Week 6

Clustering

Emre Ugur, BM 33

emre.ugur@boun.edu.tr

http://www.cmpe.boun.edu.tr/~emre/courses/cmpe462

cmpe462@listeci.cmpe.boun.edu.tr

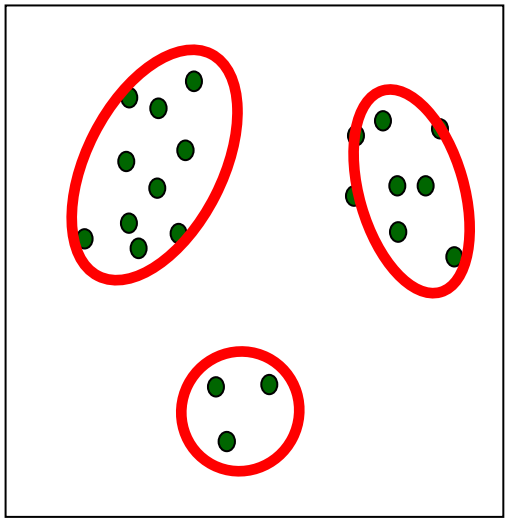
Acknowledgements

- Dimensionality reduction: adapted from textbook materials
- Clustering: adapted from Alexander Ihler's Machine Learning course material

Unsupervised learning

- Supervised learning
- Predict target value (“y”) given features (“x”)

- Unsupervised learning
- Understand patterns of data (just “x”)
- Useful for many reasons
- Data mining (“explain”)
- Representation (feature generation or selection)

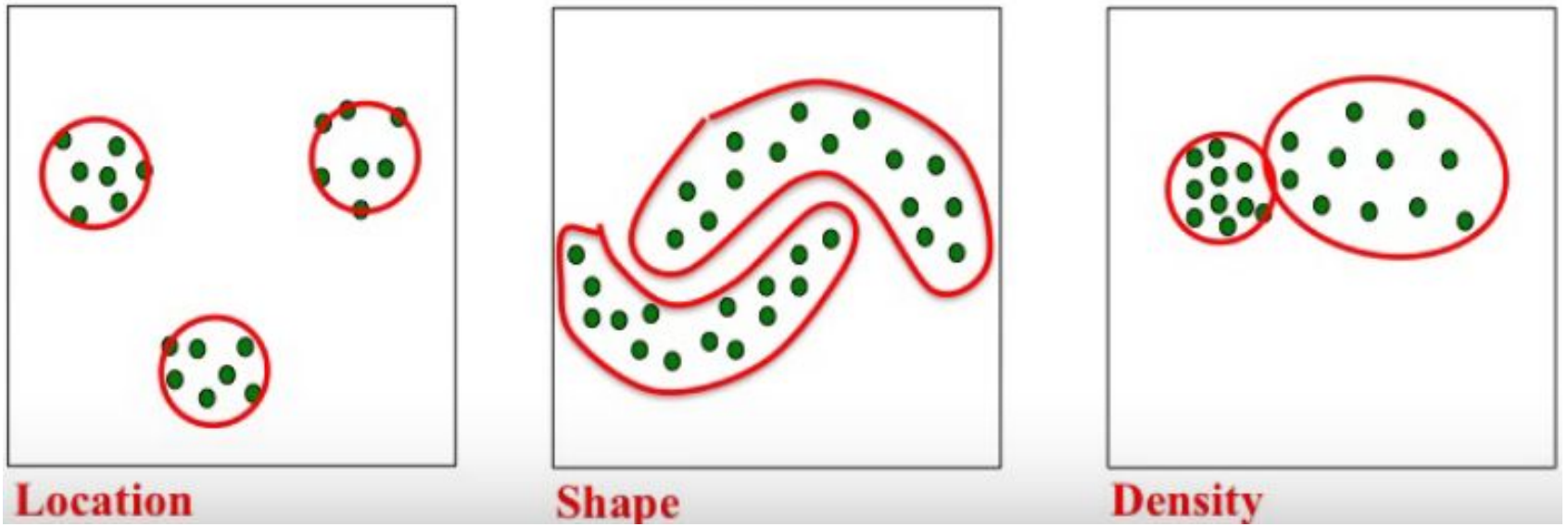


- One example: *clustering*
- *Describe data by discrete “groups” with some characteristics*

Clustering and Data Compression

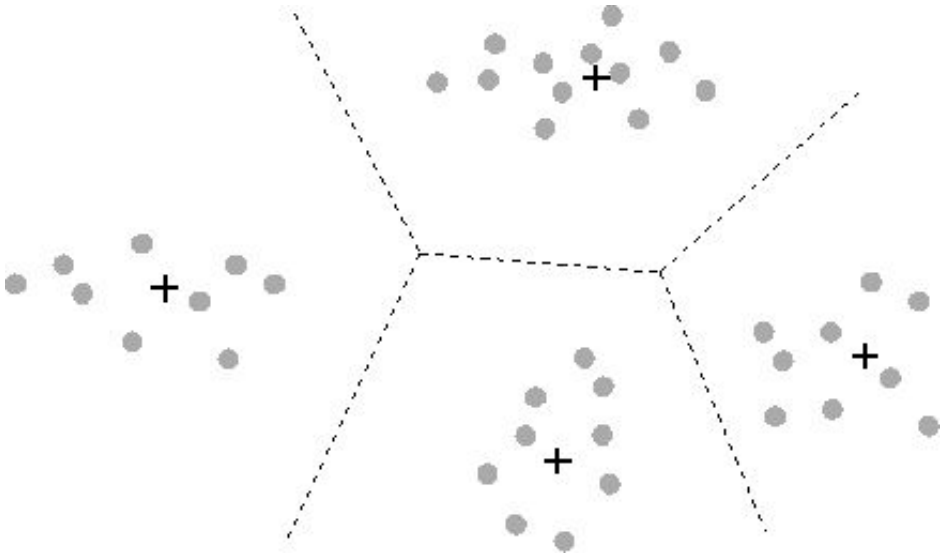
- Cluster describes data by “groups”
- The meaning of groups may vary by data

- Examples



Clustering and Data Compression

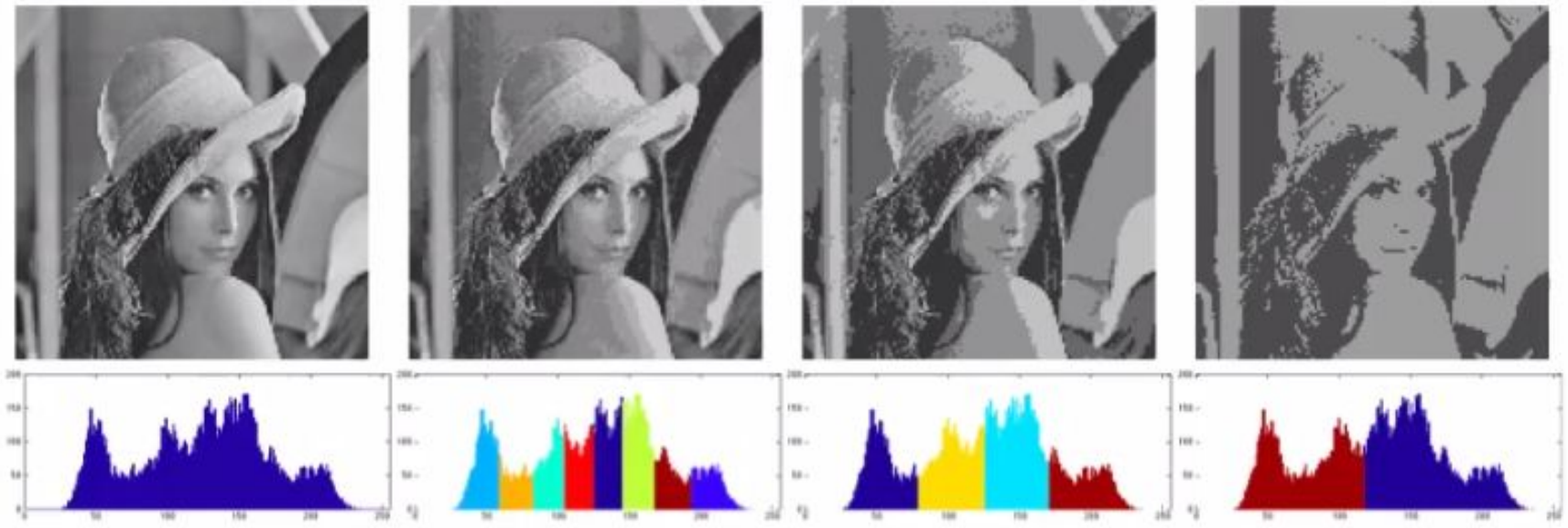
- Clustering is related to vector quantization
- Dictionary of vectors (the cluster centers)
- Each original value represented using a dictionary index
- Each center “claims” a nearby region (Voronoi region)



Clustering and Data Compression

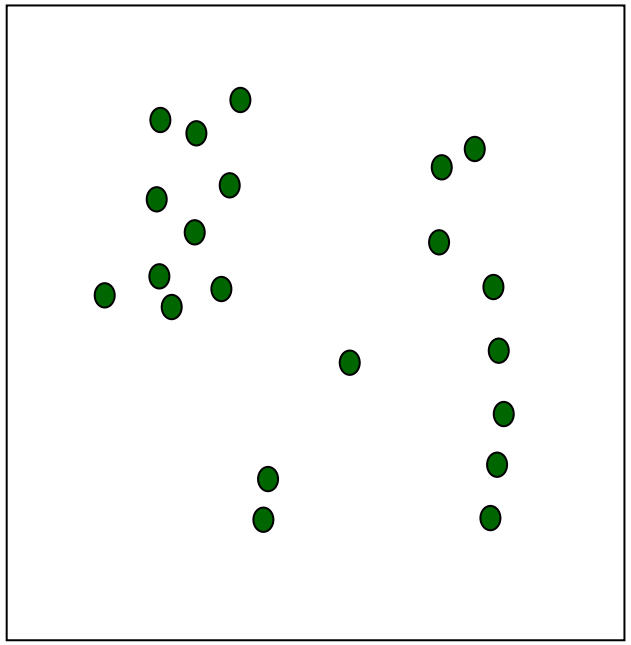
- Clustering is related to vector quantization
- Dictionary of vectors (the cluster centers)
- Each original value represented using a dictionary index
- Each center “claims” a nearby region (Voronoi region)

- Example in 1D: cluster pixels' grayscale values



Hierarchical Agglomerative Clustering

Initially, every datum is a cluster



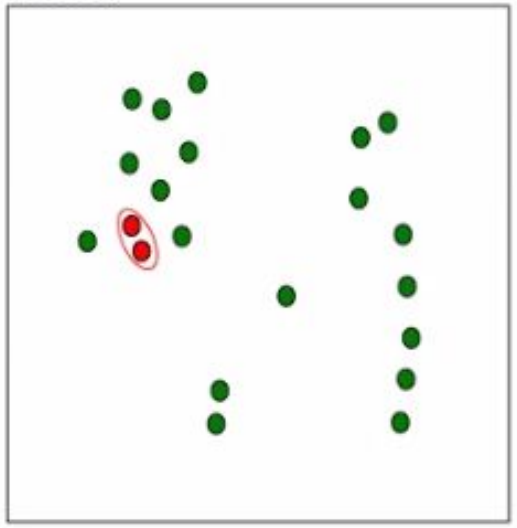
- Another simple clustering alg

- Define a distance between clusters
- Initialize: every example is a cluster
- Iterate:
  - Compute distances between all clusters (store for efficiency)
  - Merge two closest clusters
- Save both clustering and *sequence* of cluster ops
- “Dendrogram”

