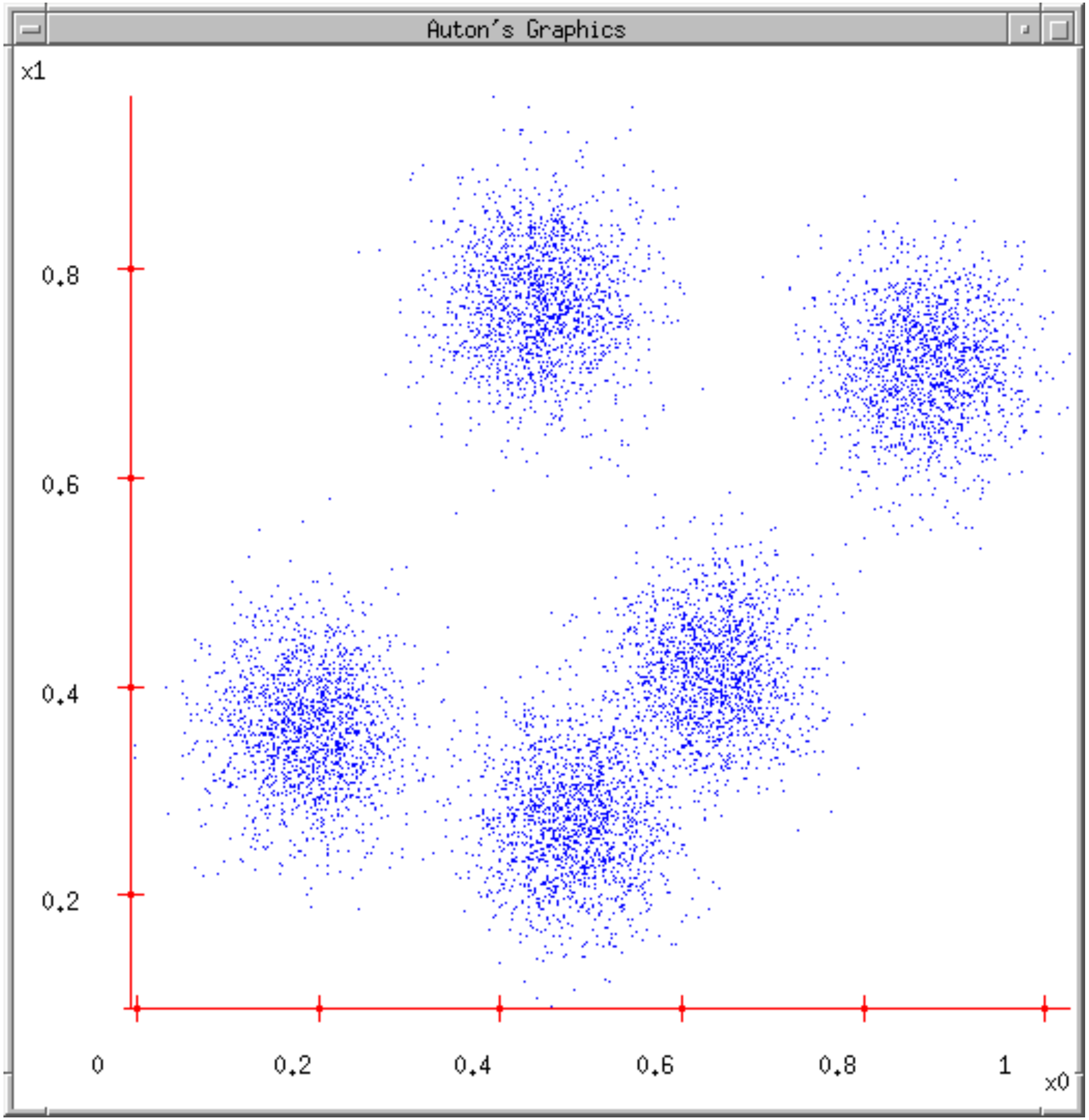
# CSE 575: Statistical Machine Learning

Jingrui He

CIDSE, ASU

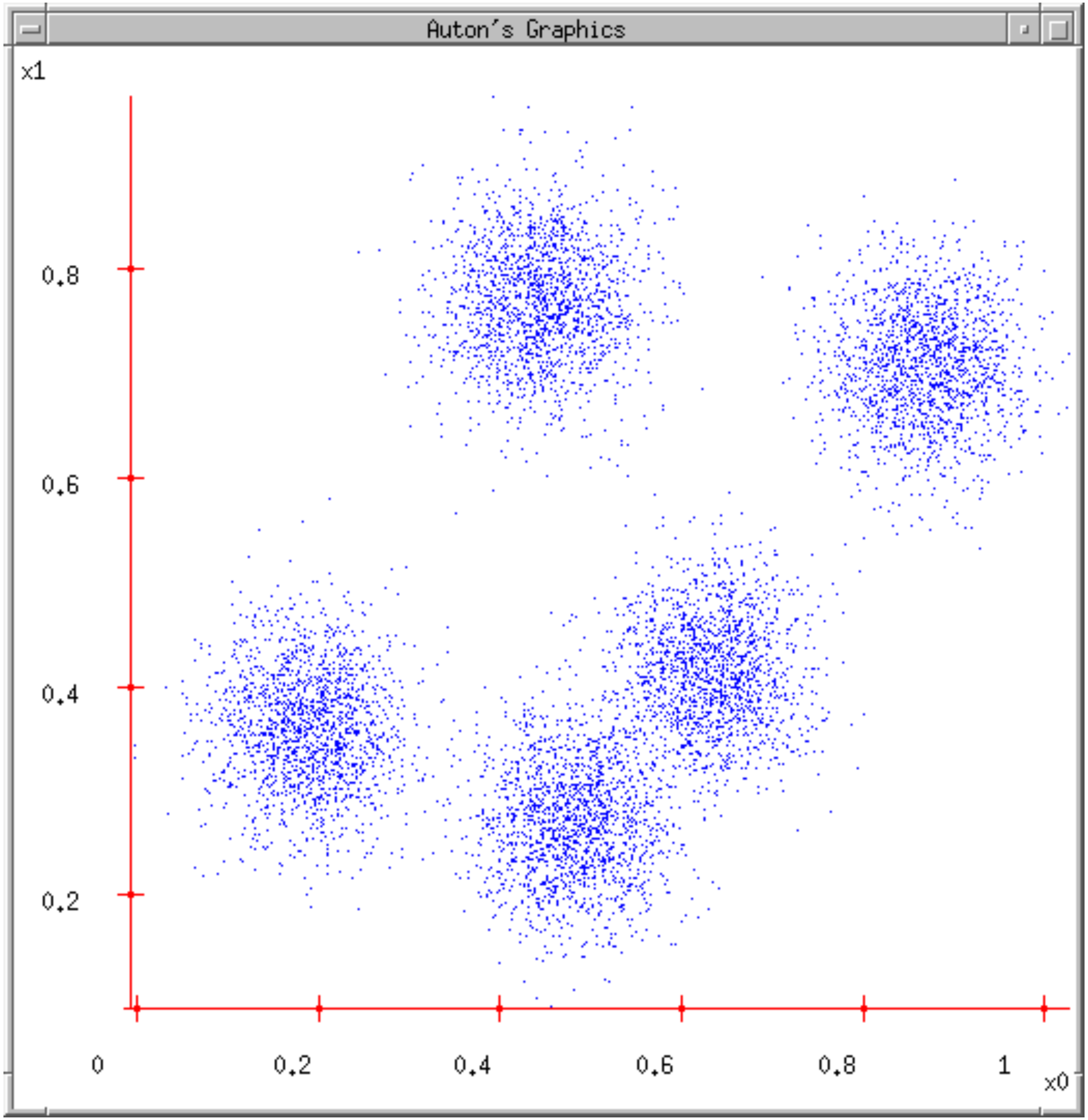
# K-means, GMM, EM

# Some Data



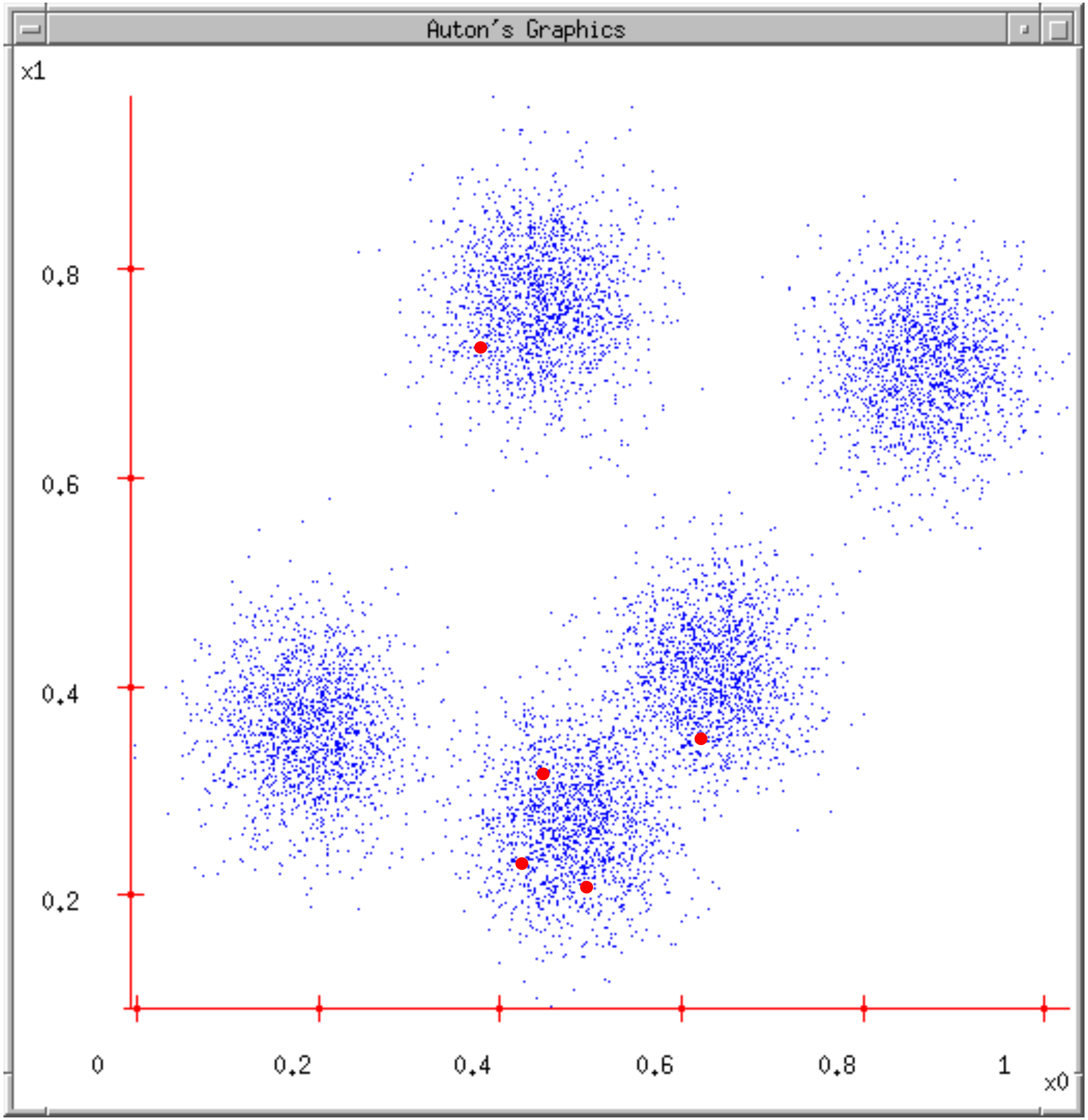
# K-means

1. Ask user how many clusters they'd like. *(e.g. k=5)*



