ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Εξόρυξη γνώσης από Βάσεις Δεδομένων και τον Παγκόσμιο Ιστό

Ενότητα # 4: Unsupervised Learning (Clustering)

Διδάσκων: Μιχάλης Βαζιργιάννης

Τμήμα: Προπτυχιακό Πρόγραμμα Σπουδών "Πληροφορικής"

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο

ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

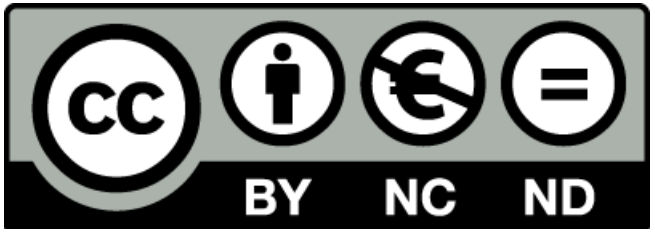
Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Οικονομικό Πανεπιστήμιο Αθηνών**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Οι εικόνες προέρχονται … .



Σκοποί ενότητας

Εισαγωγή και εξοικείωση με τις μεθόδους, Clustering, K-means, Expectation Maximization (EM), Spectral Clustering.

Περιεχόμενα ενότητας

- Clustering
- K-means
- Expectation Maximization (EM)
- Spectral Clustering

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Clustering

Μάθημα: Εξόρυξη γνώσης από Βάσεις Δεδομένων και τον Παγκόσμιο Ιστό

Ενότητα # 4: Unsupervised Learning (Clustering)

Διδάσκων: Μιχάλης Βαζιργιάννης

Τμήμα: Προπτυχιακό Πρόγραμμα Σπουδών “Πληροφορικής”

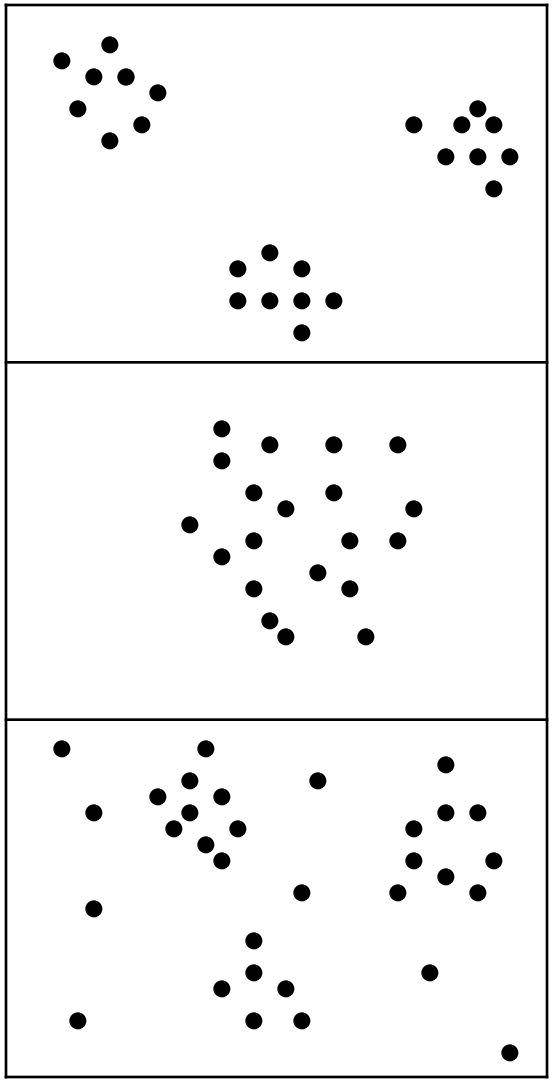
Supervised vs. Unsupervised Learning

- **Unsupervised learning (clustering)**
  - The class labels of training data are unknown
  - Given a set of measurements, observations, etc. establish the existence of clusters in the data
- **Supervised learning (classification)**
  - **Supervision:** The training data (observations, measurements, etc.) are accompanied by labels indicating the class of the observations
  - New data is classified based on the training set
- **Semi-supervised clustering**
  - Learning approaches that use **user input** (i.e. constraints or labeled data)
  - Clusters are defined so that user-constraints are satisfied

Clustering

- “automated detection of group structure in data”
  - Typically: partition N data points into K groups (clusters) such that the points in each group are more similar to each other than to points in other groups
  - descriptive technique (contrast with predictive)
  - for real-valued vectors, clusters can be thought of as clouds of points in p-dimensional space

Clustering



Sometimes easy

Sometimes impossible

and sometimes in between

Why is Clustering useful?

- "Discovery" of new knowledge from data
  - Contrast with supervised classification (where labels are known)
  - Long history in the sciences of categories, taxonomies, etc
  - Can be very useful for summarizing large data sets
    - For large n and/or high dimensionality

- Applications of clustering
  - Discovery of new types of galaxies in astronomical data
  - Clustering of genes with similar expression profiles
  - Cluster pixels in an image into regions of similar intensity
  - Segmentation of customers for an e-commerce store
  - Clustering of documents produced by a search engine
  - …. many more

General Issues in Clustering

- Representation:
  - What types of clusters are we looking for?
- Score:
  - The criterion to compare one clustering to another
- Optimization
  - Generally, finding the optimal clustering is NP-hard
    - Greedy algorithms to optimize score are widely used

- Other issues
  - Distance function, D(x(i),x(j)) critical aspect of clustering, both
    - distance of pairs of objects
    - distance of objects from clusters
  - How is K selected?
  - Different types of data
    - Real-valued versus categorical
    - Attribute-valued vectors vs. n^2 distance matrix

Clustering Methods

- **Partitional algorithms**
  - K-Means, PAM, CLARA, CLARANS [Ng and Han, VLDB 1994]
- **Hierarchical algorithms**
  - CURE [Guha et al, SIGMOD'98], BIRCH [Zhang et al, SIGMOD'96], CHAMELEON [IEEE Computer, 1999]
- **Density based algorithms**
  - DENCLUE [Hinneburg, Keim, KDD'98], DBSCAN [Ester et al, KDD 96]
- **Subspace Clustering**
  - CLIQUE [Agrawal et al, SIGMOD'98], PROCLUS [Agrawal et al, SIGMOD'99], ORCLUS: [Aggarwal, and Yu, SIGMOD' 00], DOC: [Procopiuc, Jones, Agarwal, and Murali, SIGMOD'02]
- **Locally adaptive clustering techniques**
  - LAC
- **Spectral clustering**
  - [Ng, Jordan, Weiss], [Shi/Malik], [Scott/Longuet-Higgins], [Perona/ Freeman]

Partitional Algorithms: Basic Concept

- **Partitional method:**
  - Partition the data set into a set of **k** disjoint partitions (clusters).

- **Problem Definition:**
  - Given an integer **k**, find a partitioning of **k** clusters that optimizes the chosen partitioning criterion

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

K-means

Μάθημα: Εξόρυξη γνώσης από Βάσεις Δεδομένων και τον Παγκόσμιο Ιστό

Ενότητα # 4: Unsupervised Learning (Clustering)

Διδάσκων: Μιχάλης Βαζιργιάννης

Τμήμα: Προπτυχιακό Πρόγραμμα Σπουδών “Πληροφορικής”

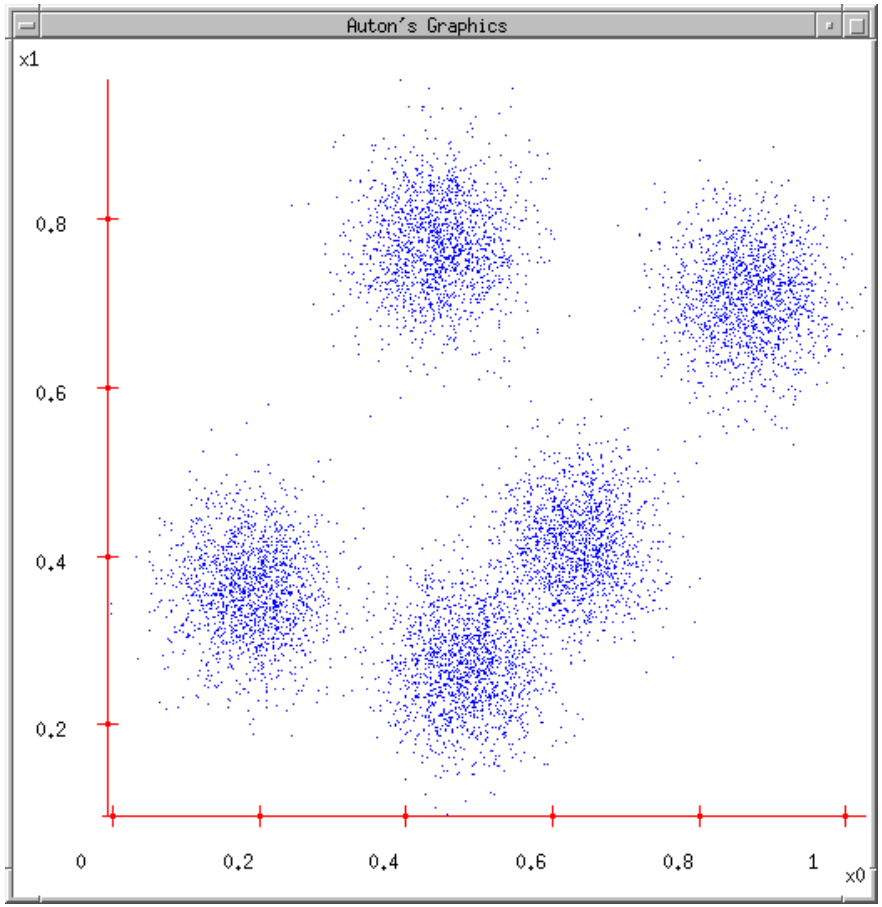
K-means Clustering

- basic idea:
  - Score = wc(C) = sum-of-squares within cluster distance
  - start with randomly chosen cluster centers $c_1 \ldots c_k$
  - repeat until no cluster memberships change:
    - assign each point x to cluster with nearest center
      - find smallest $d(\underline{x},\underline{c}_i)$, over all $\underline{c}_1 \ldots \underline{c}_k$
    - recompute cluster centers over data assigned to them
      - $\underline{c}_i = 1/(n_i) \sum_{x \in Ci} \underline{x}$
- algorithm terminates (finite number of steps)
  - decreases Score(X,C) each iteration membership changes
- converges to local maxima of Score(X,C)
  - not necessarily the global maxima …
  - different initial centers (seeds) can lead to diff local maxs

K-means Complexity

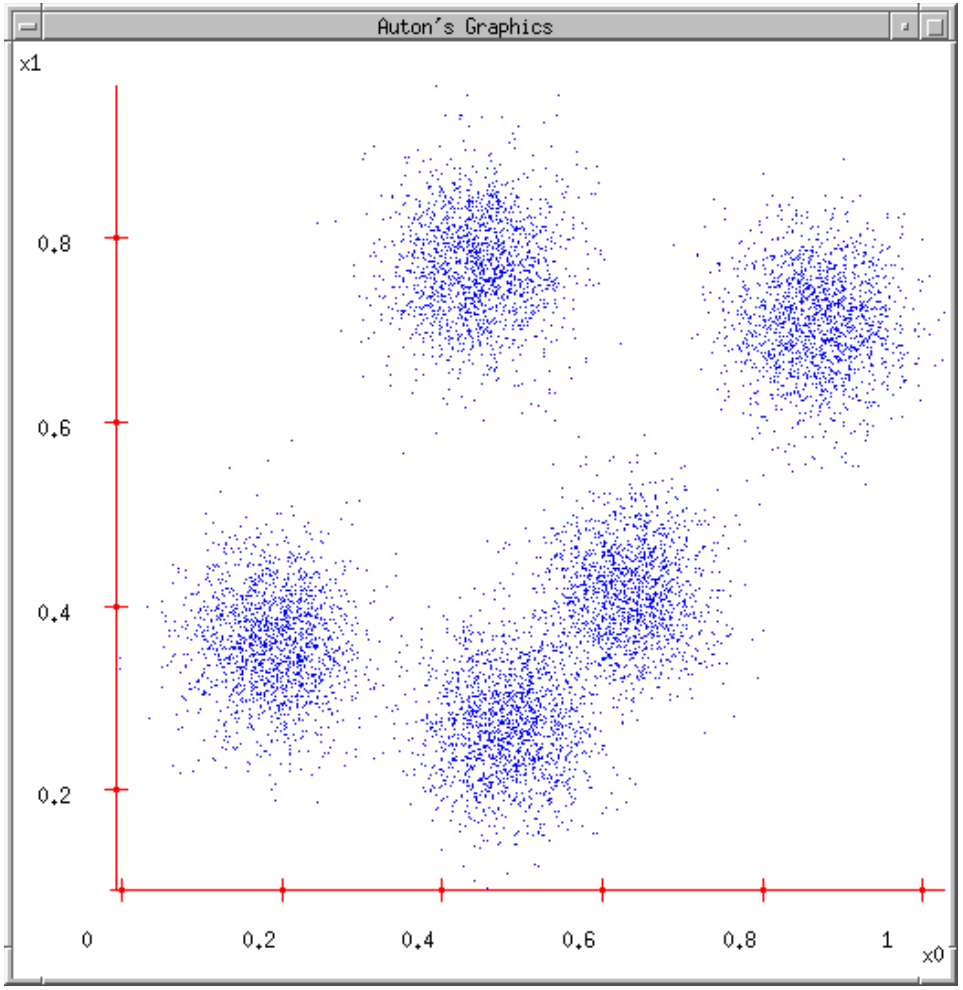
- **time complexity:** O(I e n k) << exhaustive's n^k
  - I = number of interations (steps)
  - e = cost of distance computation (e=p for Euclidian dist)
- **speed-up tricks** (especially useful in early iterations)
  - use nearest x(i)'s as cluster centers instead of mean
    - reuse of cached dists from size n^2 dist mat D (lowers effective "e")
    - k-medoids: use one of x(i)'s as center because mean not defined
  - recompute centers as points reassigned
    - useful for large n (like online neural nets) & more cache efficient
  - PCA: reduce effective "e" and/or fit more of X in RAM
  - "condense": reduce "n" by replace group with prototype
  - even more clever data structures (see work by Andrew Moore, CMU)

K-means example (courtesy of Andrew Moore, CMU)



