Only covered supervised learning

$X \to \mathbb{R}$ regression

$X \to \{0,1,\dots,k\}$ classification

Training data included labels

Clustering
K-means

Machine Learning – CSE546

Carlos Guestrin

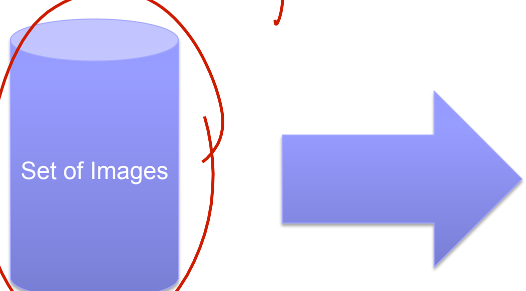
University of Washington

November 4, 2014

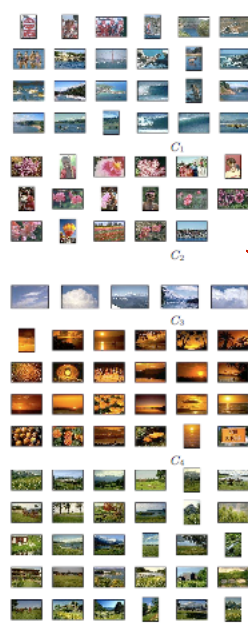
©Carlos Guestrin 2005-2014

Clustering images

given no labels



organize data into themes



beaches

flowers

©Carlos Guestrin 2005-2014

[Goldberger et al.]

K-means

d-dim vectors

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

Repeat until convergence: no point changes cluster membership

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

fix μ, opt C

- **Recenter**: $\mu_i^{(t+1)}$ becomes centroid of its point:

fix C, opt μ

  - $\mu_i^{(t+1)} \leftarrow \arg\min_\mu \sum_{j:C^{(t)}(j)=i} ||\mu - x_j||^2$ sum of points in cluster i

$$\mu_i^{(t+1)} = \frac{\sum_{j:C(j)=i} x_j}{|\{j : C(j)=i\}|}$$

  - Equivalent to $\mu_i \leftarrow$ average of its points!

©Carlos Guestrin 2005-2014

Mixtures of Gaussians

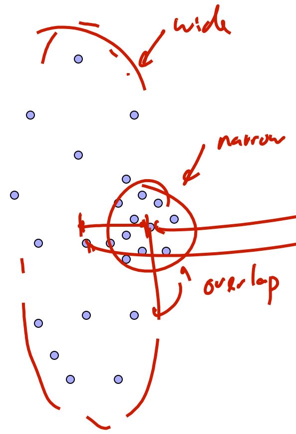
Machine Learning – CSE546
Carlos Guestrin
University of Washington

November 4, 2014

©Carlos Guestrin 2005-2014

(One) bad case for k-means

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



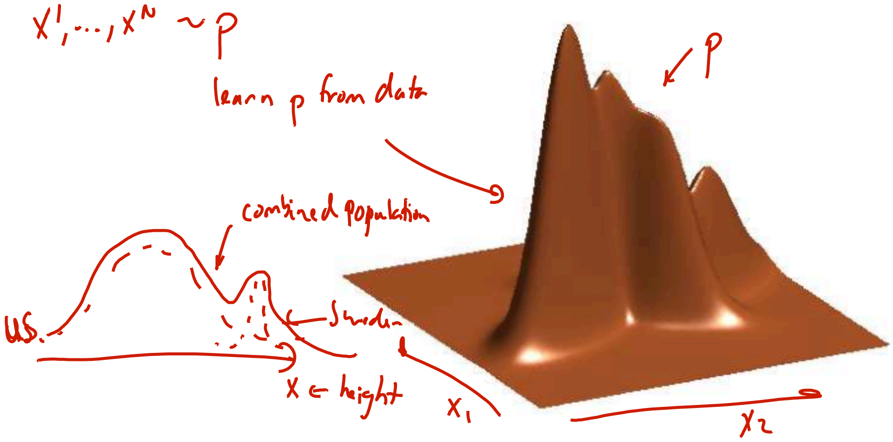
centers not enough to characterize membership