Iteration 1

Builds up a sequence of clusters ("hierarchical")

Data:



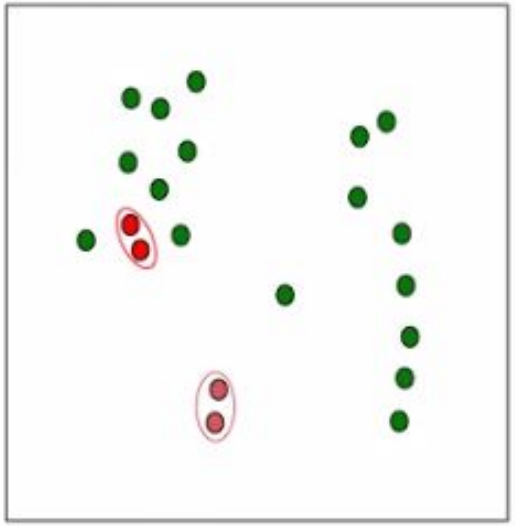
Dendrogram:

Height of the join
indicates dissimilarity

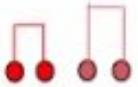
Iteration 2

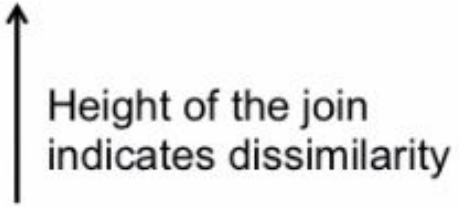
Builds up a sequence of clusters ("hierarchical")

Data:



Dendrogram:

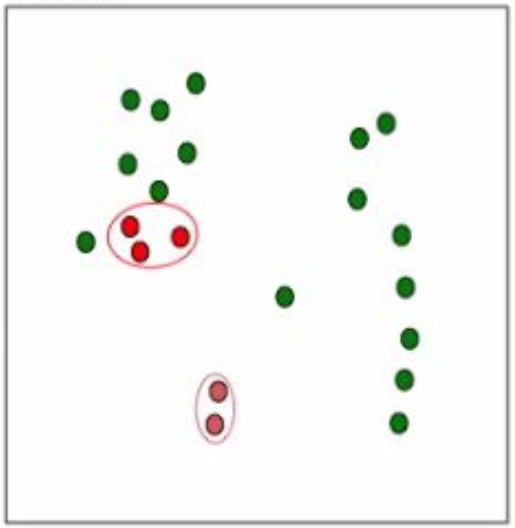




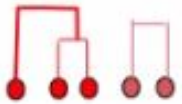
Iteration 3

Builds up a sequence of clusters ("hierarchical")

Data:



Dendrogram:

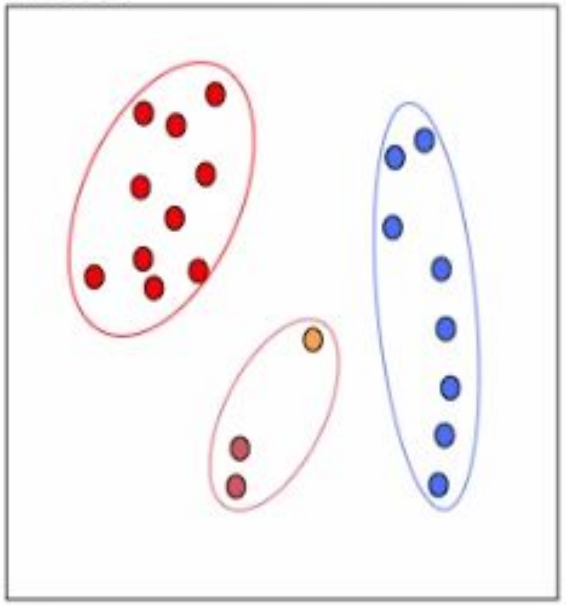


Height of the join indicates dissimilarity

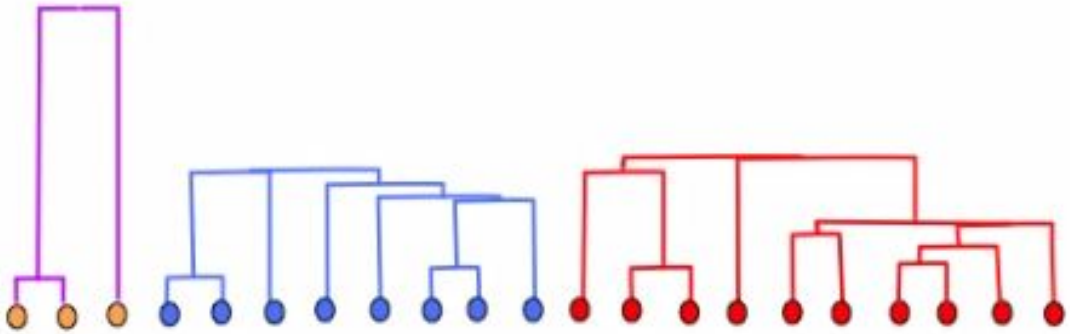
In matlab: "linkage" function (stats toolbox)

Eventually

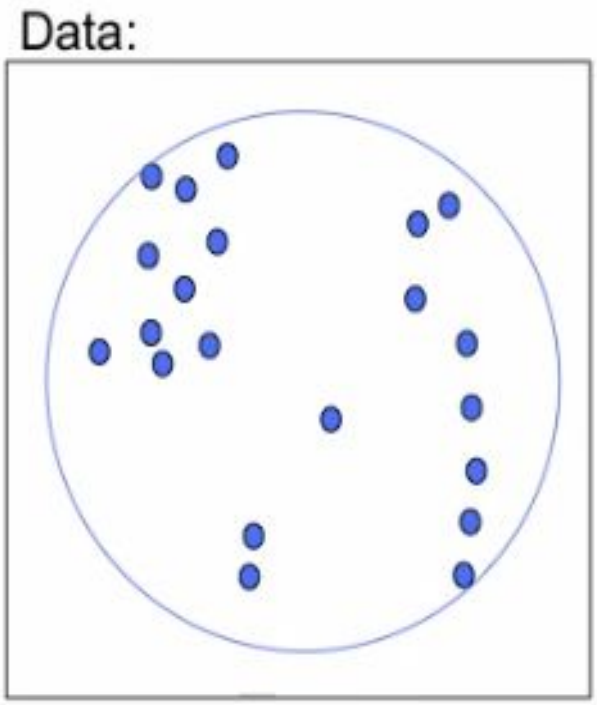
Data:

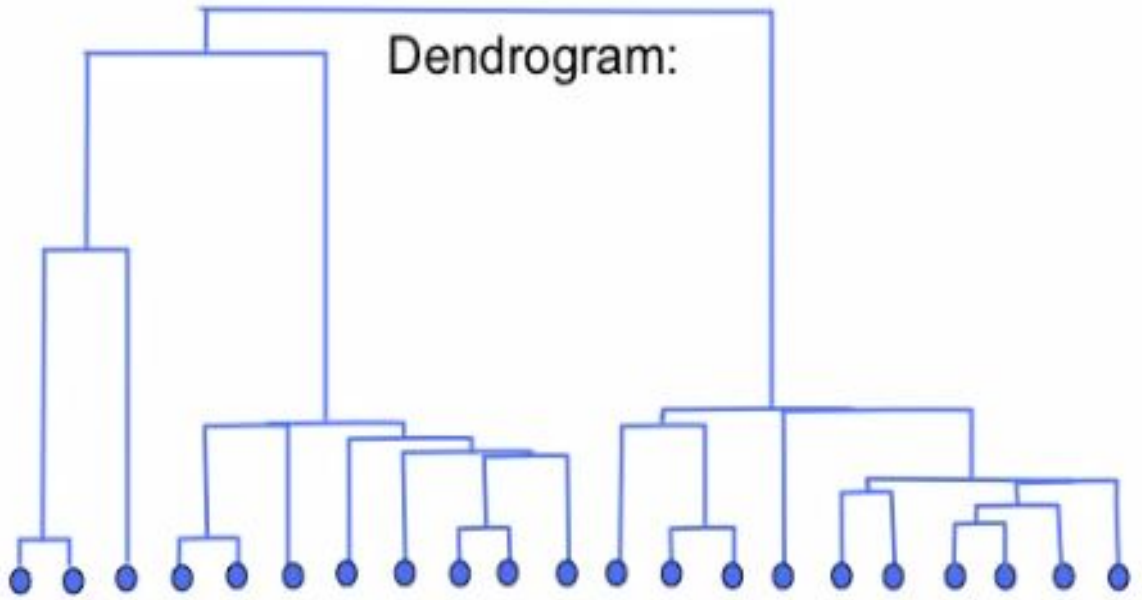


Dendrogram:



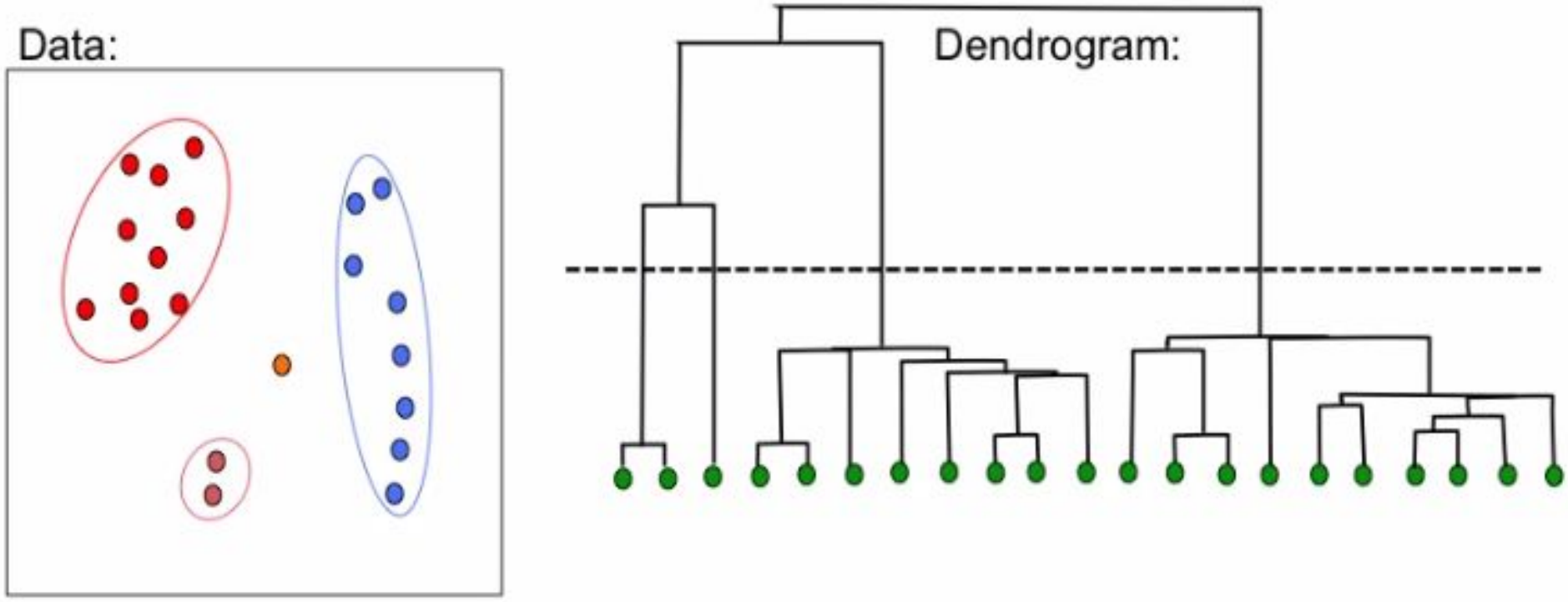
Dendrogram





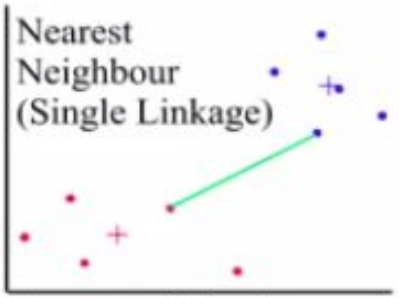
From dentogram to clusters

Given the sequence, can select a number of clusters or a dissimilarity threshold:



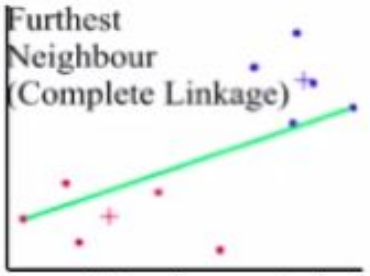
Cluster distances

$$D_{\min}(C_i, C_j) = \min_{x \in C_i,\ y \in C_j} \|x - y\|^2$$



produces minimal spanning tree.

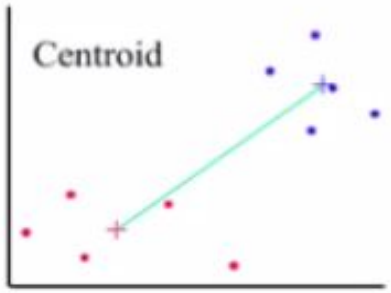
$$D_{\max}(C_i, C_j) = \max_{x \in C_i,\ y \in C_j} \|x - y\|^2$$



avoids elongated clusters.

$$D_{\text{avg}}(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{x \in C_i,\ y \in C_j} \|x - y\|^2$$

$$D_{\text{means}}(C_i, C_j) = \|\mu_i - \mu_j\|^2$$

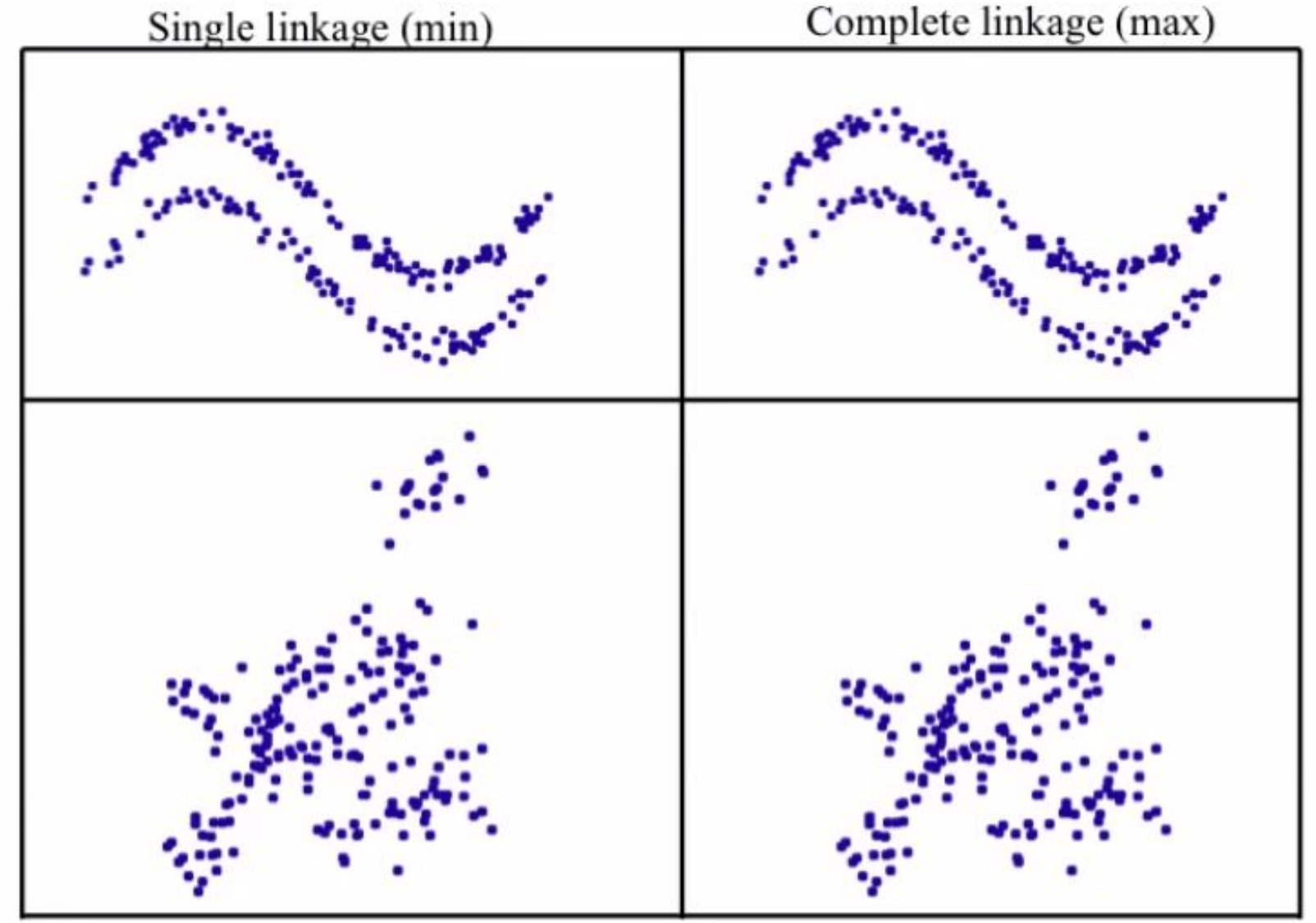


Need:

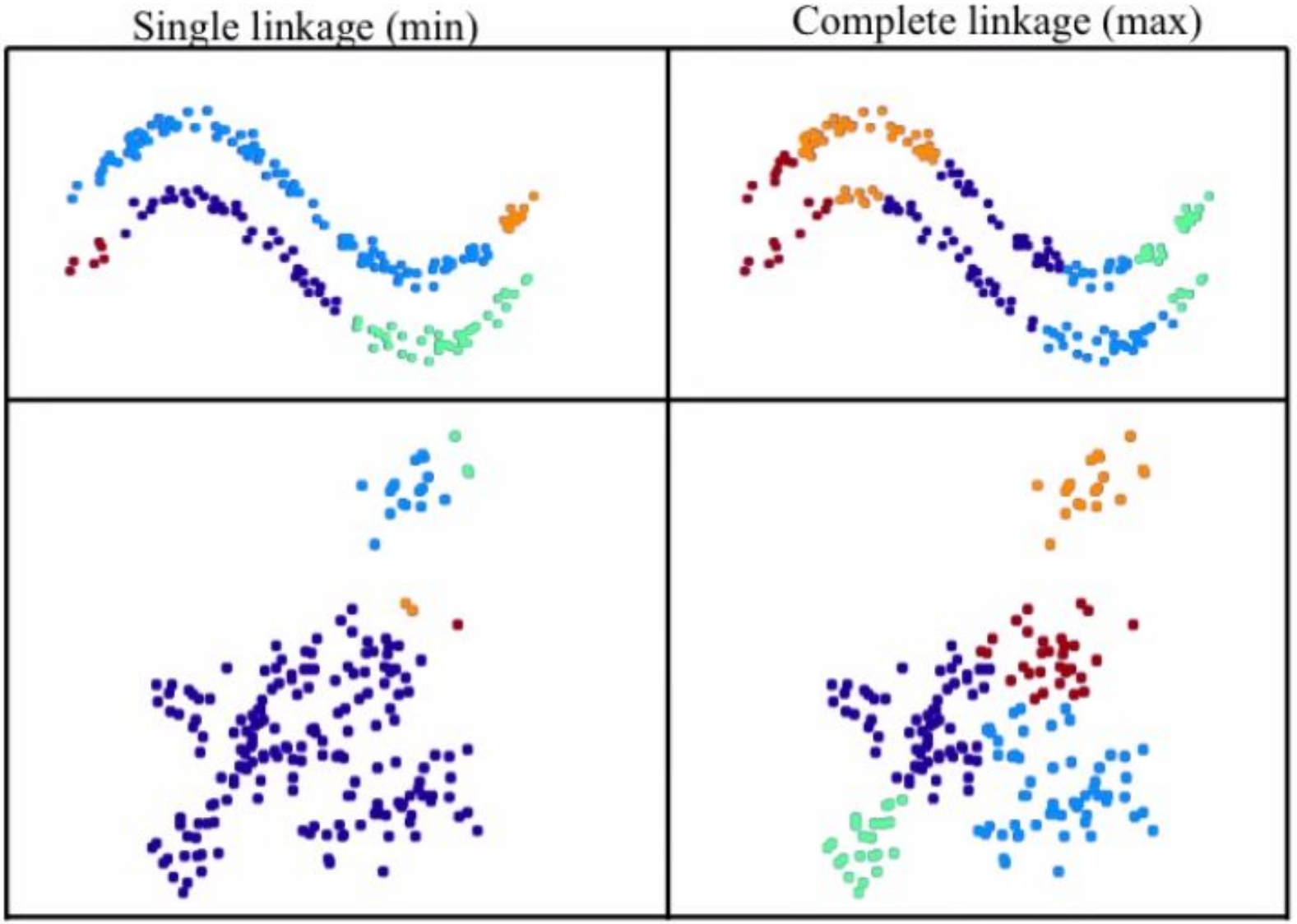
D(A,C) → D(A+B,C)

D(B,C) →

Cluster Distances

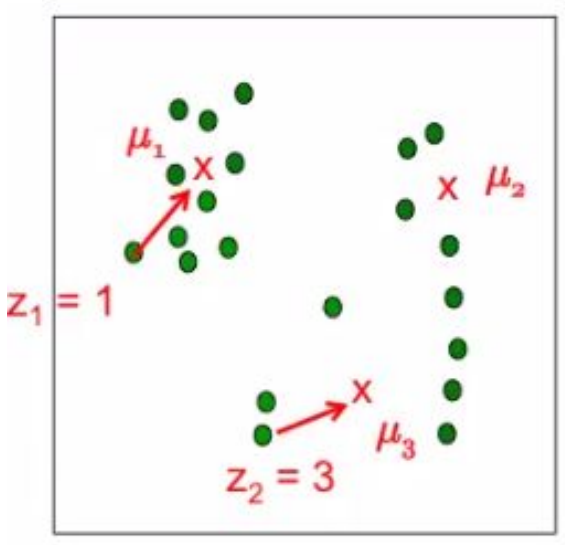


Cluster Distances



K-Means Clustering

- A simple clustering algorithm
- Iterate between
    - Updating the assignment of data to clusters
    - Updating the cluster's summarization
- Suppose we have K clusters, c=1..K
- Represent clusters by locations $^{1}_{c}$
- Example i has features x_i
- Represent assignment of i^{th} example z_i 2 1..K



