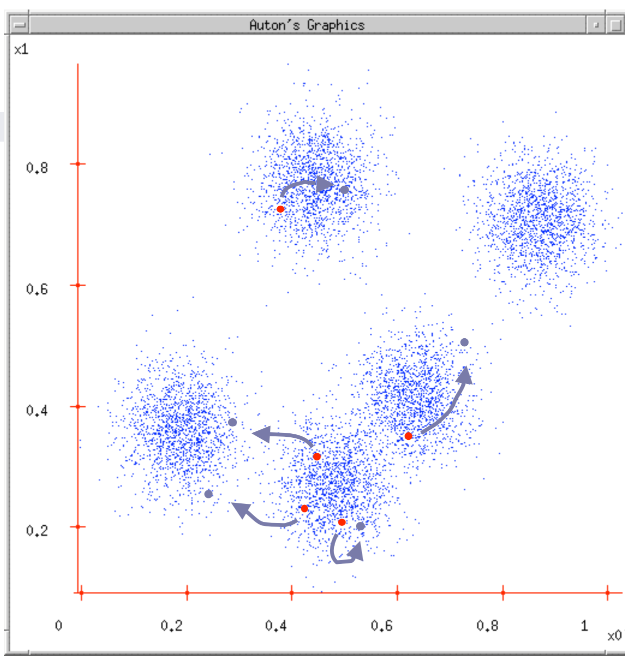
EM

Machine Learning – 10701/15781
Carlos Guestrin
Carnegie Mellon University

November 19th, 2007

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…
5. …and jumps there
6. …Repeat until terminated!



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: center of point j is closest to j
  - $C^{(t)}(j) \leftarrow \arg\min_i ||\mu_i - x_j||^2$
- **Recenter**: μ_i becomes centroid of its point:
  - $\mu_i^{(t+1)} \leftarrow \arg\min_\mu \sum_{j:C(j)=i} ||\mu - x_j||^2$ opt $\mu_i = \frac{\sum_{j:C(j)=i} x_j}{\sum_{j:C(j)=i} 1}$ is the mean!! ☺
  - Equivalent to $\mu_i \leftarrow$ average of its points!

Does K-means converge??? Part 2

- Optimize potential function:

$$\min_\mu \min_C F(\mu, C) = \min_\mu \min_C \sum_{i=1}^{k} \sum_{j:C(j)=i} ||\mu_i - x_j||^2$$

$C = \bar{C}$

- Fix C, optimize μ

$$\min_\mu \sum_{i=1}^{K} \sum_{j:\bar{C}(j)=i} ||\mu_i - x_j||^2$$

$$= \sum_{i=1}^{K} \underbrace{\min_{\mu_i} \sum_{j:\bar{C}(j)=i} ||\mu_i - x_j||^2}_{\mu_i \text{ is mean of points in cluster } i}$$

$\Rightarrow$ recenter in K-means.

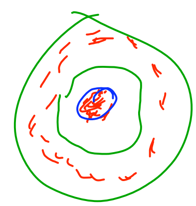
Coordinate descent algorithms

$$\min_\mu \min_C F(\mu, C) = \min_\mu \min_C \sum_{i=1}^{k} \sum_{j:C(j)=i} ||\mu_i - x_j||^2$$

- Want: $\min_a \min_b F(a,b)$
- Coordinate descent:
  - fix a, minimize b
  - fix b, minimize a
  - repeat
- Converges!!!
  - if F is bounded
  - to a (often good) local optimum
    - as we saw in applet (play with it!)

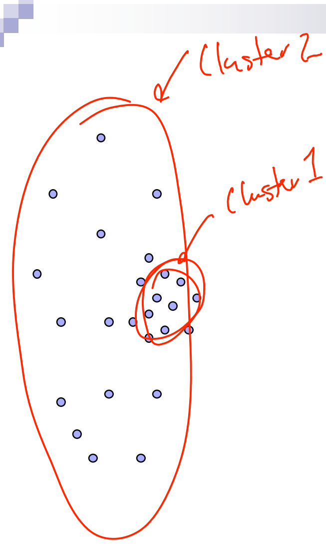
F(a,b)

∀a,b F(a,b) ≥



- K-means is a coordinate descent algorithm!

(One) bad case for k-means



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

Gaussian Bayes Classifier Reminder

likelihood functions   prior   multinomial

$$P(y=i\,|\,\mathbf{x}_j) = \frac{p(\mathbf{x}_j\,|\,y=i)\boxed{P(y=i)}}{p(\mathbf{x}_j)}$$

$P(Y=1) = 0.2$
$P(Y=2) = 0.7$
$P(Y=3) = 0.1$

$$P(y=i\,|\,\mathbf{x}_j) \propto \underbrace{\frac{1}{(2\pi)^{m/2}\,\|\Sigma_i\|^{1/2}}}_{\text{constant}}\underbrace{\exp\left[-\frac{1}{2}\left(\mathbf{x}_j-\mu_i\right)^T\Sigma_i^{-1}\left(\mathbf{x}_j-\mu_i\right)\right]}_{P(x_j|y=i)}P(y=i)$$