©Carlos Guestrin 2005-2014

Density Estimation

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

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

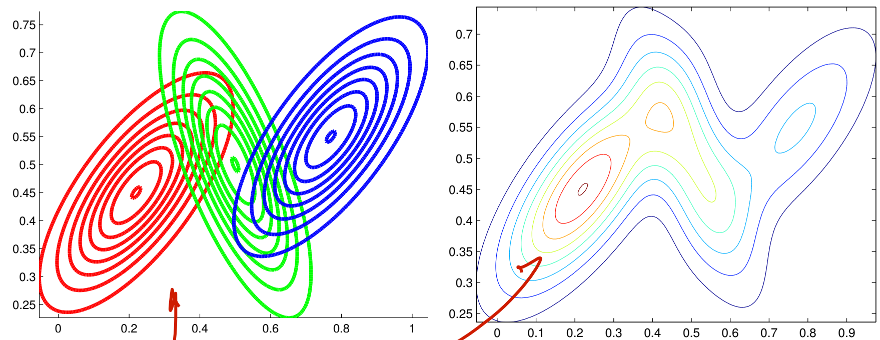


©Carlos Guestrin 2005-2014

Density as Mixture of Gaussians

- Approximate density with a mixture of Gaussians

Mixture of 3 Gaussians *Contour Plot of Joint Density*



original

p sum with weights π_i of each Gaussian

©Carlos Guestrin 2005-2014

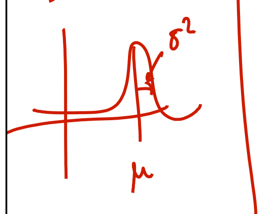
Gaussians in *d* Dimensions

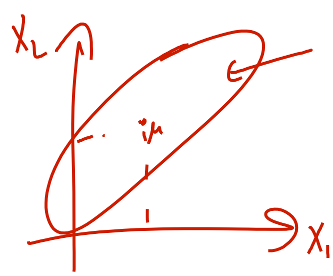
mean vector

covariance matrix

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

1D $P(x) \propto e^{-\frac{(\mu-x)^2}{2\sigma^2}}$





$$\Sigma = \begin{pmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{21} & \sigma_2^2 \end{pmatrix}$$

$\sigma_{12} = \sigma_{21}$

©Carlos Guestrin 2005-2014

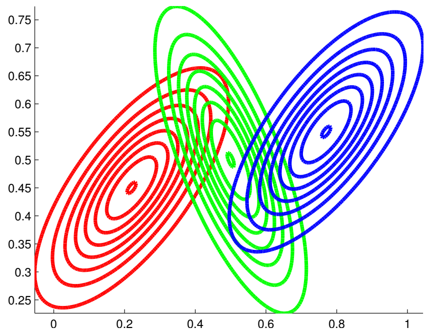
Density as Mixture of Gaussians

- Approximate density with a mixture of Gaussians

$\pi_1, \pi_2 \cdots \pi_k$ weights

$\sum_{i=1}^{k} \pi_i = 1$

Mixture of 3 Gaussians



$$P: \; p(x^j|\pi, \mu, \Sigma) = \sum_{i=1}^{k} \pi_i \, N(x^j|\mu_i, \Sigma_i)$$

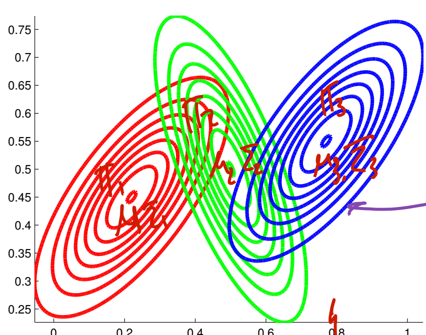
In 1D

©Carlos Guestrin 2005-2014

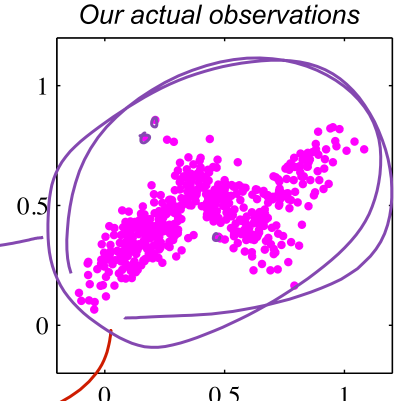
Density as Mixture of Gaussians

- Approximate with density with a mixture of Gaussians

Mixture of 3 Gaussians



Our actual observations



recover original densities How??