K-Means Clustering

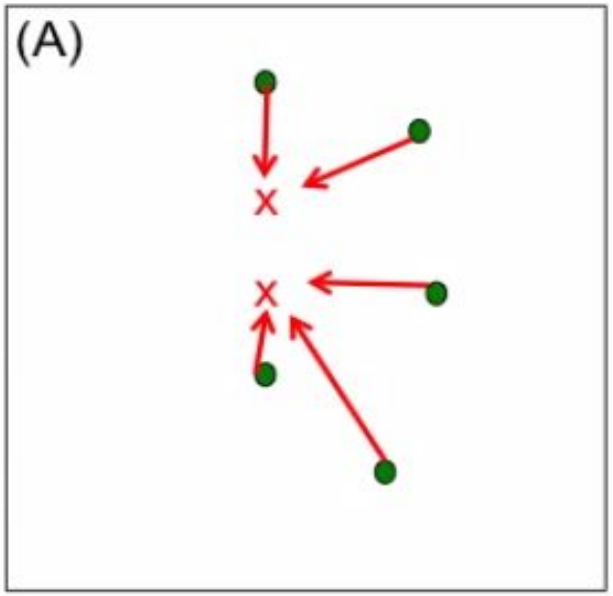
Iterate until convergence:

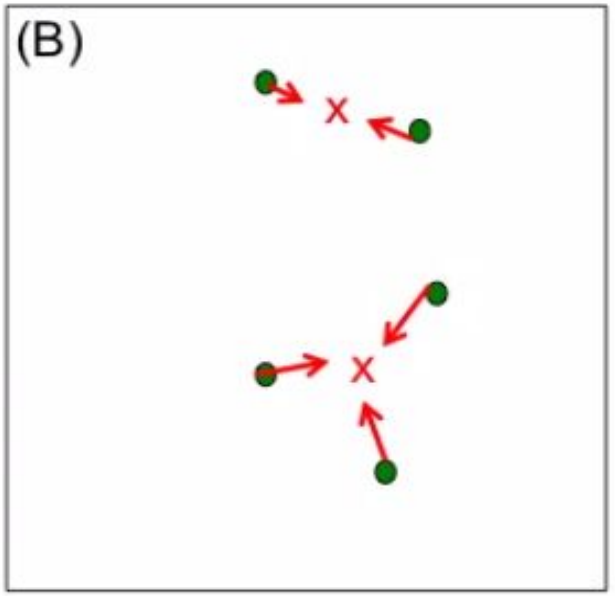
- (A) For each datum, find the closest cluster

$$z_i = \arg\min_c \|x_i - \mu_c\|^2 \qquad \forall i$$

- (B) Set each cluster to the mean of all assigned data:

$$\forall c, \qquad \mu_c = \frac{1}{m_c} \sum_{i \in S_c} x_i \qquad S_c = \{i : z_i = c\},\ m_c = |S_c|$$





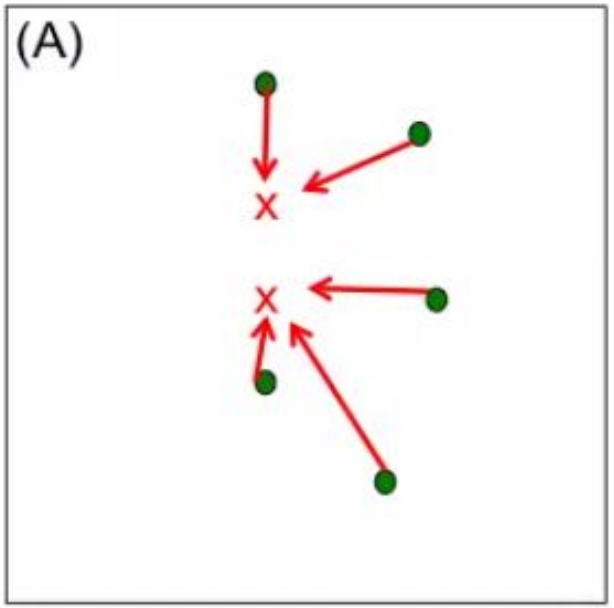
K-Means Clustering

- Optimizing the cost function:

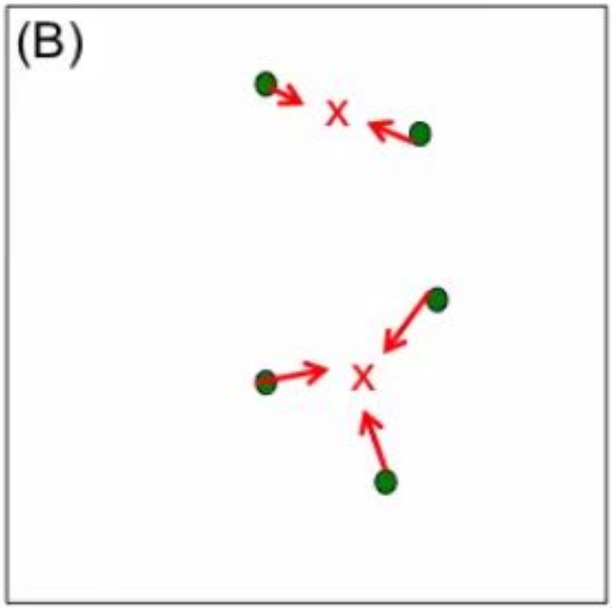
$$C(\underline{z}, \underline{\mu}) = \sum_i \|x_i - \mu_{z_i}\|^2$$

- Coordinate descent:

Over the cluster assignments:
Only one term in sum depends on z_i
Minimized by selecting closest μ_c



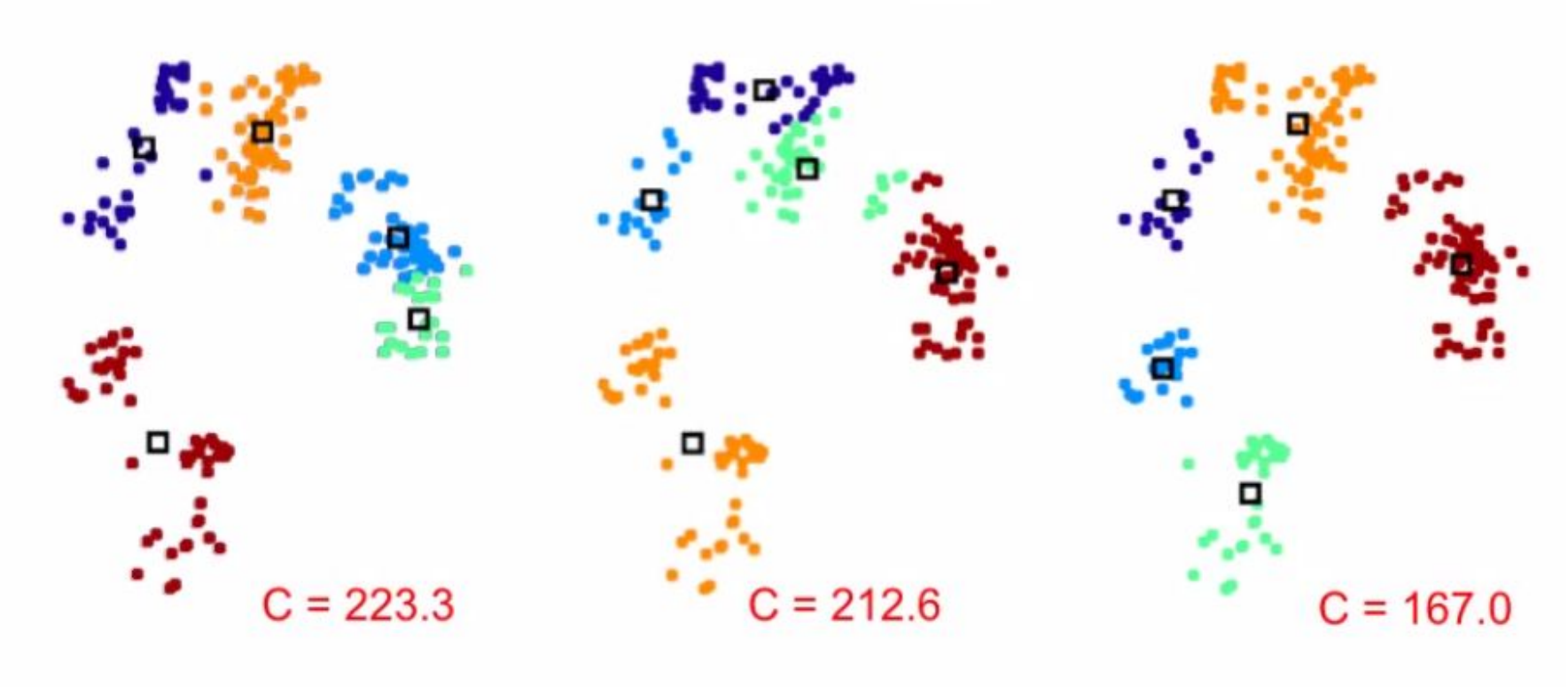
Over the cluster centers:
Cluster c only depends on x_i with z_i=c
Minimized by selecting the mean



K-Means clustering

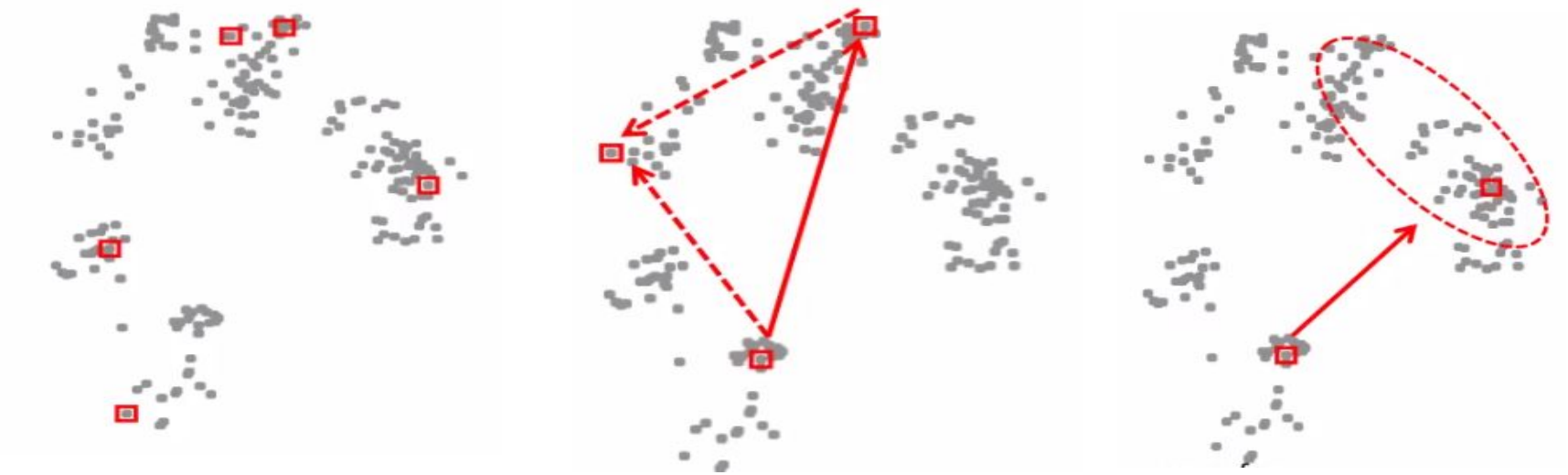
- As with any descent method, beware of local minima
- Algorithm behavior depends significantly on initalization

$$C(\underline{z}, \underline{\mu}) = \sum_i \|x_i - \mu_{z_i}\|^2$$



K-Means clustering

- As with any descent method, beware of local minima
- Algorithm behavior depends significantly on initalization
- Random: ensures centers are near data,
- may choose nearby points
- Distance-based: find the point farthest from the clusters so far
- may choose outliers
- Random+distance: choose next points far but randomly

