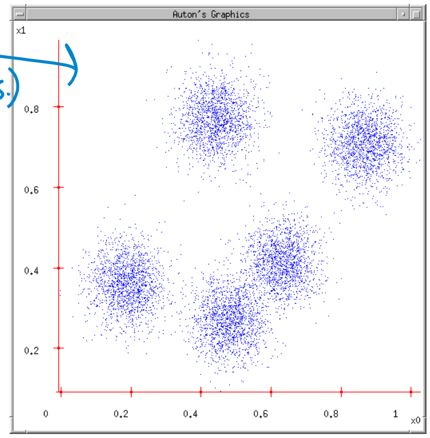
November 6, 2013

## Next… back to Density Estimation

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

Goal is to fit a MoG (mix. of Gauss.) to this data

Learn: $\{\pi_k, \mu_k, \Sigma_k\}$

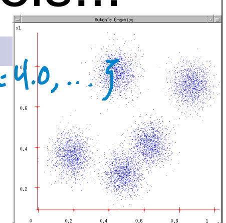


## But we don't see class labels!!!

In classification $x^i = \{GPA = 3.9,\ ML\ Grade = 4.0, \dots\}$

$z^i = \{Role = VP\}$



- MLE:
  - $\underset{\pi,\mu,\Sigma}{\text{argmax}} \prod_i P(z^i, x^i)$

    class labels ↗ ↖ features

- But we don't know $z^i$ ← "nuissance variable"
- Maximize marginal likelihood:
  - $\underset{\pi,\mu,\Sigma}{\text{argmax}} \prod_i P(x^i) = \text{argmax} \prod_i \sum_{k=1}^{K} P(z^i=k, x^i)$

    only obs. this

    $\underbrace{p(x^i \mid z^i = k)}_{N(x^i; \mu_k, \Sigma_k)}\ \underbrace{P(z^i = k)}_{\pi_k}$

Sum/avg. out unobserved variables

Sum role = {VP, Engineer, Barista}

weigh by prob. $P(z = VP)$, $P(z = Eng.)$, $P(z = barista)$

# Special case: spherical Gaussians and hard assignments

$P(\mathbf{x}^i \mid z^i = k)$ $\quad$ $P(z^i=k)$

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

$\mathbf{x}^i \in \mathbb{R}^d$

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

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

$$\Sigma_k = \begin{bmatrix} \sigma^2 & & 0 \\ & \ddots & \\ 0 & & \sigma^2 \end{bmatrix} = \sigma^2 I$$

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

$\rightarrow P(z^i=k) = \begin{cases} 1 & C(i)=k \\ 0 & \text{otherwise} \end{cases}$

want to max this:

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

- Same as K-means!!!

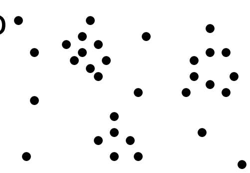
$$\max_{C,\mu,\sigma} \prod_{i=1}^{N} e^{-\frac{1}{2\sigma^2}\|x^i - \mu_{C(i)}\|^2} = \max \ln \prod = \max \sum -\frac{1}{2\sigma^2}\|x^i - \mu_{C(i)}\|^2$$

$$= \min_{C,\mu,\sigma} \sum_{i=1} \|x^i - \mu_{C(i)}\|^2 \leftarrow \text{exactly k-means obj.}$$



# EM: "Reducing" Unsupervised Learning to Supervised Learning

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



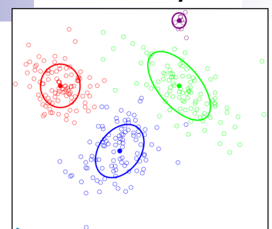
easy

- 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

iterate until convergence like in k-means

# Generic Mixture Models

MoG Example:



- Observations: $x^1, \ldots, x^N$ with $x^i \in \mathbb{R}^d$
- Parameters:

$\pi = [\pi_1, \ldots, \pi_K]$ mix. weights

$\phi = \{\phi_1, \ldots, \phi_K\}$ like. params, (e.g. $\phi_k = \{\mu_k, \Sigma_k\}$) for MoG

defines cluster

$\theta = \{\pi, \phi\}$

- Likelihood:

$$p(x^i \mid \theta) = \sum_{k=1}^{K} \pi_k \, p(x^i \mid \phi_k)$$

e.g. $N(x^i; \mu_k, \Sigma_k)$

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

©Emily Fox 2013 19

# ML Estimate of Mixture Model Params

- Marginal Log likelihood

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

$$p(X \mid \theta) = \prod_i p(x^i \mid \theta) = \prod_i \sum_{z^i} p(x^i, z^i \mid \theta)$$

- Want ML estimate

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

- Neither convex nor concave and local optima

©Emily Fox 2013 20

$\sum_i \log \pi_{z_i} = \sum_{j=1}^{K} \sum_{i:z_i=j} \log \pi_j = \sum_{j=1}^{K} N_j \log \pi_j \qquad N_j = |\{i : z^i = j\}|$

# If "complete" data were observed…

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

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

$$= \sum_{j=1}^{K} \sum_{i:z^i=j} \log p(x^i \mid z^i = j, \phi_j) + \sum_{j=1}^{K-1} N_j \log \pi_j + N_K \log\left(1 - \sum_{j=1}^{K-1} \pi_j\right)$$

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

$\sum \pi_j = 1$

$$\hat{\phi}_k = \arg\max_{\phi_k} \sum_{i:z^i=k} \log p(x^i \mid z^i = k, \phi_k) \qquad \hat{\pi}_k = \frac{N_k}{N} \quad k = 1, \dots, K-1$$

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

$$\begin{cases} \hat{\mu}_k = \frac{1}{N_k} \sum_{i:z^i=k} x_i \\ \hat{\Sigma}_k = \frac{1}{N_k} \sum_{i:z^i=k} x_i x_i^T - \hat{\mu}_k \hat{\mu}_k^T \end{cases} \qquad \hat{\pi}_k = \frac{N_k}{N}$$





# 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

$z^i \xrightarrow{\text{max}} \hat{\theta} \xrightarrow{\text{est.}} z^i \qquad \hat{\theta} = \{\hat{\phi}_k, \pi_k\}$

- Example: MoG (derivation soon… + HW)
  1. Infer "responsibilities" ← prev. iter.

soft weights

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

  2. Optimize parameters

$$\max \text{ w.r.t. } \pi_k : \quad \hat{\pi}_k^{(t)} = \frac{1}{N} \sum_i r_{ik} = \frac{r_k}{N}$$ ← soft counts!

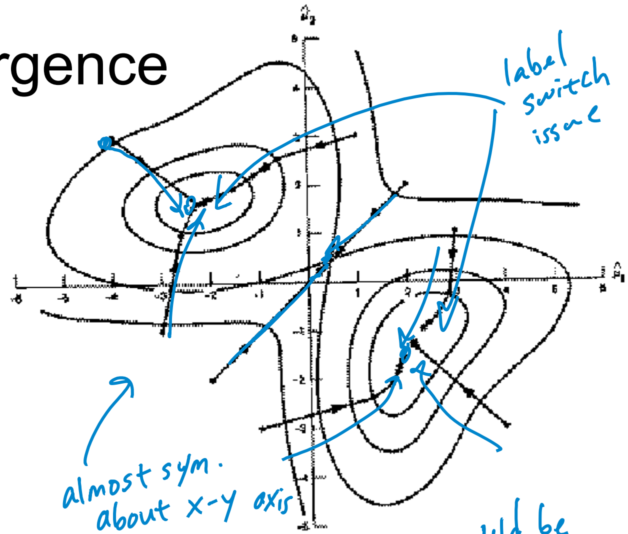
$$\max \text{ w.r.t. } \mu_k, \Sigma_k :$$

$$\hat{\mu}_k^{(t)} = \frac{\sum r_{ik} x_i}{N}$$ ← weighted mean

$$\hat{\Sigma}_k^{(t)} = \frac{1}{r_k} \sum r_{ik} x_i x_i^T - \hat{\mu}_k^{(t)} \hat{\mu}_k^{(t)T}$$

# E.M. Convergence

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



label switch issue

almost sym. about x-y axis

(under sym. prior on classes, would be exactly sym.)