$x_j \in \mathbb{R}^d$

$$P(x_j|y=i) \propto e^{-\frac{1}{2}(x_j-\mu_i)^T\Sigma_i^{-1}(x_j-\mu_i)}$$

mean vector

$$\mu_i = \begin{bmatrix}\mu_{i,1}\\ \mu_{i,2}\\ \vdots\\ \mu_{i,d}\end{bmatrix}$$

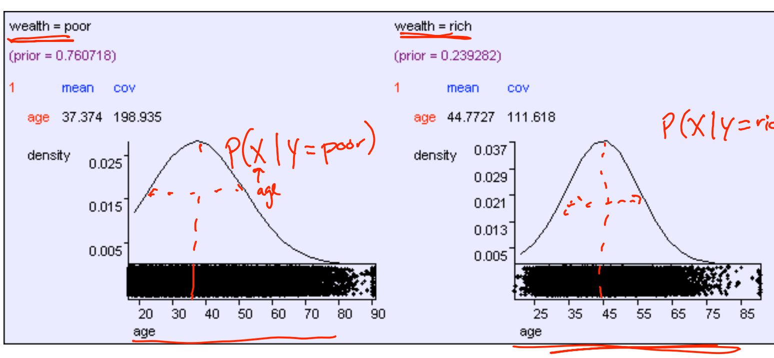
$\mu_{i,2} \leftarrow$ mean value of second dimension

covariance between features 1 and 2

$$\Sigma_i = \begin{bmatrix}\sigma_{i,11} & \sigma_{i,12} & \sigma_{i,13} & \\ \sigma_{i,21} & \sigma_{i,22} & \cdots & \\ & & \ddots & \\ & & & \sigma_{i,dd}\end{bmatrix}$$

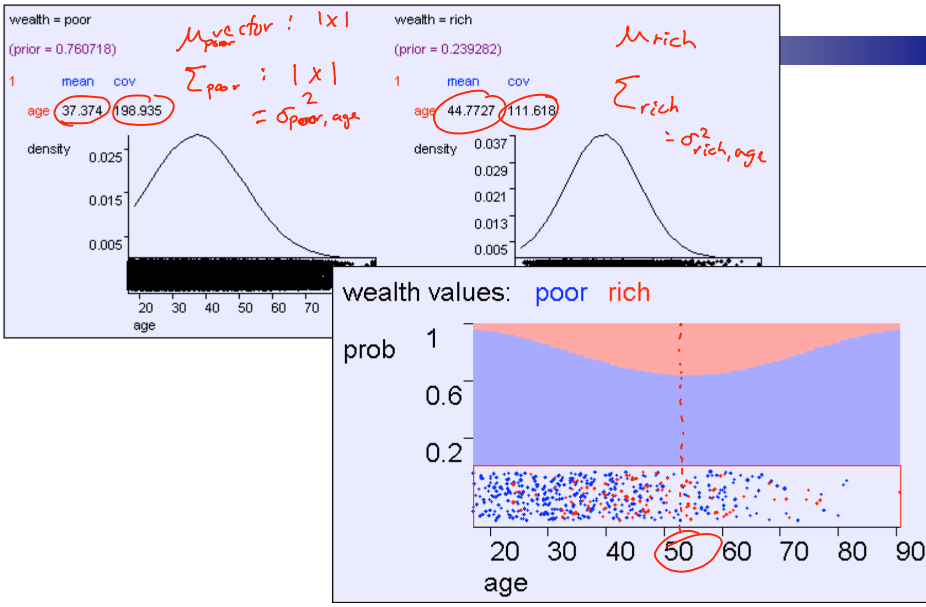
covariance matrix

Predicting wealth from age



P(X|Y=rich)

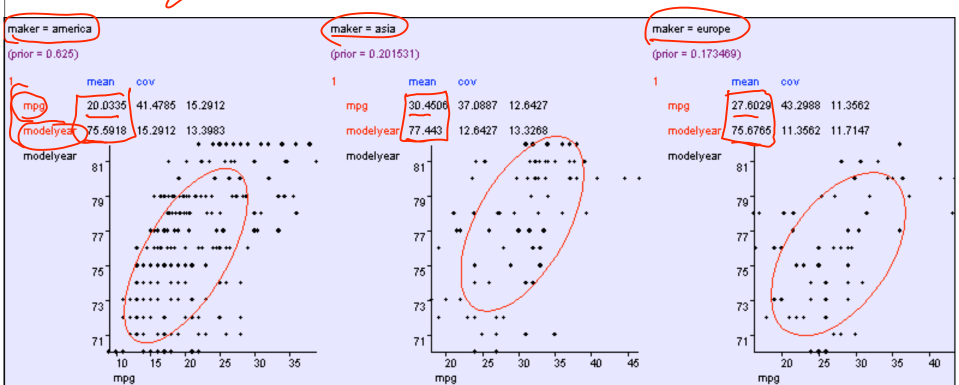
Predicting wealth from age



Learning modelyear , mpg ---> maker

$$\Sigma_i = \begin{pmatrix} \sigma^2_1 & \sigma_{12} & \cdots & \sigma_{1m} \\ \sigma_{12} & \sigma^2_2 & \cdots & \sigma_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{1m} & \sigma_{2m} & \cdots & \sigma^2_m \end{pmatrix}$$