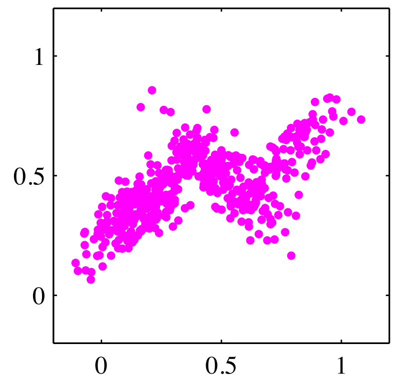
C. Bishop, Pattern Recognition & Machine Learning

©Carlos Guestrin 2005-2014

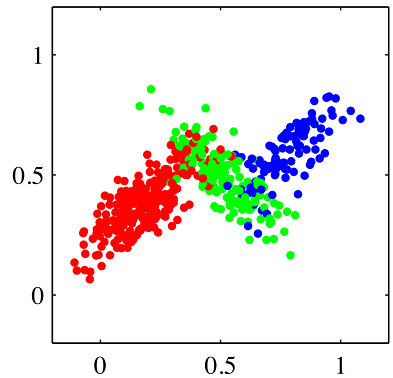
Clustering our Observations

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

Our actual observations



Complete data labeled by true cluster assignments



C. Bishop, Pattern Recognition & Machine Learning

©Carlos Guestrin 2005-2014

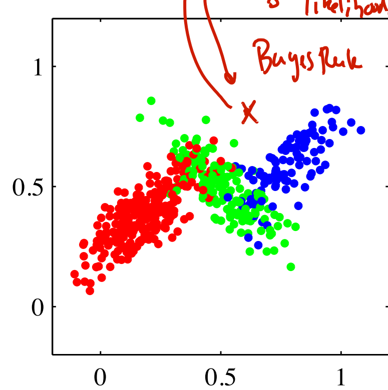
Clustering our Observations

$P(z^i | x^i, \pi, \mu, \Sigma) = \dfrac{P(x^i | z^i, \pi, \mu, \Sigma)\, P(z^i | \pi)}{P(x^i | \pi, \mu, \Sigma)}$

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

posterior of & likelihood prior

Bayes Rule



Complete data labeled by true cluster assignments

- Introduce latent cluster indicator variable z^i

$C(i) \to z^i$

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

$P(z^i = j) = \pi_j$

- Then we have

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

if we had observed the z^i, estimating params would be easy

C. Bishop, Pattern Recognition & Machine Learning

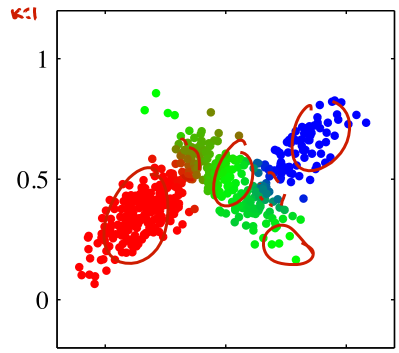
©Carlos Guestrin 2005-2014

$P(z|x) = \frac{P(x|z)\,P(z)}{P(x)}$

Clustering our Observations

- We must infer the cluster assignments from the observations

$\sum_{k=1}^{K} r_{ik} = 1$



Soft assignments to clusters

"responsibilities"

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

point i, cluster k

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

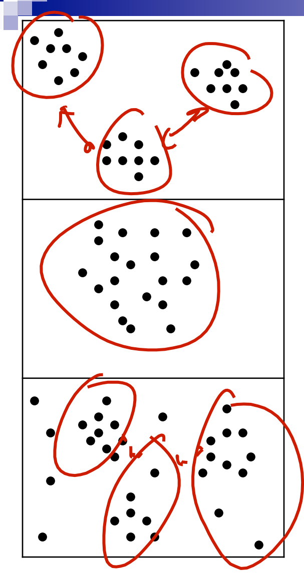
$$= \frac{\pi_k \, N(x^i | \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \, N(x^i | \mu_j, \Sigma_j)}$$

if I had params, the "classify" step is simple.