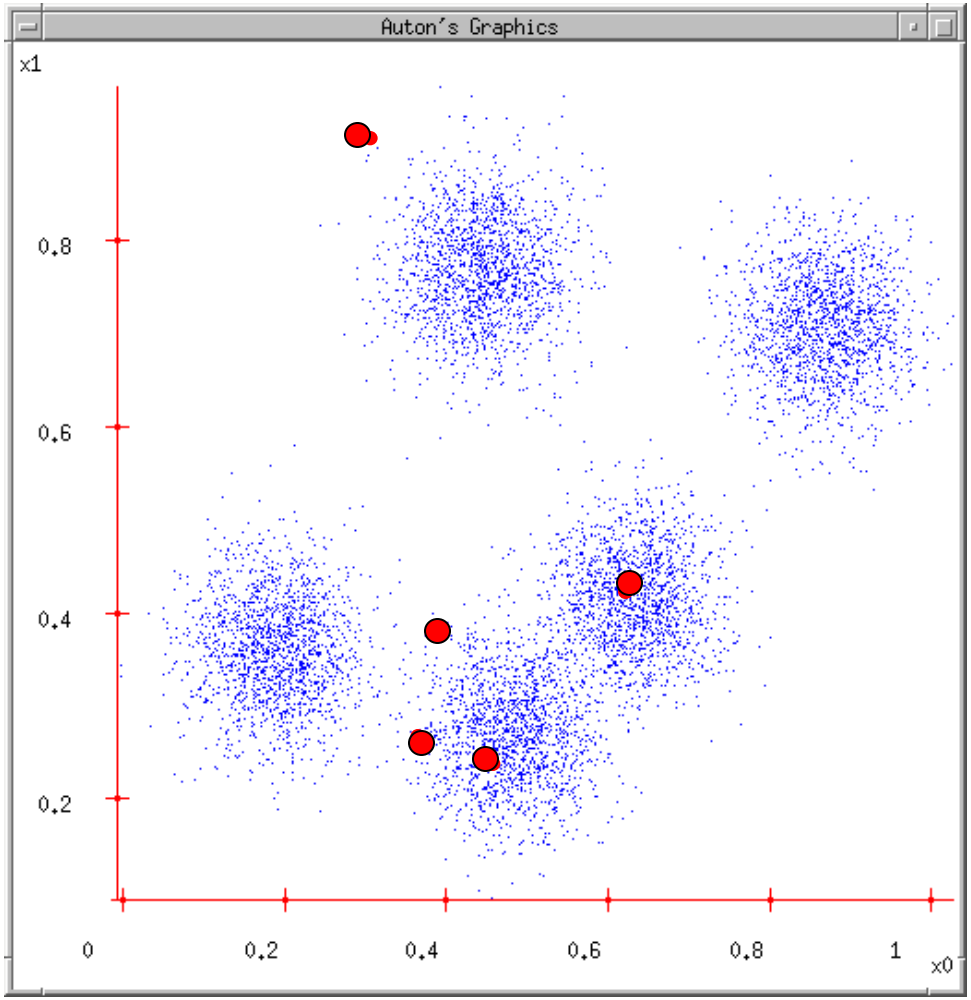
K-means

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



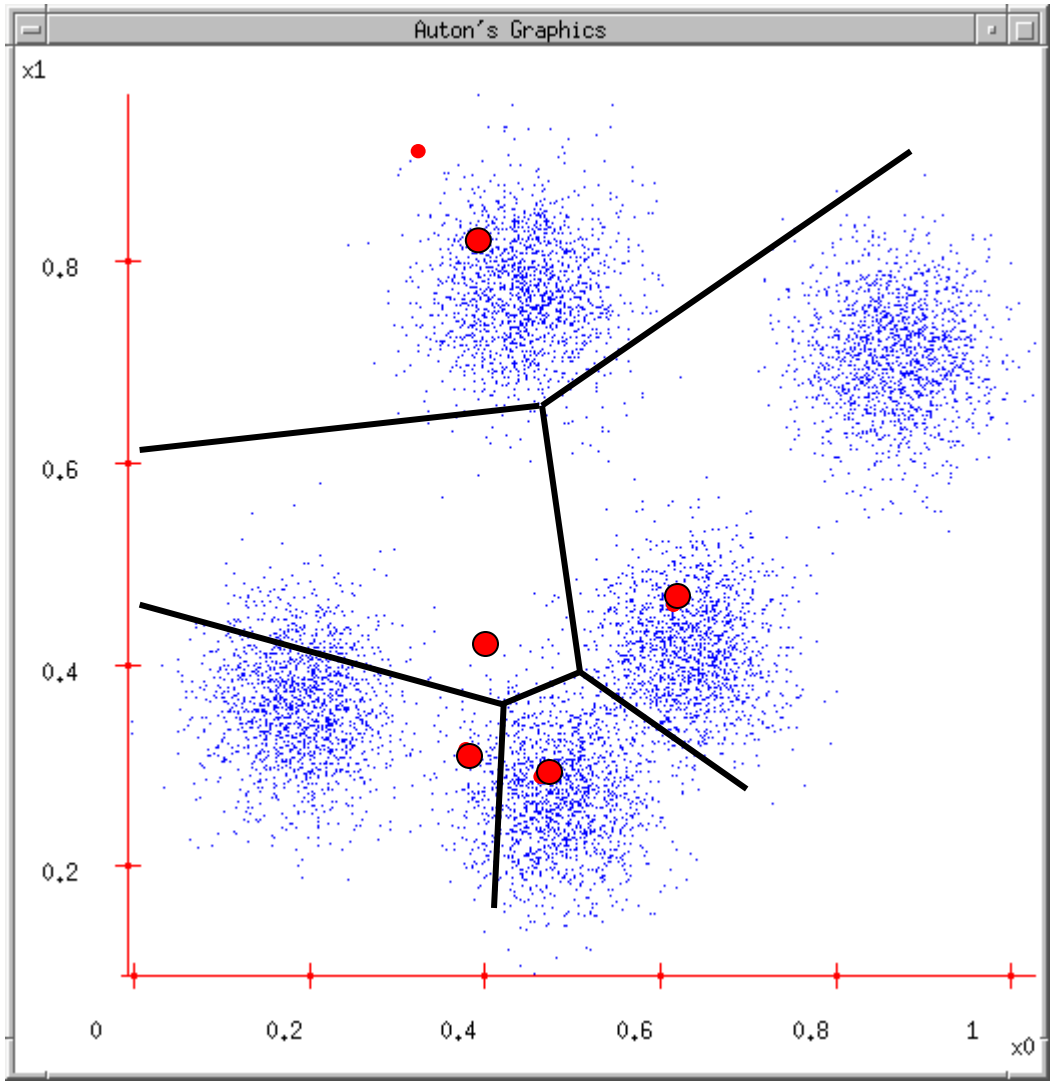
K-means

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



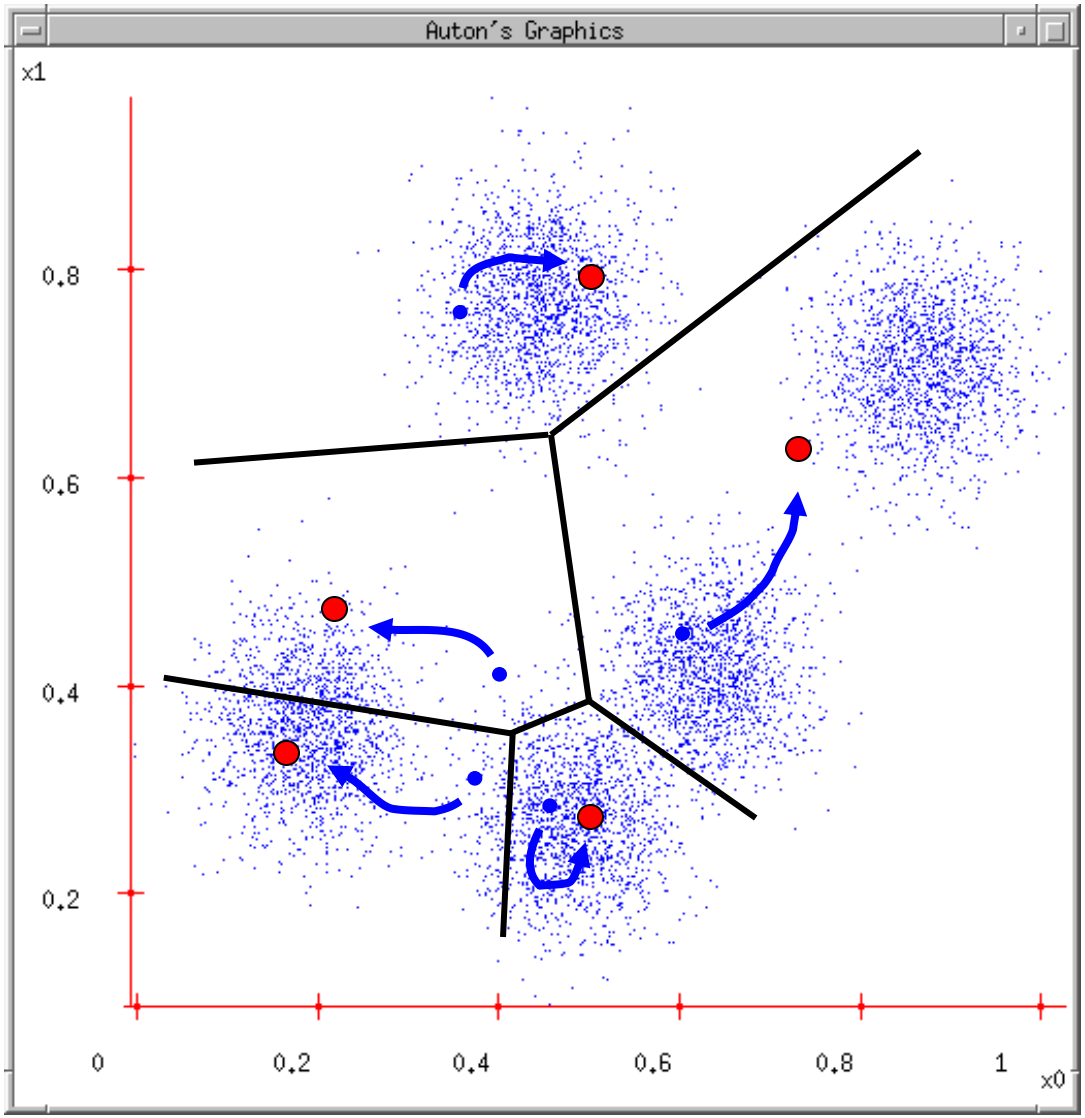
K-means

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



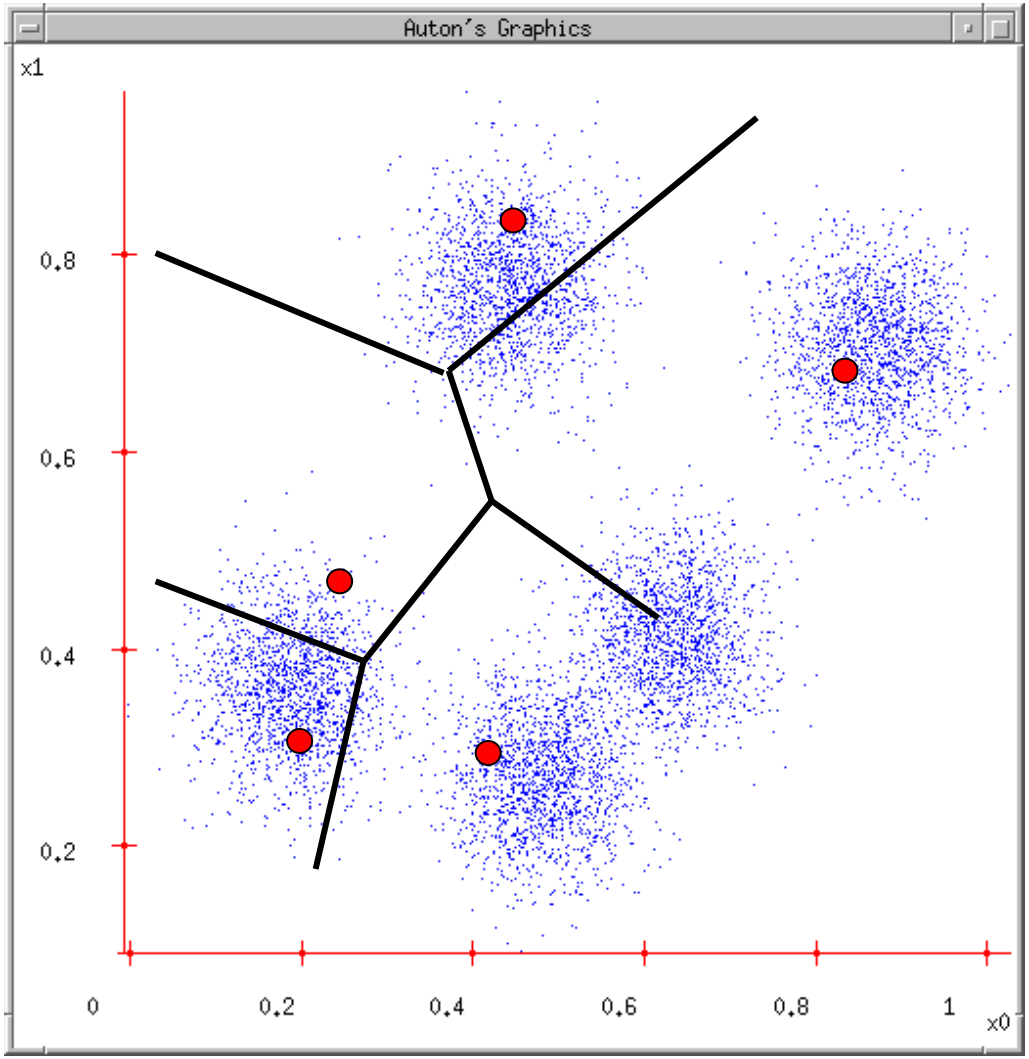
K-means

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



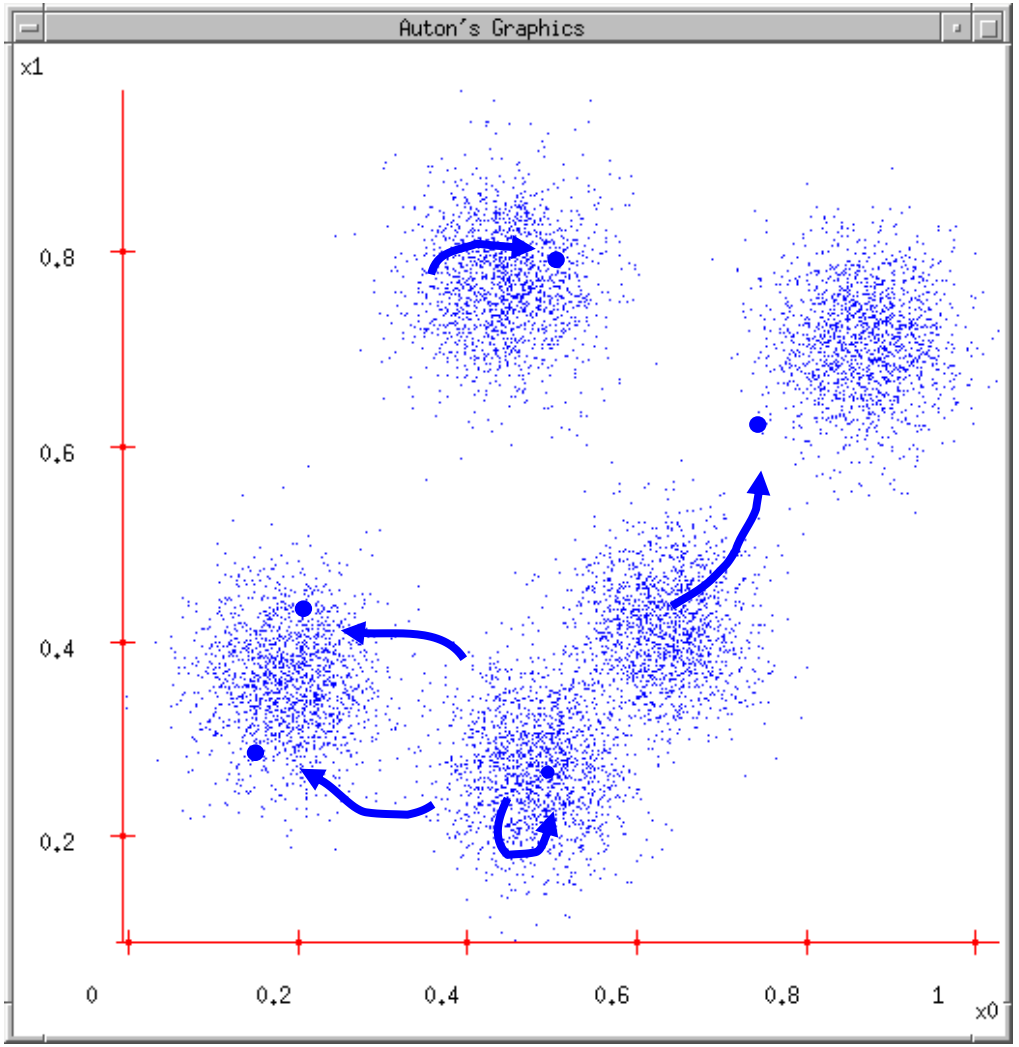
K-means

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. New Centers => new boundaries
6. Repeat until no change!



K-means

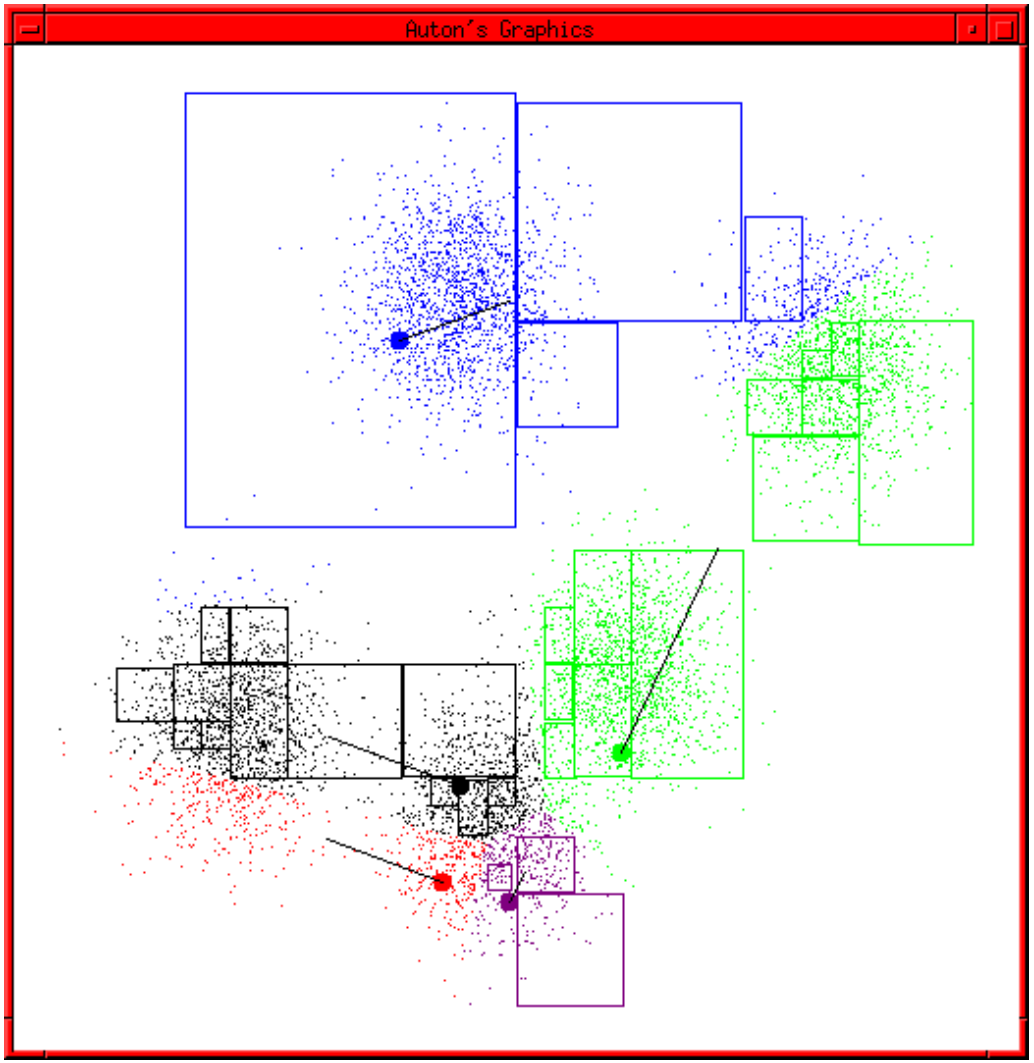
1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns…
5. …and jumps there
6. …Repeat until terminated!