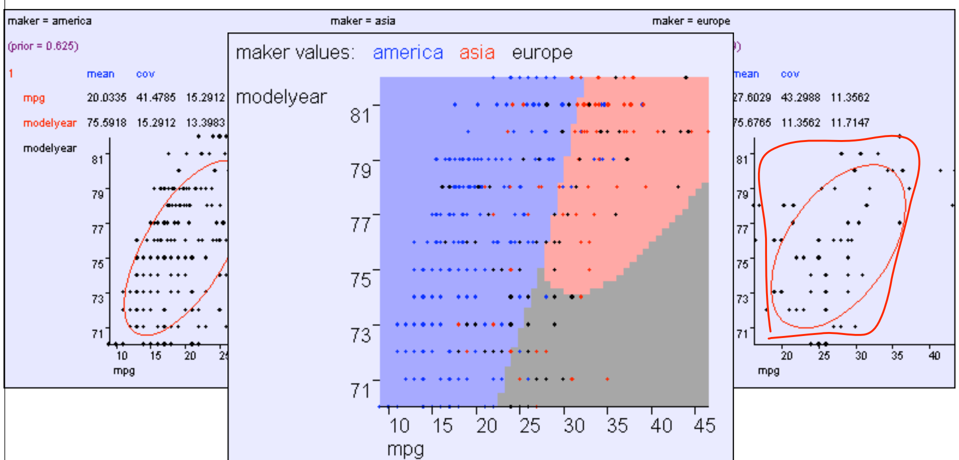
$\Sigma_{america} = 2\times2$



General: $O(m^2)$ parameters

per class

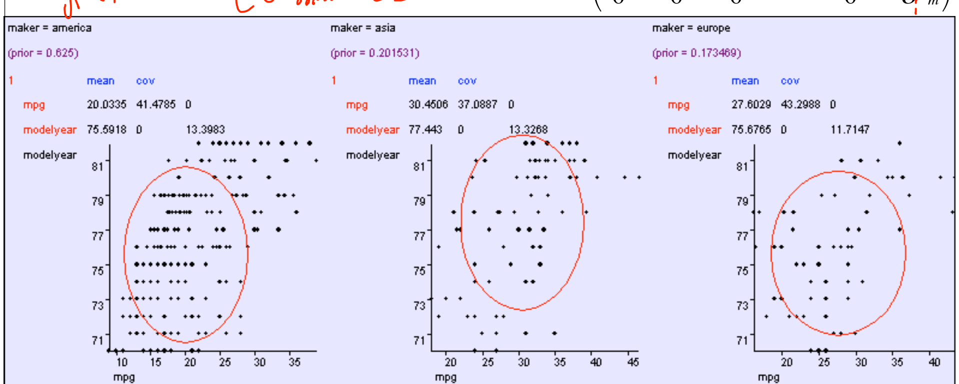
$$\Sigma_i = \begin{pmatrix} \sigma^2{}_1 & \sigma_{12} & \cdots & \sigma_{1m} \\ \sigma_{12} & \sigma^2{}_2 & \cdots & \sigma_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{1m} & \sigma_{2m} & \cdots & \sigma^2{}_m \end{pmatrix}$$



Aligned: $O(m)$ parameters

Features are indep. given class [Gaussian NB]

$$\Sigma_i = \begin{pmatrix} \sigma^2_i{}_1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & \sigma^2_i{}_2 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \sigma^2_i{}_3 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \sigma^2_i{}_{m-1} & 0 \\ 0 & 0 & 0 & \cdots & 0 & \sigma^2_i{}_m \end{pmatrix}$$

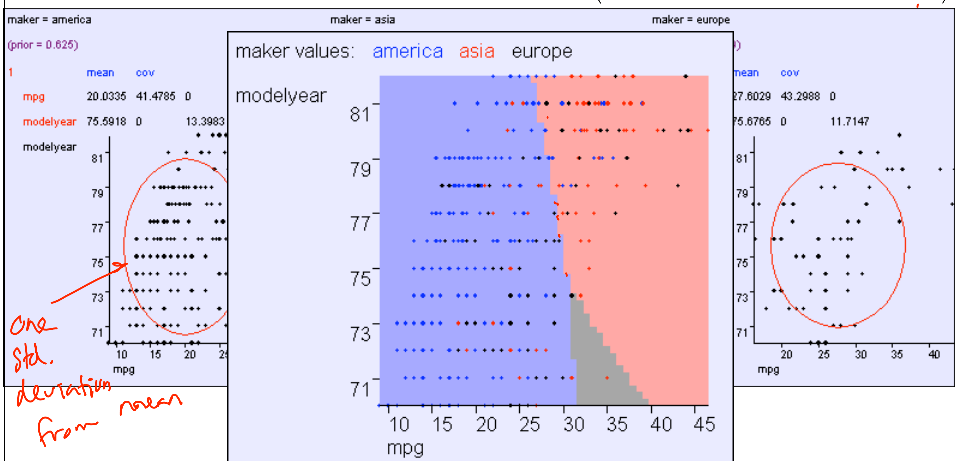


axis aligned Gaussians

Aligned: *O(m)* parameters

per class

$$\Sigma_i = \begin{pmatrix} \sigma_{i1}^2 & 0 & 0 & \cdots & 0 & 0 \\ 0 & \sigma_{i2}^2 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \sigma_{i3}^2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \sigma_{i,m-1}^2 & 0 \\ 0 & 0 & 0 & \cdots & 0 & \sigma_{i,m}^2 \end{pmatrix}$$