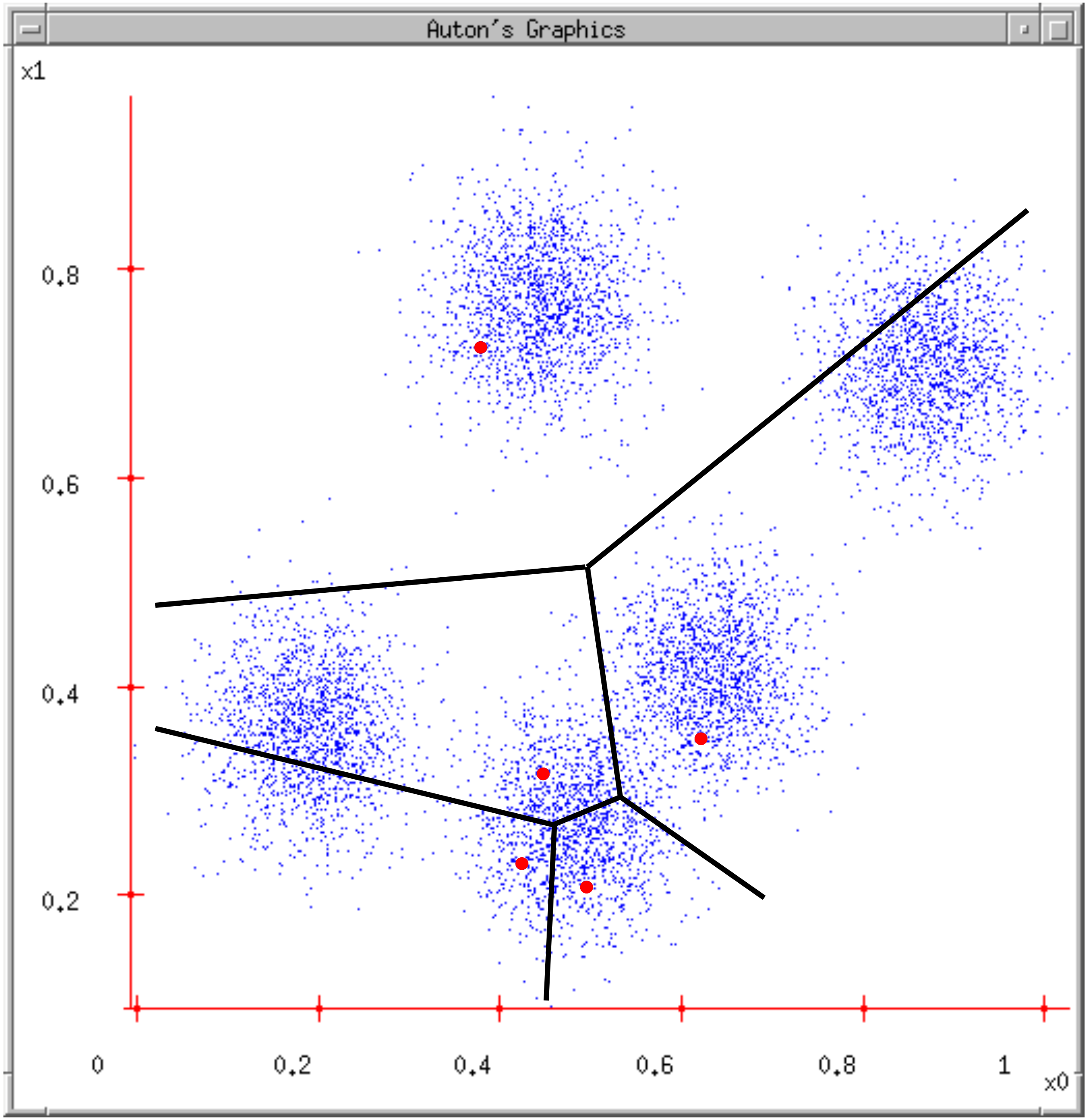
# K-means

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations



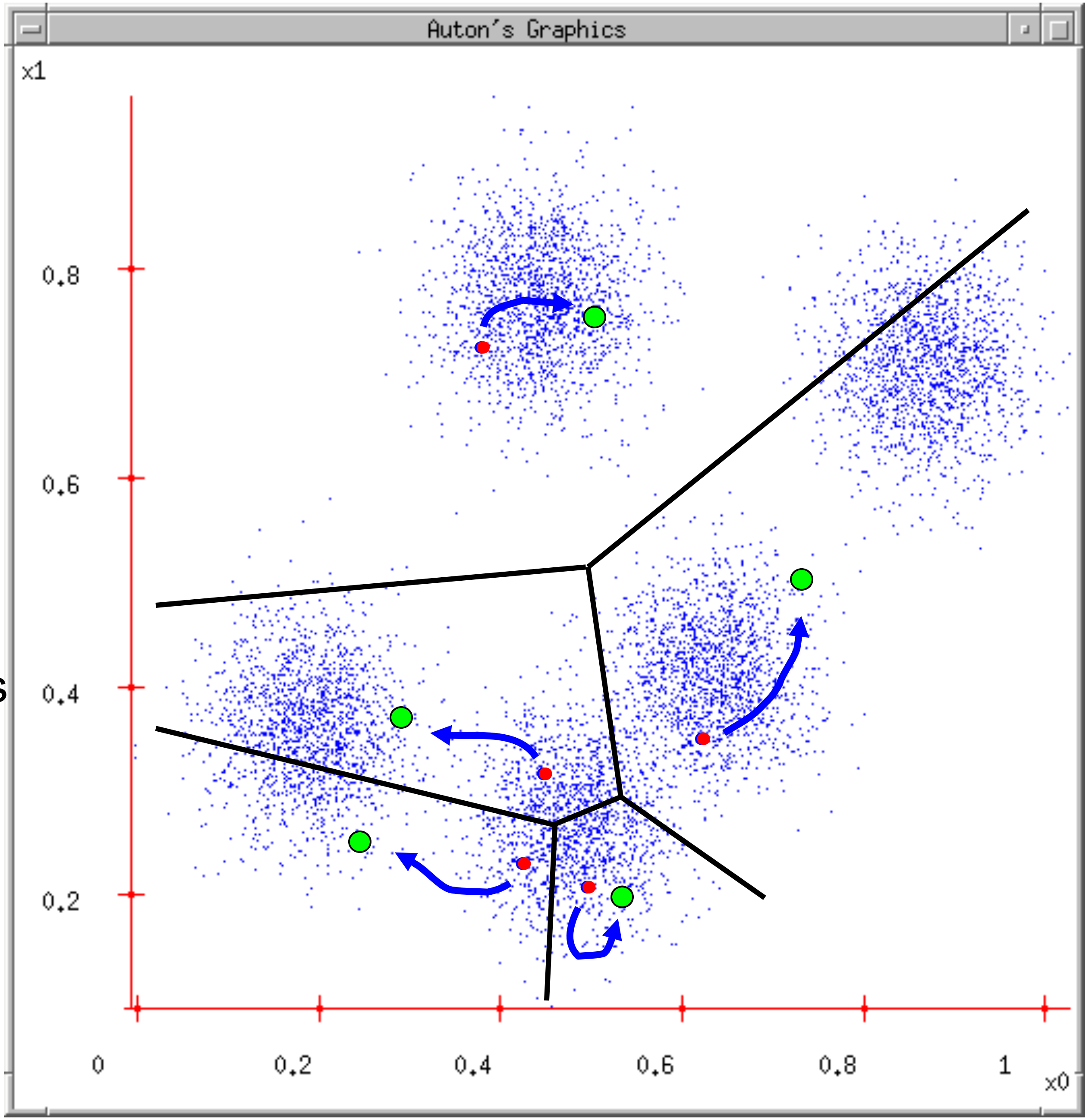
# K-means

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each data point finds out which Center it's closest to (Thus each Center "owns" a set of data points)