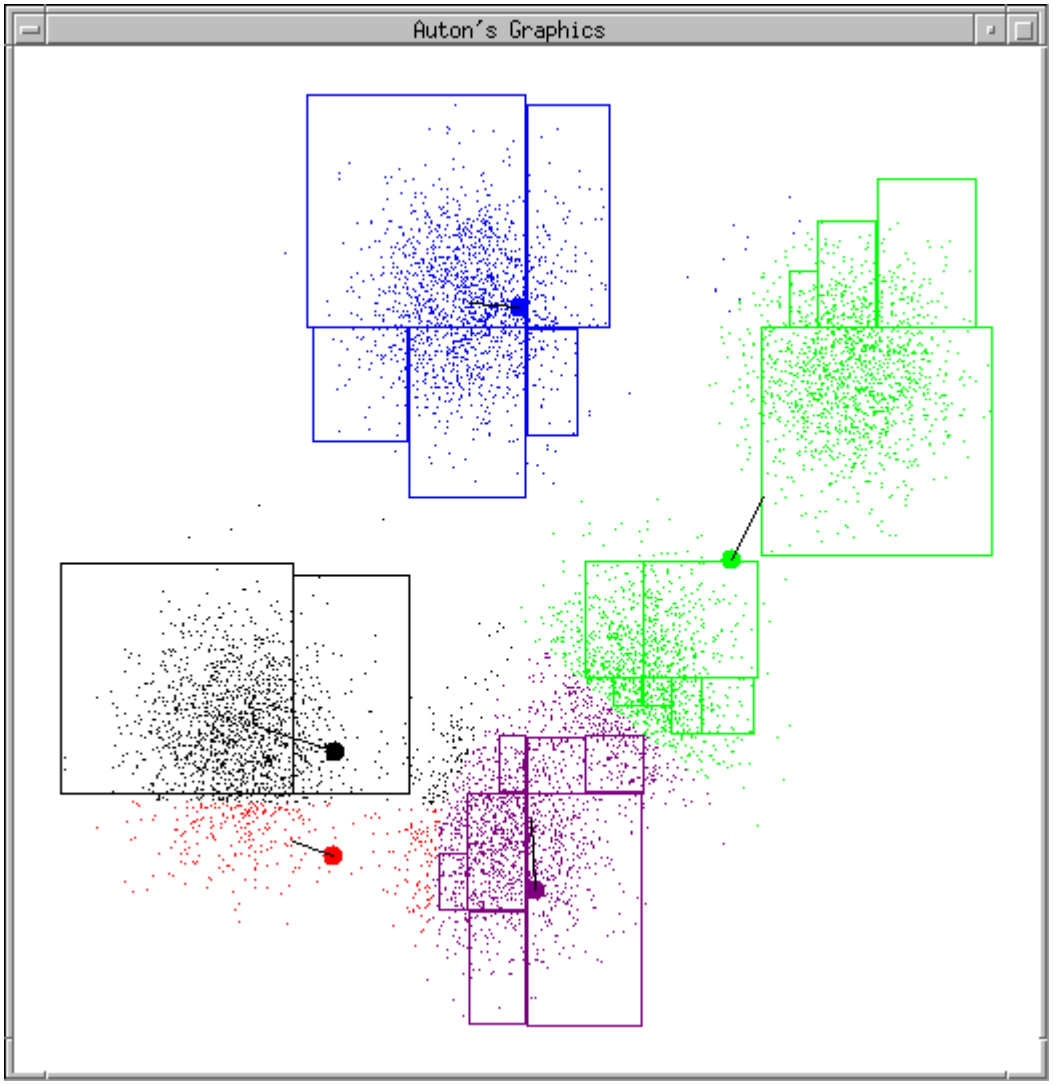
Accelerated Computations

Example generated by Pelleg and Moore's accelerated k-means

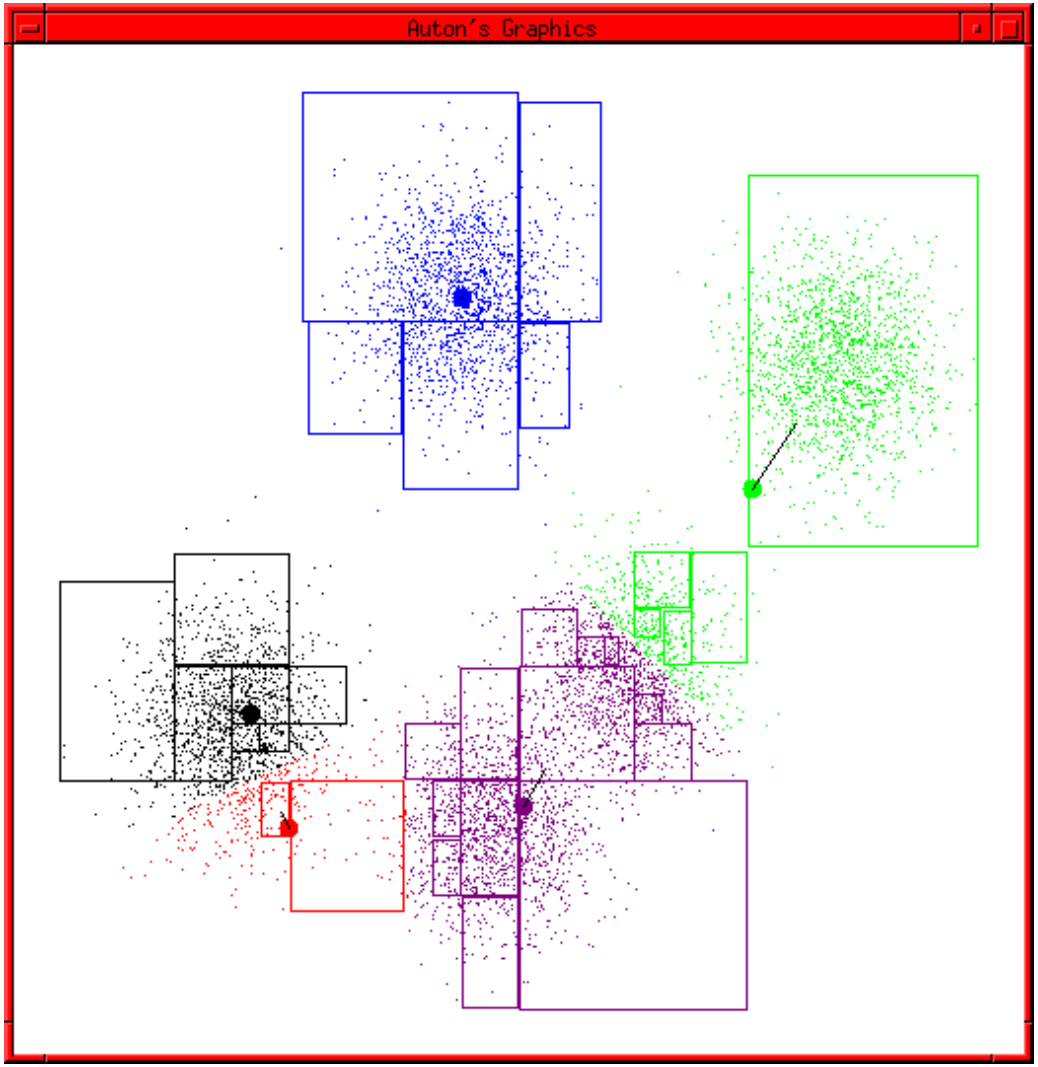
Dan Pelleg and Andrew Moore. Accelerating Exact k-means Algorithms with Geometric Reasoning. Proc. Conference on Knowledge Discovery in Databases 1999, (KDD99) (available on www.autonlab.org/pap.html*)*