C. Bishop, Pattern Recognition & Machine Learning

©Carlos Guestrin 2005-2014

Unsupervised Learning: not as hard as it looks



Sometimes easy

Sometimes impossible

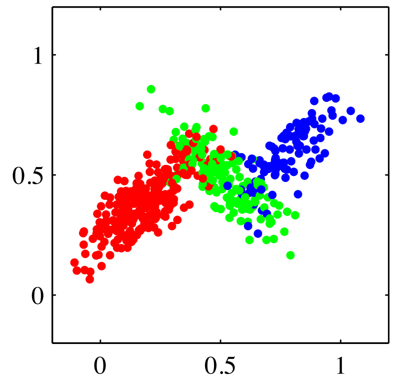
and sometimes in between

©Carlos Guestrin 2005-2014

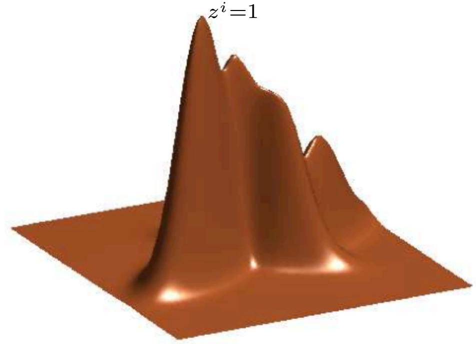
Summary of GMM Concept

- 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

©Carlos Guestrin 2005-2014 15

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 :

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

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

©Carlos Guestrin 2005-2014 16

Expectation Maximization

- coord. ascent alg.
- generalizes k-means

Machine Learning – CSE546

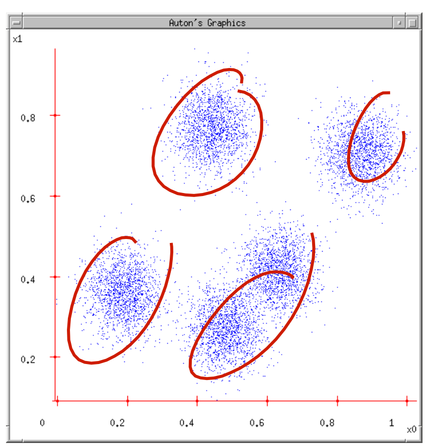
Carlos Guestrin
University of Washington

November 6, 2014

©Carlos Guestrin 2005-2014

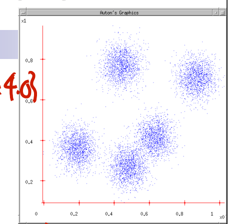
Next… back to Density Estimation

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



©Carlos Guestrin 2005-2014

But we don't see class labels!!!



- MLE: in classification: $x^i = \{GPA{:}3.9, MLGrade{:}4.0\}$, $z^i = \{Role = VP\}$
  - argmax $\prod_i P(z^i,x^i)$
    π, μ, Σ simple convex
- But we don't know z^i ← "latent" or "nuissance" variable
- Maximize marginal likelihood: $P(z^i=k) = \pi_k$, $P(x^i|z^i=k) \sim N(x^i|\mu_k,\Sigma_k)$
  - argmax $\prod_i P(x^i)$ = argmax $\prod_i \sum_{k=1}^K P(z^i=k,x^i)$
    π, μ, Σ only over observables
    sum out latent variables
    sum over Role = {VP, DS, Barista, ...}
    weigh by prob of each role
    non-convex & hard ≈ coordinate or gradient descent

©Carlos Guestrin 2005-2014

Special case: spherical Gaussians and hard assignments

max $\frac{1}{2\sigma}f$ ≡ min f max f ≡ max $\log f$

$N(x^i|\mu_k,\Sigma_k)$ π_k Role GPA/MLGrade

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

general Gaussian

- If P(X|z=k) is spherical, with same σ for all classes: $\Sigma_k = \begin{bmatrix} \sigma^2 & & 0 \\ & \sigma^2 & \\ 0 & & \ddots \end{bmatrix}$

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

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

$r_{ik} = \begin{cases} 1 & \text{if } C(i)=k \\ 0 & \text{otherwise} \end{cases}$

max$_{\mu,\sigma,C}$ MLE:

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

same as k-means objective!!

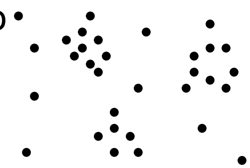
- Same as K-means!!!

$$\max_{C,\mu,\sigma} \sum_{i=1}^{N} -\frac{1}{2\sigma^2}\|x^i - \mu_{C(i)}\|^2 \equiv \min_{C,\mu} \sum_{i=1}^{N} \|x^i - \mu_{C(i)}\|^2$$