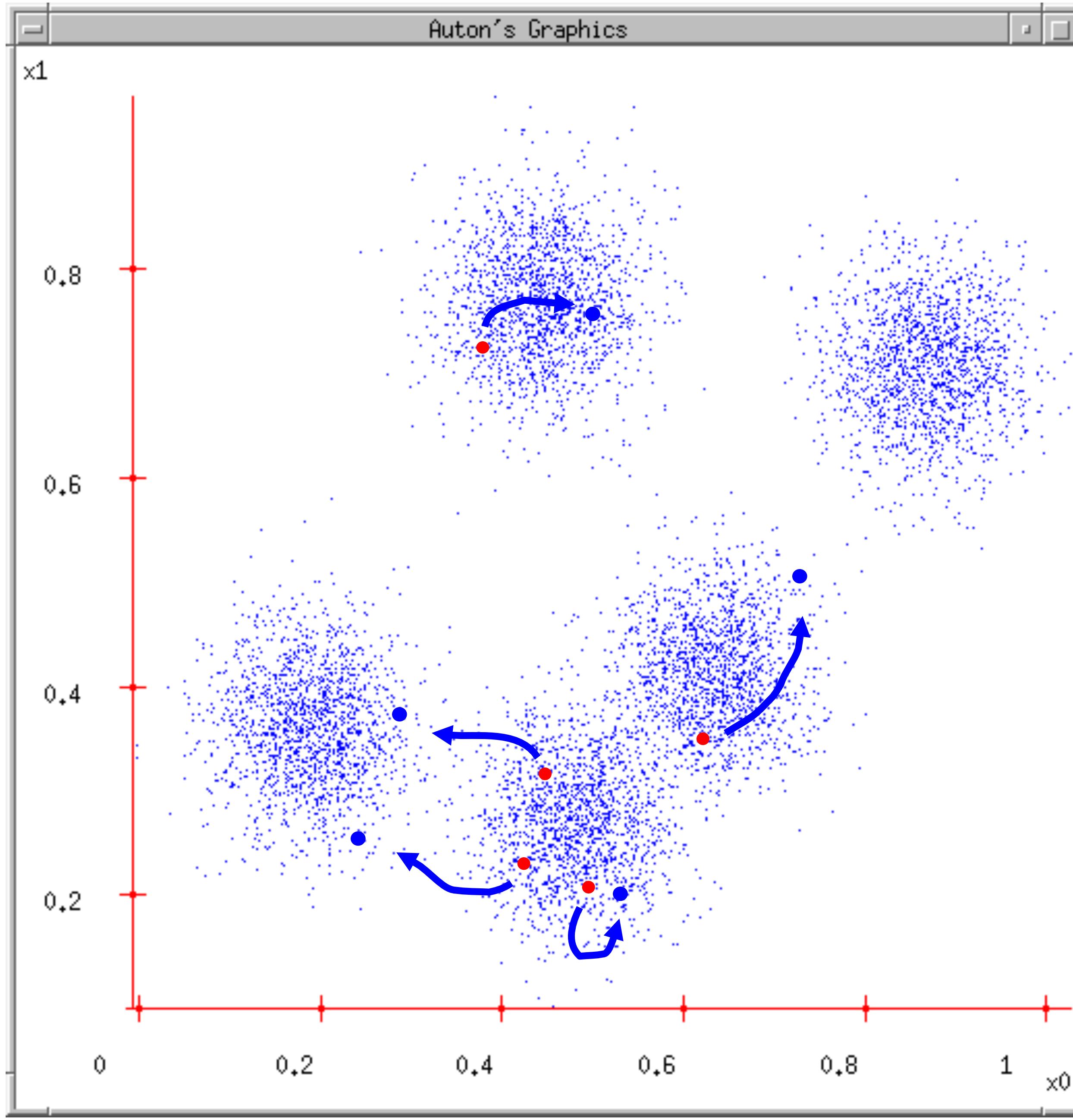
# K-means

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



# K-means

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each data point 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:
  - $C^{(t)}(j) \leftarrow \arg\min_i ||\mu_i - x_j||^2$

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

# What is K-means optimizing?

- Potential function $F(\boldsymbol{\mu}, C)$ of centers $\boldsymbol{\mu}$ and point allocations C:
  - $F(\mu, C) = \sum_{j=1}^{m} ||\mu_{C(j)} - x_j||^2$
- Optimal K-means:
  - $\min_{\boldsymbol{\mu}} \min_{C} F(\boldsymbol{\mu}, C)$

# Does K-means converge??? Part 1

- Optimize potential function:

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

- Fix $\mu$, optimize C

# Does K-means converge??? Part 2

- Optimize potential function:

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

- Fix C, optimize $\mu$

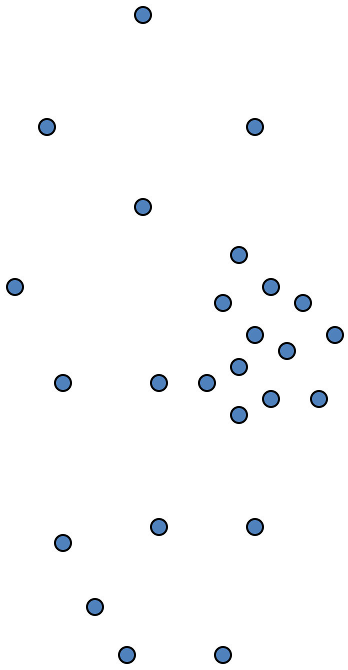
# Coordinate descent algorithms

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

- 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

- K-means is a coordinate descent algorithm!

# (One) bad case for k-means



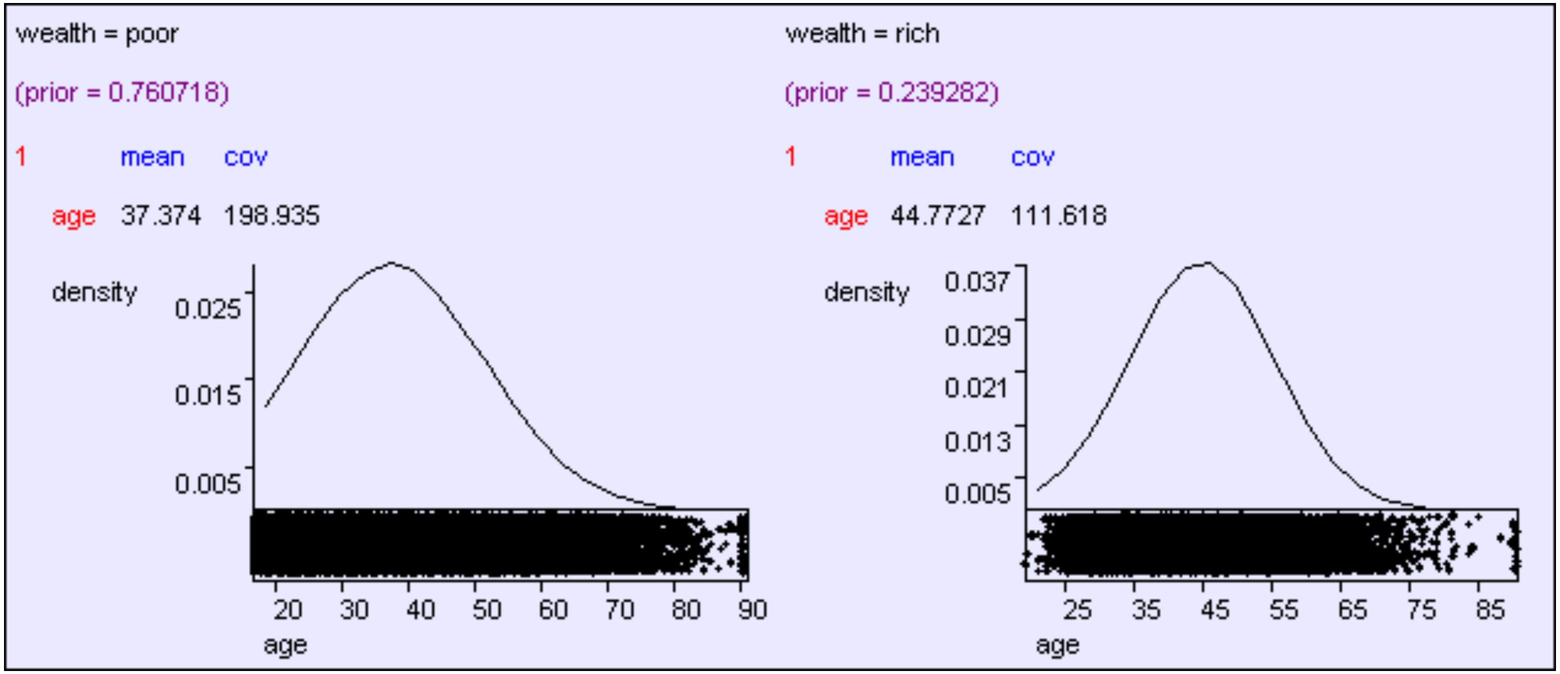
- Clusters may overlap
- Some clusters may be "wider" than others

# Gaussian Bayes classifier reminder

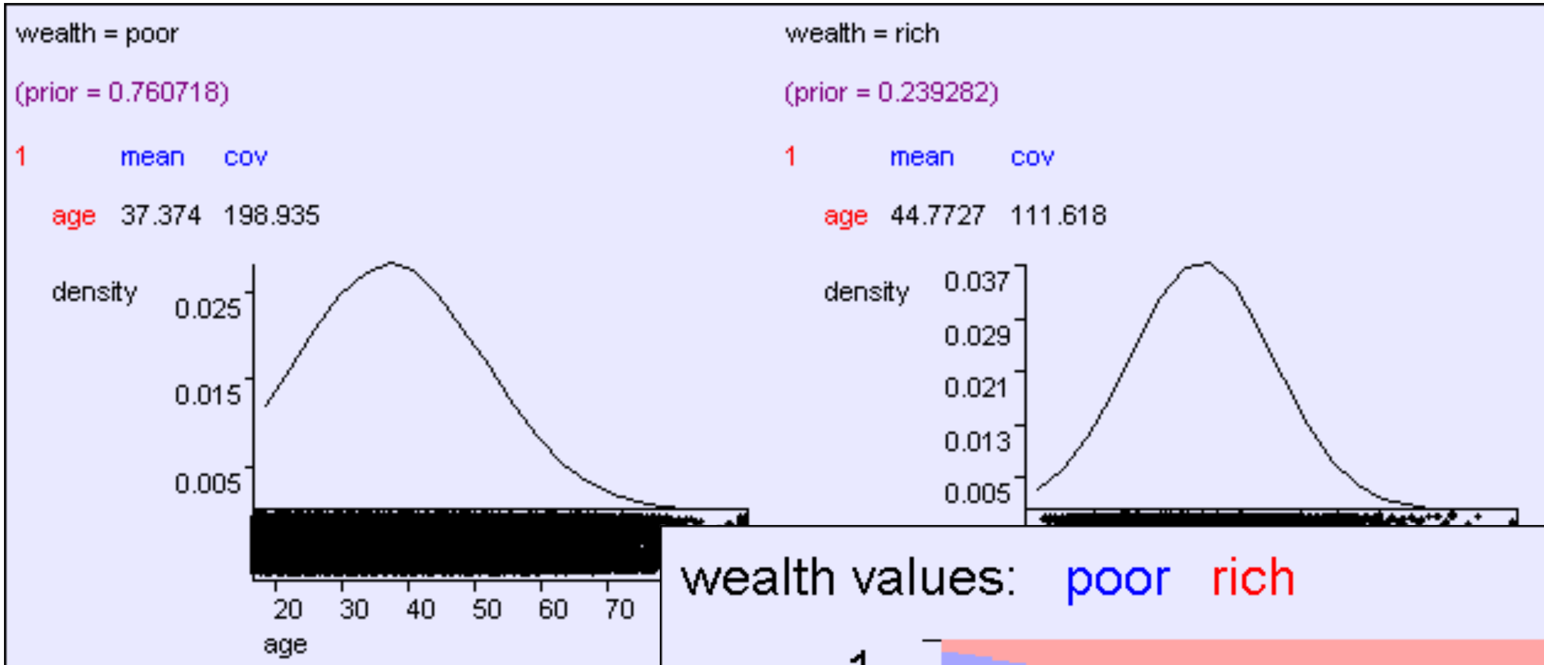
$$P(y=i \mid \mathbf{x}_j) = \frac{p(\mathbf{x}_j \mid y=i)P(y=i)}{p(\mathbf{x}_j)}$$