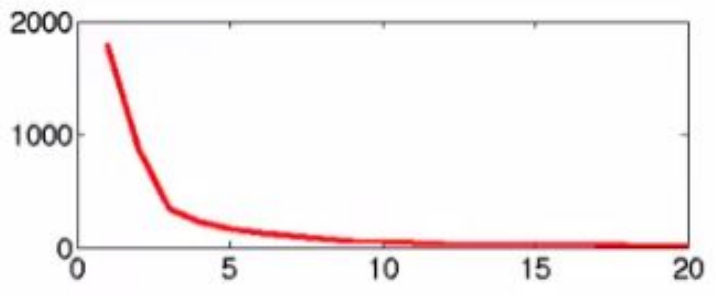


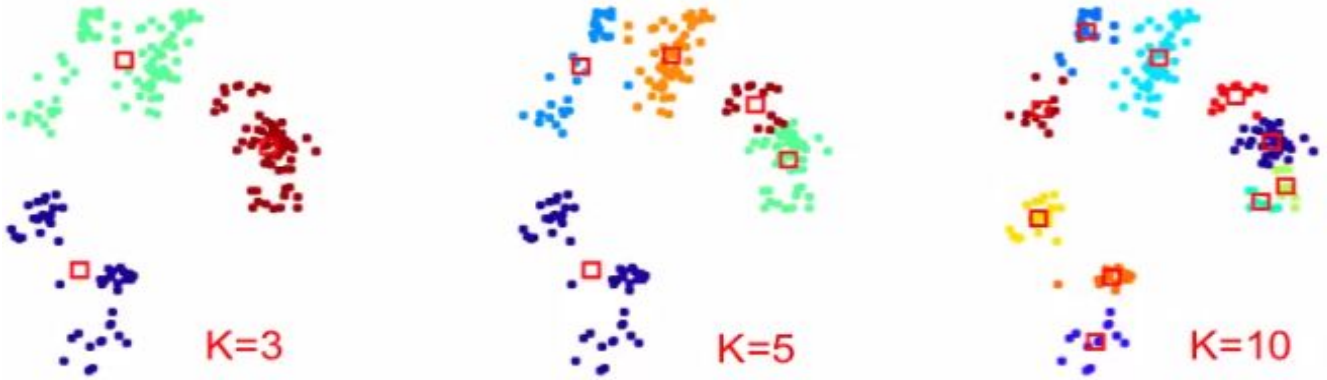
Choosing the number of clusters

- With cost function

$$C(\underline{z}, \underline{\mu}) = \sum_i \|x_i - \mu_{z_i}\|^2$$

what is the optimal value of k?
(can increasing k ever increase the cost?)





- One solution is to penalize for complexity

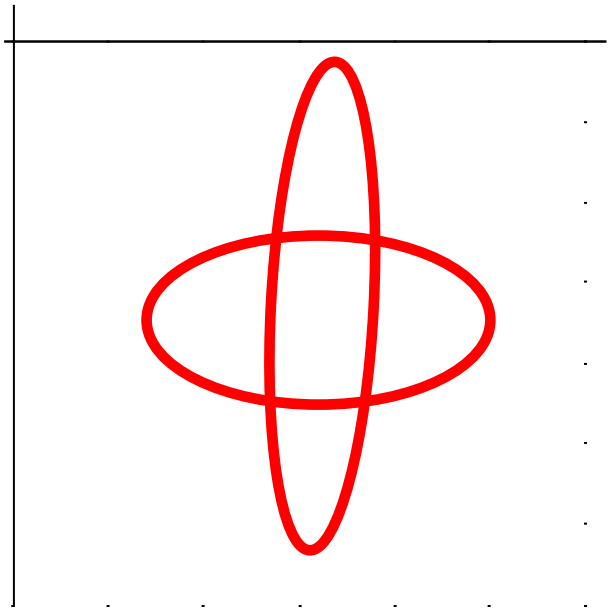
$$J(\underline{z}, \underline{\mu}) = \log\left[\frac{1}{m\,d}\sum_i \|x_i - \mu_{z_i}\|^2\right] + k\frac{\log m}{m}$$

X-means method

Mixtures of Gaussians

- K-means algorithm
    - Assigned each example to exactly one cluster
    - What if clusters are overlapping?
        - Hard to tell which cluster is right
    - Maybe we should try to remain uncertain
    - Used Euclidean distance
    - What if cluster has a non-circular shape?

- Gaussian mixture models
    - Clusters modeled as Gaussians
        - Not just by their mean
    - EM algorithm: assign data to cluster with some *probability*



Revisit k-means

- EM'ish algorithm, define unobserved latent variables z_i, cluster membership

Iterate until convergence:

- (A) For each datum, find the closest cluster

$$z_i = \arg\min_c \|x_i - \mu_c\|^2 \qquad \forall i$$

Compute expected value of latent variable z_i based on model params

- (B) Set each cluster to the mean of all assigned data:

$$\forall c, \qquad \mu_c = \frac{1}{m_c} \sum_{i \in S_c} x_i \qquad S_c = \{i : z_i = c\},\ m_c = |S_c|$$

Minimize error / maximize likelihood

$$C(\underline{z}, \underline{\mu}) = \sum_i \|x_i - \mu_{z_i}\|^2$$ neters)

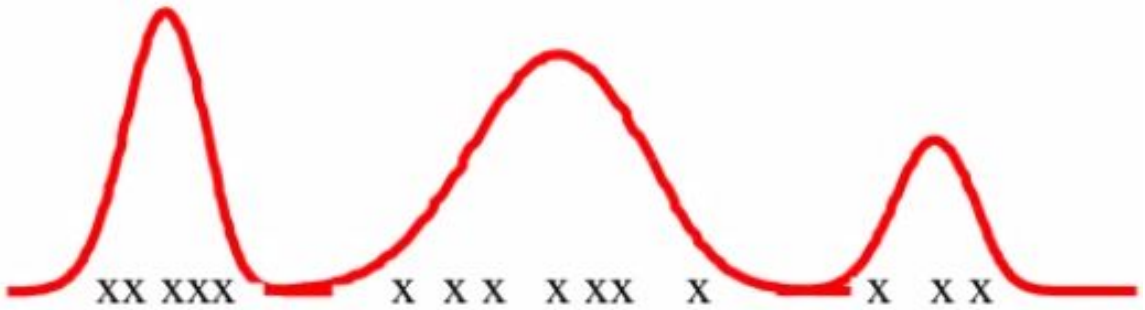
Lectu

Mixtures of Gaussians

Start with parameters describing each cluster

Mean μ_c, variance σ_c, "size" π_c

Probability distribution: $p(x) = \sum_c \pi_c \mathcal{N}(x\ ;\ \mu_c, \sigma_c)$



Mixtures of Gaussians

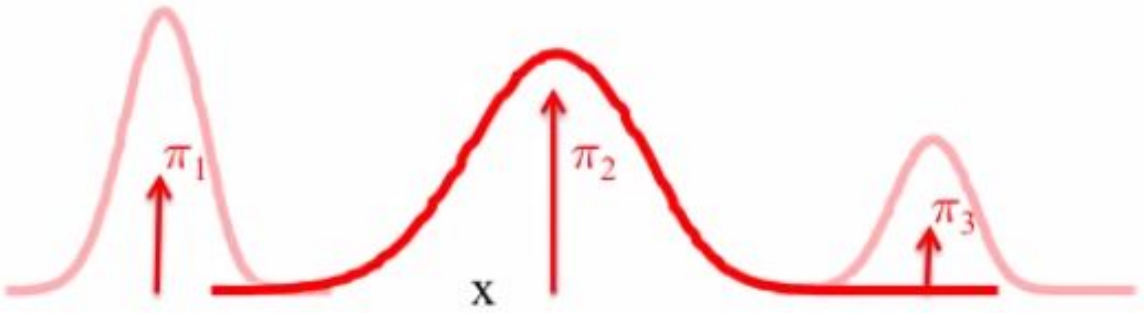
Start with parameters describing each cluster

Mean μ_c, variance σ_c, "size" π_c

Probability distribution: $p(x) = \sum_c \pi_c \, \mathcal{N}(x \; ; \; \mu_c, \sigma_c)$

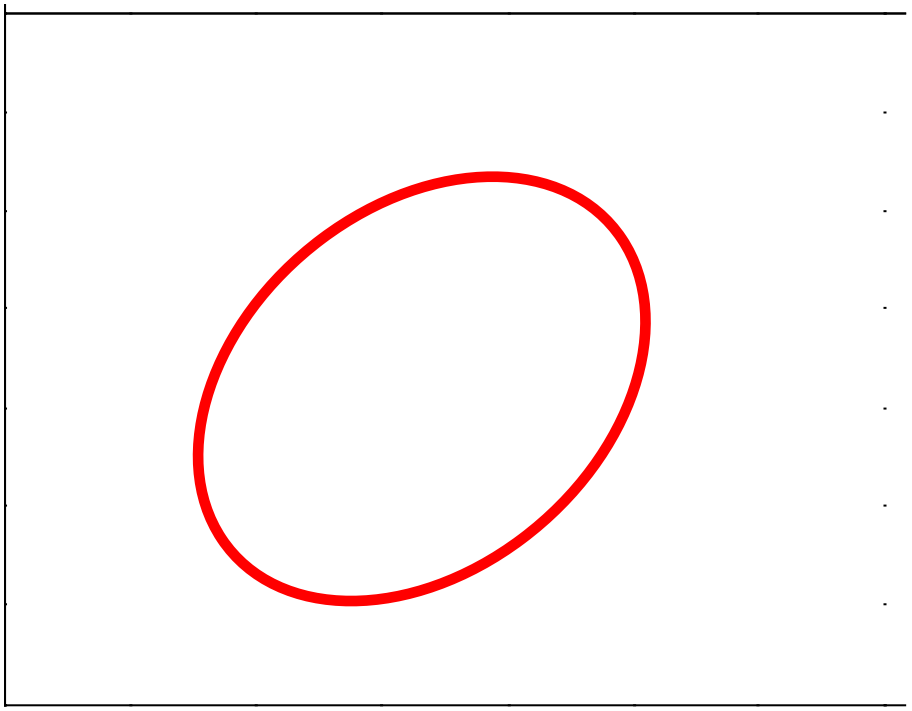
$p(z = c) = \pi_c$ Select a mixture component with probability π

$p(x|z = c) = \mathcal{N}(x \; ; \; \mu_c, \sigma_c)$ Sample from that component's Gaussian