©Carlos Guestrin 2005-2014

EM: "Reducing" Unsupervised Learning to Supervised Learning

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



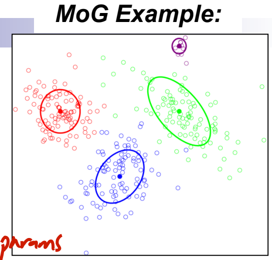
- Expectation-Maximization (EM)
  - E □ Guess assignment of points to classes
    - In standard ("soft") EM: each point associated with prob. of being in each class
  - M □ Recompute model parameters
  - □ Iterate

coordinate descent

like k-means

Generic Mixture Models

MoG Example:



- Observations: $X^1, \ldots, X^N$ with e.g. $x^i \in \mathbb{R}^d$
- Parameters: $\pi = \{\pi_1, \ldots \pi_K\}$ mix weights

  $\phi = \{\phi_1, \ldots \phi_K\}$ mix component params

  e.g. $\phi_k = \{\mu_k, \Sigma_k\}$

  $\theta = \{\pi, \phi\}$
- Likelihood:

$$P(x^i \mid \theta) = \sum_{k=1}^{K} \pi_k \, P(x^i \mid \phi_k) \longleftarrow N(x^i \mid \mu_k, \Sigma_k)$$

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

ML Estimate of Mixture Model Params

- Log likelihood

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

- Want ML estimate

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

- Neither convex nor concave and local optima

If "complete" data were observed…

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

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

Simply Counts

$$= \sum_{i=1}^{N} \log \underbrace{P(z^i|\theta)}_{\pi_{z^i}} + \sum_{i=1}^{N} \underbrace{\log P(x^i|z^i,\theta)}_{\text{Params for each component}}$$

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

$$\hat{\pi}_k = \frac{\overbrace{Count(z^i=k)}^{=N_k}}{N}$$

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

$$\hat{\mu}_k = \frac{1}{N_k} \sum_{i: z^i = k} x^i$$, Similarly for Σ_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

  $\hat{\theta}$ z^i (responsibilities)
- Example: MoG (derivation soon…)
  1. Infer "responsibilities"

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

weighted data

  2. Optimize parameters

$$\max \text{ w.r.t. } \pi_k : \quad \hat{\pi}_k^{(t)} = \frac{1}{N} \sum_{i=1}^{N} r_{ik}$$

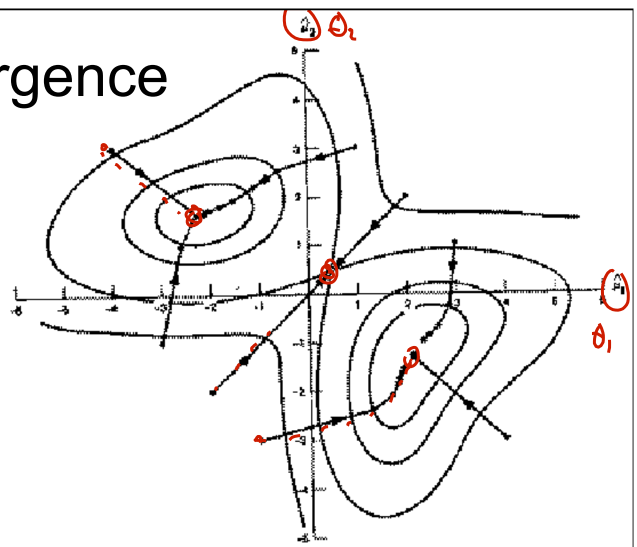
$$\max \text{ w.r.t. } \mu_k, \Sigma_k :$$

$$\mu_k^{(t)} = \frac{1}{N} \sum_{i=1}^{N} r_{ik} x^i \quad , \text{ similarly for } \Sigma_k$$

weighted mean

E.M. Convergence

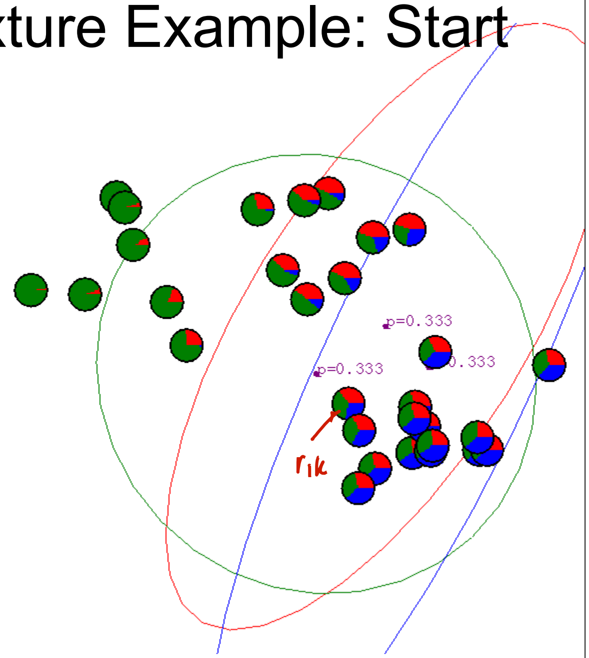
- EM is coordinate ascent on an interesting potential function
- Coord. ascent for bounded pot. func. ➔ convergence to a local optimum guaranteed