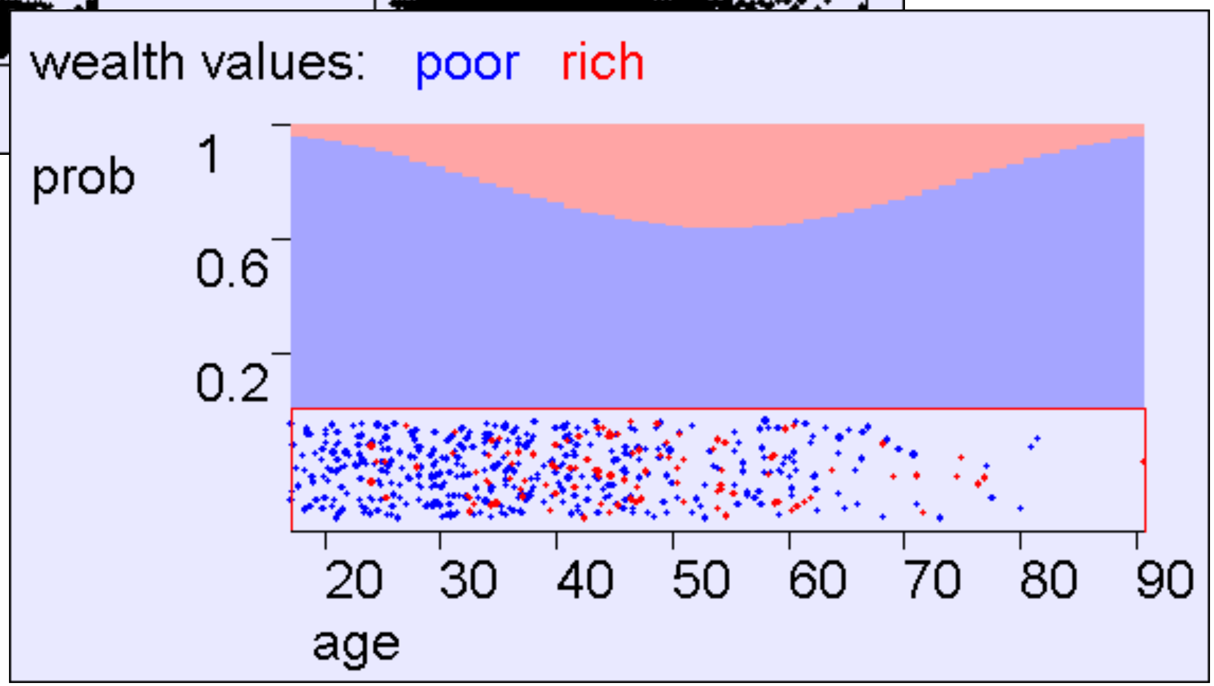
$$P(y=i \mid \mathbf{x}_j) \propto \frac{1}{(2\pi)^{m/2} \| \Sigma_i \|^{1/2}} \exp\left[-\frac{1}{2}\left(\mathbf{x}_j - \mu_i\right)^T \Sigma_i^{-1}\left(\mathbf{x}_j - \mu_i\right)\right] P(y=i)$$

