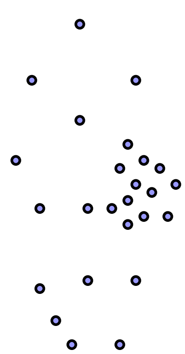
Mixtures of Gaussians

Machine Learning – CSE546
Sham Kakade
University of Washington

November 15, 2016

©Sham Kakade 2016

(One) bad case for k-means



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

©Sham Kakade 2016

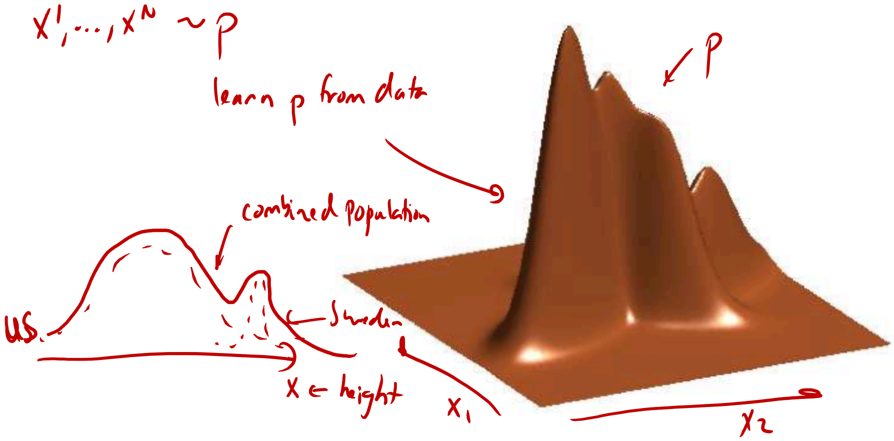
Density Estimation

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

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

learn p from data

combined population

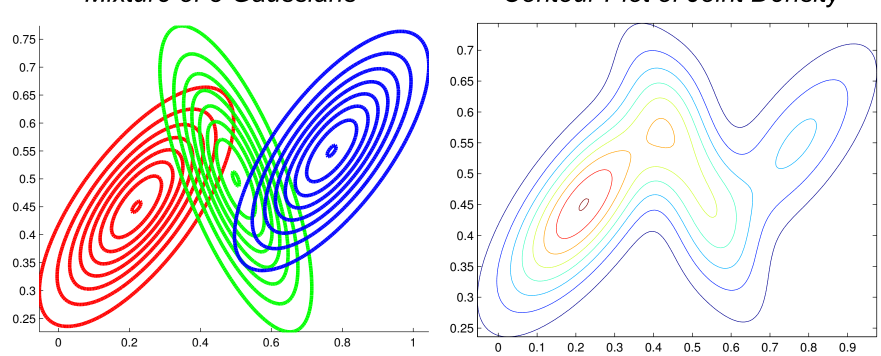


 ©Sham Kakade 2016

Density as Mixture of Gaussians

- Approximate density with a mixture of Gaussians

Mixture of 3 Gaussians *Contour Plot of Joint Density*



 ©Sham Kakade 2016

Gaussians in *d* Dimensions

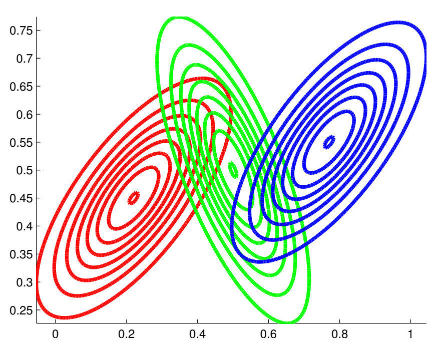
$$P(\mathbf{x}) = \frac{1}{(2\pi)^{m/2} \|\Sigma\|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\mu)^T \Sigma^{-1} (\mathbf{x}-\mu)\right]$$