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

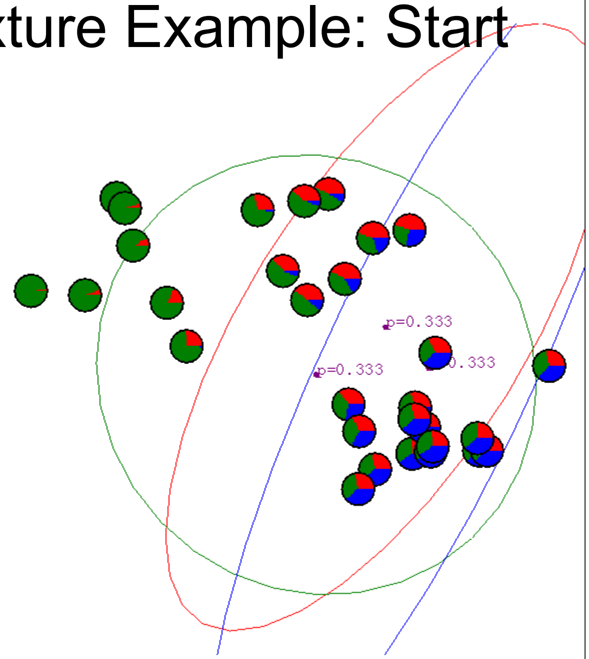




# Gaussian Mixture Example: Start

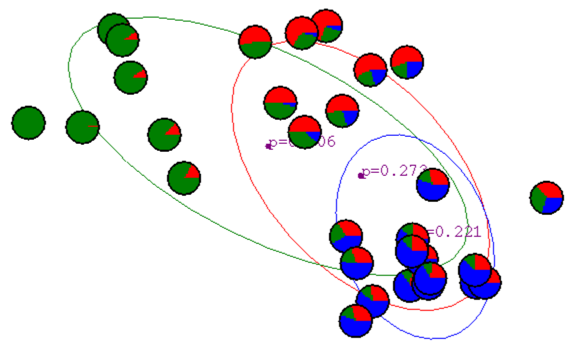
start with initial est. of $\pi^{(0)}, \theta^{(0)}$

→ lead to initial "responsibilities"