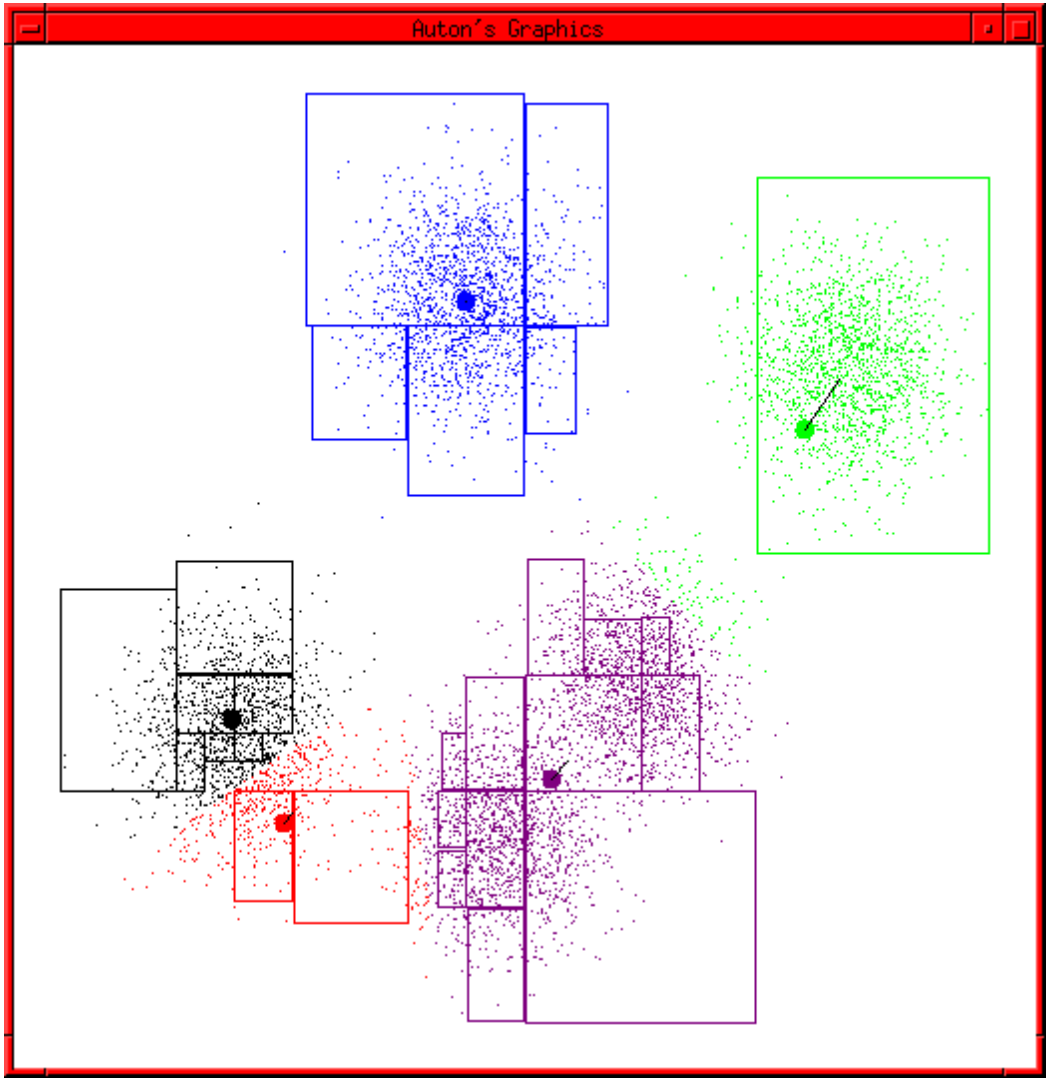
K-means continues…



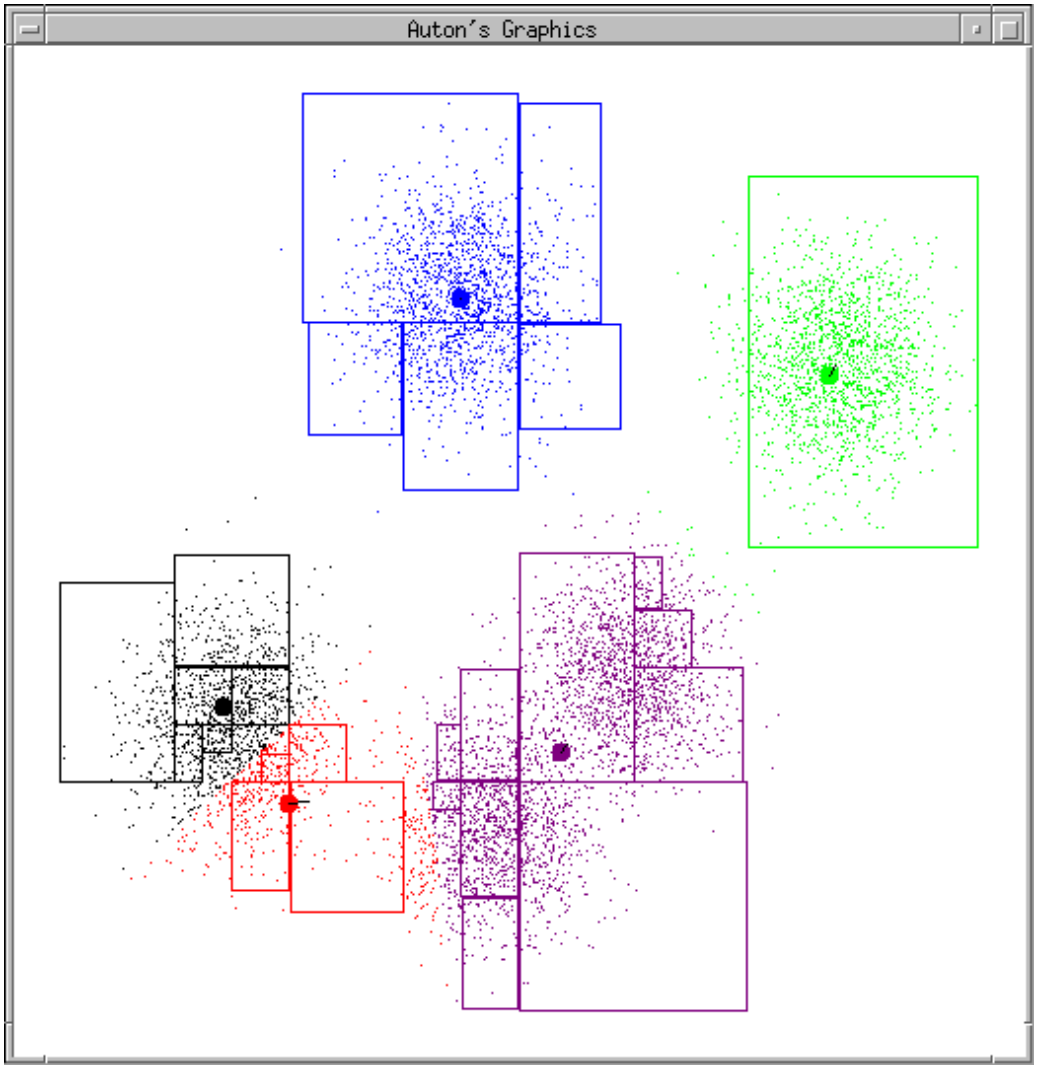
K-means continues…



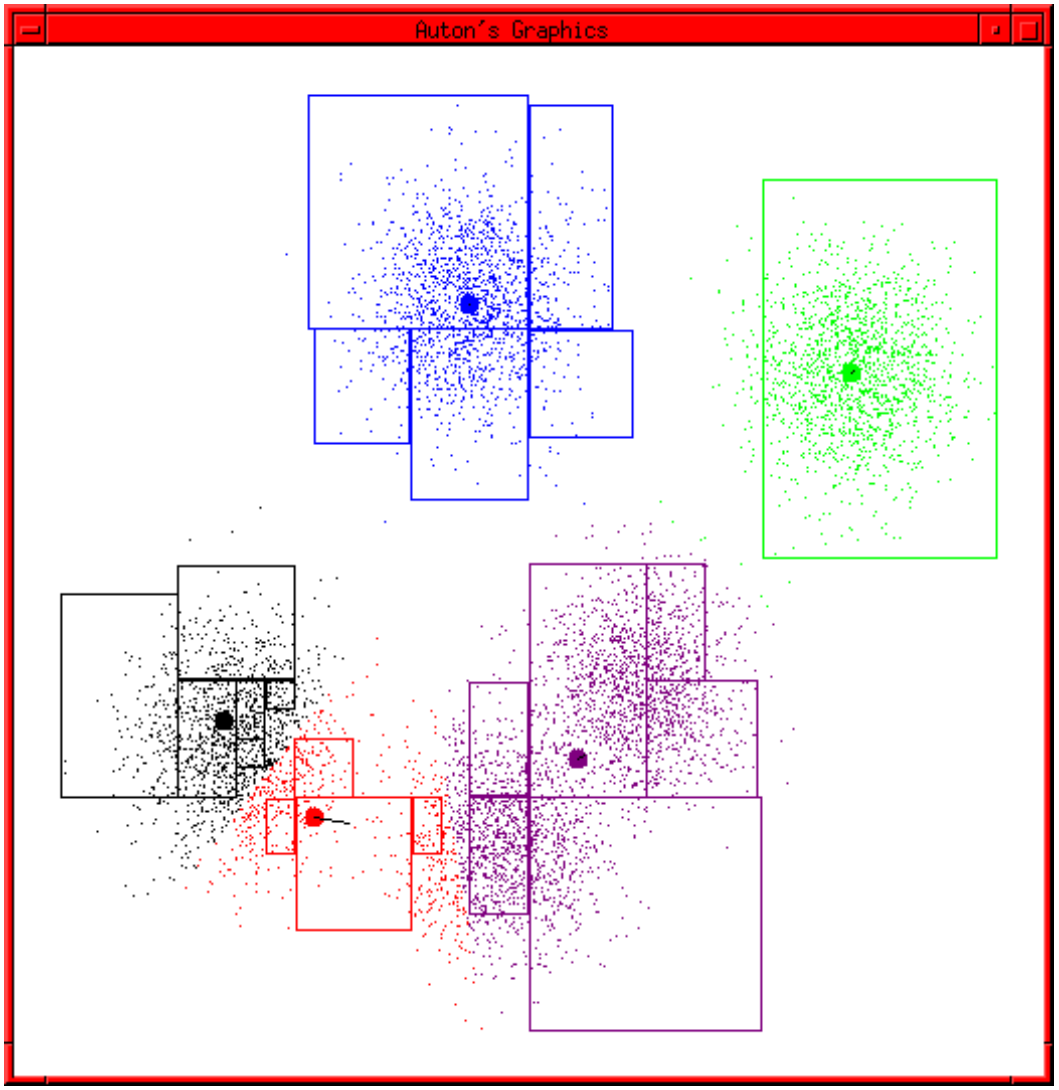
K-means continues…



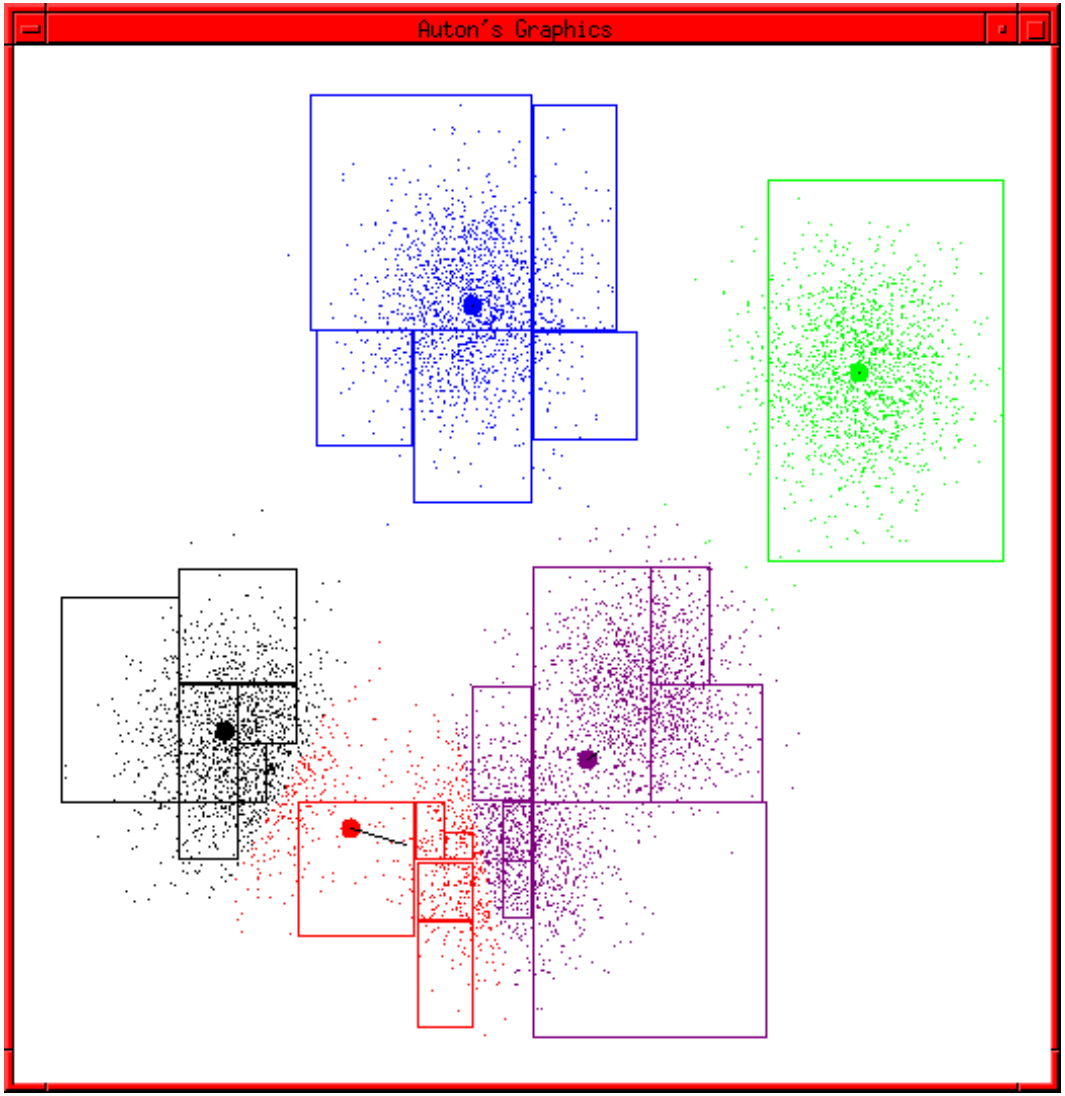
K-means continues…



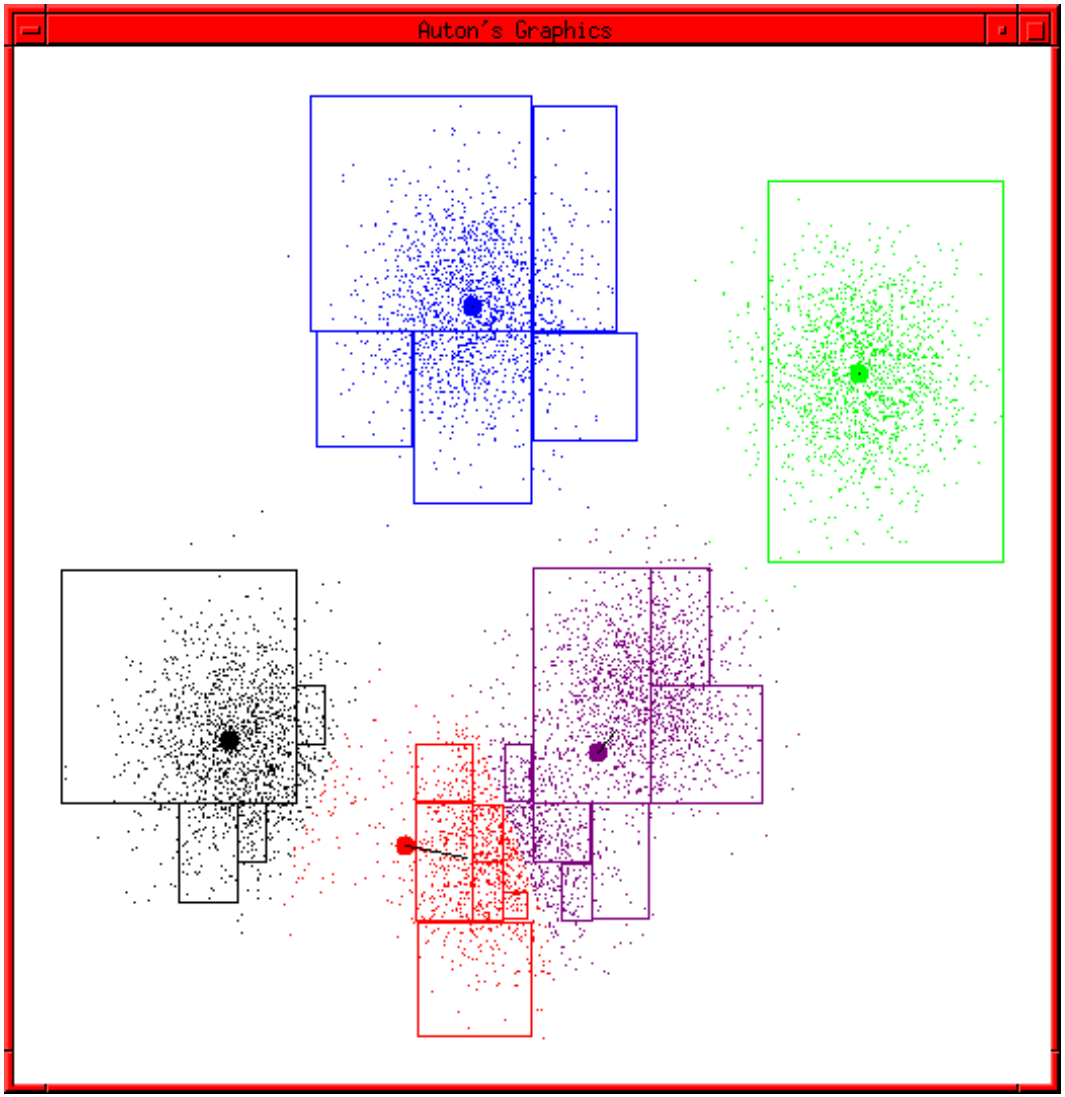
K-means continues…



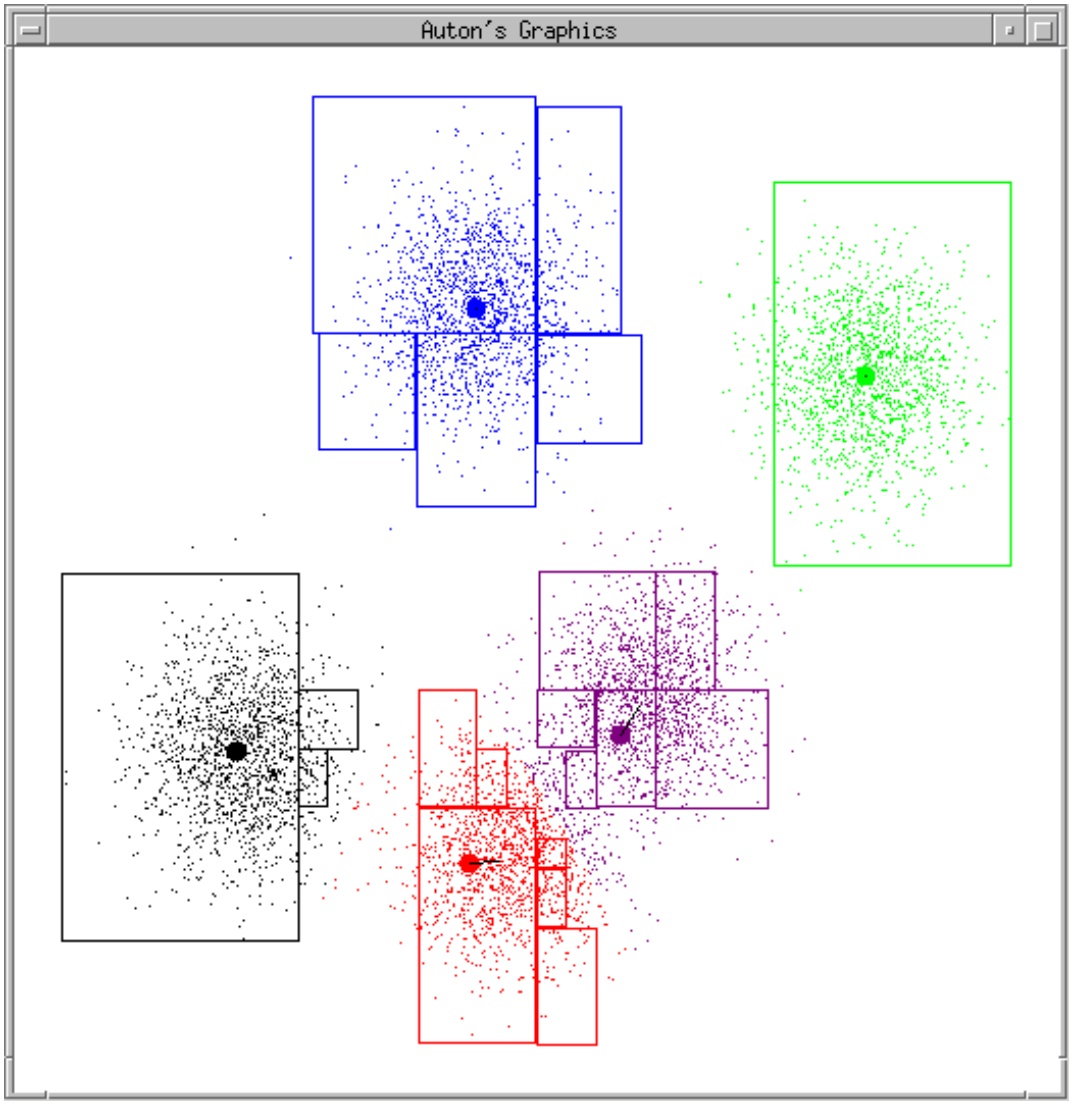
K-means continues…



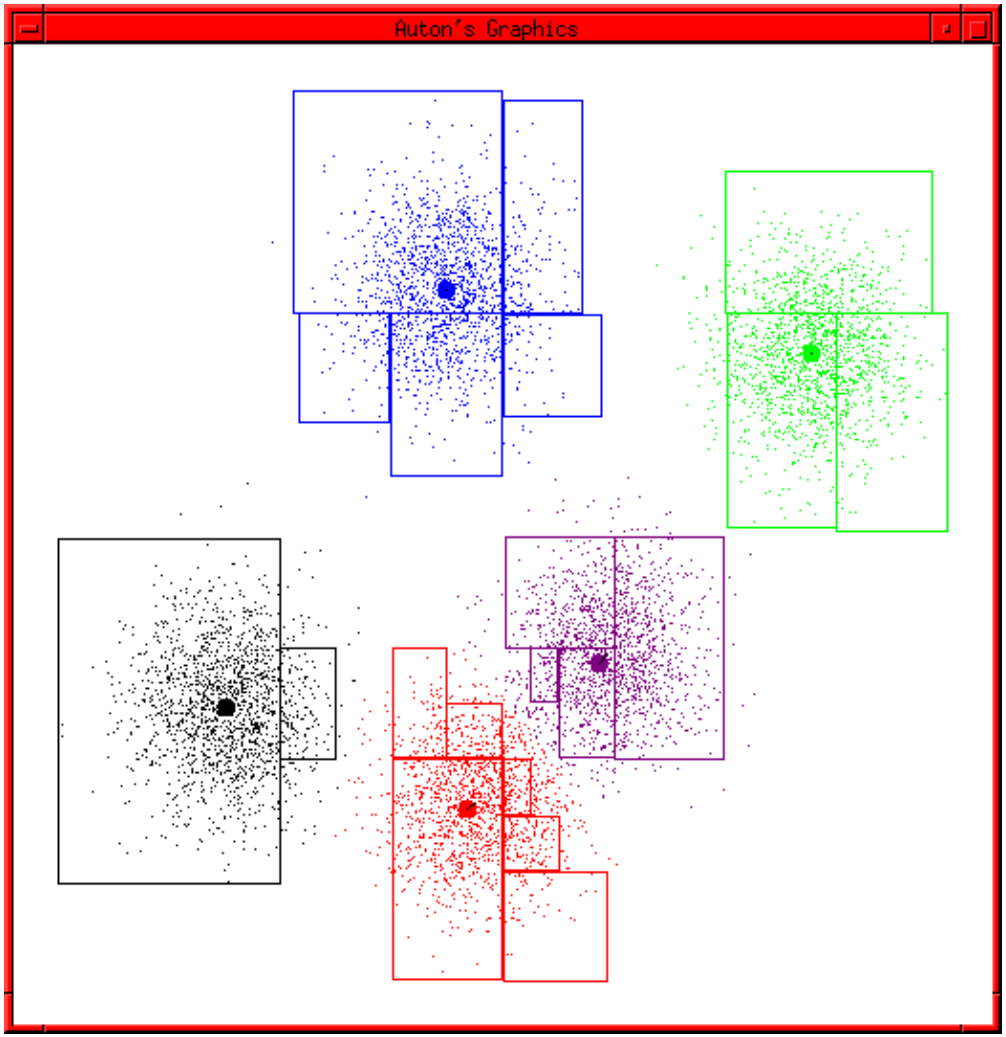
K-means continues…



K-means continues…



K-means terminates



Image

Clusters on color

K-means clustering of RGB (3 value) pixel color intensities, K = 11 segments
(courtesy of David Forsyth, UC Berkeley)

Issues in K-means clustering

- Simple, but useful
  - tends to select compact “isotropic” cluster shapes
  - can be useful for initializing more complex methods
  - many algorithmic variations on the basic theme
- Choice of distance measure
  - Euclidean distance
  - Weighted Euclidean distance
  - Many others possible
- Selection of K
  - “screen diagram” - plot SSE versus K, look for knee
    - Limitation: may not be any clear K value

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Expectation Maximization (EM)

Μάθημα: Εξόρυξη γνώσης από Βάσεις Δεδομένων και τον Παγκόσμιο Ιστό

Ενότητα # 4: Unsupervised Learning (Clustering)

Διδάσκων: Μιχάλης Βαζιργιάννης

Τμήμα: Προπτυχιακό Πρόγραμμα Σπουδών “Πληροφορικής”

Expectation Maximization

- (EM) algorithm is an iterative method for finding maximum likelihood estimates of parameters in statistical models that depend on unobserved latent variables.
- Assume *X* the data observed - We assume the data are produced by *K* different classes/processes represented by respective weights there fore w_k

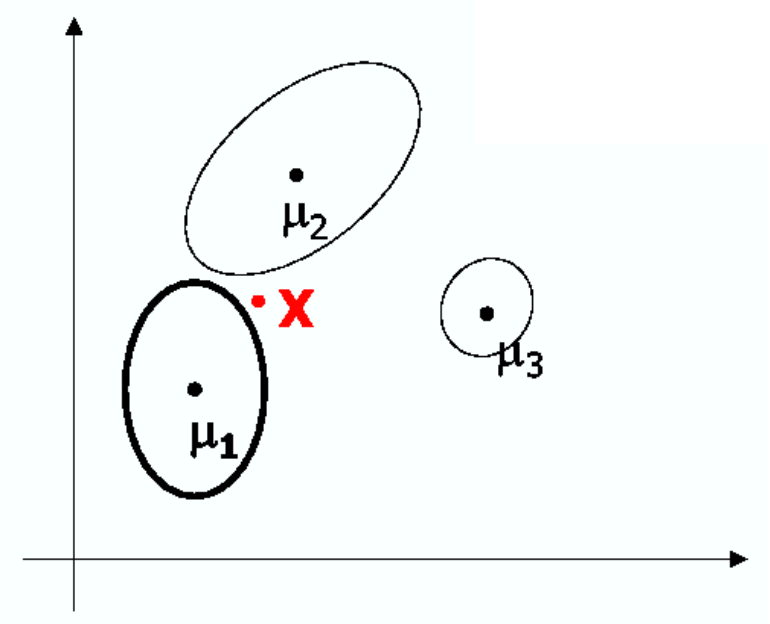
$$f(x) = \sum_k w_k f_k(x|\theta_k)$$

Gaussian Mixture Models (GMM)

- Assume the the components are normal distrubutions $N(\mu_k, \Sigma_k)$
  - often assume diagonal covariance: $\Sigma_{jj} = \sigma_j^2$, $\Sigma_{i \neq j} = 0$
  - or sometimes even simpler: $\Sigma_{jj} = \sigma^2$, $\Sigma_{i \neq j} = 0$

- $f(x) = \sum_{k=1 \ldots K} w_k \, f_k(x; \theta_k)$ with $\theta_k = <\mu_k , \Sigma_k>$ or $<\mu_k , \sigma_k>$

- <u>generative</u> model:
  - - randomly choose a component
  - selected with probability w_k
  - - generate $x \sim N(\mu_k, \sigma_k)$
  - - note: μ_k & σ_k both d-dim vectors



Learning Mixture Models from Data

- **Score function** Log-likelihood $L(\theta)$
  - $L(\theta) = log\ p(X|\theta) = log\ \sum_H p(X,H|\theta)$
  - H = hidden variables (cluster memberships of each x)
  - $L(\theta)$ cannot be optimized directly

- **EM Procedure**
  - General technique for maximizing log-likelihood with missing data
  - For mixtures
    - *E-step*: compute "memberships" $p(k \mid x) = w_k\ f_k(x;\theta_k) / f(x)$
    - *M-step*: pick a new θ to max expected data log-likelihood
    - Iterate: guaranteed to climb to (local) maximum of $L(\theta)$

Expectation maximization (EM)

The Expectation-maximization algorithm computes missing memberships of data points in a chosen distribution model.