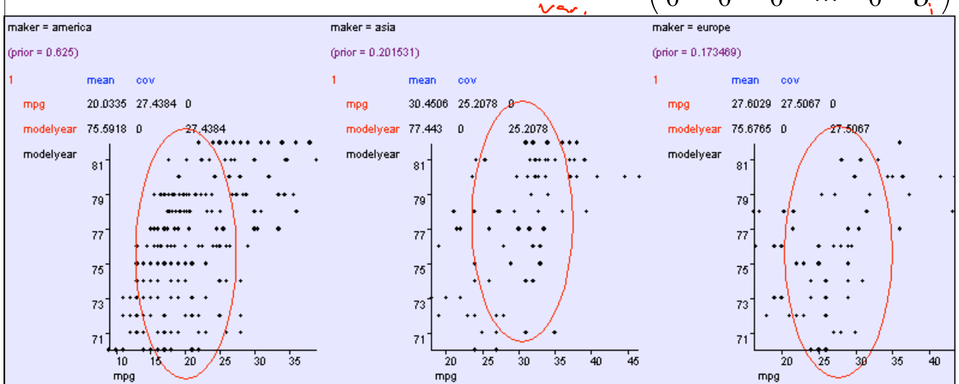
One std. deviation from mean

Spherical: *O(1)* cov parameters

per class

every feature has same var.

$$\Sigma = \begin{pmatrix} \sigma_i^2 & 0 & 0 & \cdots & 0 & 0 \\ 0 & \sigma_i^2 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \sigma_i^2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \sigma_i^2 & 0 \\ 0 & 0 & 0 & \cdots & 0 & \sigma_i^2 \end{pmatrix}$$

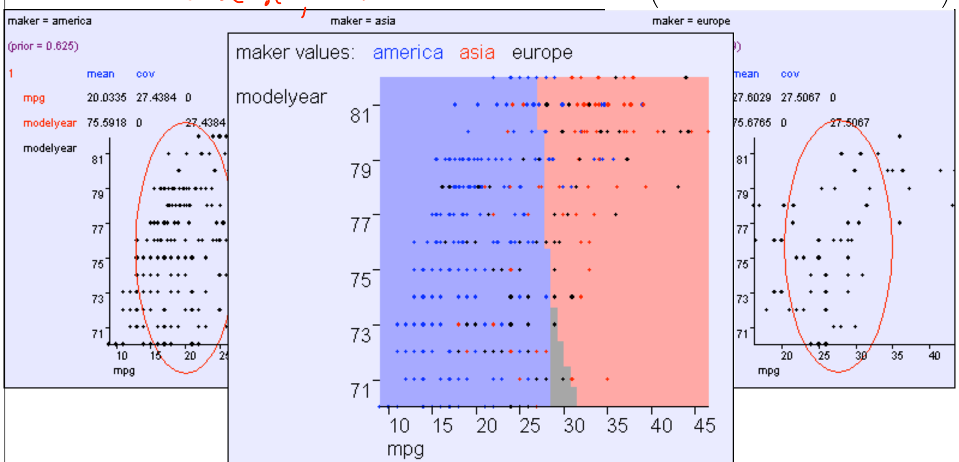


Spherical Gaussians

Spherical: *O(1)* cov parameters

$$\Sigma = \begin{pmatrix} \sigma^2 & 0 & 0 & \cdots & 0 & 0 \\ 0 & \sigma^2 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \sigma^2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \sigma^2 & 0 \\ 0 & 0 & 0 & \cdots & 0 & \sigma^2 \end{pmatrix}$$

Carlos does not endorse, however....



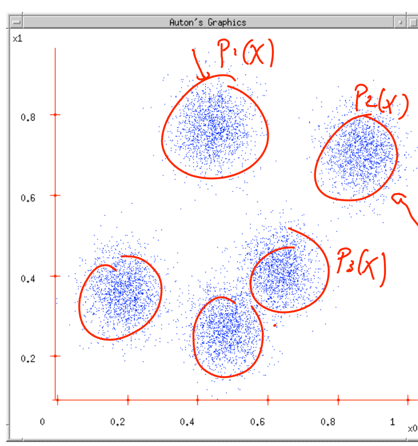
Next… back to Density Estimation

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

$P(x) = \sum_i w_i P_i(x)$

Gaussian mixture model

P_i is some gaussian with mean μ_i covariance Σ_i



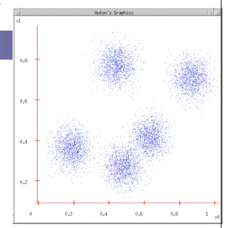
if data were labeled

x_i, y_i

⇓

for each i would know which bump

But we don't see class labels!!!