# After first iteration

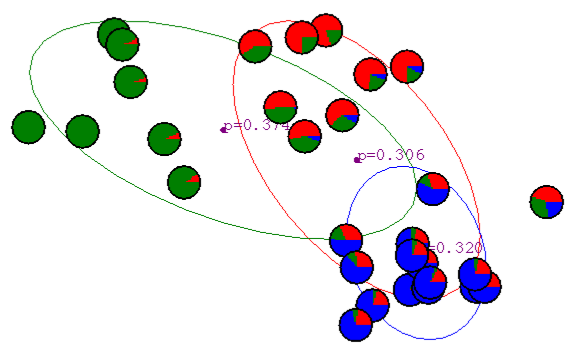
max. like
given soft
assignments



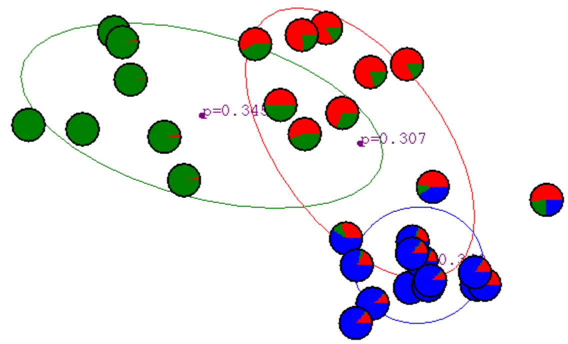

→ use new
$\pi^{(1)}$, $\phi^{(1)}$
to compute
new $r_{ik}$



# After 2nd iteration

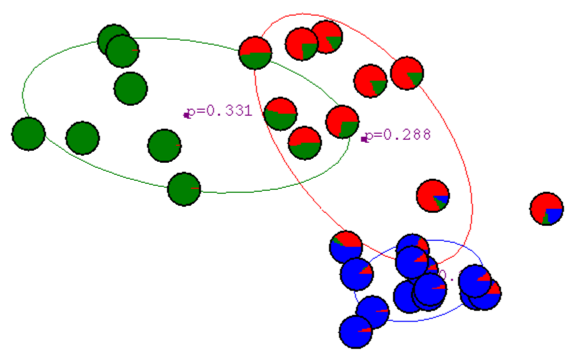


# After 3rd iteration

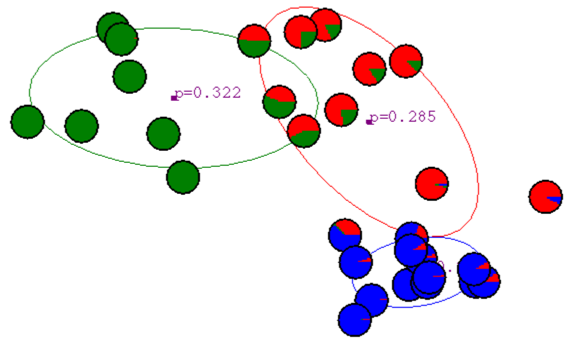





# After 4th iteration

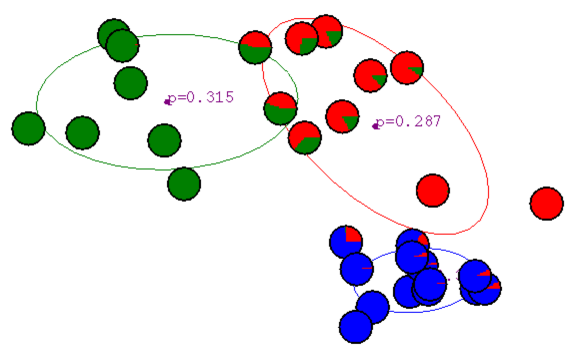


# After 5th iteration




©Emily Fox 2013 29

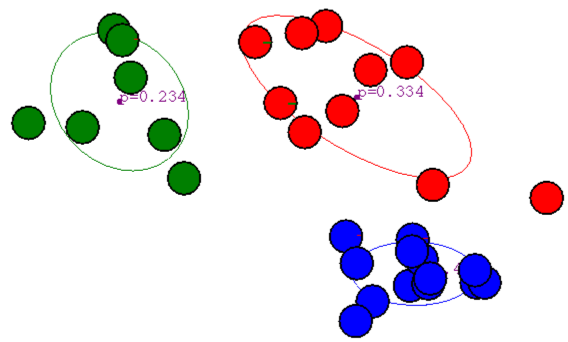
# After 6th iteration




©Emily Fox 2013 30

# After 20th iteration

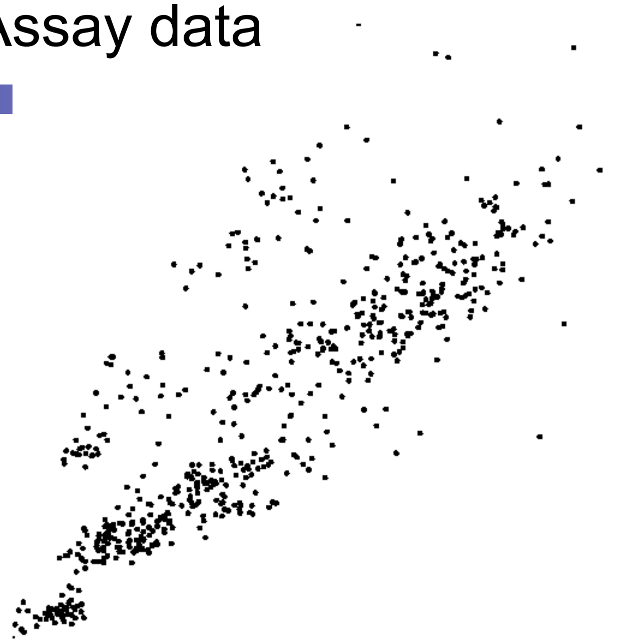
looks pretty good



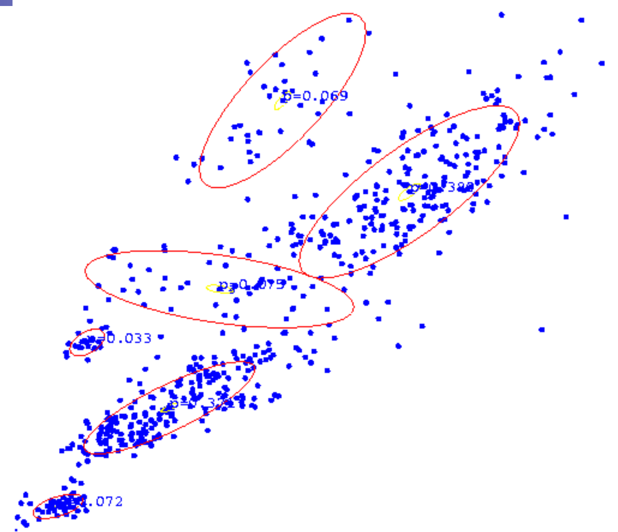




# Some Bio Assay data



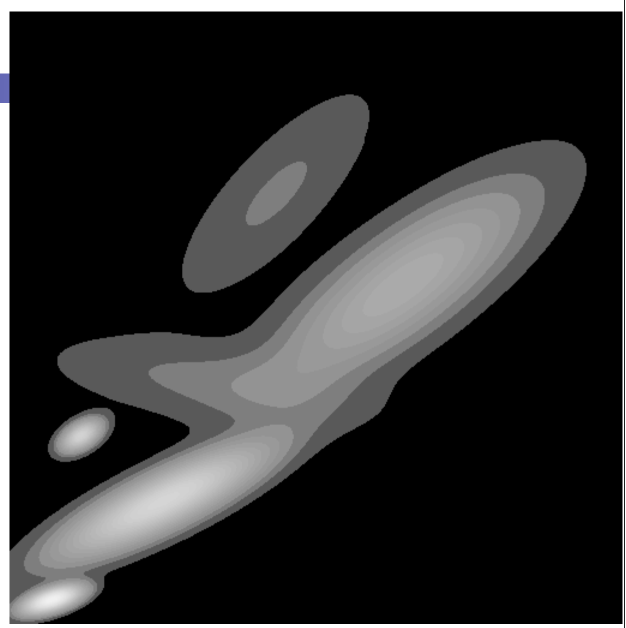
# GMM clustering of the assay data







# Resulting Density Estimator



# Expectation Maximization (EM) – Setup

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

- Model: $x$ observable – *"incomplete" data* ← what we have
  $y$ not (fully) observable – *"complete" data* ← what we wish we had
  $\theta$ parameters

  introduce →

- Interested in maximizing (wrt $\theta$):

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