# Predicting wealth from age



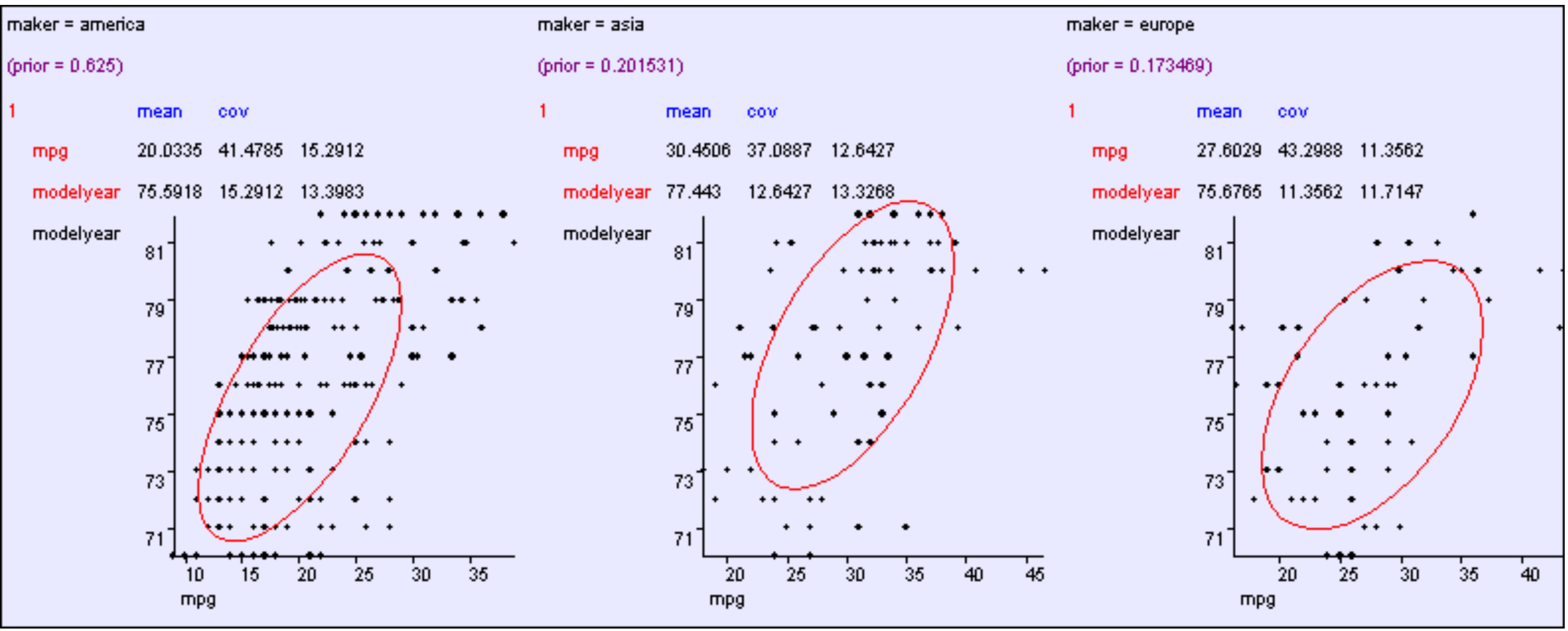
# Predicting wealth from age



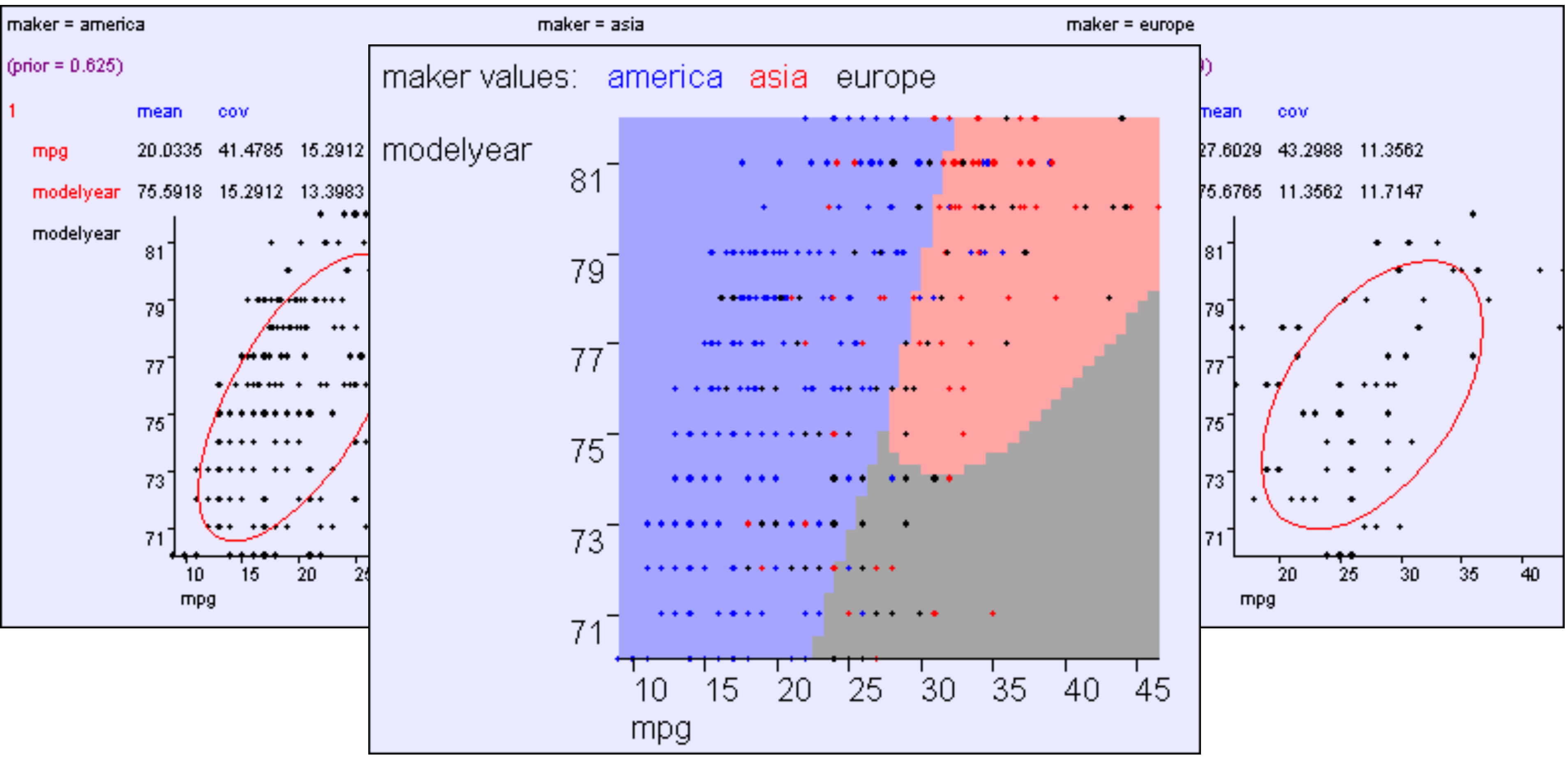
# Learning modelyear, mpg ---> maker

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



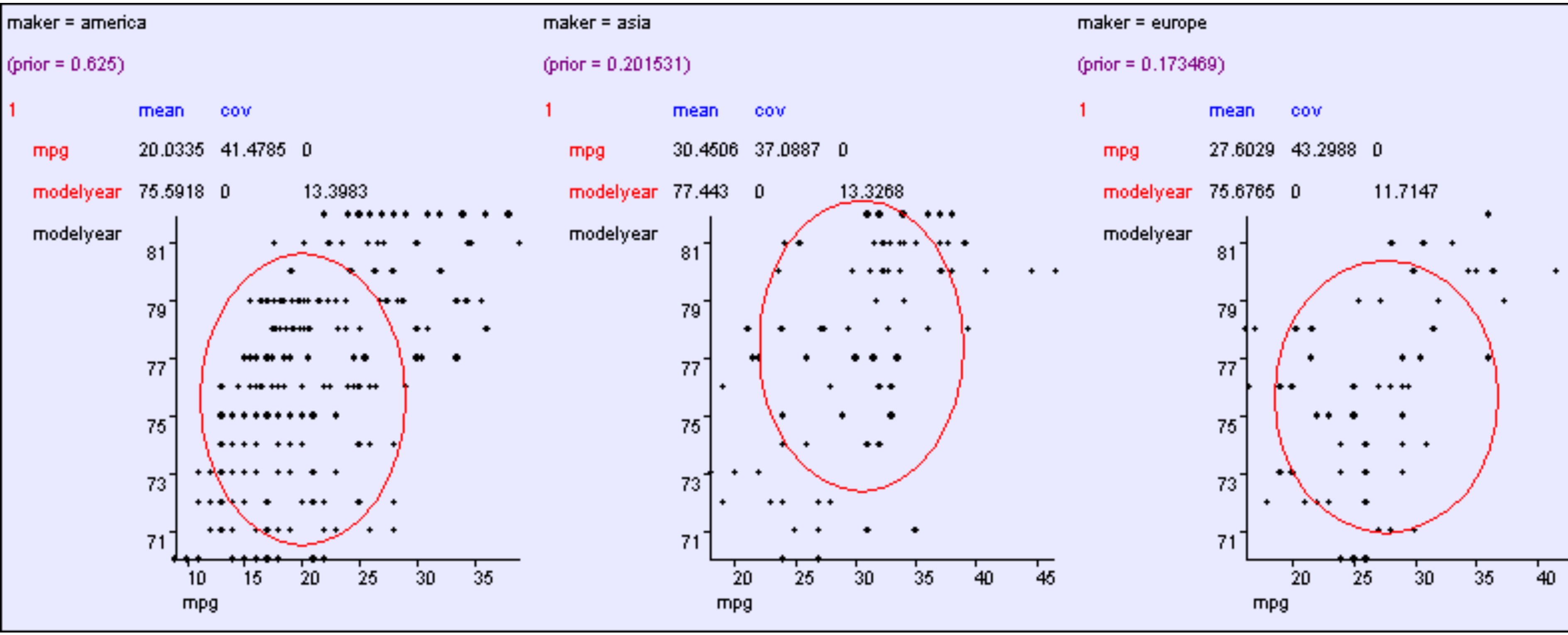
# General: $O(m^2)$ parameters

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



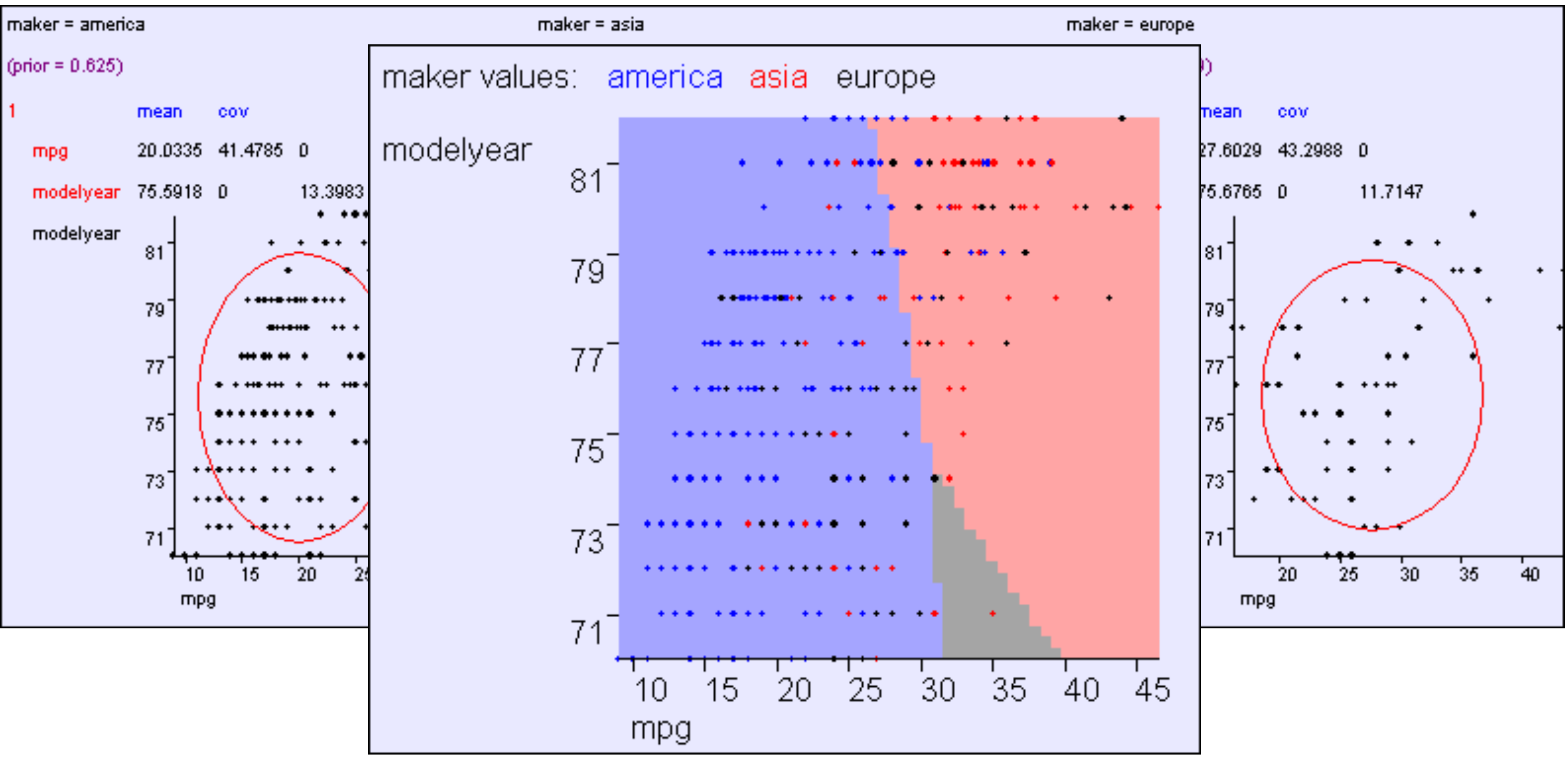
# Aligned: *O(m)* parameters

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



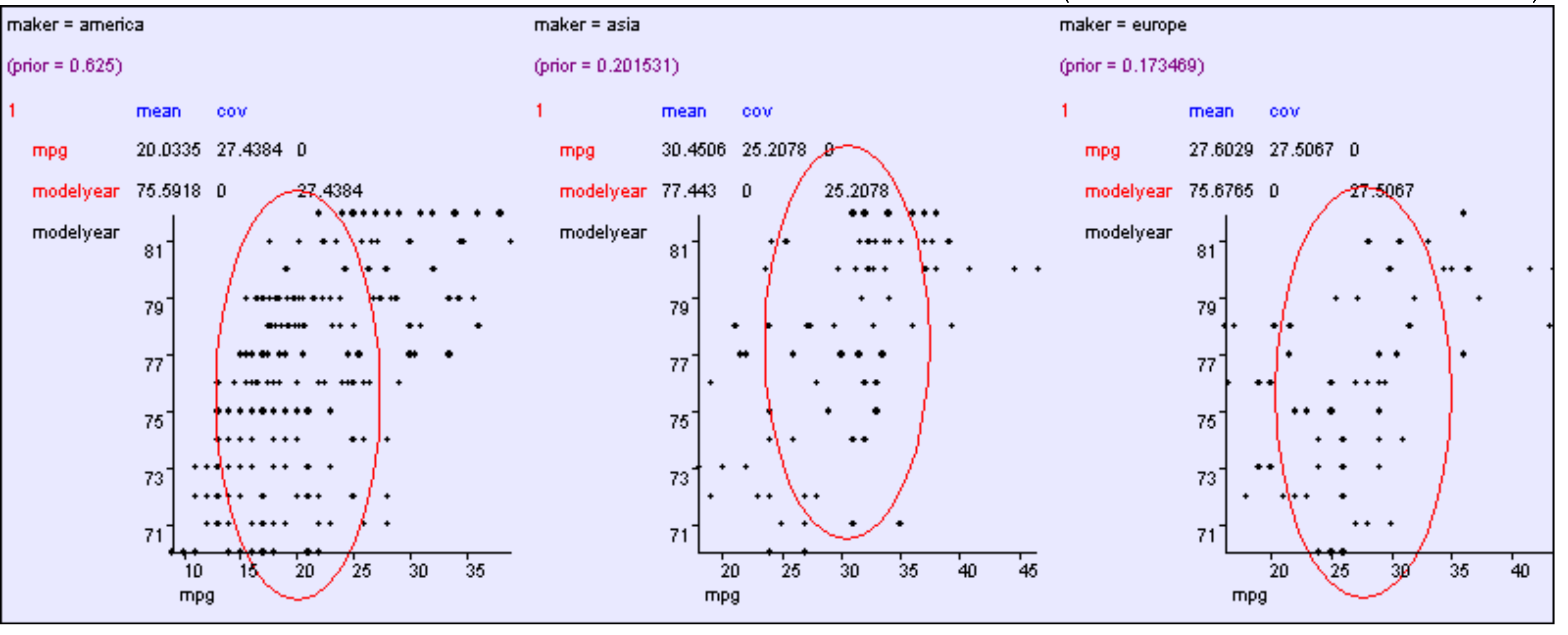
# Aligned: *O(m)* parameters

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



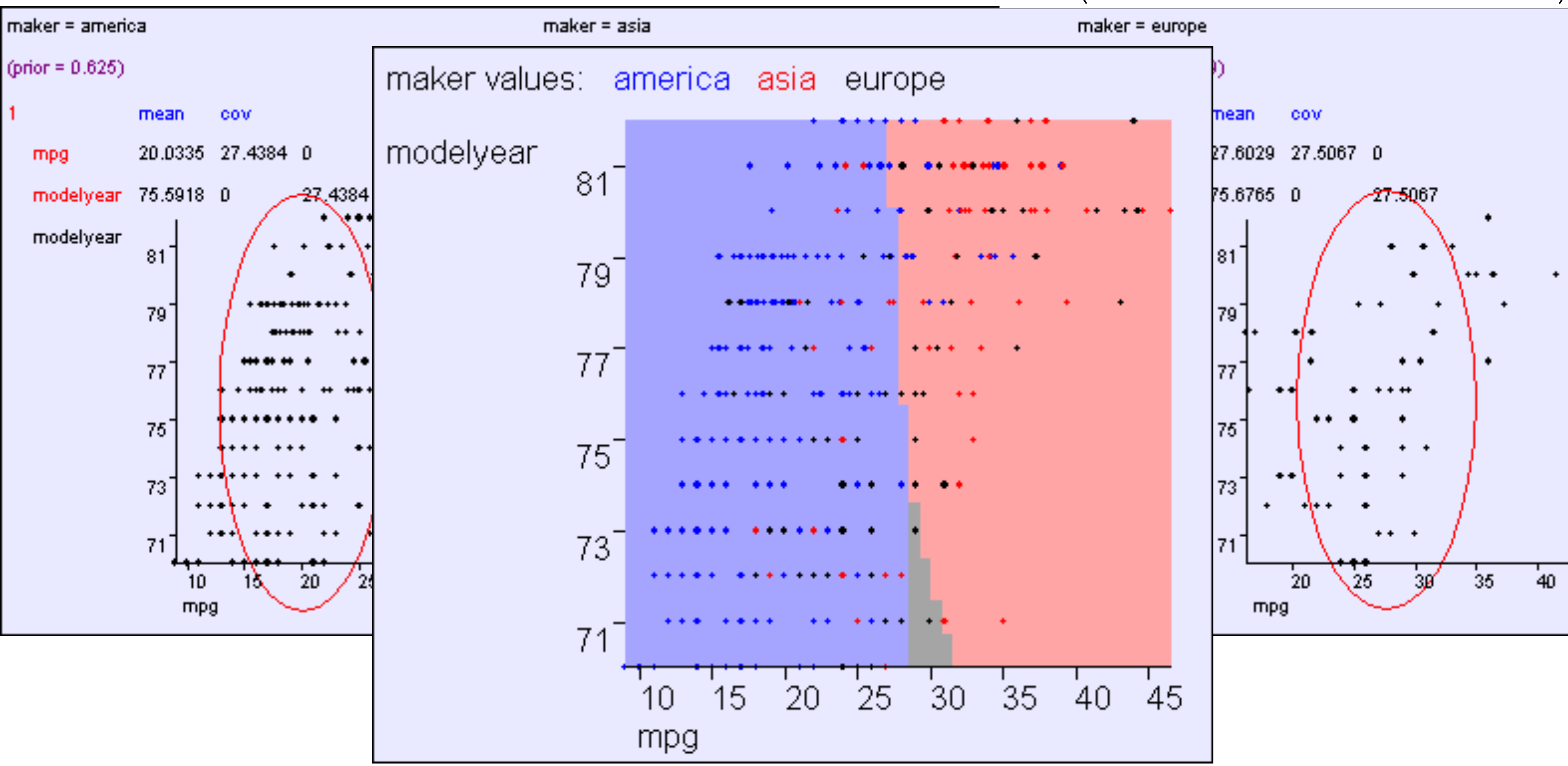
# Spherical: *O(1)* cov parameters

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



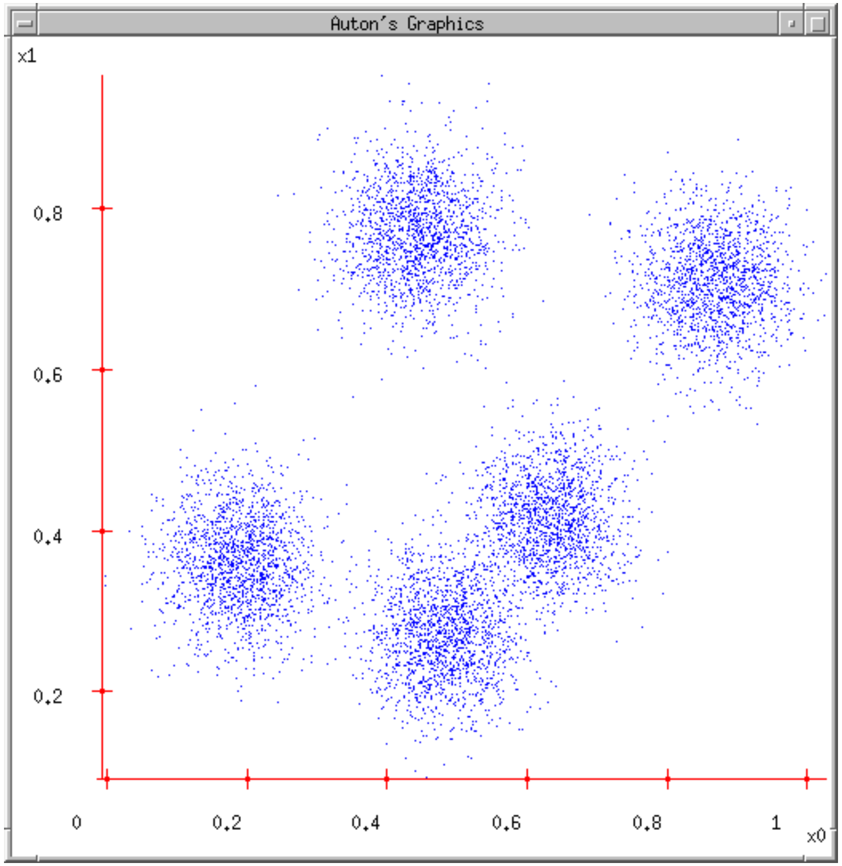