Multivariate Gaussian models

$$\mathcal{N}(\underline{x}\,;\,\underline{\mu},\Sigma) = \frac{1}{(2\pi)^{d/2}}|\Sigma|^{-1/2}\exp\left\{-\frac{1}{2}(\underline{x}-\underline{\mu})^T\Sigma^{-1}(\underline{x}-\underline{\mu})\right\}$$



Maximum Likelihood estimates

$$\hat{\mu} = \frac{1}{N}\sum_i x^{(i)}$$

$$\hat{\Sigma} = \frac{1}{N}\sum_i (x^{(i)}-\hat{\mu})^T(x^{(i)}-\hat{\mu})$$

We'll model each cluster using one of these Gaussian "bells"…

EM Algorithm: E-step

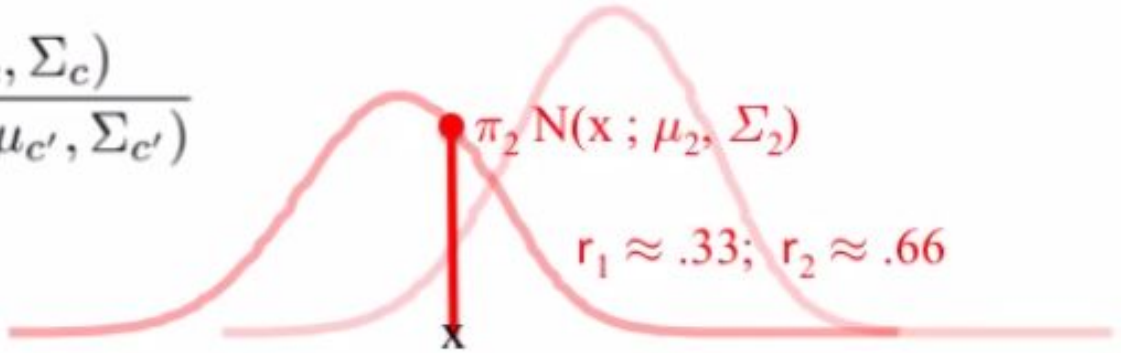
Start with clusters: Mean μ_c, Covariance Σ_c, "size" π_c

E-step ("Expectation")

- For each datum (example) x_i,
- Compute "r_{ic}", the probability that it belongs to cluster c
  - Compute its probability under model c
  - Normalize to sum to one (over clusters c)

responsibility / soft membership

$$r_{ic} = \frac{\pi_c \mathcal{N}(x_i \; ; \; \mu_c, \Sigma_c)}{\sum_{c'} \pi_{c'} \mathcal{N}(x_i \; ; \; \mu_{c'}, \Sigma_{c'})}$$



- If x_i is very likely under the c^{th} Gaussian, it gets high weight
- Denominator just makes r's sum to one

EM Algorithm: M-step

- Start with assignment probabilities r_{ic}
- Update parameters: mean μ_c, Covariance Σ_c, "size" π_c

- M-step ("Maximization")
  - For each cluster (Gaussian) z = c,
  - Update its parameters using the (weighted) data points

$$m_c = \sum_i r_{ic}$$ Total responsibility allocated to cluster c

$$\pi_c = \frac{m_c}{m}$$ Fraction of total assigned to cluster c

$$\mu_c = \frac{1}{m_c} \sum_i r_{ic} x^{(i)}$$ Weighted mean of assigned data

$$\Sigma_c = \frac{1}{m_c} \sum_i r_{ic} (x^{(i)} - \mu_c)^T (x^{(i)} - \mu_c)$$ Weighted covariance of assigned data (use new weighted means here)

Expectation-Maximization

Each step increases the log-likelihood of our model

$$\log p(\underline{X}) = \sum_i \log \left[\sum_c \pi_c \, \mathcal{N}(x_i \; ; \; \mu_c, \Sigma_c) \right]$$

(we won't derive this here, though)

Iterate until convergence

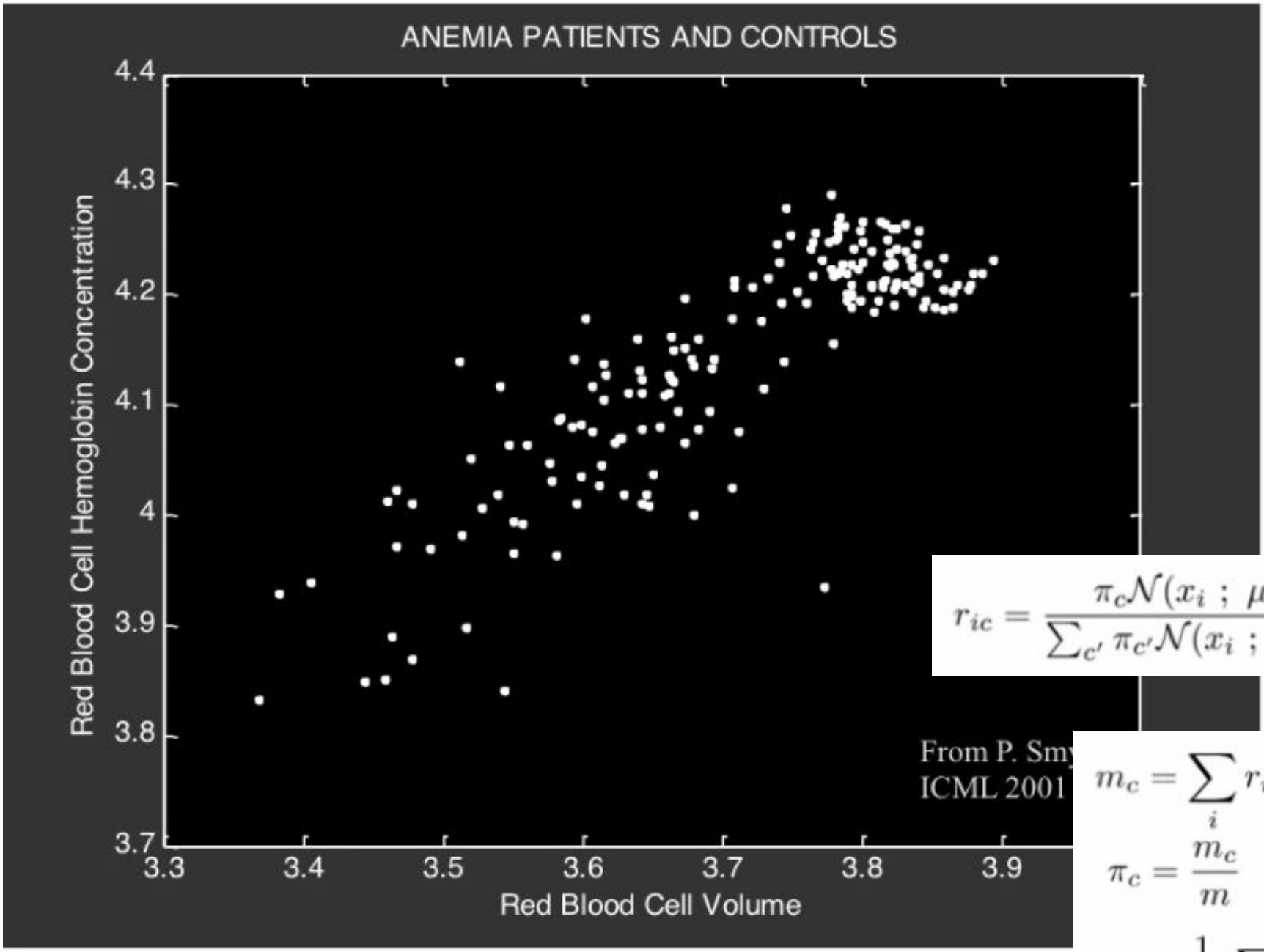
- Convergence guaranteed – another ascent method
- Local optima: initialization often important

What should we do

- If we want to choose a single cluster for an "answer"?
- With new data we didn't see during training?

Choosing the number of clusters

- Can use penalized likelihood of training data (like k-means
- True probability model: can use log-likelihood of test data, log p(x')



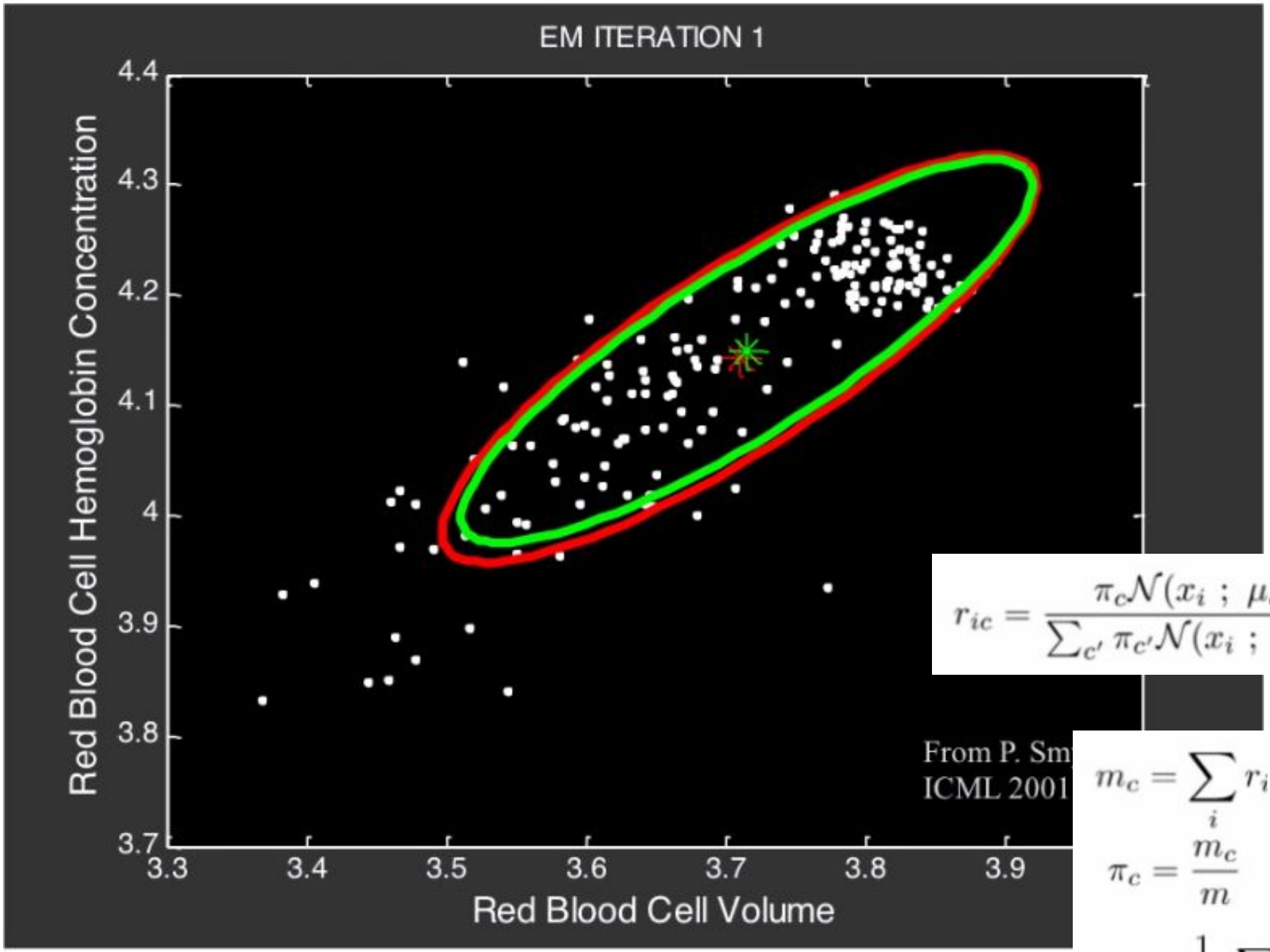
From P. Smy
ICML 2001

$$r_{ic} = \frac{\pi_c \mathcal{N}(x_i \; ; \; \mu_c, \Sigma_c)}{\sum_{c'} \pi_{c'} \mathcal{N}(x_i \; ; \; \mu_{c'}, \Sigma_{c'})}$$