- MLE:
  - argmax $\prod_j P(y_j,x_j)$

don't observe y

- But we don't know y_j's!!! don't know bump
- Maximize marginal likelihood:
  - argmax $\prod_j P(x_j)$ = argmax $\prod_j \sum_{i=1}^k P(y_j=i,x_j)$

part I observe

Special case: spherical Gaussians and hard assignments

$\sigma_{i,1}^2=\sigma_{i,2}^2$ $\quad \Sigma_i = \begin{bmatrix} \sigma_{i,1} & & 0 \\ & \sigma_{i,2} & \\ 0 & & \ddots \; \sigma_{i,d} \end{bmatrix}$

$\Sigma_i = \Sigma_j$

$$P(y=i\mid \mathbf{x}_j) \propto \frac{1}{(2\pi)^{m/2}\,\|\Sigma_i\|^{1/2}}\exp\left[-\frac{1}{2}\left(\mathbf{x}_j-\mu_i\right)^T\Sigma_i^{-1}\left(\mathbf{x}_j-\mu_i\right)\right]P(y=i)$$

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

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

- If each x_j belongs to one class C(j) (hard assignment), marginal likelihood:

$p(x_j, Y=i) = 0 \;\; \forall i \neq C(j)$

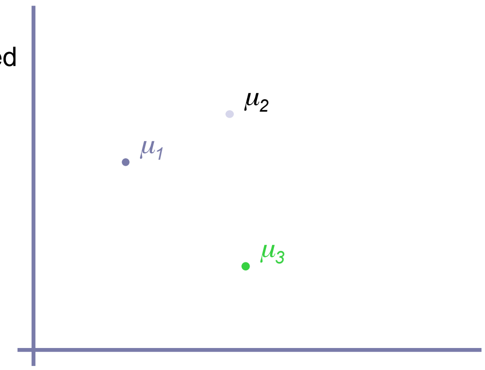
$$\log \prod_{j=1}^{m}\sum_{i=1}^{k}P(\mathbf{x}_j,y=i) \propto \log \prod_{j=1}^{m}\exp\left[-\frac{1}{2\sigma^2}\left\|\mathbf{x}_j-\mu_{C(j)}\right\|^2\right]$$

- Same as K-means!!!

$$= \sum_{j=1}^{m}\left[-\frac{1}{2\sigma^2}\|x_j-\mu_{C(j)}\|^2\right] = -\frac{1}{2\sigma^2}\underbrace{\sum_{j=1}^{m}\|x_j-\mu_{C(j)}\|^2}_{\text{K-means objective}}$$

The GMM assumption

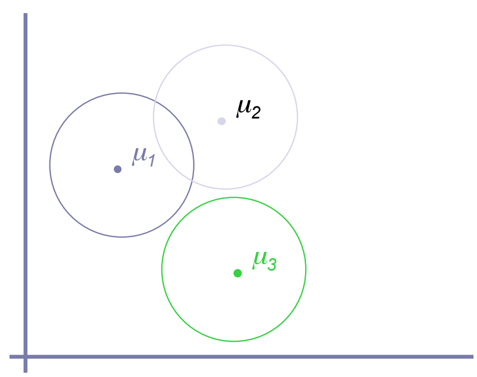
- There are k components
- Component *i* has an associated mean vector μ_i



The GMM assumption

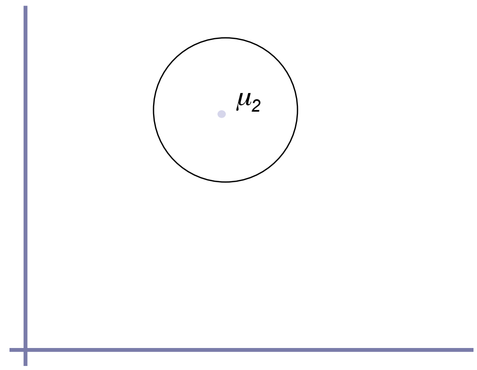
- There are k components
- Component *i* has an associated mean vector μ_i
- Each component generates data from a Gaussian with mean μ_i and covariance matrix $\sigma^2 \mathbf{I}$

Each data point is generated according to the following recipe:



The GMM assumption

- There are k components
- Component *i* has an associated mean vector μ_i
- Each component generates data from a Gaussian with mean μ_i and covariance matrix $\sigma^2 \boldsymbol{I}$



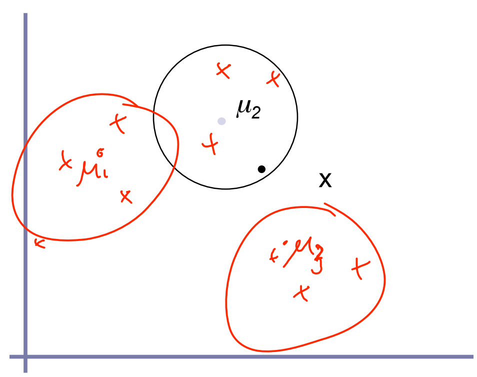
Each data point is generated according to the following recipe:

1. Pick a component at random: Choose component i with probability *P(y=i)*

The GMM assumption

with K-means model

- There are k components
- Component *i* has an associated mean vector μ_i
- Each component generates data from a Gaussian with mean μ_i and covariance matrix $\sigma^2 \boldsymbol{I}$



Each data point is generated according to the following recipe:

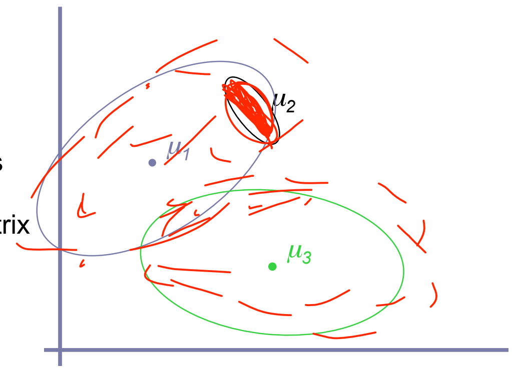
1. Pick a component at random: Choose component i with probability *P(y=i)*
2. Datapoint ~ $N(\mu_i, \sigma^2 \boldsymbol{I})$

The General GMM assumption

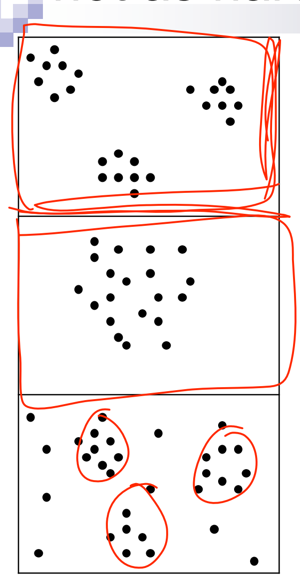
- There are k components
- Component *i* has an associated mean vector μ_i
- Each component generates data from a Gaussian with mean μ_i and covariance matrix Σ_i

Each data point is generated according to the following recipe:

1. Pick a component at random: Choose component i with probability *P(y=i)*
2. Datapoint ~ N(μ_i, Σ_i)



Unsupervised Learning: not as hard as it looks



Sometimes easy

Sometimes impossible

and sometimes in between

*IN CASE YOU'RE WONDERING WHAT THESE DIAGRAMS ARE, THEY SHOW 2-d UNLABELED DATA (**X** VECTORS) DISTRIBUTED IN 2-d SPACE. THE TOP ONE HAS THREE VERY CLEAR GAUSSIAN CENTERS*

Marginal likelihood for general case

$$P(y=i \mid \mathbf{x}_j) \propto \frac{1}{(2\pi)^{m/2} \|\Sigma_i\|^{1/2}} \exp\left[-\frac{1}{2}\left(\mathbf{x}_j - \mu_i\right)^T \Sigma_i^{-1}\left(\mathbf{x}_j - \mu_i\right)\right] P(y=i)$$

- Marginal likelihood:

sum over possible clusters

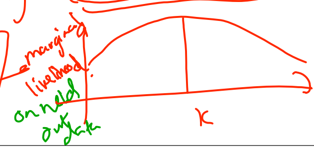
$$\prod_{j=1}^{m} P(\mathbf{x}_j) = \prod_{j=1}^{m}\sum_{i=1}^{k} P(\mathbf{x}_j, y=i)$$

$$= \prod_{j=1}^{m}\sum_{i=1}^{k} \underbrace{\frac{1}{(2\pi)^{m/2} \|\Sigma_i\|^{1/2}} \exp\left[-\frac{1}{2}\left(\mathbf{x}_j - \mu_i\right)^T \Sigma_i^{-1}\left(\mathbf{x}_j - \mu_i\right)\right]}_{\text{likelihood}} \underbrace{P(y=i)}_{\text{prior}}$$

pick k using cross validation (gencly will pick more clusters than you "want")

doesn't work for K-means... !

"other criteria"



Special case 2: spherical Gaussians and soft assignments

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

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

likelihood function

- Uncertain about class of each x_j (soft assignment), marginal likelihood:

$$\prod_{j=1}^{m}\sum_{i=1}^{k} P(\mathbf{x}_j, y=i) \propto \prod_{j=1}^{m}\sum_{i=1}^{k} \exp\left[-\frac{1}{2\sigma^2}\left\|\mathbf{x}_j - \mu_i\right\|^2\right] P(y=i)$$

each point belongs to more than one cluster : prob. $P(Y_j = i)$

Unsupervised Learning: Mediumly Good News

We now have a procedure s.t. if you give me a guess at **μ**$_1$, **μ**$_2$ .. **μ**$_k$,

I can tell you the prob of the unlabeled data given those **μ**'s.

$P(Y_j=i \mid X_j) \leftarrow$ Bayes rule

Suppose ***x***'s are 1-dimensional.