# Spherical: *O(1)* cov parameters

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

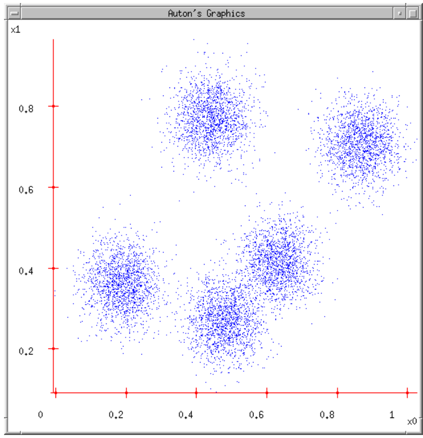


# Density estimation

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



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




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

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

# Special case: spherical Gaussians and hard assignments

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

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

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

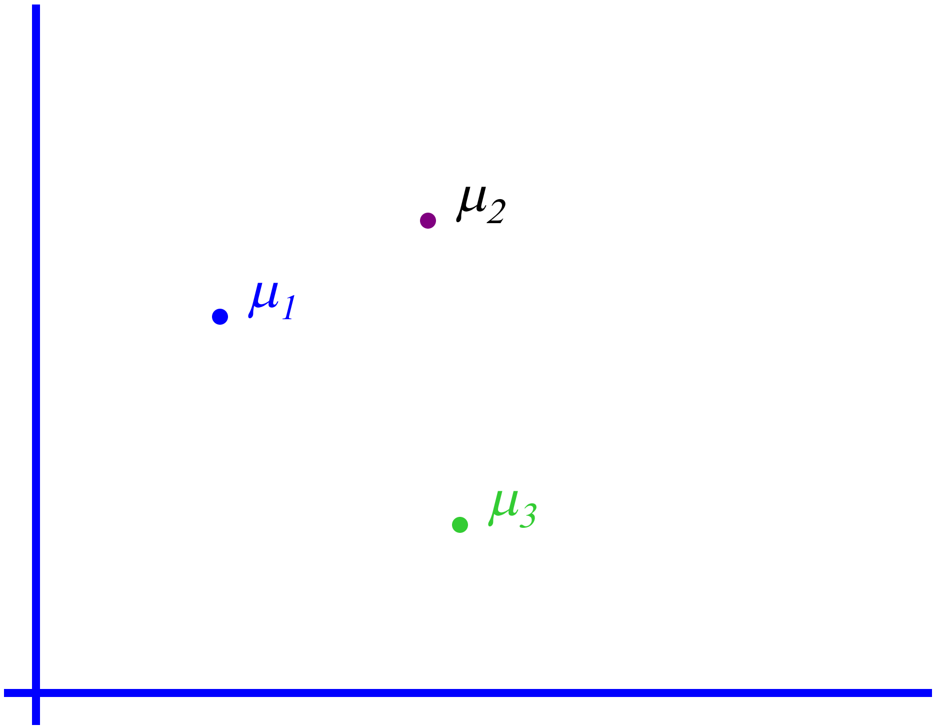
- If each $x_j$ belongs to one class C(j) (hard assignment), marginal likelihood:

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

- Same as K-means!!!