$$m_c = \sum_i r_i$$

$$\pi_c = \frac{m_c}{m}$$

$$\mu_c = \frac{1}{m_c} \sum_i r_{ic} x^{(i)}$$

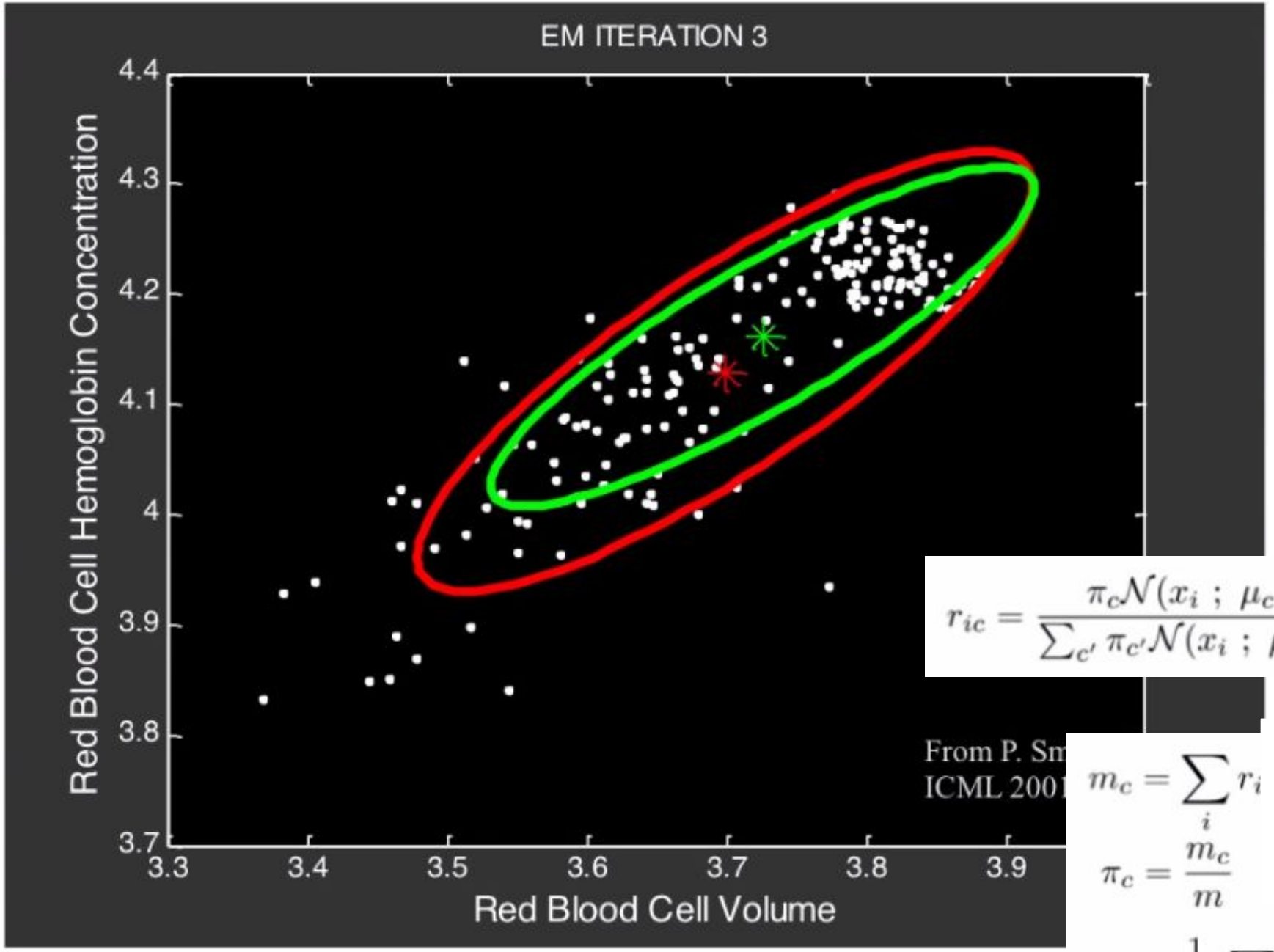


$$r_{ic} = \frac{\pi_c \mathcal{N}(x_i \; ; \; \mu_c, \Sigma_c)}{\sum_{c'} \pi_{c'} \mathcal{N}(x_i \; ; \; \mu_{c'}, \Sigma_{c'})}$$

$$m_c = \sum_i r_i$$

$$\pi_c = \frac{m_c}{m}$$

$$\mu_c = \frac{1}{m_c} \sum_i r_{ic} x^{(i)}$$



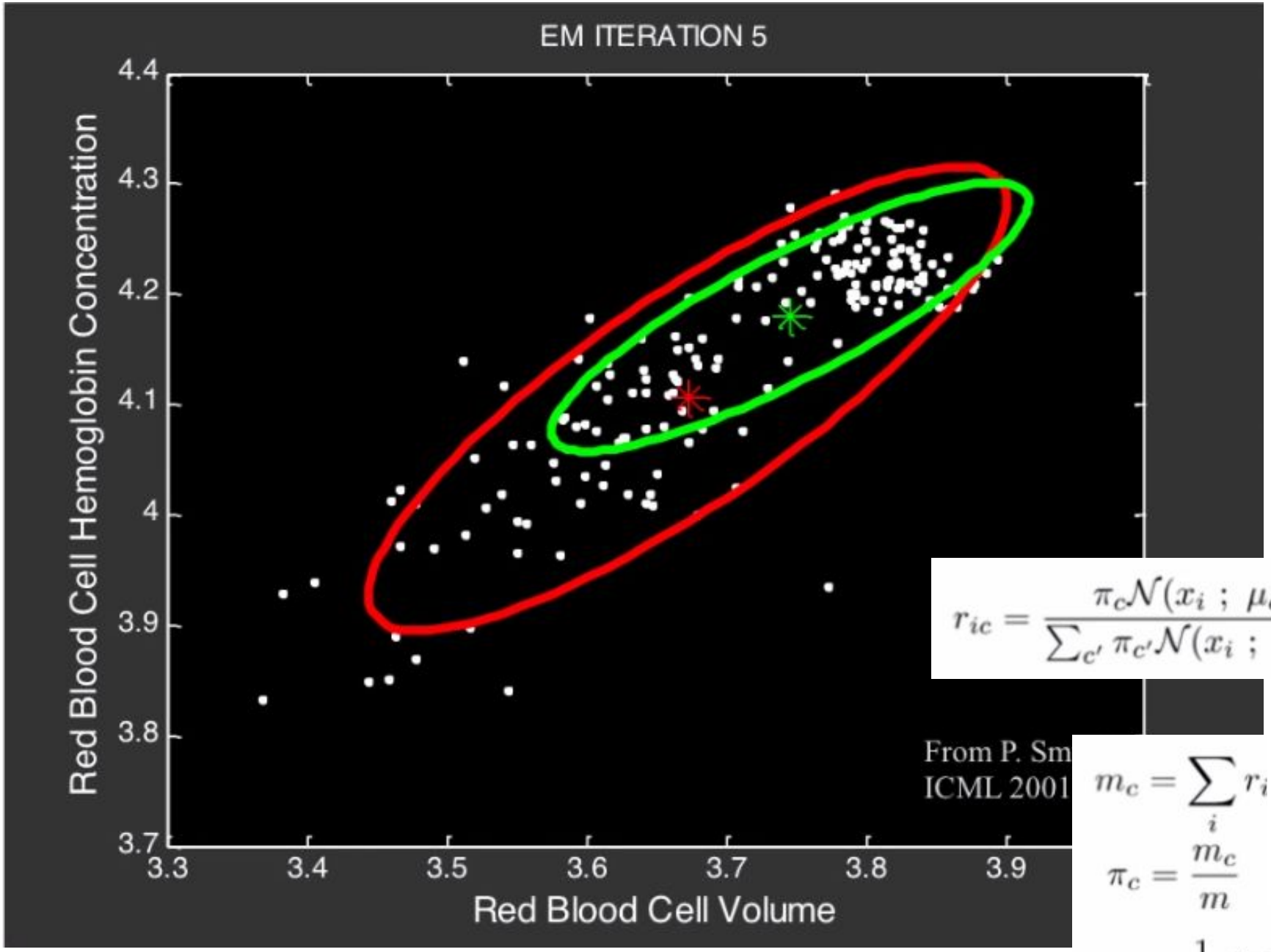
From P. Sm
ICML 200

$$r_{ic} = \frac{\pi_c \mathcal{N}(x_i \; ; \; \mu_c, \Sigma_c)}{\sum_{c'} \pi_{c'} \mathcal{N}(x_i \; ; \; \mu_{c'}, \Sigma_{c'})}$$