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

Gaussian Mixture Example: Start

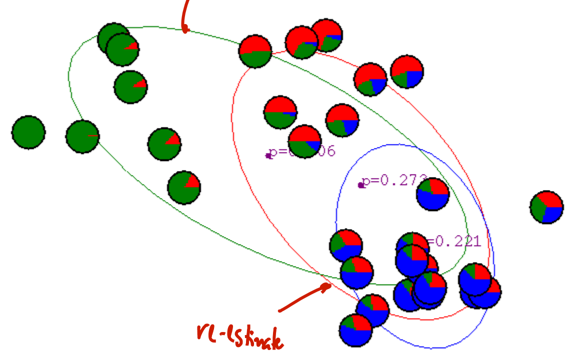
guess some $\theta^{(0)}$



©Carlos Guestrin 2005-2014

After first iteration

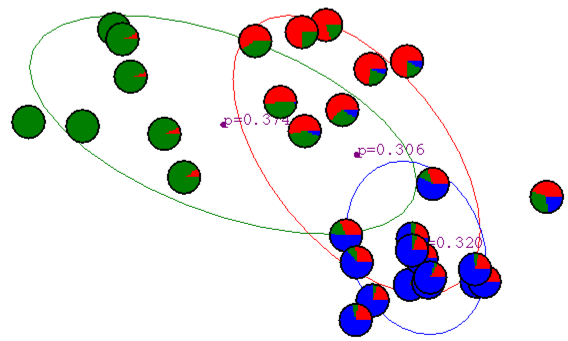
re-estimate π, μ, σ



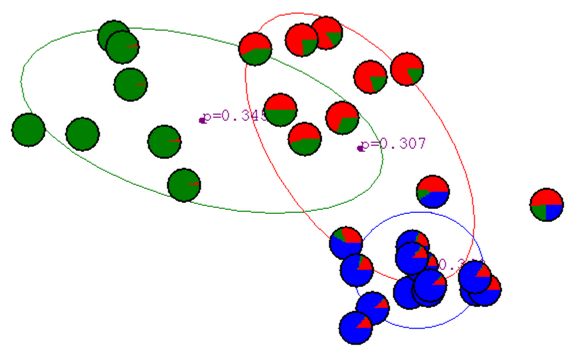
re-estimate r_{ik}

©Carlos Guestrin 2005-2014

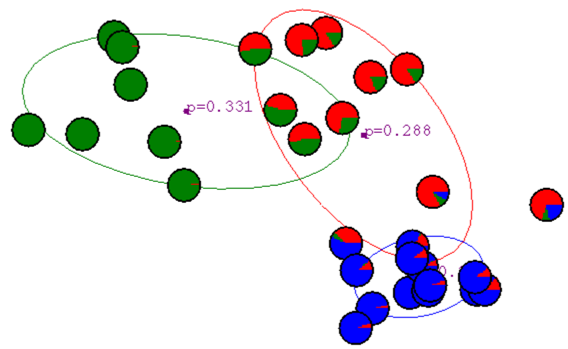