- **Expectation step**
  - initial guesses for the parameters in our mixture model,
  - compute "partial membership" of each data point in each constituent distribution.
  - By calculating expectation for the membership variables of each data point.
- **Example.**
- Data set resulting from a sum of two Gaussian distributions.

$$P(x_i) = (1 - f)\mathrm{N}(x_i \mid \mu_1, \sigma) + f\mathrm{N}(x_i \mid \mu_2, \sigma)$$

- ***f*** is the mixing coefficient in (0,1], assume **σ** is known and constant.
- For each data point *i*, compute a membership value for each of the two Gaussians

$$y_{1,i}(x_i) = \frac{(1 - f)\mathrm{N}(x_i \mid \mu_1, \sigma)}{(1 - f)\mathrm{N}(x_i \mid \mu_1, \sigma) + f\mathrm{N}(x_i \mid \mu_2, \sigma)}$$

  - and similarly for $y_{2,i}$

Expectation maximization (EM)

- **The maximization step**
- With expectation values for group membership
- - Re-compute estimates of distribution parameters.

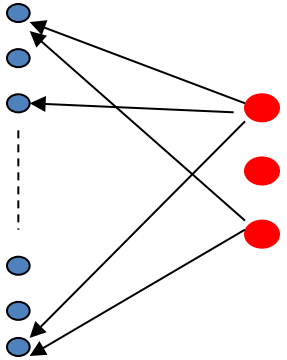
$$\begin{cases} f = \frac{\sum_i y_{i,2}}{N} \\ \\ \mu_1 = \frac{\sum_i y_{i,1} x_i}{\sum_i y_{i,1}} \end{cases}$$

  - *N* is the total number of data points.
- - back to the Expectation step: Re-compute new membership values.
- - repeated until change in the mixture model parameters below threshold

The EM Step

E step: Compute memberships $p(x_i/\vartheta_k)$

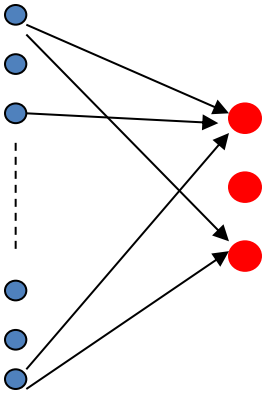
n data points



Current K clusters and parameters

M step: Compute q_ι, given n data points and memberships

n data points

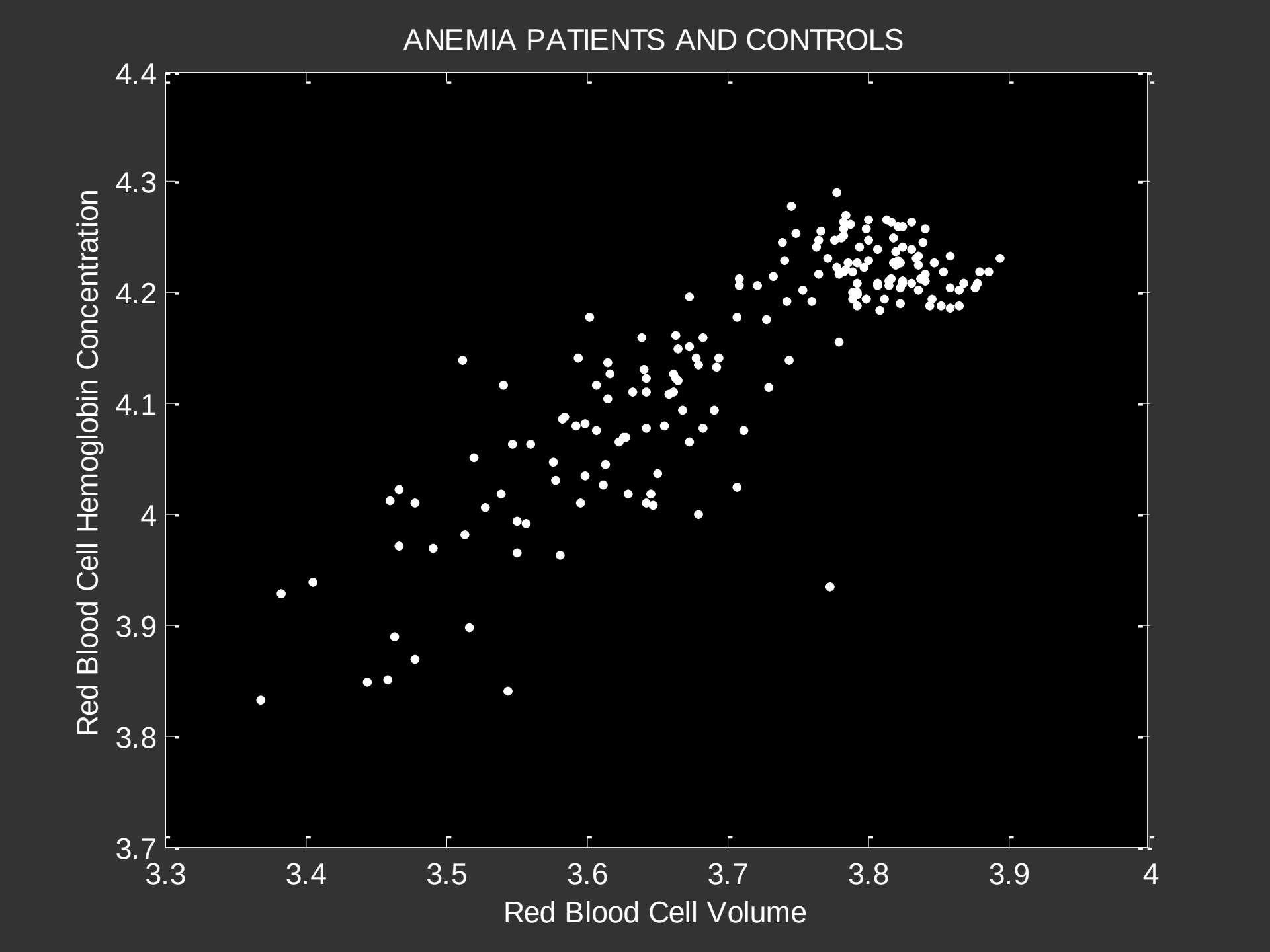


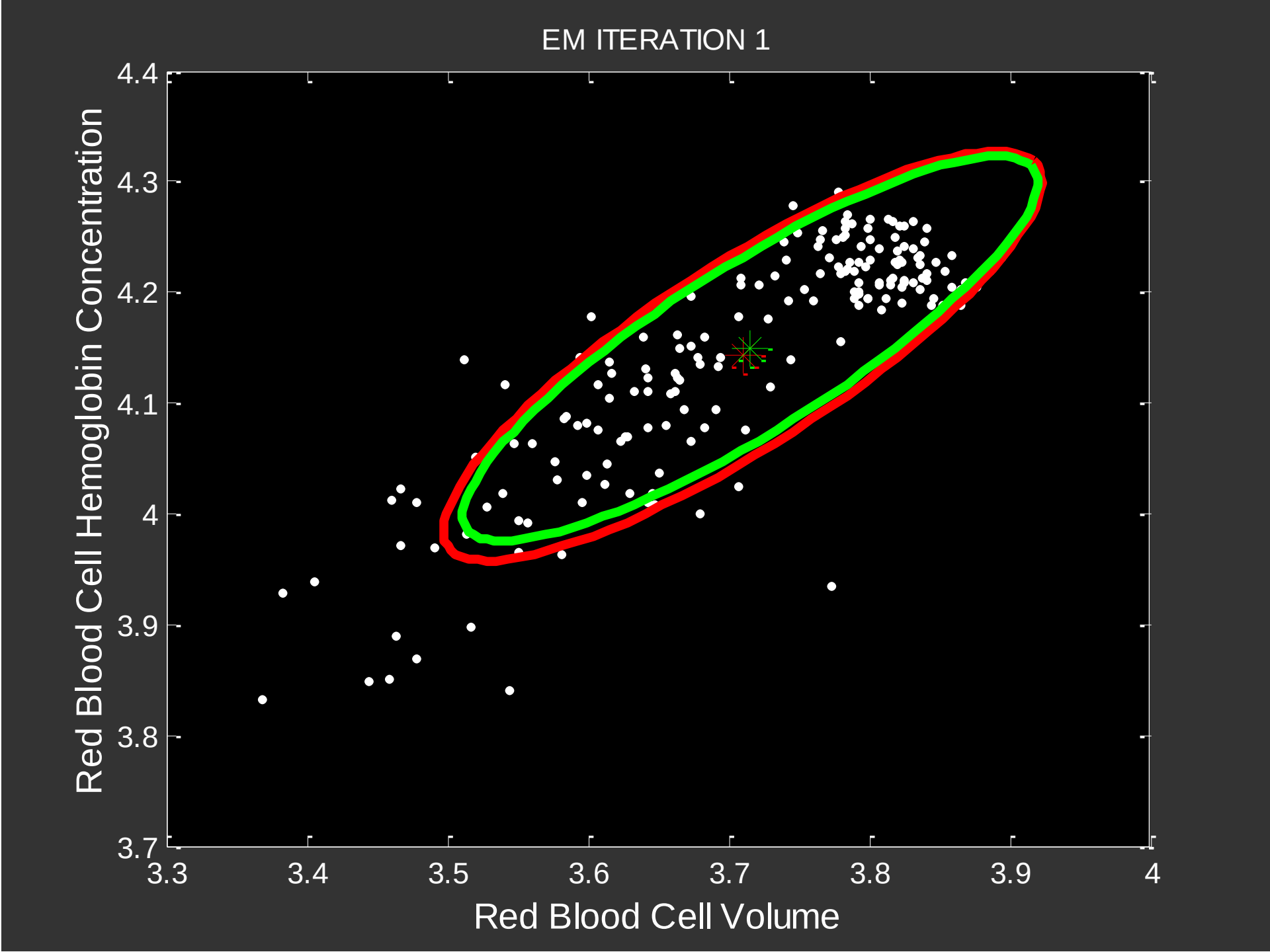
New parameters for the K clusters

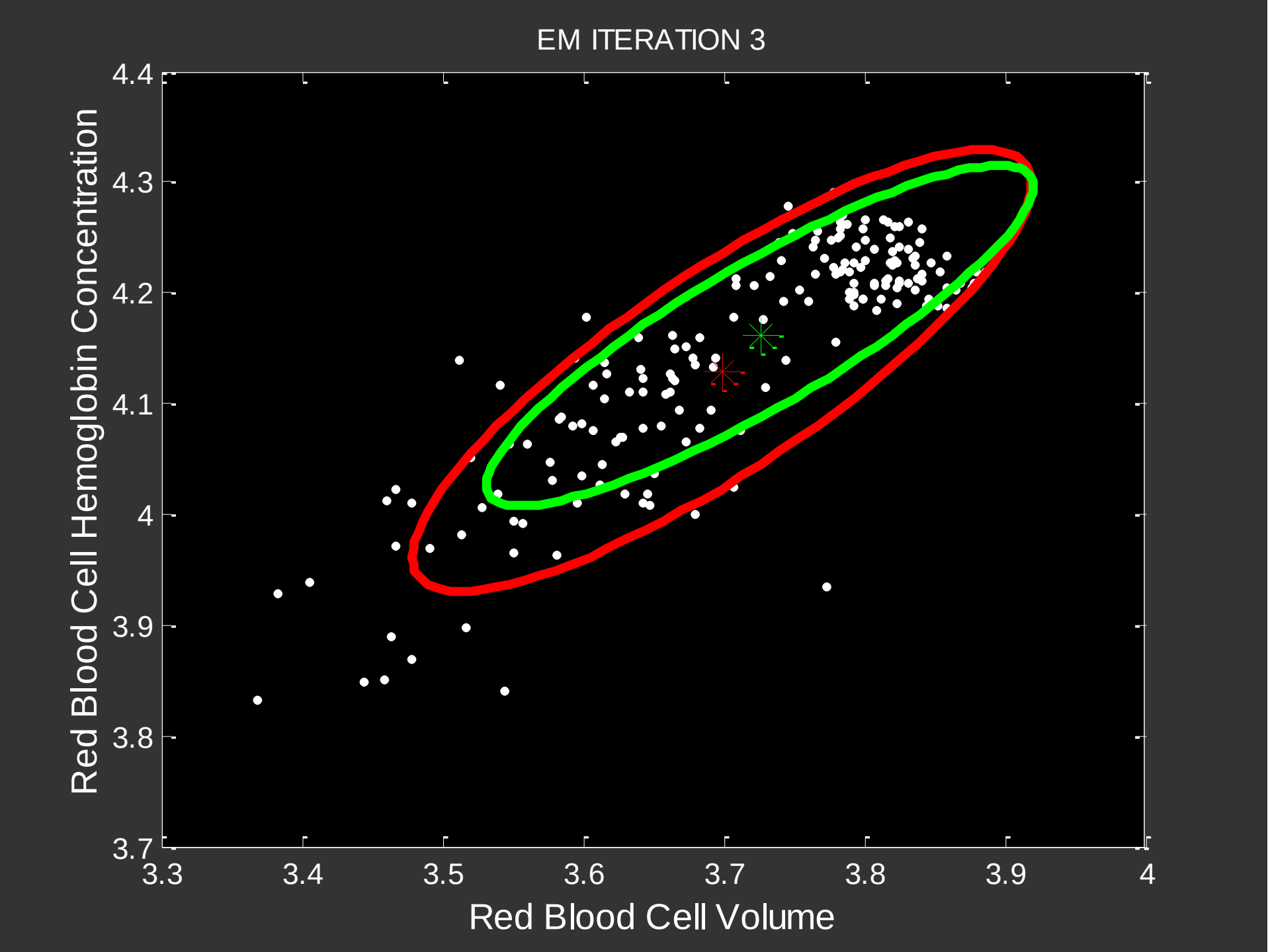
Comments on Mixtures and EM Learning

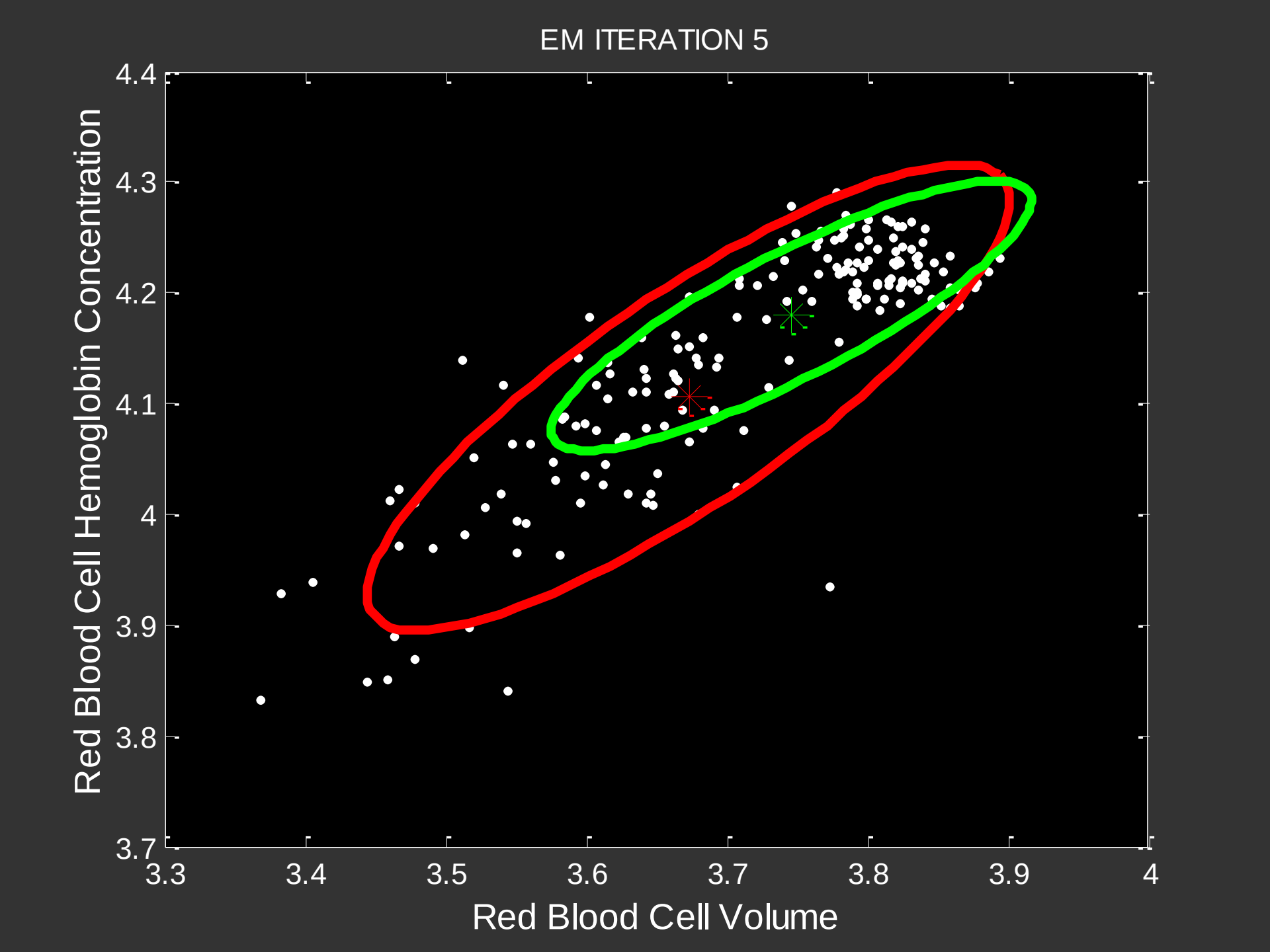
- - Complexity of each EM iteration
  - Depends on the probabilistic model being used
    - e.g., for Gaussians, E-step: $O(nK)$, M-step: $O(Knp^2)$
  - Sometimes E or M-step is not closed form
    - => can requires numerical methods at each iteration