©Sham Kakade 2016

Density as Mixture of Gaussians

- Approximate density with a mixture of Gaussians

Mixture of 3 Gaussians



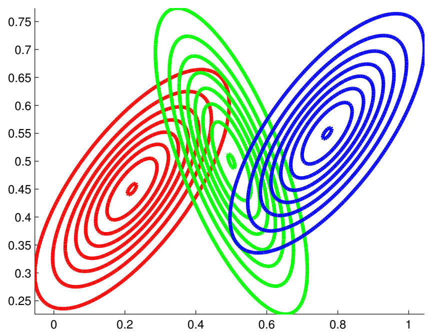
$$p(x^i|\pi, \mu, \Sigma) =$$

©Sham Kakade 2016

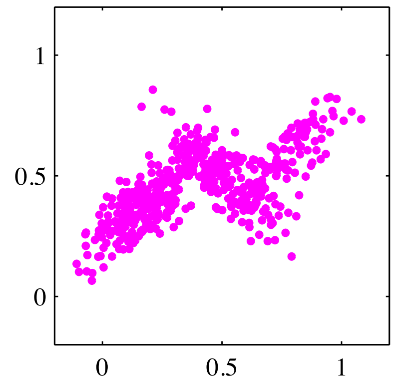
Density as Mixture of Gaussians

- Approximate with density with a mixture of Gaussians

Mixture of 3 Gaussians



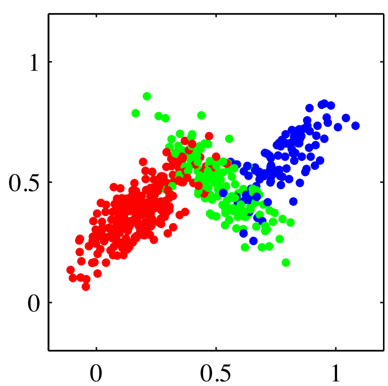
Our actual observations



C. Bishop, Pattern Recognition & Machine Learning

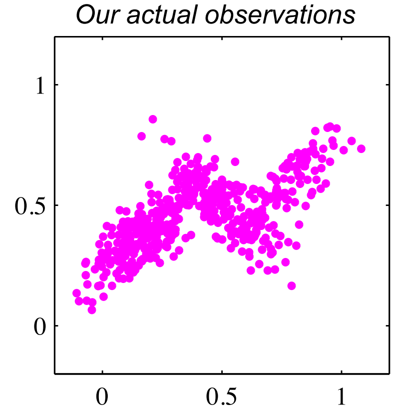
Clustering our Observations

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



Complete data labeled by true cluster assignments

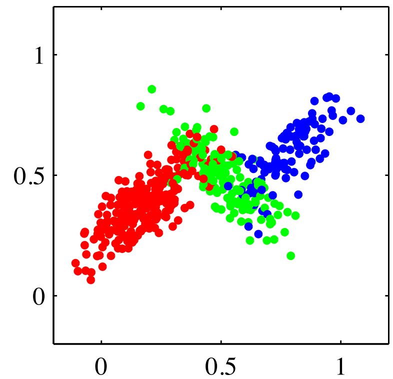
Our actual observations



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

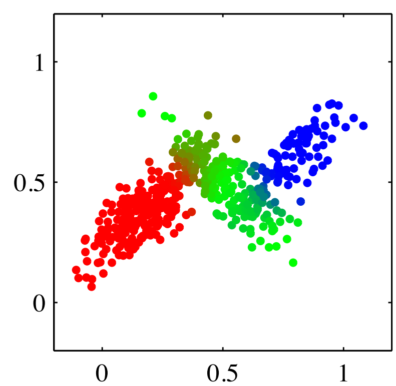
$$p(x^i|z^i,\pi,\mu,\Sigma) =$$

C. Bishop, Pattern Recognition & Machine Learning

©Sham Kakade 2016

Clustering our Observations

- We must infer the cluster assignments from the observations



Soft assignments to clusters

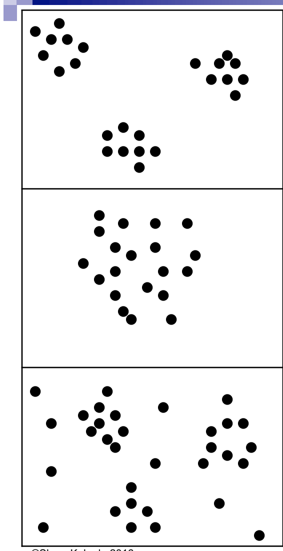
- Posterior probabilities of assignments to each cluster *given* model parameters:

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

C. Bishop, Pattern Recognition & Machine Learning

©Sham Kakade 2016

Unsupervised Learning: not as hard as it looks



Sometimes easy

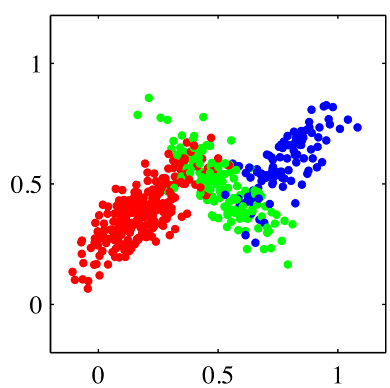
Sometimes impossible

and sometimes in between

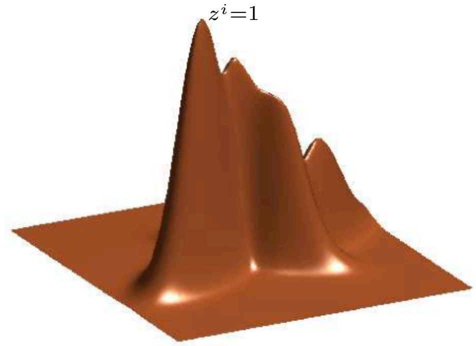
Summary of GMM Concept

- 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 :

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

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

©Sham Kakade 2016

Expectation Maximization

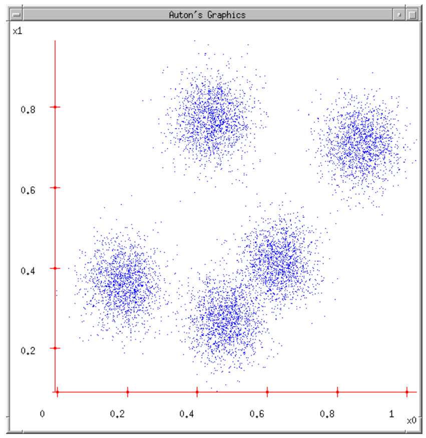
Machine Learning – CSE546
Sham Kakade
University of Washington

November 15, 2016

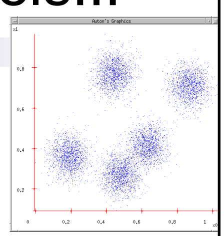
©Sham Kakade 2016

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:

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

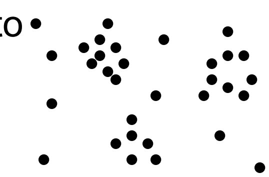
- If each x^i belongs to one class C(i) (hard assignment), marginal likelihood:

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

- Same as K-means!!!

EM: "Reducing" Unsupervised Learning to Supervised Learning

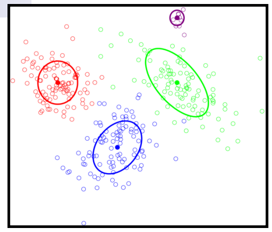
- If we knew assignment of points to classes ➔ Supervised Learning!



- 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"

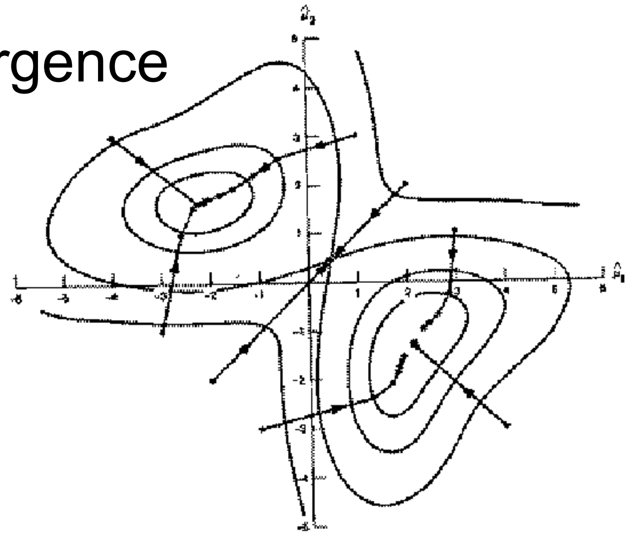
$$r_{ik} = p(z^i = k \mid x^i, \hat{\theta}^{(t-1)}) =$$