- Special case: ← non-invertible, deterministic fcn

$$x = g(y)$$

e.g. $y = \begin{bmatrix} z \\ x \end{bmatrix}$ ← class labels ← obs. in standard mix. models



# Expectation Maximization (EM) – Derivation

- Step 1
  - Rewrite desired likelihood in terms of complete data terms

$p(y, x \mid \theta)$ ← b/c $x = g(y)$

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

← quantity of interest

$$\Rightarrow \underbrace{\log p(x \mid \theta)}_{L_x(\theta)} = \log p(y \mid \theta) - \log p(y \mid x, \theta)$$

- Step 2
  - Assume estimate of parameters $\hat{\theta}$
  - Take expectation with respect to $p(y \mid x, \hat{\theta})$ "$E[\cdot \mid x, \hat{\theta}]$"

$$L_x(\theta) = \underbrace{E[\log p(y \mid \theta) \mid x, \hat{\theta}]}_{U(\theta, \hat{\theta})} + \underbrace{E[-\log p(y \mid x, \theta) \mid x, \hat{\theta}]}_{V(\theta, \hat{\theta})}$$

# Expectation Maximization (EM) – Derivation

- Step 3
  - Consider log likelihood of data at any $\theta$ relative to log likelihood at $\hat{\theta}$

$$L_x(\theta) - L_x(\hat{\theta}) = [U(\theta,\hat{\theta}) - U(\hat{\theta},\hat{\theta})] + [V(\theta,\hat{\theta}) - V(\hat{\theta},\hat{\theta})]$$