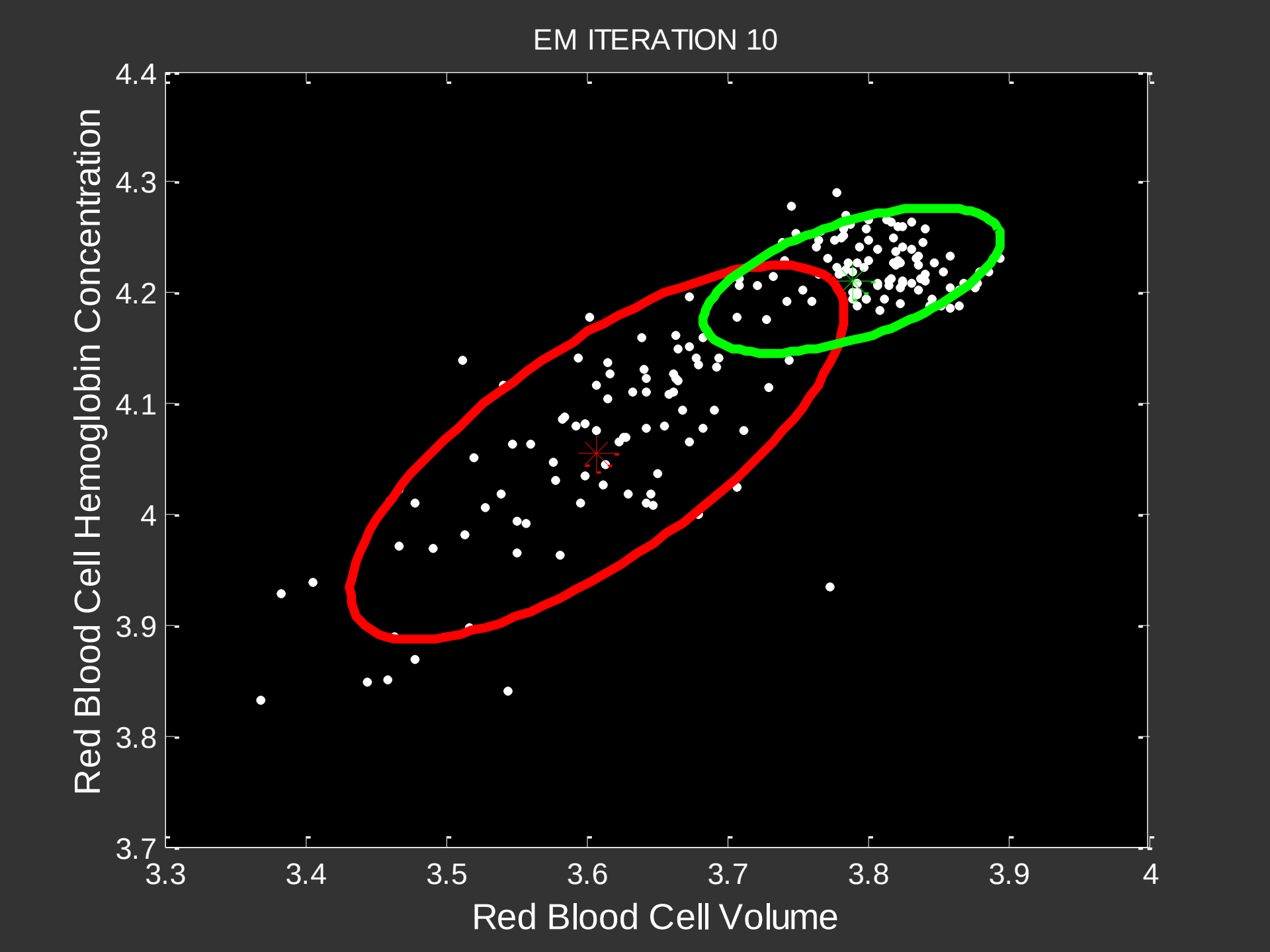
- - K-means interpretation
  - Gaussian mixtures with isotropic (diagonal, equi-variance) $\sum_k$'s
  - Approximate the E-step by choosing most likely cluster (instead of using membership probabilities)

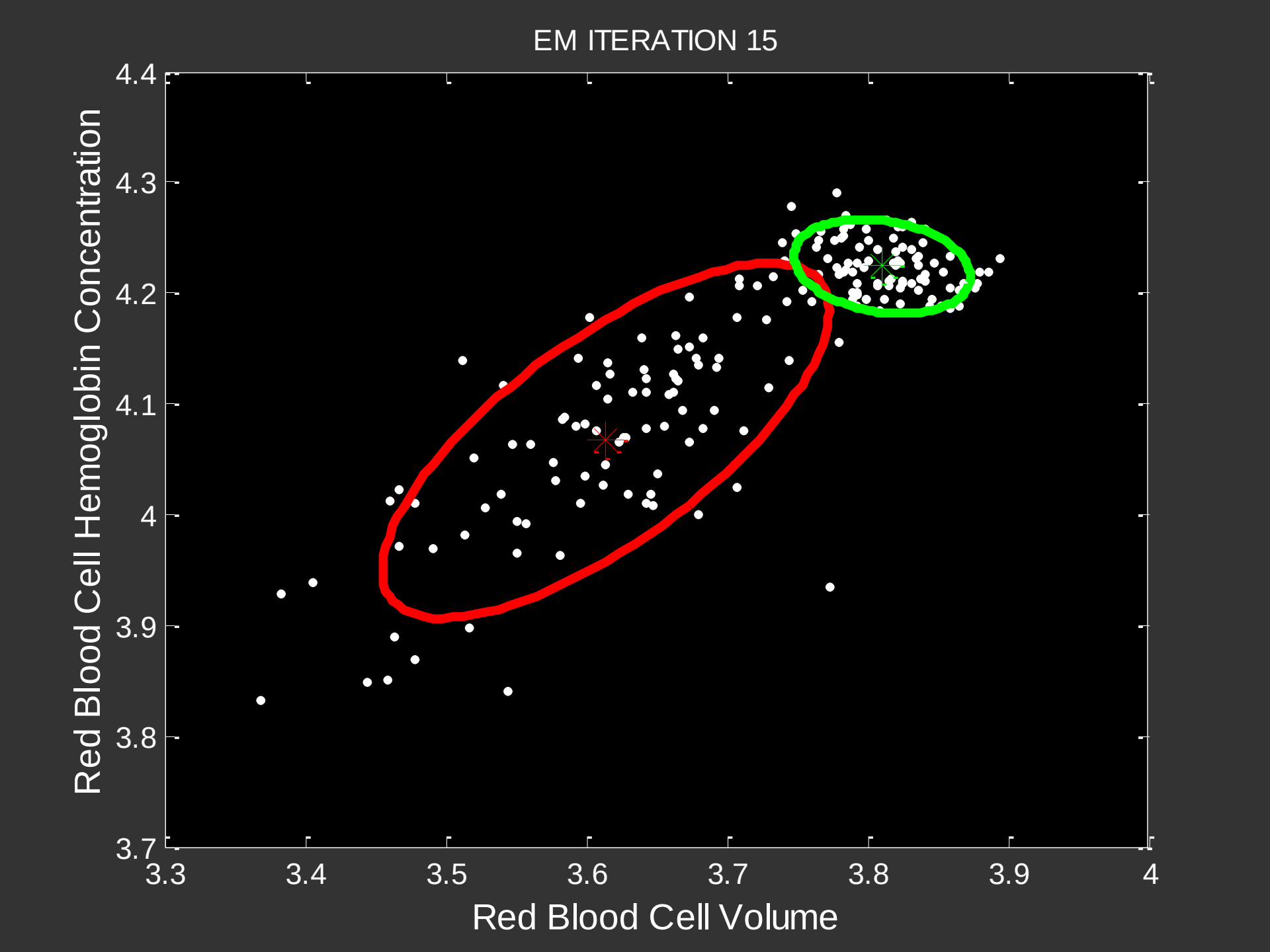

ANEMIA PATIENTS AND CONTROLS
Red Blood Cell Hemoglobin Concentration
Red Blood Cell Volume
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4

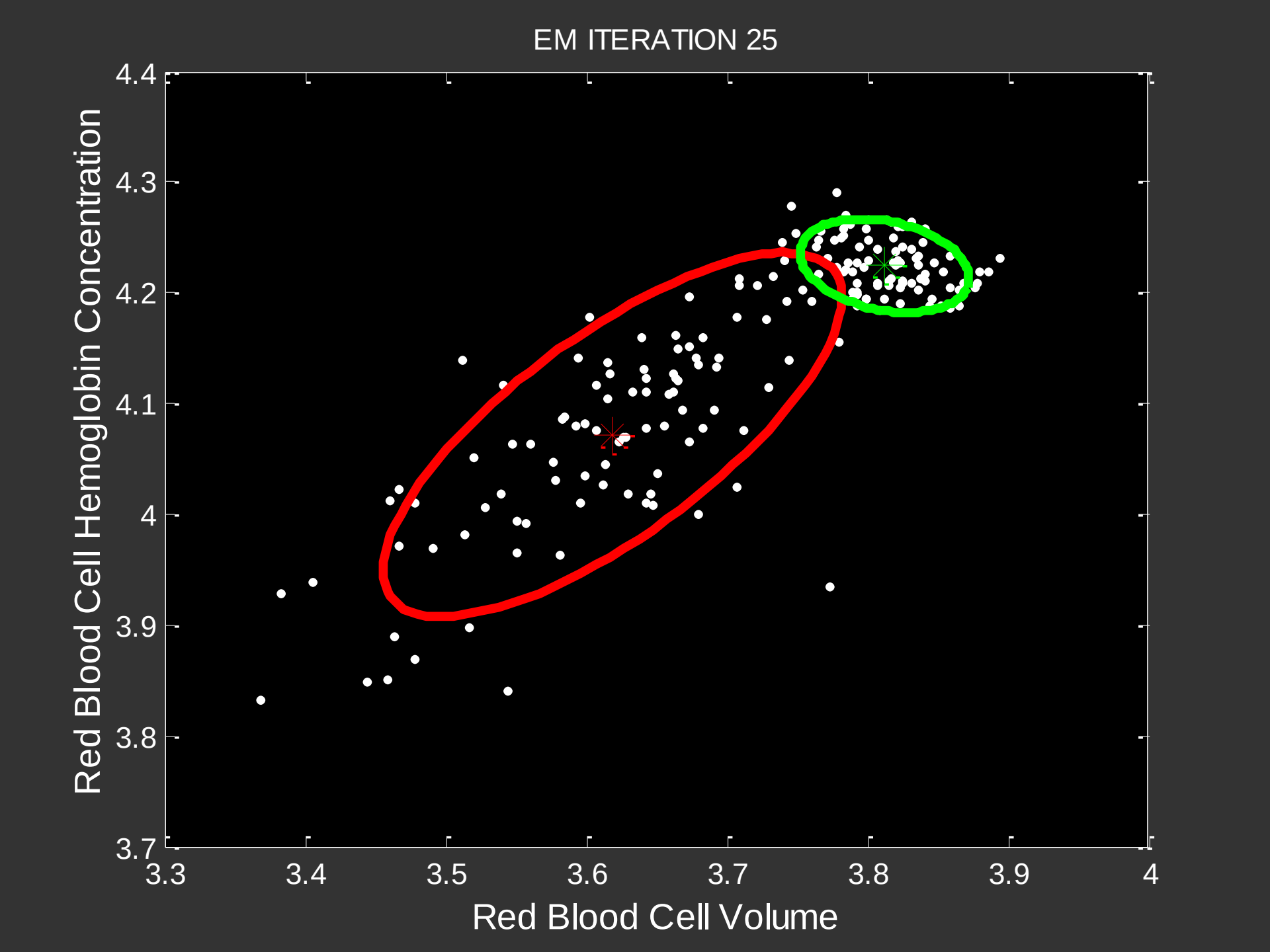

EM ITERATION 1
Red Blood Cell Hemoglobin Concentration
Red Blood Cell Volume
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4

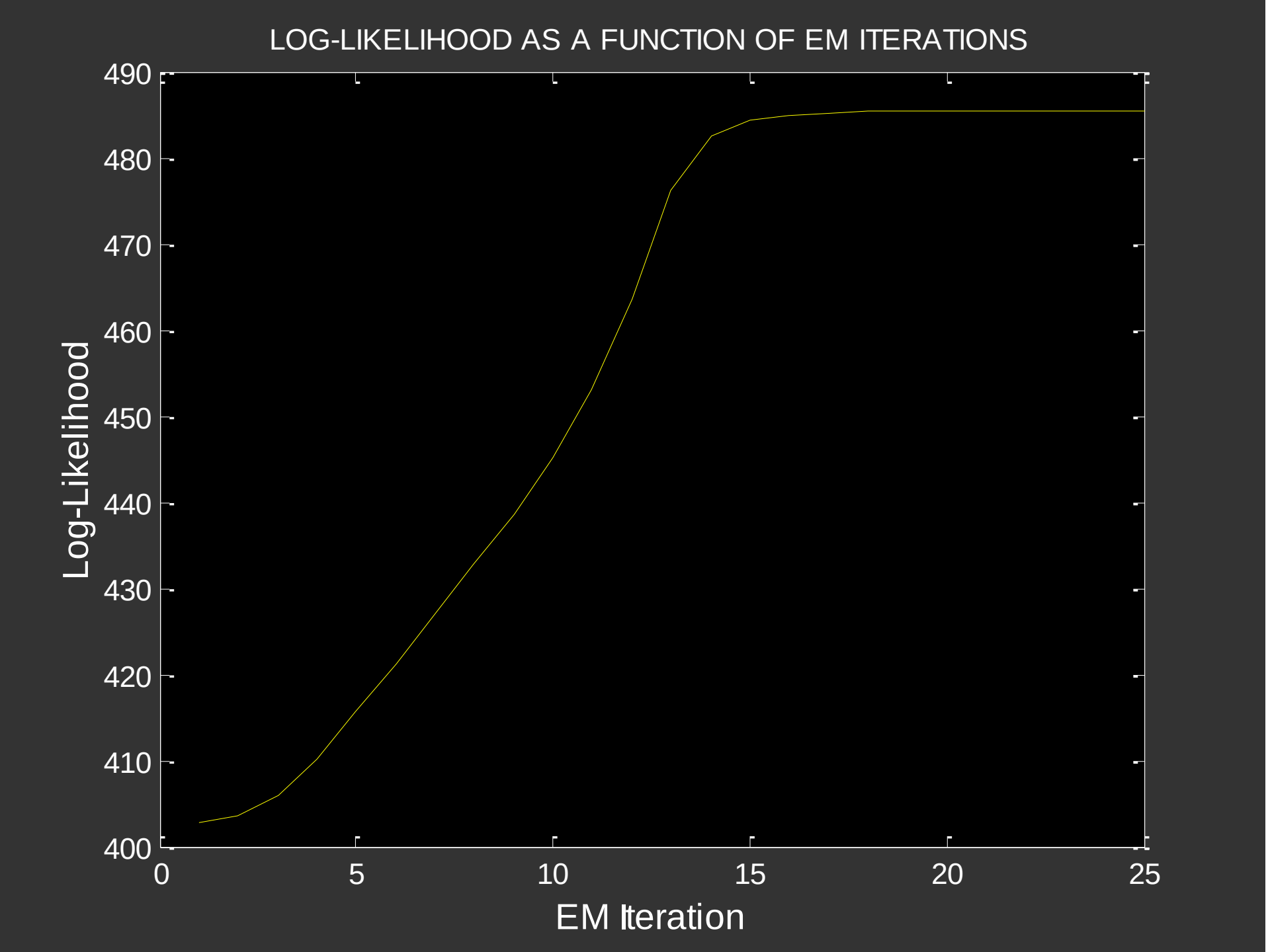

EM ITERATION 3
Red Blood Cell Hemoglobin Concentration
Red Blood Cell Volume
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4