  2. Optimize parameters

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

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

E.M. Convergence

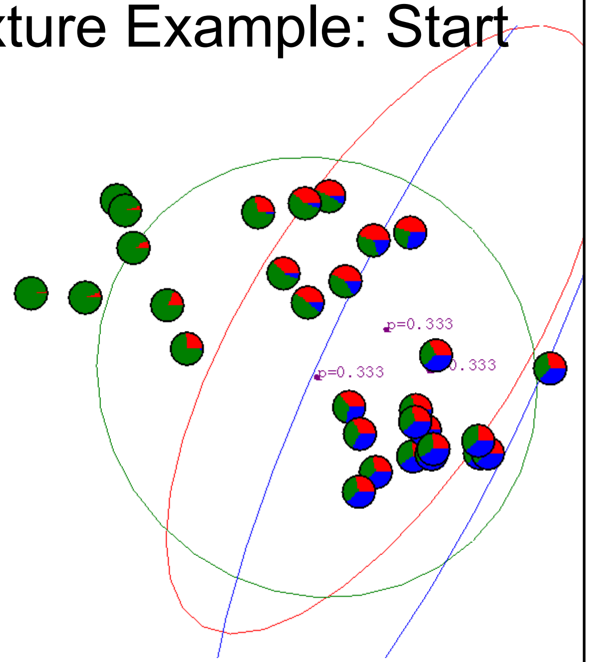


- EM is coordinate ascent on an interesting potential function
- Coord. ascent for bounded pot. func. ➔ convergence to a local optimum guaranteed