After 2nd iteration



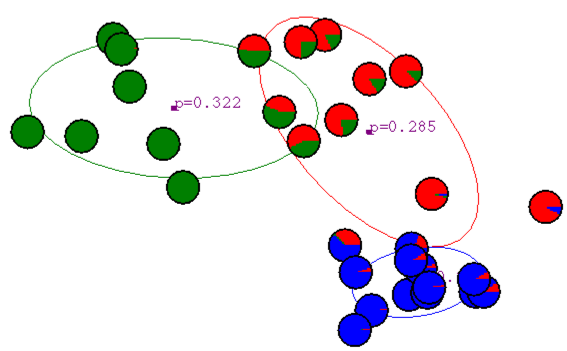
After 3rd iteration



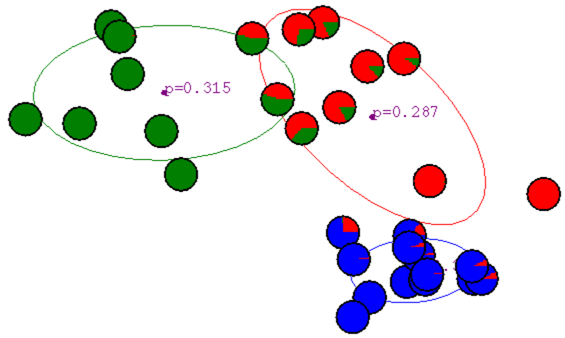
After 4th iteration



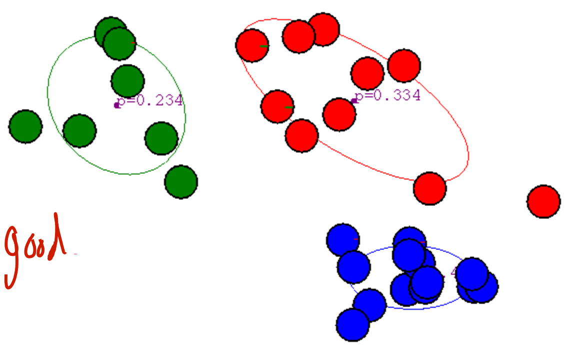
After 5th iteration



After 6th iteration

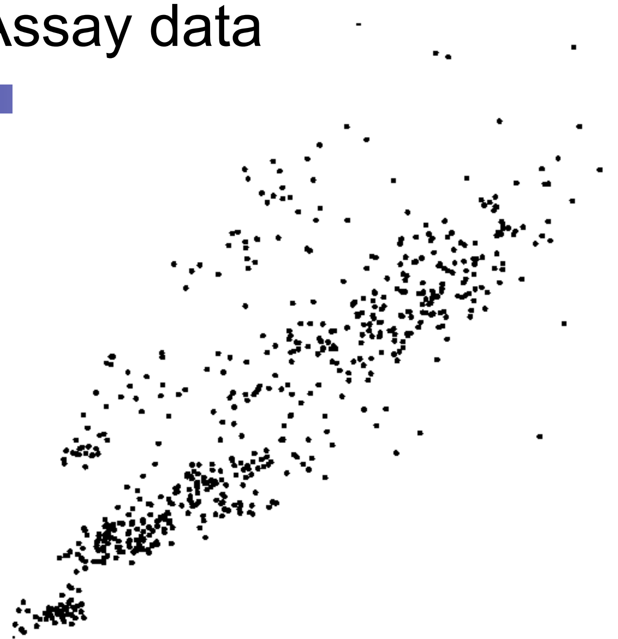


After 20th iteration

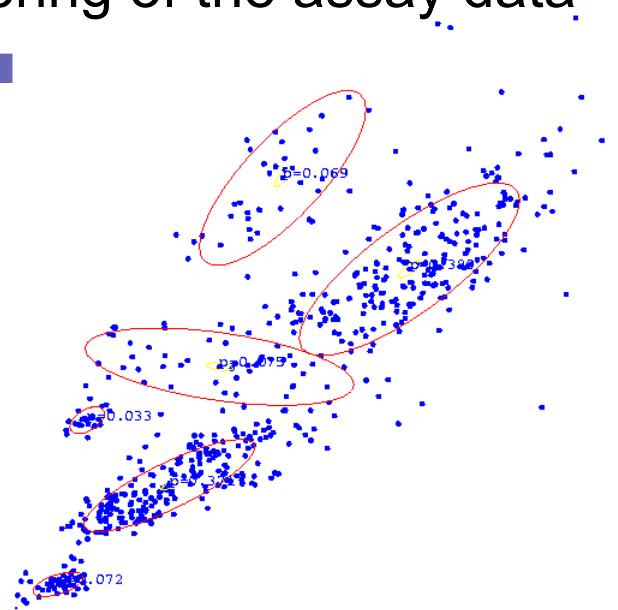


looks pretty good.