✓

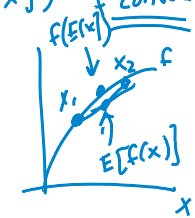
- **Aside: Gibbs Inequality** $E_p[\log p(x)] \geq E_p[\log q(x)]$

Proof: Use Jensen's Ineq. $E[f(x)] \leq f(E[x])$ for any $f$ convex

Here:

$$E_p[\log q] - E_p[\log p] = E_p\left[\log \frac{q}{p}\right]$$

$$\leq \log E_p\left[\frac{q}{p}\right] = \log \int_x \cancel{p(x)} \frac{q(x)}{\cancel{p(x)}} dx = 0$$






# Expectation Maximization (EM) – Derivation

$$L_x(\theta) - L_x(\hat{\theta}) = [U(\theta,\hat{\theta}) - U(\hat{\theta},\hat{\theta})] + \underbrace{[V(\theta,\hat{\theta}) - V(\hat{\theta},\hat{\theta})]}_{\geq 0}$$

- Step 4
  - Determine conditions under which log likelihood at $\theta$ exceeds that at $\hat{\theta}$

Using Gibbs inequality:

$$V(\theta,\hat{\theta}) = E[-\log p(y \mid x,\theta) \mid x,\hat{\theta}] \geq E[-\log p(y \mid x,\hat{\theta}) \mid x,\hat{\theta}]$$

$$= V(\hat{\theta},\hat{\theta}) \quad \forall \theta$$

If $U(\theta,\hat{\theta}) \geq U(\hat{\theta},\hat{\theta})$

choosing $\theta$ s.t. this is true means we're moving in the right direction (or at least not wrong)

Then

$$L_x(\theta) \geq L_x(\hat{\theta})$$

## Motivates EM Algorithm

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

- **E-Step**

  Compute $U(\theta, \hat{\theta}^{(t)}) = E[\log p(y \mid \theta) \mid x, \hat{\theta}^{(t)}]$

- **M-Step**

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

From before, $U(\hat{\theta}^{(t+1)}, \hat{\theta}^{(t)}) \geq U(\hat{\theta}^{(t)}, \hat{\theta}^{(t)})$

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




## Example – Mixture Models

- **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.

$p(x^i, z^i \mid \theta) = \pi_{z^i} p(x^i \mid \phi_{z^i}) =$

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

# Coordinate Ascent Behavior

- Bound log likelihood:

$$L_x(\theta) = U(\theta, \hat{\theta}^{(t)}) + V(\theta, \hat{\theta}^{(t)})$$
$$\geq$$
$$L_x(\hat{\theta}^{(t)}) = U(\hat{\theta}^{(t)}, \hat{\theta}^{(t)}) + V(\hat{\theta}^{(t)}, \hat{\theta}^{(t)})$$

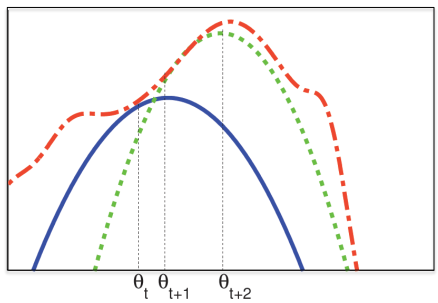



Figure from KM textbook



# Comments on EM

- Since Gibbs inequality is satisfied with equality only if *p*=*q*, any step that changes $\theta$ should strictly **increase likelihood**
- In practice, can replace the **M-Step** with increasing *U* instead of maximizing it (**Generalized EM**)
- Under certain conditions (e.g., in exponential family), can show that EM **converges to a stationary point** of $L_x(\theta)$
- Often there is a **natural choice for *y*** … has physical meaning
- If you want to choose any *y*, not necessarily *x=g(y)*, replace $p(y \mid \theta)$ in *U* with $p(y, x \mid \theta)$

## 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



## 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
- Be happy with this kind of probabilistic analysis
- Remember, E.M. can get stuck in local minima, and empirically it <u>DOES</u>
- EM is coordinate ascent