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

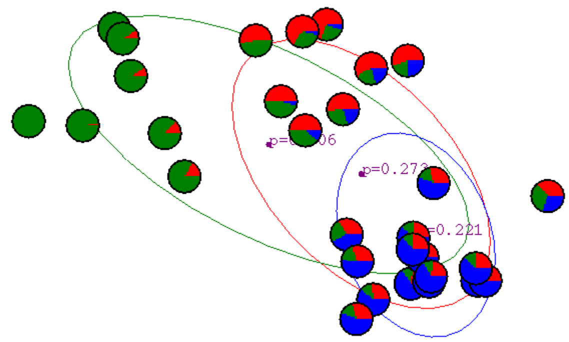
©Sham Kakade 2016

Gaussian Mixture Example: Start



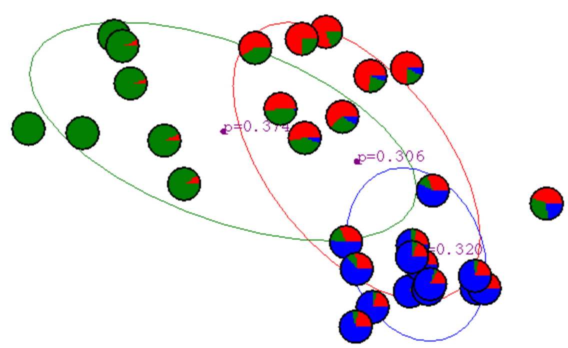
©Sham Kakade 2016

After first iteration



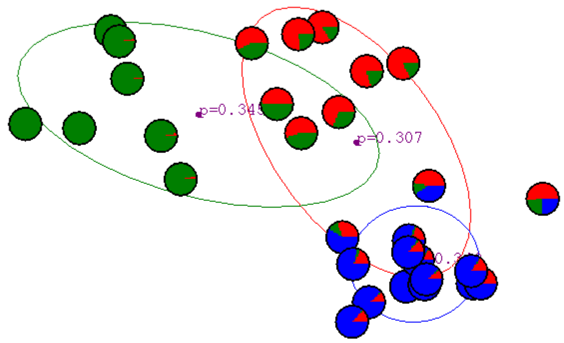
©Sham Kakade 2016

After 2nd iteration



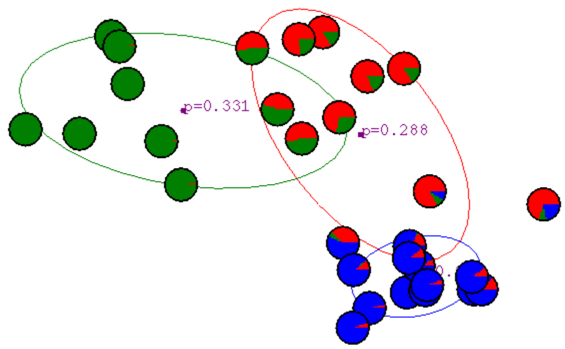
©Sham Kakade 2016

After 3rd iteration



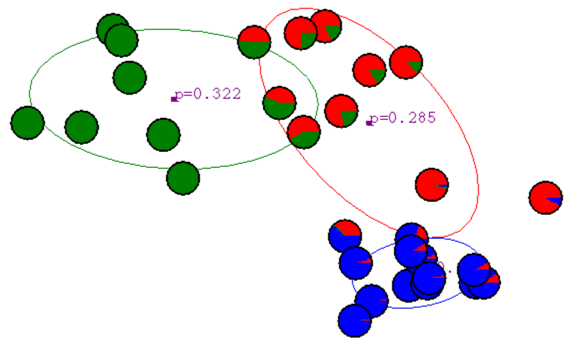
©Sham Kakade 2016

After 4th iteration

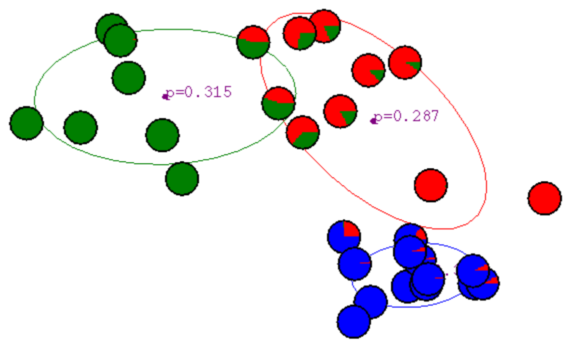


©Sham Kakade 2016

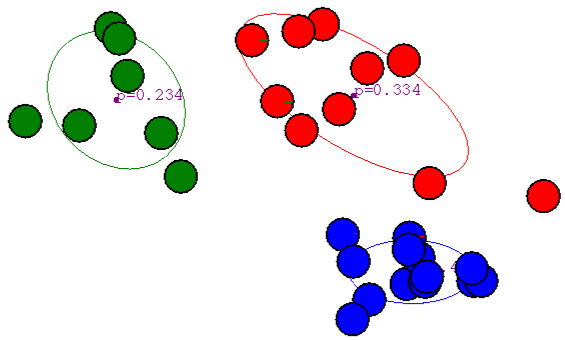
After 5th iteration



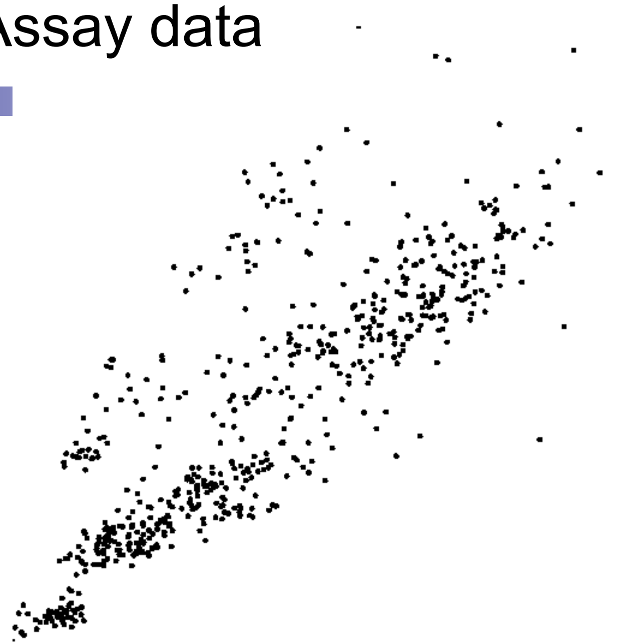
After 6th iteration



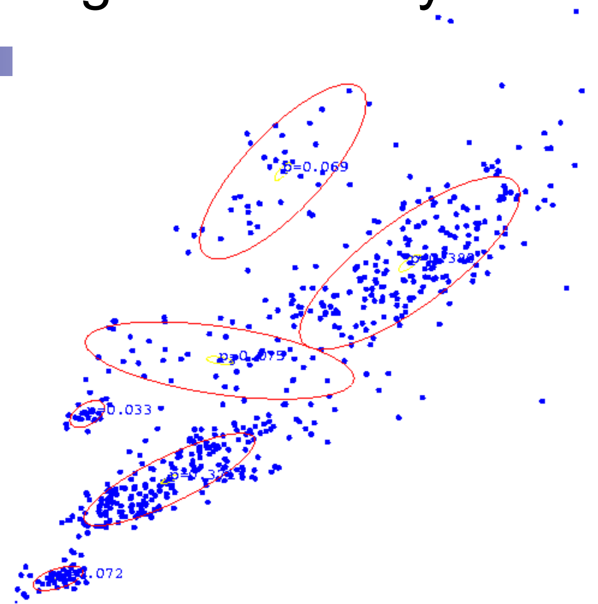
After 20th iteration



Some Bio Assay data

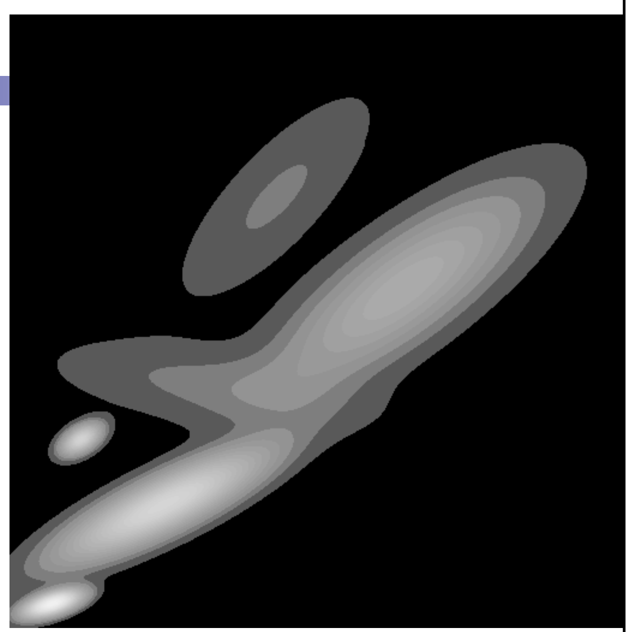


GMM clustering of the assay data



©Sham Kakade 2016

Resulting Density Estimator



©Sham Kakade 2016

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

©Sham Kakade 2016

Expectation Maximization (EM) – Setup

- More broadly applicable than just to mixture models considered so far
- Model: x observable – *"incomplete" data*
  y not (fully) observable – *"complete" data*
  θ parameters
- Interested in maximizing (wrt θ):

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

- Special case:

$$x = g(y)$$

©Sham Kakade 2016

Expectation Maximization (EM) – Derivation

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

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

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

©Sham Kakade 2016

Expectation Maximization (EM) – Derivation

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

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

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

  Proof:

©Sham Kakade 2016

Expectation Maximization (EM) – Derivation

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

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

  Using Gibbs inequality:

  If

  Then

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

Motivates EM Algorithm

- Initial guess:
- Estimate at iteration *t*:
- **E-Step**

  Compute

- **M-Step**

  Compute

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:

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

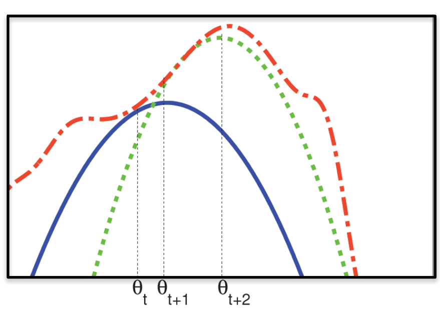


Figure from KM textbook

Comments on EM

- Since Gibbs inequality is satisfied with equality only if *p=q*, any step that changes θ 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 DOES
- EM is coordinate ascent

©Sham Kakade 2016