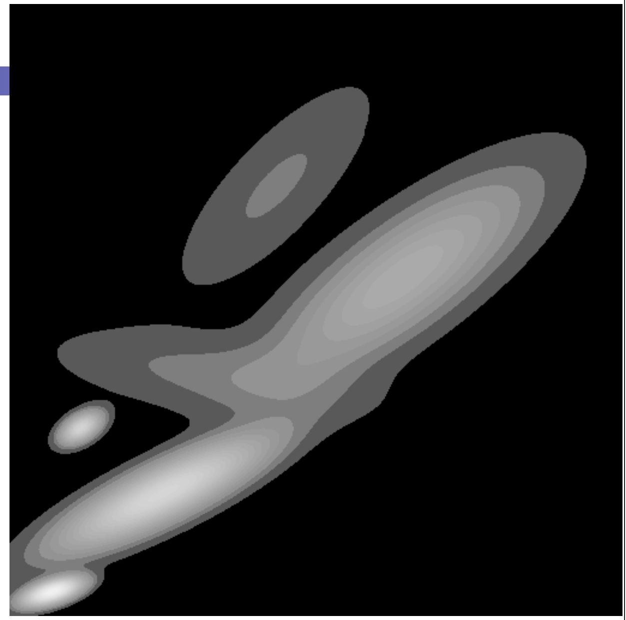
Some Bio Assay data



GMM clustering of the assay data



Resulting Density Estimator



©Carlos Guestrin 2005-2014

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

©Carlos Guestrin 2005-2013

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

  D k means
- Can be quite important to quality of solution in practice

©Carlos Guestrin 2005-2014 39

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

©Carlos Guestrin 2005-2014 40

Expectation Maximization (EM) – Setup

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

what we have

- Model: x observable – *"incomplete" data*

  introduce latent para y not (fully) observable – *"complete" data* ← what we wish we had

  θ parameters

- Interested in maximizing (wrt θ):

$$\max_{\theta} \; p(x \mid \theta) = \sum_y p(x, y \mid \theta)$$

- Special case:

$$x = g(y)$$

$$y = \begin{bmatrix} z \\ x \end{bmatrix}$$

hidden cluster labels

observation

Expectation Maximization (EM) – Derivation

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

$$p(y \mid \theta) = p(y \mid x, \theta) \underbrace{p(x \mid \theta)}$$

quantity we want to maximize

what we want to max

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

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

don't observe y, so average over y

$E_y[\cdot \mid x, \hat{\theta}]$

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

Expectation Maximization (EM) – Derivation

Jensens inequality: f is concave

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

θ from previous iteration

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

new θ we are trying to maximize

$$L_x(\theta) - L_x(\hat{\theta}) = \underbrace{[U(\theta,\hat{\theta}) - U(\hat{\theta},\hat{\theta})]}_{\text{focus on max this}} + \underbrace{[V(\theta,\hat{\theta}) - V(\hat{\theta},\hat{\theta})]}_{\ge 0}$$

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

Proof:

?

Expectation Maximization (EM) – Derivation

$$\underbrace{L_x(\theta) - L_x(\hat{\theta})}_{\text{want to max}} = \underbrace{[U(\theta,\hat{\theta}) - U(\hat{\theta},\hat{\theta})]}_{\text{focus}} - \underbrace{[V(\theta,\hat{\theta}) - V(\hat{\theta},\hat{\theta})]}_{\ge 0}$$

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

Using Gibbs inequality:

$$V(\theta,\hat{\theta}) = E[-\log P(y|x,\theta) \,|\, x,\hat{\theta}] \ge E[-\log P(y|x,\hat{\theta}) \,|\, x,\hat{\theta}] = V(\hat{\theta},\hat{\theta})$$

If $\underbrace{U(\theta,\hat{\theta}) \ge U(\hat{\theta},\hat{\theta})}$ choose θ such that U increases

Then

$$\underbrace{L_x(\theta) \ge L_x(\hat{\theta})}_{\text{making some progress}}$$

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

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:

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

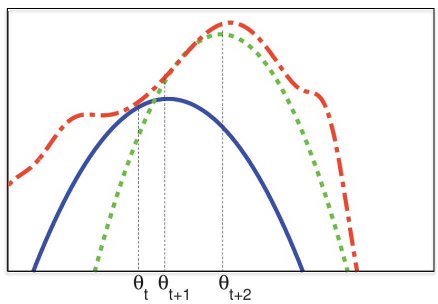


Figure from KM textbook

©Carlos Guestrin 2005-2014

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

©Carlos Guestrin 2005-2014

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

©Carlos Guestrin 2005-2014

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

©Carlos Guestrin 2005-2014