There are two classes; w_1 and w_2

$P(y_1) = 1/3$ $P(y_2) = 2/3$ $\sigma = 1$. ← Know these

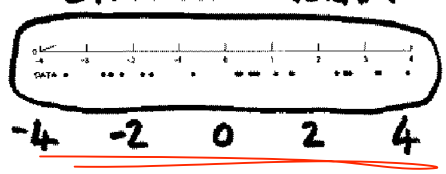
compute $\prod_j P(x_j)$

There are 25 unlabeled datapoints

x_1 = 0.608
x_2 = -1.590
x_3 = 0.235
x_4 = 3.949
:
x_{25} = -0.712

(From Duda and Hart)

DATA SCATTERGRAM

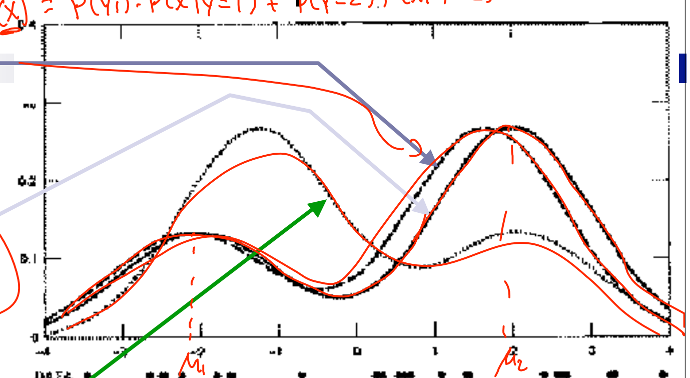


Duda & Hart's Example

$P(X) = P(Y_1).P(X \mid Y=1) + P(Y=2).P(X \mid Y=2)$

We can graph the prob. dist. function of data given our μ_1 and μ_2 estimates.

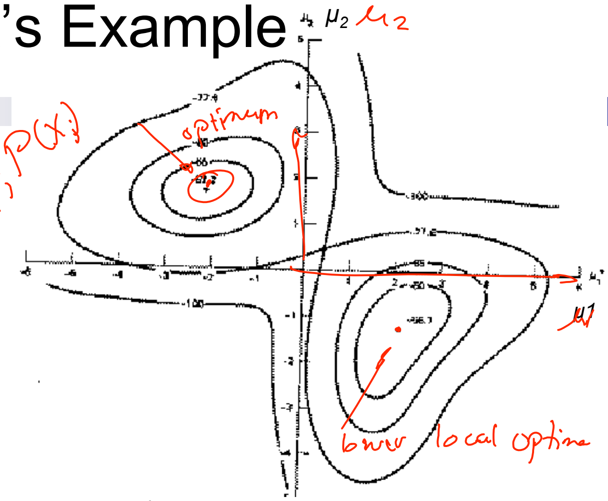
We can also graph the true function from which the data was randomly generated.



- They are close. Good.
- The 2nd solution tries to put the "2/3" hump where the "1/3" hump should go, and vice versa. local optima
- In this example unsupervised is almost as good as supervised. If the x_1 .. x_{25} are given the class which was used to learn them, then the results are (μ_1=-2.176, μ_2=1.684). Unsupervised got (μ_1=-2.13, μ_2=1.668).

Duda & Hart's Example

max marginal $\prod_i P(x_i)$
μ_1, μ_2 (likelihood)



Graph of
log P(x_1, x_2 .. x_{25} | μ_1, μ_2)
against μ_1 (→) and μ_2 (↑)

Max likelihood = (μ_1 =-2.13, μ_2 =1.668)

Local minimum, but very close to global at (μ_1 =2.085, μ_2 =-1.257)*

* corresponds to switching y_1 with y_2.

Finding the max likelihood $\mu_1, \mu_2 .. \mu_k$

We can compute P(data | $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2 .. \boldsymbol{\mu}_k$)
How do we find the $\boldsymbol{\mu}_i$'s which give max. likelihood?

- The normal max likelihood trick:
  Set $\frac{\partial}{\partial \mu_i}$ log Prob (….) = 0
  and solve for μ_i's.
  \# Here you get non-linear non-analytically-solvable equations
- Use gradient descent
  Often slow but doable
- Use a much faster, cuter, and recently very popular method…

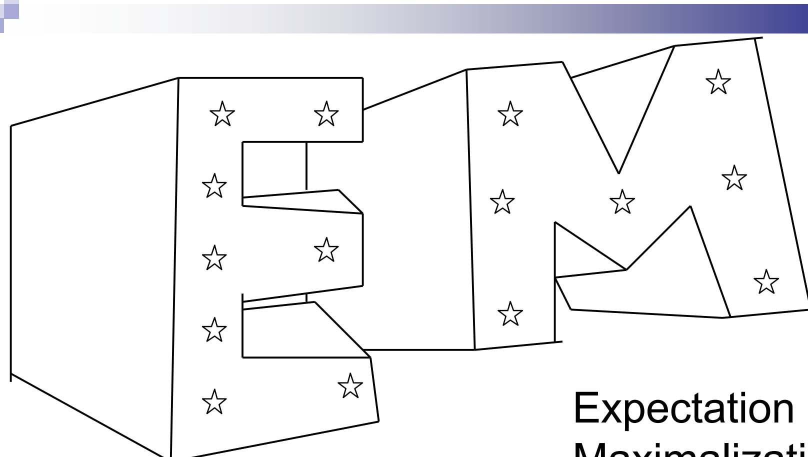
not convex
(previous slide)

Announcements

- HW5 out later today…
  - Due December 5th by 3pm to Monica Hopes, Wean 4619
- Project:
  - Poster session: NSH Atrium, Friday 11/30, 2-5pm 4:45pm
    - Print your poster early!!!
      - SCS facilities has a poster printer, ask helpdesk
      - Students from outside SCS should check with their departments
      - It's OK to print separate pages
    - We'll provide pins, posterboard and an easel
      - Poster size: 32x40 inches
    - Invite your friends, there will be a prize for best poster, by popular vote

arrive 15 mins early to set up!