$$m_c = \sum_i r_i$$

$$\pi_c = \frac{m_c}{m}$$

$$\mu_c = \frac{1}{m_c} \sum_i r_{ic} x^{(i)}$$



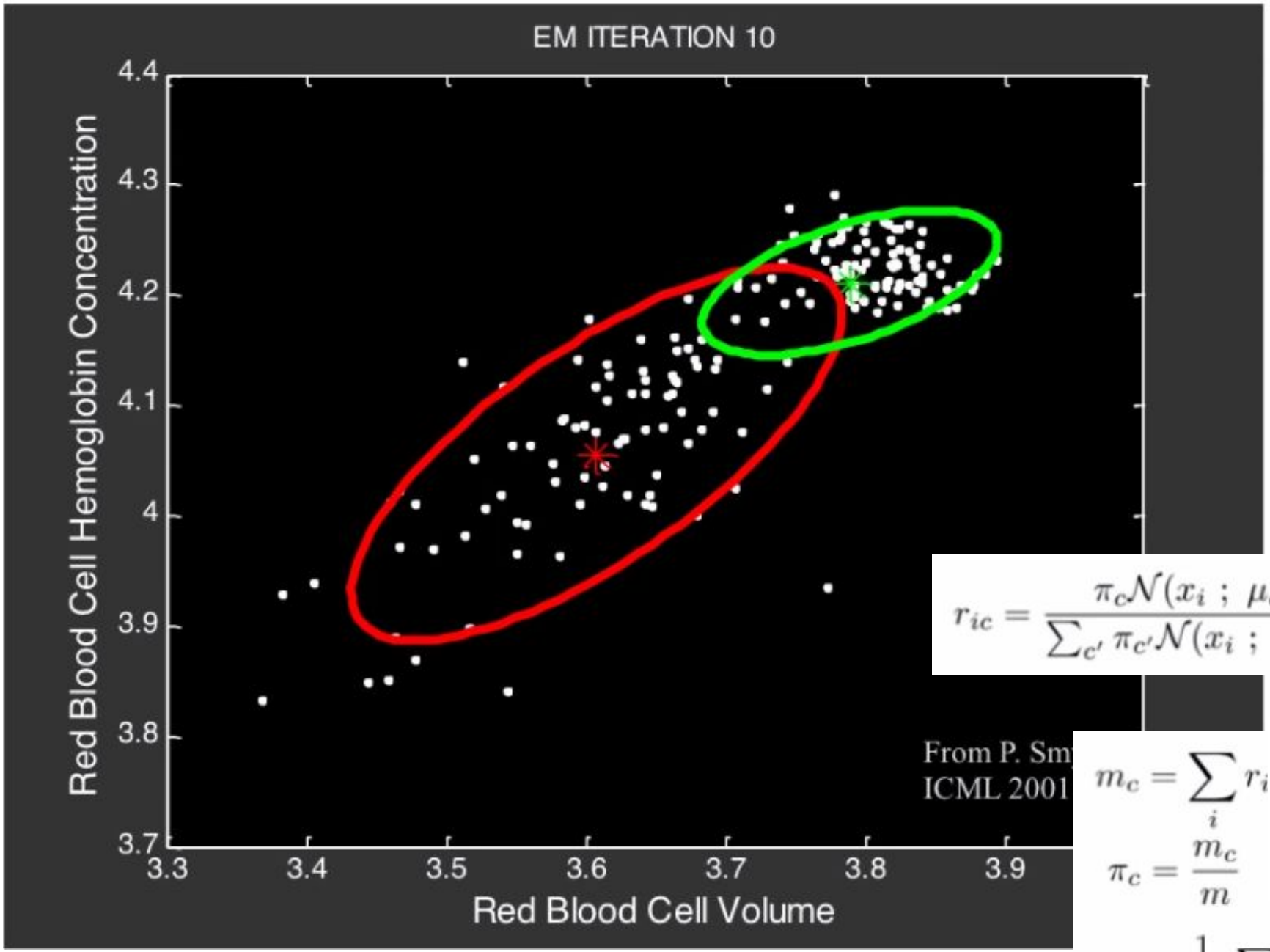
From P. Sm
ICML 2001

$$r_{ic} = \frac{\pi_c \mathcal{N}(x_i \; ; \; \mu_c, \Sigma_c)}{\sum_{c'} \pi_{c'} \mathcal{N}(x_i \; ; \; \mu_{c'}, \Sigma_{c'})}$$

$$m_c = \sum_i r_i$$

$$\pi_c = \frac{m_c}{m}$$

$$\mu_c = \frac{1}{m_c} \sum_i r_{ic} x^{(i)}$$

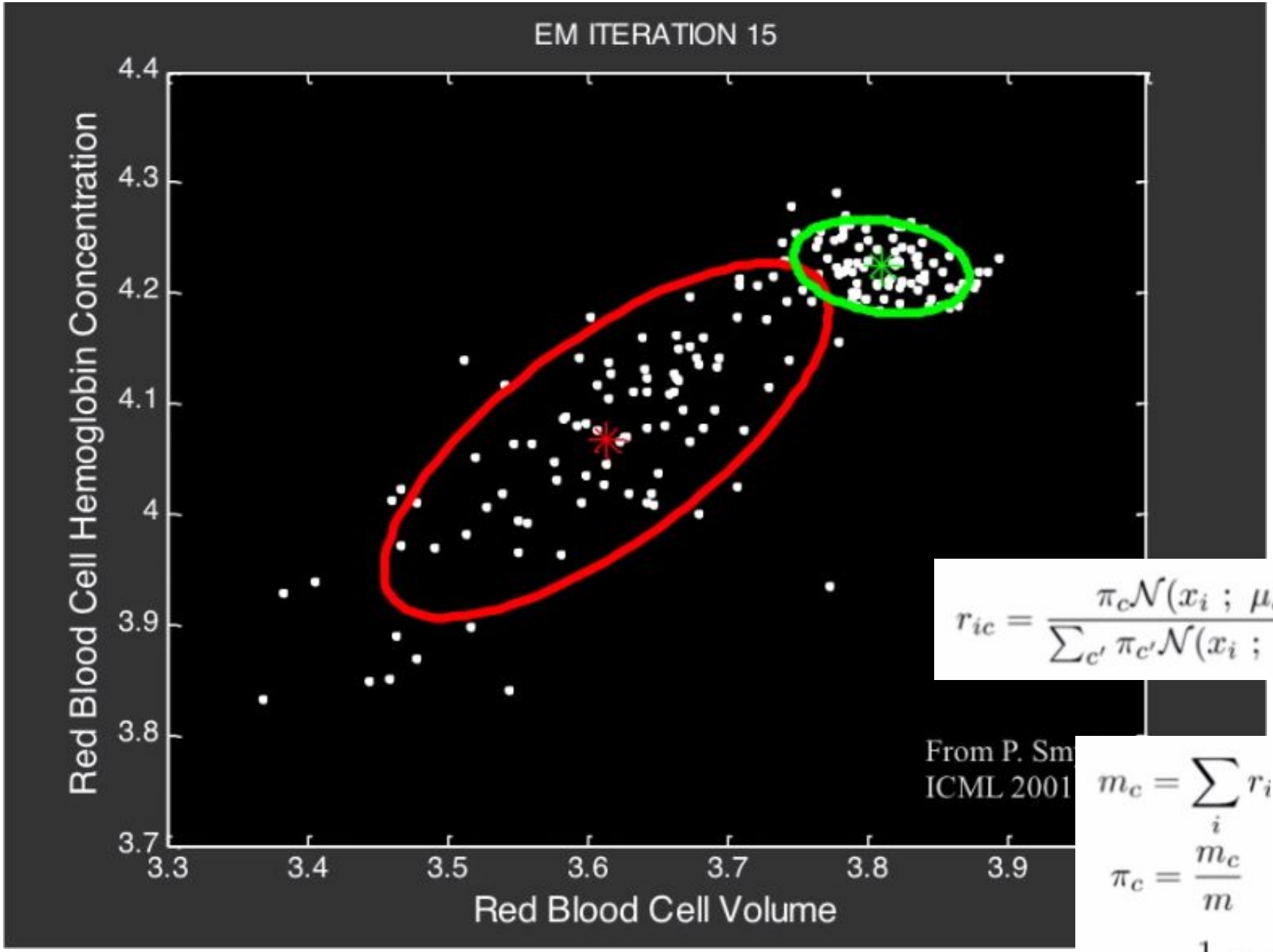


$$r_{ic} = \frac{\pi_c \mathcal{N}(x_i \; ; \; \mu_c, \Sigma_c)}{\sum_{c'} \pi_{c'} \mathcal{N}(x_i \; ; \; \mu_{c'}, \Sigma_{c'})}$$

$$m_c = \sum_i r_i$$

$$\pi_c = \frac{m_c}{m}$$

$$\mu_c = \frac{1}{m_c} \sum_i r_{ic} x^{(i)}$$

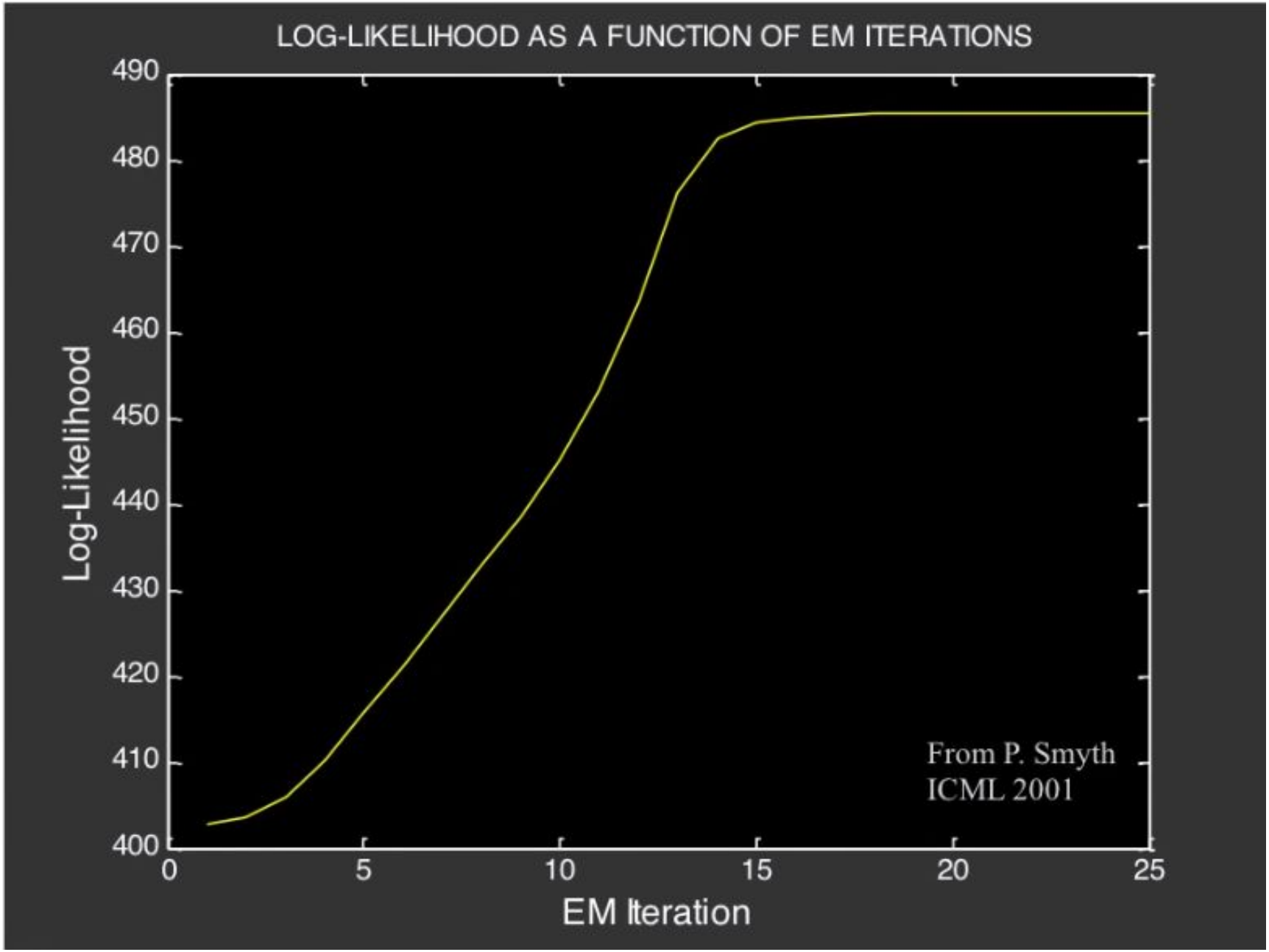


$$r_{ic} = \frac{\pi_c \mathcal{N}(x_i \; ; \; \mu_c, \Sigma_c)}{\sum_{c'} \pi_{c'} \mathcal{N}(x_i \; ; \; \mu_{c'}, \Sigma_{c'})}$$

$$m_c = \sum_i r_i$$

$$\pi_c = \frac{m_c}{m}$$

$$\mu_c = \frac{1}{m_c} \sum_i r_{ic} x^{(i)}$$


LOG-LIKELIHOOD AS A FUNCTION OF EM ITERATIONS
Log-Likelihood
EM Iteration
490
480
470
460
450
440
430
420
410
400
0
5
10
15
20
25
From P. Smyth
ICML 2001