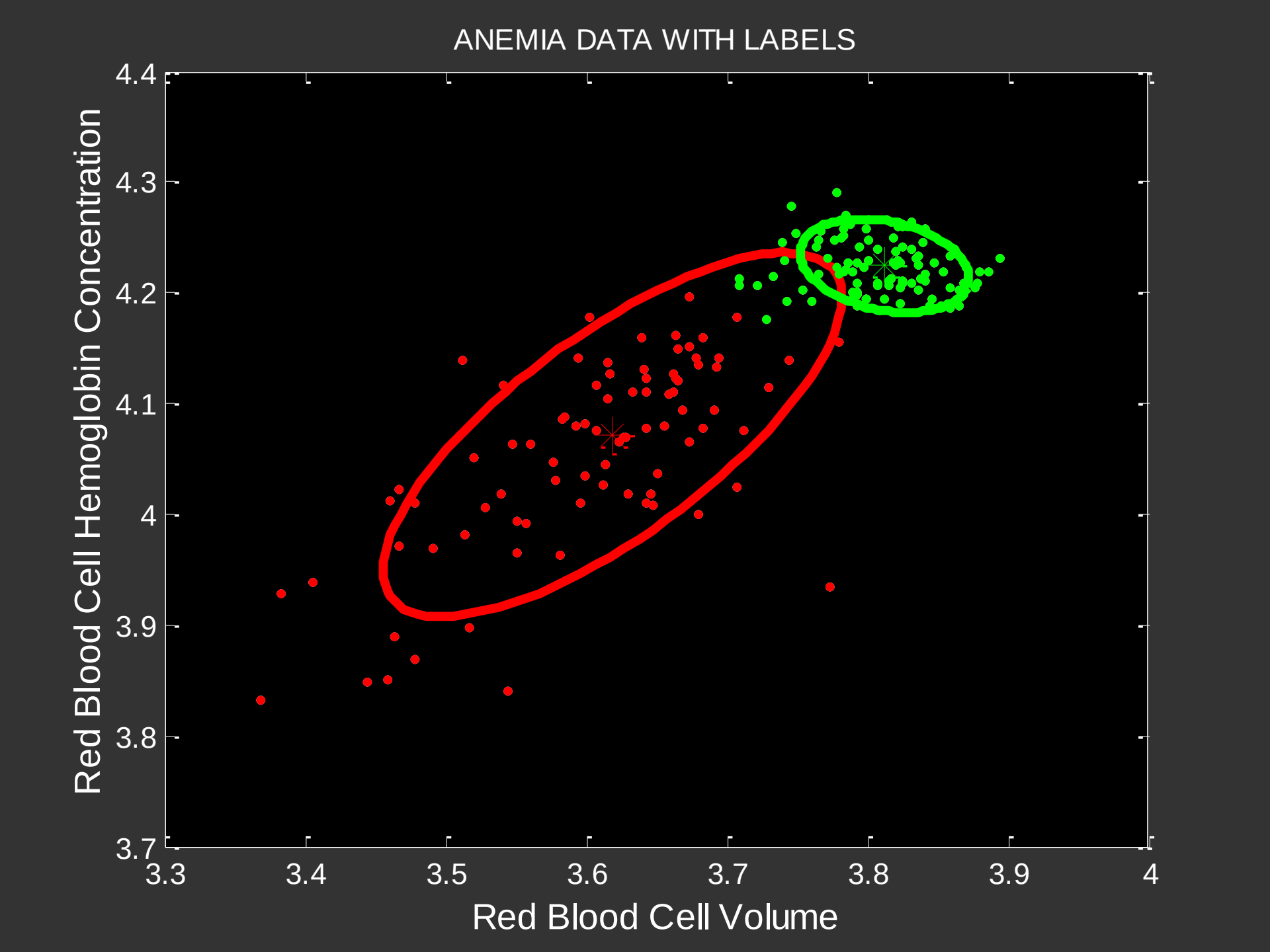

EM ITERATION 5
Red Blood Cell Hemoglobin Concentration
Red Blood Cell Volume
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4


EM ITERATION 10
Red Blood Cell Hemoglobin Concentration
Red Blood Cell Volume
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4


EM ITERATION 15
Red Blood Cell Hemoglobin Concentration
Red Blood Cell Volume
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4

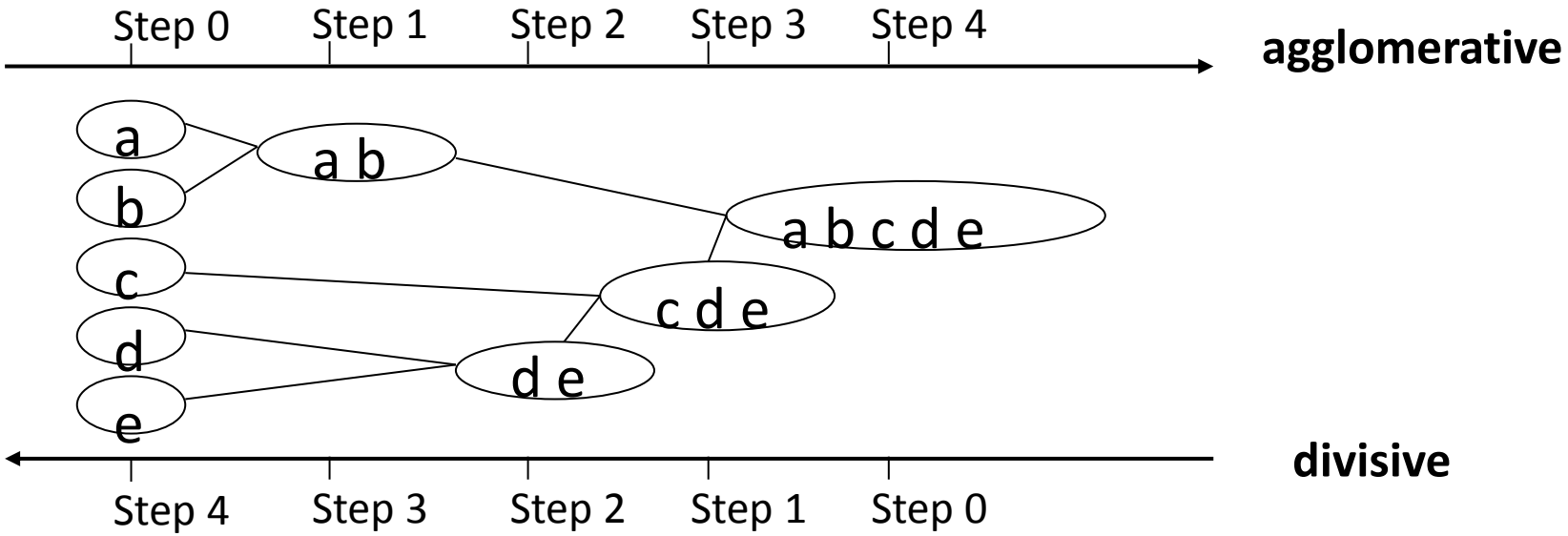

EM ITERATION 25
Red Blood Cell Hemoglobin Concentration
Red Blood Cell Volume
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4


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


ANEMIA DATA WITH LABELS
Red Blood Cell Hemoglobin Concentration
Red Blood Cell Volume
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4

Hierarchical Clustering

- **Two basic approaches:**
  - merging smaller clusters into larger ones (agglomerative),
  - splitting larger clusters (divisive)
- **visualize both via "dendograms"**
  - shows nesting structure
  - merges or splits = tree nodes



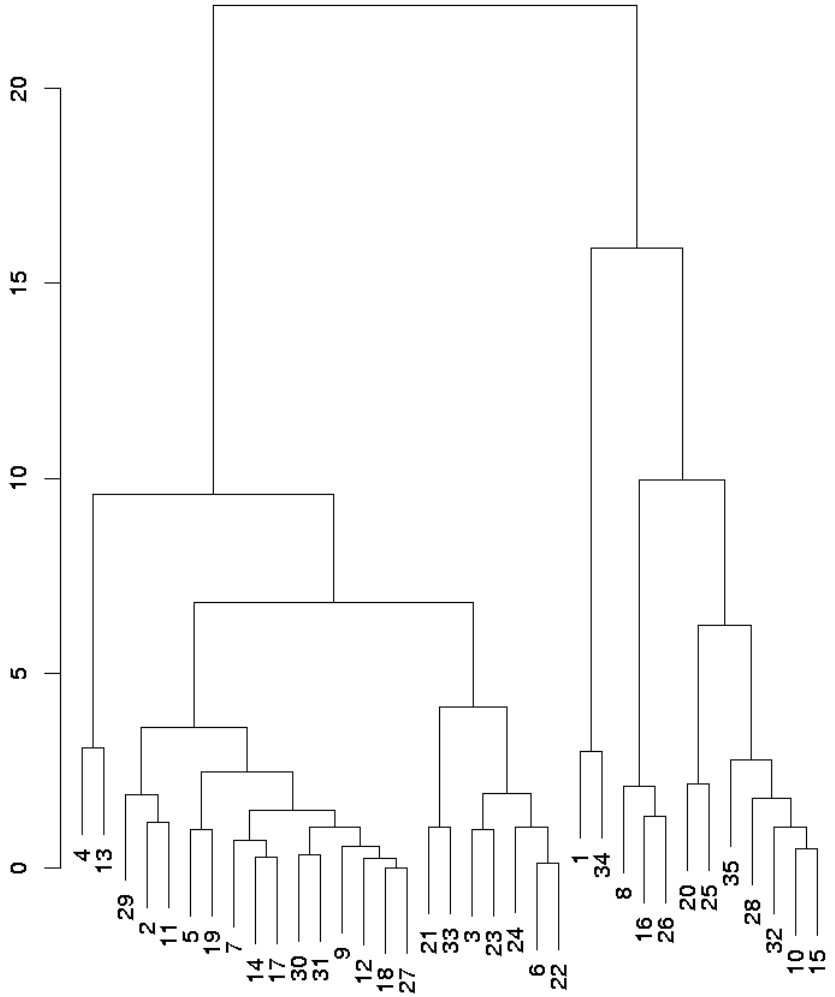
Hierarchical Clustering: Complexity

- **Quadratic algorithms**
- **Running time** can be improved using
  - sampling

    [Guha et al, SIGMOD 1998]

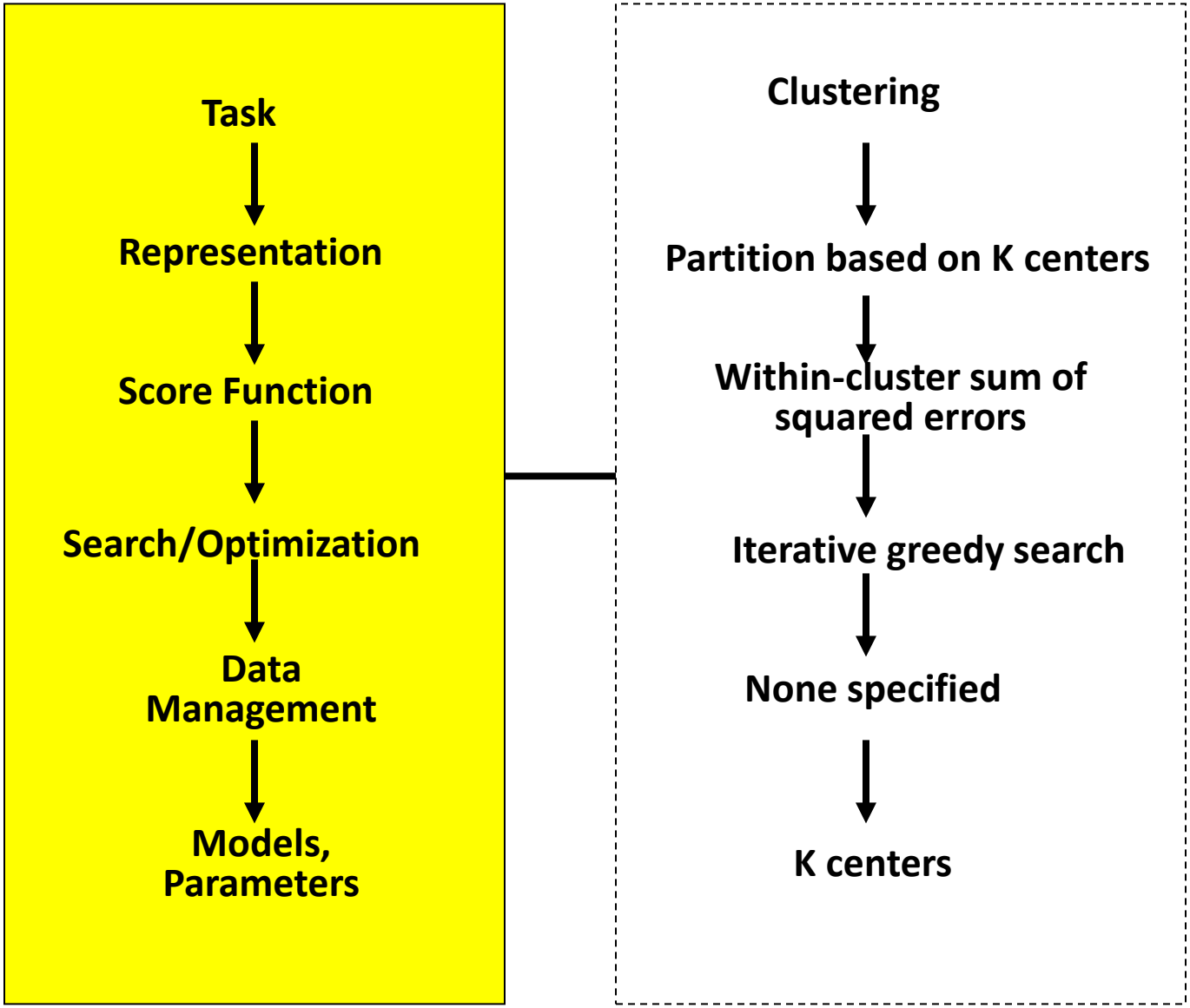
    [Kollios et al, ICDE 2001]

  or using the triangle inequality (when it holds)