- Last lecture:
  - Thursday, 11/29, 5-6:20pm, Wean 7500



Expectation Maximalization

The E.M. Algorithm

DETOUR

- We'll get back to unsupervised learning soon
- But now we'll look at an even simpler case with hidden information
- The EM algorithm
  - Can do trivial things, such as the contents of the next few slides
  - An excellent way of doing our unsupervised learning problem, as we'll see
  - Many, many other uses, including learning BNs with hidden data

Silly Example

Let events be "grades in a class"

w_1 = Gets an A	$P(A) = ½$
w_2 = Gets a B	$P(B) = \mu$
w_3 = Gets a C	$P(C) = 2\mu$
w_4 = Gets a D	$P(D) = ½-3\mu$

(Note $0 \le \mu \le 1/6$)

Assume we want to estimate μ from data. In a given class there were

a A's
b B's
c C's
d D's

What's the maximum likelihood estimate of μ given a,b,c,d ?

$\hat{\mu}_{MLE}$

Trivial Statistics

P(A) = ½ P(B) = μ P(C) = 2μ P(D) = ½-3μ

P(*a,b,c,d* | μ) = K(½)^{a}(μ)^{b}(2μ)^{c}(½-3μ)^{d}

const

log P(*a,b,c,d* | μ) = log K + *a*log ½ + *b*log μ + *c*log 2μ + *d*log (½-3μ)

$$\text{FOR MAX LIKE } \mu, \text{ SET } \frac{\partial \text{LogP}}{\partial \mu} = 0$$

$$\frac{\partial \text{LogP}}{\partial \mu} = \frac{b}{\mu} + \frac{2c}{2\mu} - \frac{3d}{1/2 - 3\mu} = 0$$

$$\text{Gives max like } \mu = \frac{b+c}{6(b+c+d)}$$

So if class got

A	B	C	D
14	6	9	10

$$\text{Max like } \mu = \frac{1}{10}$$

Boring, but true!

Same Problem with Hidden Information

REMEMBER
P(A) = ½
P(B) = μ
P(C) = 2μ
P(D) = ½-3μ

Someone tells us that

Number of High grades (A's + B's) = *h*

Number of C's = *c*

Number of D's = *d*

What is the max. like estimate of μ now?

A ∈ $\bar{a}$ people

B ∈ h − $\bar{a}$ people

what's ?

if I knew $\bar{a}$

$$\mu = \frac{h - \bar{a} + c}{6(h - \bar{a} + c + d)}$$

What's $\bar{a}$?

Same Problem with Hidden Information

Someone tells us that

Number of High grades (A's + B's) = *h*

Number of C's = *c*

Number of D's = *d*

What is the max. like estimate of μ now?

We can answer this question circularly:

REMEMBER
P(A) = ½
P(B) = μ
P(C) = 2μ
P(D) = ½-3μ

EXPECTATION

If we know the value of μ we could compute the expected value of *a* and *b*

iterate

Since the ratio a:b should be the same as the ratio ½ : μ

$$a = \frac{\frac{1}{2}}{\frac{1}{2}+\mu}h \qquad b = \frac{\mu}{\frac{1}{2}+\mu}h$$

MAXIMIZATION

If we know the expected values of *a* and *b* we could compute the maximum likelihood value of μ

$$\mu = \frac{b+c}{6(b+c+d)}$$

E.M. for our Trivial Problem

REMEMBER
P(A) = ½
P(B) = μ
P(C) = 2μ
P(D) = ½-3μ

We begin with a guess for μ

We iterate between EXPECTATION and MAXIMALIZATION to improve our estimates of μ and *a* and *b*.

Define $\mu^{(t)}$ the estimate of μ on the t'th iteration
$b^{(t)}$ the estimate of *b* on t'th iteration

if trueparam was $\mu(t)$

$$\mu^{(0)} = \text{initial guess}$$

$$b^{(t)} = \frac{\mu^{(t)}h}{\frac{1}{2}+\mu^{(t)}} = \mathrm{E}\left[b \mid \mu^{(t)}\right]$$

E-step

$$\mu^{(t+1)} = \frac{b^{(t)}+c}{6\left(b^{(t)}+c+d\right)}$$

$$= \text{max like est. of } \mu \text{ given } b = b^{(t)}$$

M-step

Continue iterating until converged.
Good news: Converging to local optimum is assured.
Bad news: I said "local" optimum.

E.M. Convergence

- Convergence proof based on fact that Prob(data | μ) must increase or remain same between each iteration [NOT OBVIOUS]
- But it can never exceed 1 [OBVIOUS]

So it must therefore converge [OBVIOUS]

In our example, suppose we had

$h = 20$

$c = 10$

$d = 10$

$\mu^{(0)} = 0$

t	$\mu^{(t)}$	$b^{(t)}$
0	0	0
1	0.0833	2.857
2	0.0937	3.158
3	0.0947	3.185
4	0.0948	3.187
5	0.0948	3.187
6	0.0948	3.187

$a^{(t)}$

20

Convergence is generally linear: error decreases by a constant factor each time step.