# The GMM assumption

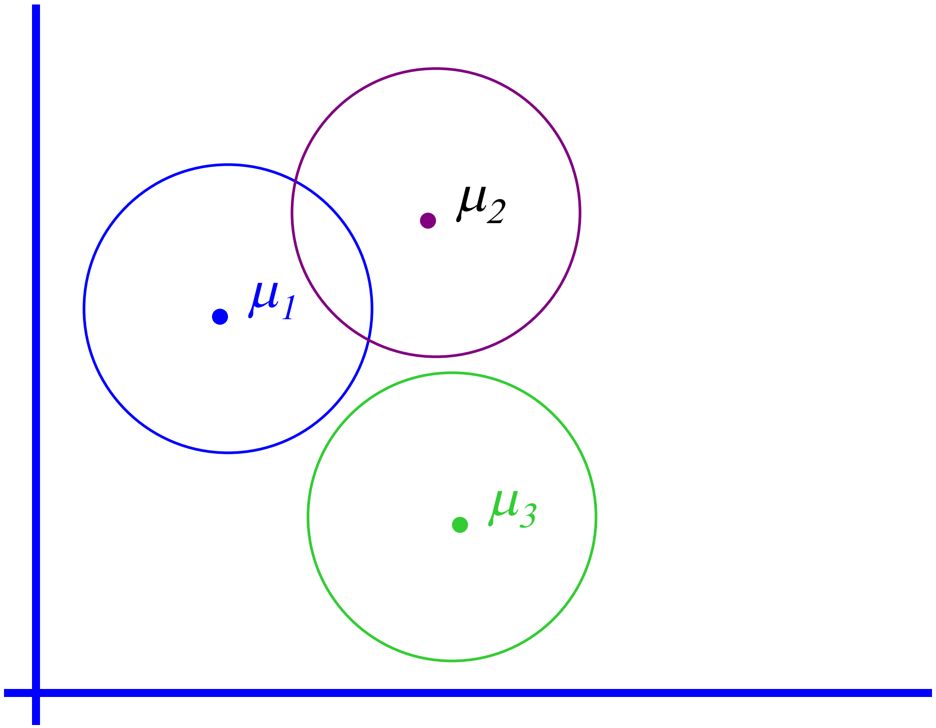
- There are k components
- Component *i* has an associated mean vector $\mu_i$



# The GMM assumption

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

Each data point is generated according to the following recipe:

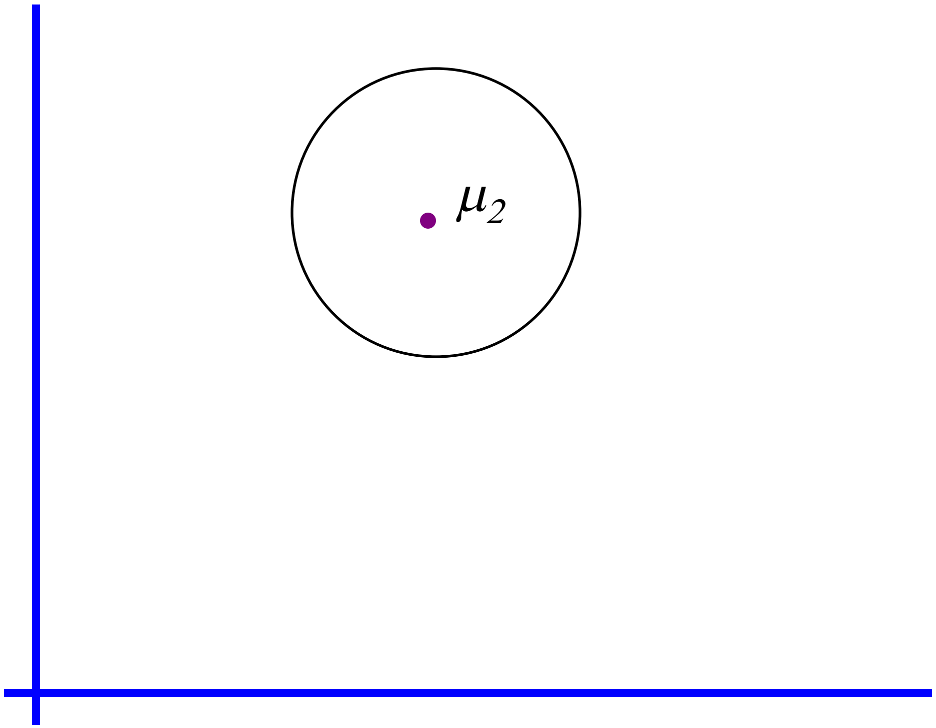


# The GMM assumption

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

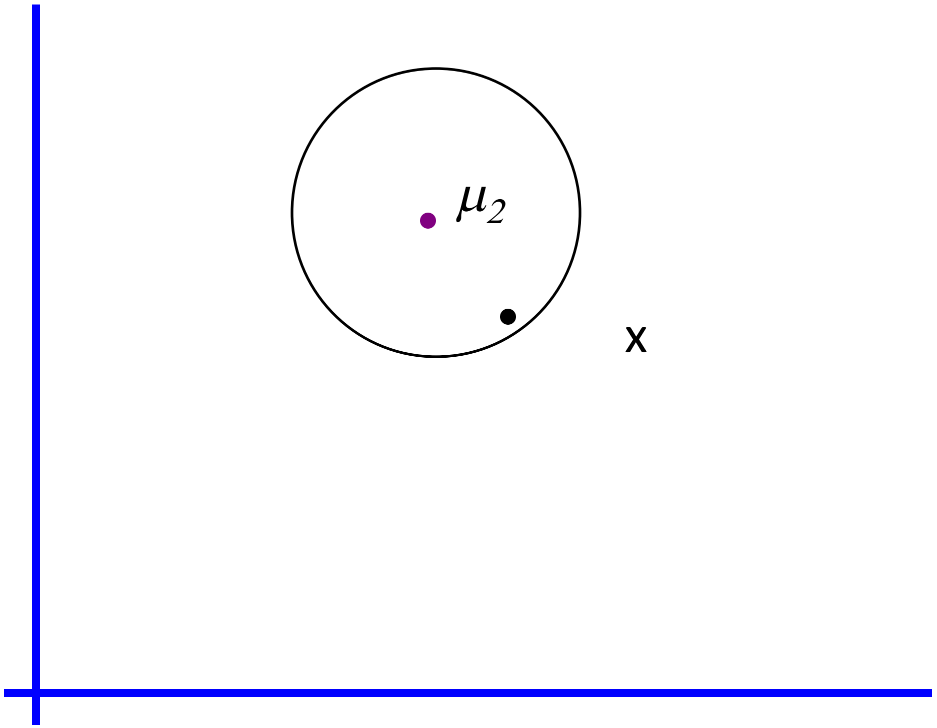
Each data point is generated according to the following recipe:

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



# The GMM assumption

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

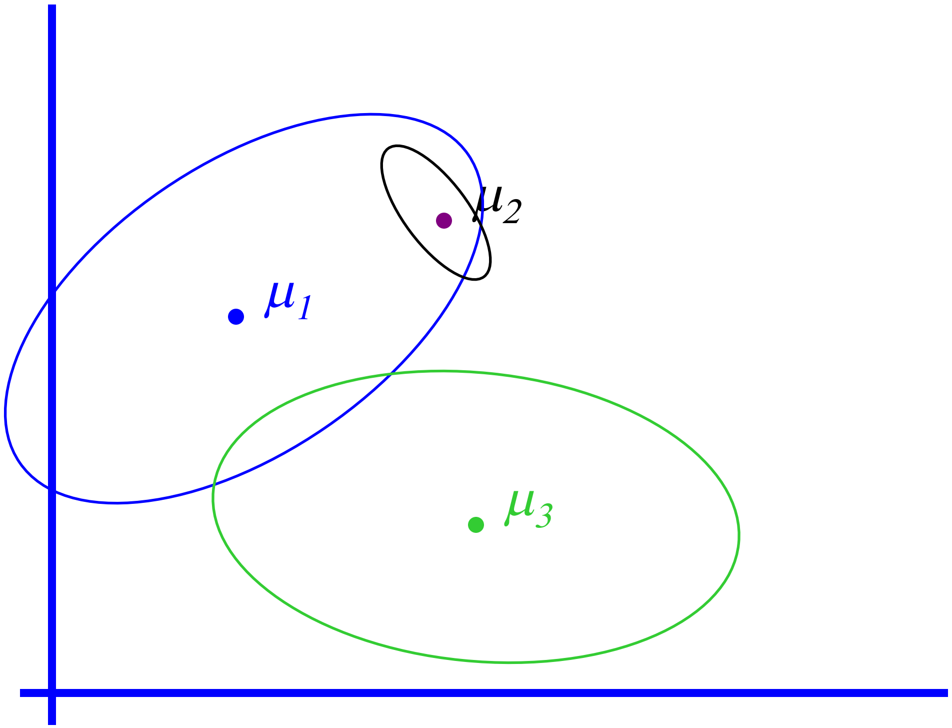



Each data point is generated according to the following recipe:

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

# The General GMM assumption

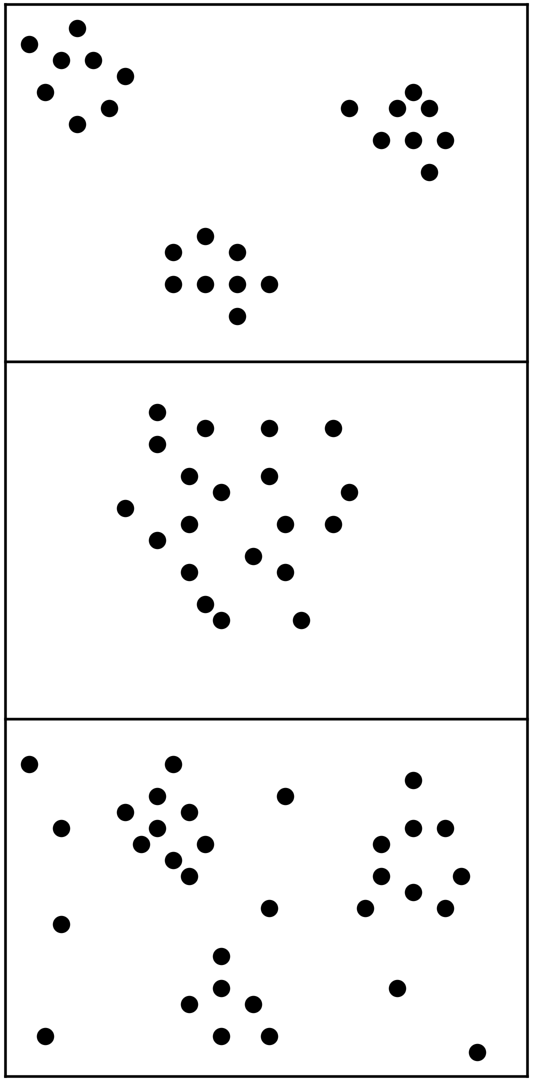
- There are k components
- Component *i* has an associated mean vector $\mu_i$
- Each component generates data from a Gaussian with mean $\mu_i$ and covariance matrix $\Sigma_i$




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. Data point ~ $\mathrm{N}(\mu_i, \Sigma_i)$

# 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:

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

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

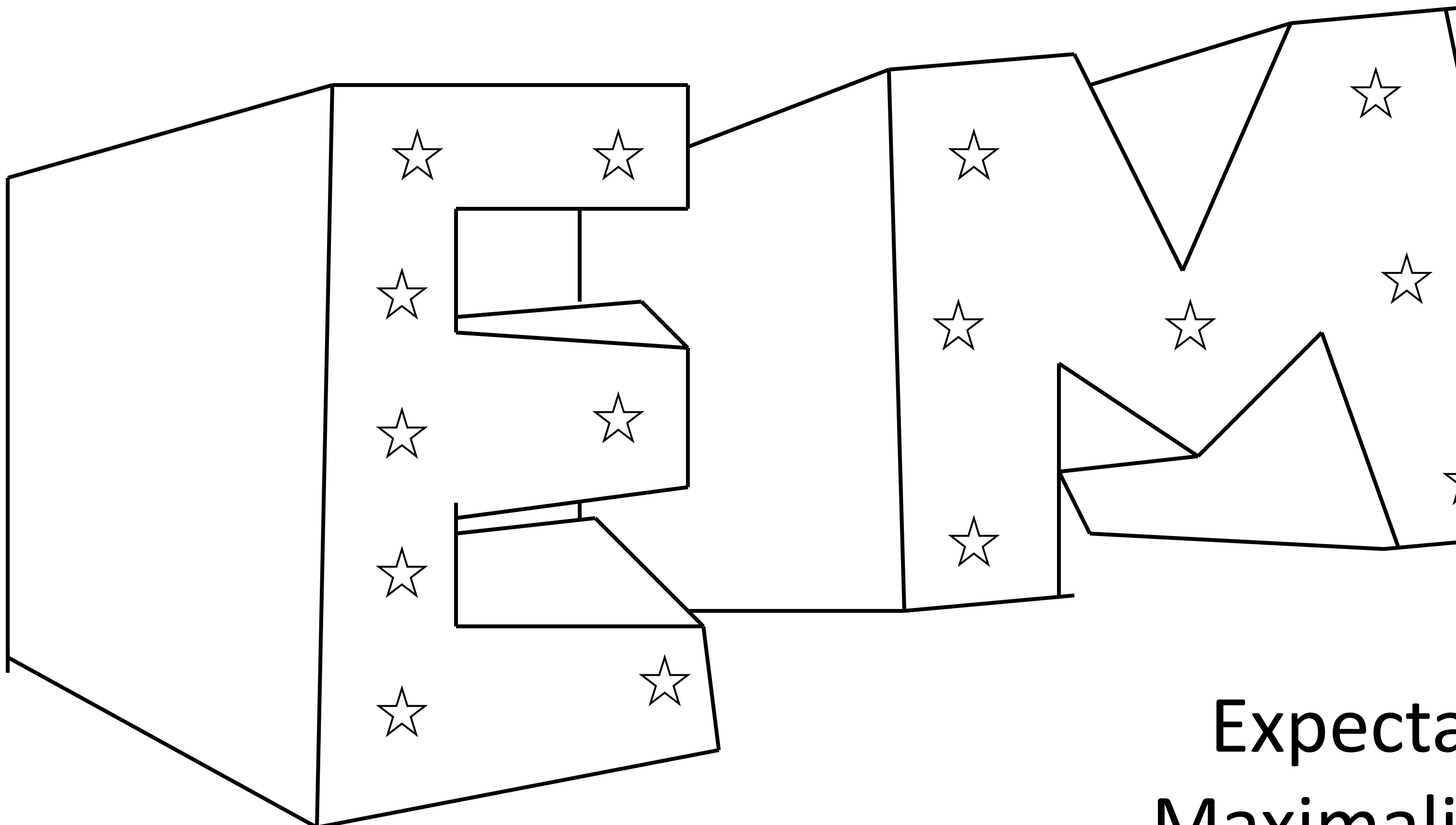
# Special case:
# spherical Gaussians and soft assignments

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

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

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

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



Expectation
Maximalization

# 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 $0 \leq \mu \leq 1/6$)

Assume we want to estimate $\mu$ 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 $\mu$ given a,b,c,d ?

# Trivial statistics

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

$P(\, a,b,c,d \mid \mu) = (½)^{a}(\mu)^{b}(2\mu)^{c}(½-3\mu)^{d}$

$\log P(\, a,b,c,d \mid \mu) = a\log ½ + b\log \mu + c\log 2\mu + d\log (½-3\mu)$

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

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

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

So if class got

| A | B | C | D |
|---|---|---|---|
| 14 | 6 | 9 | 10 |

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

Boring, but true!

# Problem with hidden information

REMEMBER

$P(A) = \frac{1}{2}$

$P(B) = \mu$

$P(C) = 2\mu$

$P(D) = \frac{1}{2} - 3\mu$

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?

# Problem with hidden information

REMEMBER

$P(A) = ½$

$P(B) = \mu$

$P(C) = 2\mu$

$P(D) = ½-3\mu$

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:

**EXPECTATION**

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

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

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

**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

We begin with a guess for μ

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

REMEMBER

P(A) = ½

P(B) = μ

P(C) = 2μ

P(D) = ½-3μ

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

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

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

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

**E-step**

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

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

**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

$h = 20$

$c = 10$

$d = 10$

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

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

| 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 |

# Back to unsupervised learning of GMMs – a simple case

A simple case:

We have unlabeled data $\boldsymbol{x}_1\ \boldsymbol{x}_2\ \ldots\ \boldsymbol{x}_m$

We know there are k classes

We know $P(y_1)\ P(y_2)\ P(y_3)\ \ldots\ P(y_k)$

We <u>don't</u> know $\boldsymbol{\mu}_1\ \boldsymbol{\mu}_2\ ..\ \boldsymbol{\mu}_k$

We can write P( data | $\boldsymbol{\mu}_1 \ldots \boldsymbol{\mu}_k$)

$$= \mathrm{p}\left(x_1 \ldots x_m \middle| \mu_1 \ldots \mu_k\right)$$

$$= \prod_{j=1}^{m} \mathrm{p}\left(x_j \middle| \mu_1 \ldots \mu_k\right)$$

$$= \prod_{j=1}^{m} \sum_{i=1}^{k} \mathrm{p}\left(x_j \middle| \mu_i\right) \mathrm{P}(y=i)$$

$$\propto \prod_{j=1}^{m} \sum_{i=1}^{k} \exp\left(-\frac{1}{2\sigma^2}\left\|x_j - \mu_i\right\|^2\right) \mathrm{P}(y=i)$$

# EM for simple case of GMMs: The E-step

- If we know $\mu_1,\ldots,\mu_k$ $\rightarrow$ easily compute prob. point $x_j$ belongs to class y=i

$$\mathrm{p}\left(y=i \middle| x_j, \mu_1 \ldots \mu_k\right) \propto \exp\left(-\frac{1}{2\sigma^2}\left\|x_j-\mu_i\right\|^2\right)\mathrm{P}(y=i)$$

# EM for simple case of GMMs: The M-step

- If we know prob. point $x_j$ belongs to class y=i

  $\rightarrow$ MLE for $\mu_i$ is weighted average

  – imagine k copies of each $x_j$, each with weight $P(y=i|x_j)$:

$$\mu_i = \frac{\sum_{j=1}^{m} P(y=i|x_j)x_j}{\sum_{j=1}^{m} P(y=i|x_j)}$$

# E.M. for GMMs

**E-step**

Compute "expected" classes of all data points for each class

$$p\left(y=i \middle| x_j, \mu_1 \ldots \mu_k\right) \propto \exp\left(-\frac{1}{2\sigma^2}\left\|x_j-\mu_i\right\|^2\right) P(y=i)$$

*Just evaluate a Gaussian at $x_j$*

**M-step**

Compute Max. like **μ** given our data's class membership distributions

$$\mu_i = \frac{\sum_{j=1}^{m} P\left(y=i \middle| x_j\right) x_j}{\sum_{j=1}^{m} P\left(y=i \middle| x_j\right)}$$