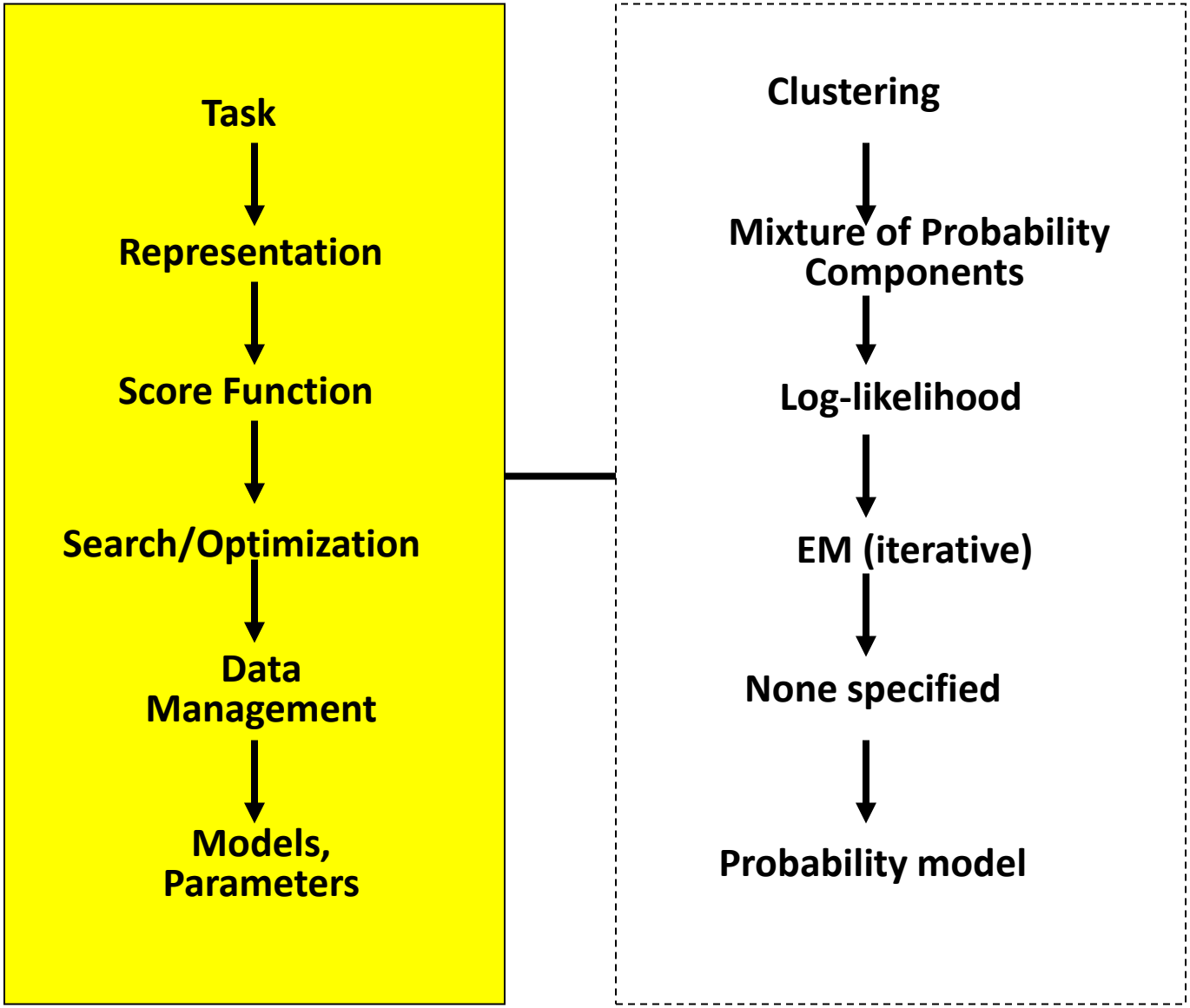


*based on slides by Padhraic Smyth UC, Irvine

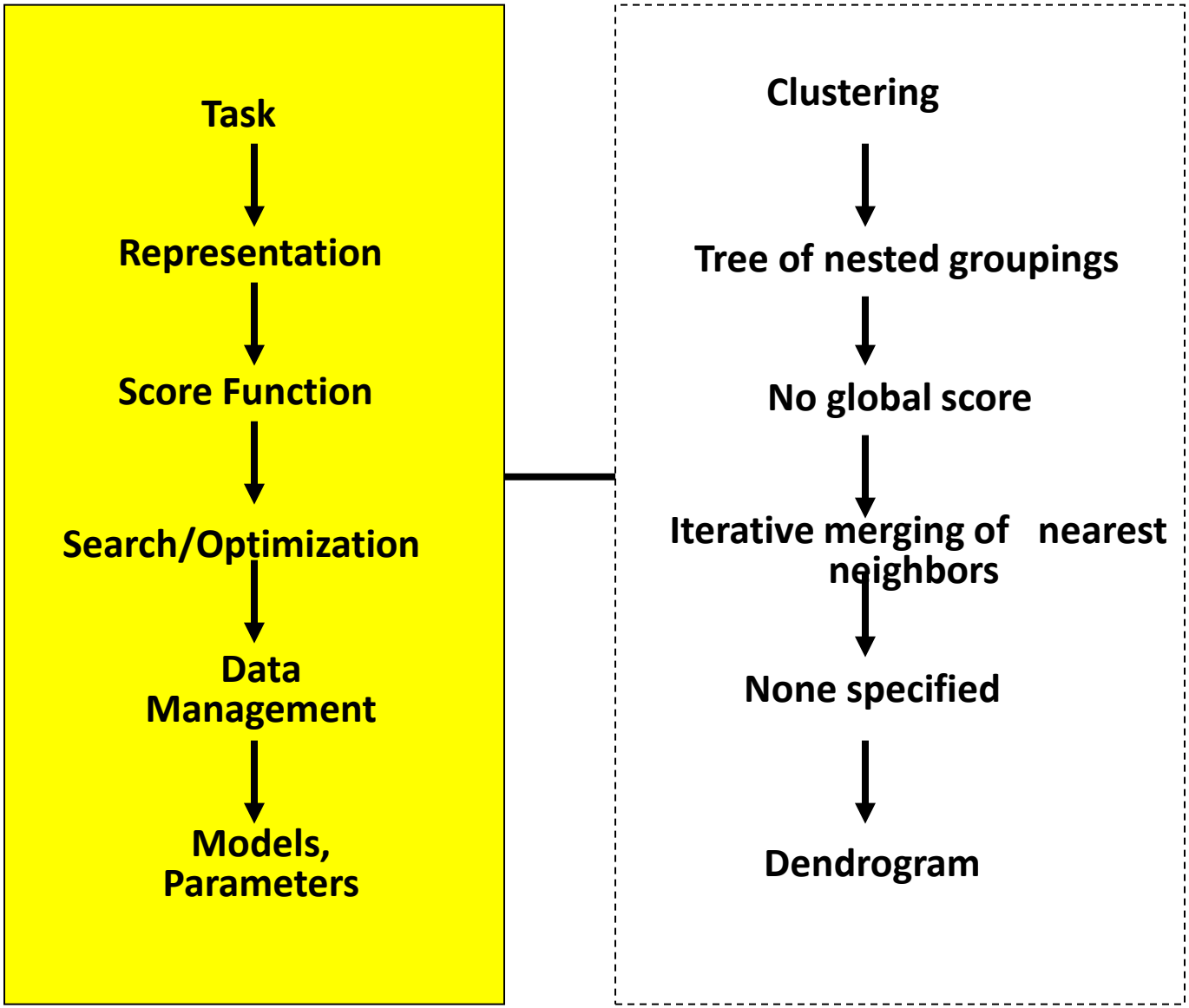
K-Means Clustering



Probabilistic Model-Based Clustering



Single-Link Hierarchical Clustering



ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Association Rules

Μάθημα: Εξόρυξη γνώσης από Βάσεις Δεδομένων και τον Παγκόσμιο Ιστό

Ενότητα # 5: Unsupervised Learning (Clustering)

Διδάσκων: Μιχάλης Βαζιργιάννης

Τμήμα: Προπτυχιακό Πρόγραμμα Σπουδών “Πληροφορικής”

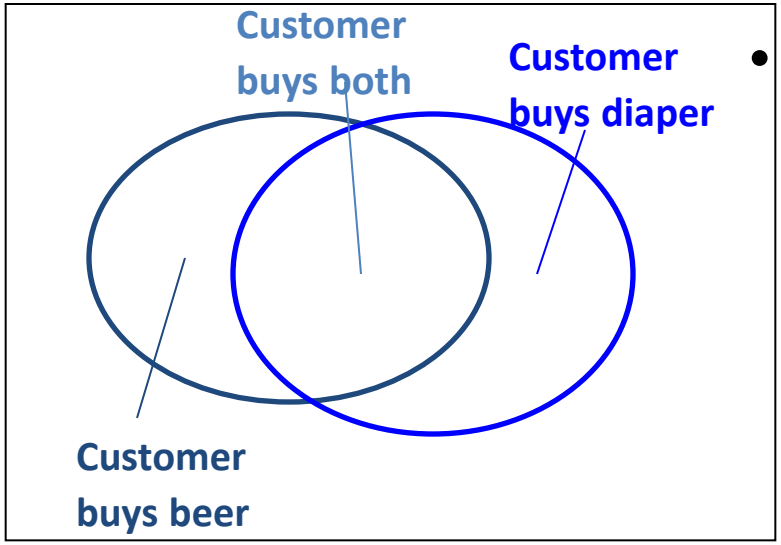
What Is Association Mining?

- Association rule mining:
  - Finding frequent patterns, associations, correlations, or causal structures among sets of items or objects in transaction databases, relational databases, and other information repositories.
- Applications:
  - Basket data analysis, cross-marketing, catalog design, loss-leader analysis, clustering, classification, etc.
- Examples.
  - Rule form: "Body $\rightarrow$ Head [support, confidence]".
  - buys(x, "diapers") $\rightarrow$ buys(x, "beers") [0.5%, 60%]
  - major(x, "CS") ^ takes(x, "DB") $\rightarrow$ grade(x, "A") [1%, 75%]

Association Rule: Basic Concepts

- Given: (1) database of transactions, (2) each transaction is a list of items (purchased by a customer in a visit)
- Find: all rules that correlate the presence of one set of items with that of another set of items
  - E.g., *98% of people who purchase tires and auto accessories also get automotive services done*
- Applications
  - $*$ $\Rightarrow$ *Maintenance Agreement* (What the store should do to boost Maintenance Agreement sales)
  - *Home Electronics* $\Rightarrow *$ (What other products should the store stocks up?)
  - Attached mailing in direct marketing
  - Detecting “ping-pong”ing of patients, faulty “collisions”

Rule Measures: Support and Confidence



Find all the rules X & Y ⇒ Z with minimum confidence and support

- support, s, probability that a transaction contains {X & Y & Z}
- confidence, c, conditional probability that a transaction having {X & Y} also contains Z

Transaction ID	Items Bought
2000	A,B,C
1000	A,C
4000	A,D
5000	B,E,F

Let minimum support 50%, and minimum confidence 50%, we have

- A ⇒ C (50%, 66.6%)
- C ⇒ A (50%, 100%)

Mining Association Rules—An Example

Transaction ID	Items Bought
2000	A,B,C
1000	A,C
4000	A,D
5000	B,E,F

Min. support 50%
Min. confidence 50%

Frequent Itemset	Support
{A}	75%
{B}	50%
{C}	50%
{A,C}	50%

For rule $A \Rightarrow C$:

support = support($\{A \Rightarrow C\}$) = 50%

confidence = support($\{A \Rightarrow C\}$)/support($\{A\}$) = 66.6%

The Apriori principle:

Any subset of a frequent itemset must be frequent

Mining Frequent Itemsets: the Key Step

- Find the *frequent itemsets*: the sets of items that have minimum support
  - A subset of a frequent itemset must also be a frequent itemset
    - i.e., if {*AB*} is a frequent itemset, both {*A*} and {*B*} should be a frequent itemset
  - Iteratively find frequent itemsets with cardinality from 1 to *k* *(k*-itemset*)*
- Use the frequent itemsets to generate association rules.

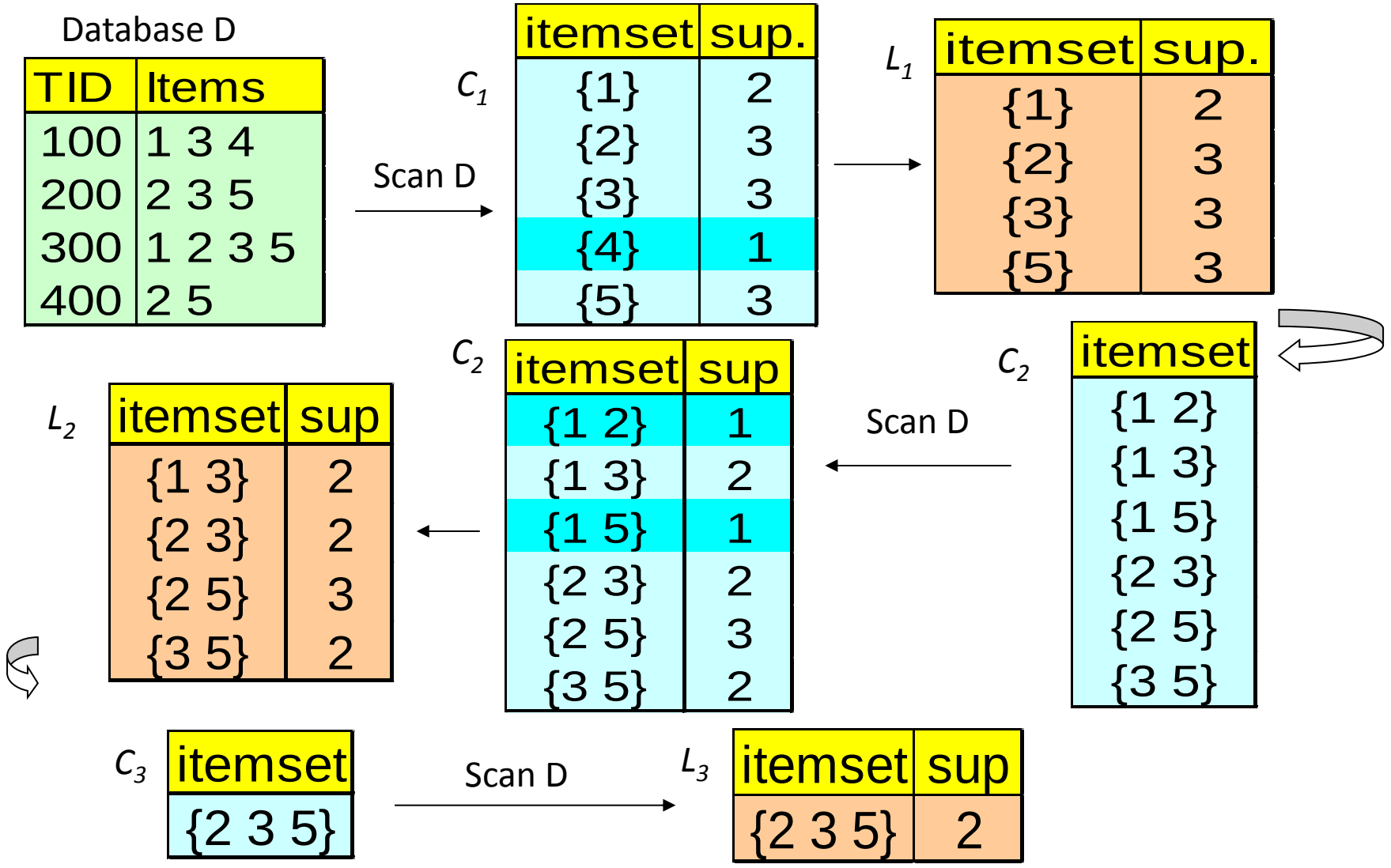
The Apriori Algorithm

- Join Step: C_k is generated by joining $L_{k-1}$with itself
- Prune Step: Any (k-1)-itemset that is not frequent cannot be a subset of a frequent k-itemset
- Pseudo-code:

C_k: Candidate itemset of size k
L_k : frequent itemset of size k

L_1 = {frequent items};
for (k = 1; L_k !=∅; k++) **do begin**
  C_{k+1} = candidates generated from L_k;
  for each transaction t in database do
    increment the count of all candidates in C_{k+1} that are contained in t
  L_{k+1} = candidates in C_{k+1} with min_support
  end
return $\cup_k L_k$;

The Apriori Algorithm — Example



Example of Generating Candidates

- L_3={*abc, abd, acd, ace, bcd*}
- Self-joining: L_3*L_3
  - *abcd* from *abc* and *abd*
  - *acde* from *acd* and *ace*
- Pruning:
  - *acde* is removed because *ade* is not in L_3
- C_4={*abcd*}

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Spectral Clustering

Μάθημα: Εξόρυξη γνώσης από Βάσεις Δεδομένων και τον Παγκόσμιο Ιστό

Ενότητα # 4: Unsupervised Learning (Clustering)

Διδάσκων: Μιχάλης Βαζιργιάννης

Τμήμα: Προπτυχιακό Πρόγραμμα Σπουδών “Πληροφορικής”

Spectral Clustering

Given Graph G=(V,E) undirected:

Vertex Set V={v_1,…..v_n}, Edge e_{ij} between v_i and v_j

we assume weight $w_{ij}>0$ for e_{ij}

|V| : number of vertices

d_i degree of v_i : $d_i = \sum_{v_j \in V} w_{ij}$

$$v(V) = \sum_{v_i \in V} d_i$$

for $A \subset V$ $\overline{A} = V - A$

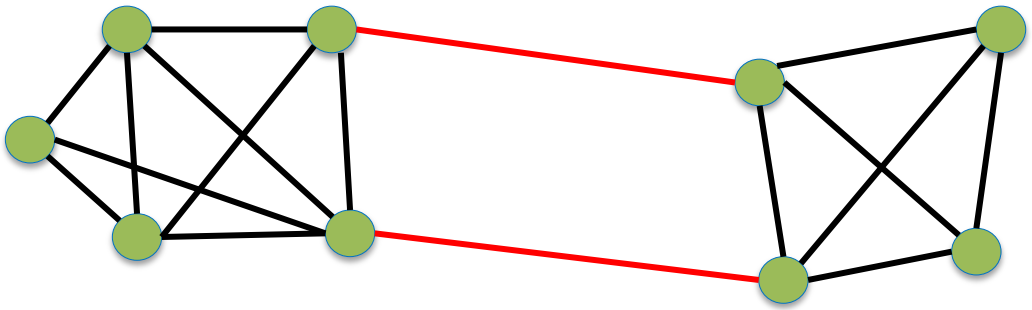
Given A, B $\subset V$ & A $\cap$ $B = \emptyset$ $w(A,B) = \sum_{v_i \in A, vj \in B} w_{ij}$

D: Diagonal matrix where D(i,i)=d_i

W: Adjacency matrix W(i,j)=w_{ij}

Graph-Cut

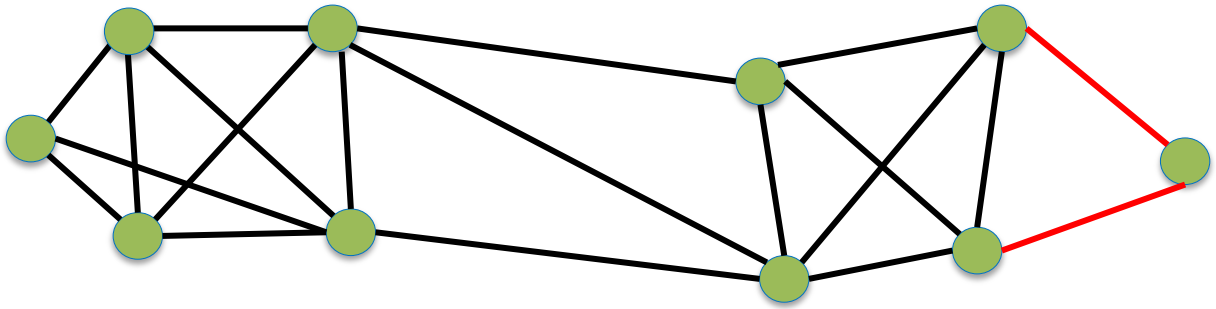
- For k clusters:
    - undirected graph:1/2 we count twice each edge



- Min-cut: Minimize the edges' weight a cluster shares with the rest of the graph

Min-Cut

- Easy for k=2 : Mincut(A_1,A_2)
  - Stoer and Wagner: “A Simple Min-Cut Algorithm”
- In practice one vertex is separated from the rest
  - The algorithm is drawn to outliers



Normalized Graph Cuts

We can normalize by the size of the cluster (size of sub-graph) :

number of Vertices (Hagen and Kahng, 1992):

$$Ratiocut(A_1, \dots Ak) = \sum_{i=1}^{k} \frac{cut(Ai, \overline{A_i})}{|Ai|}$$

sum of weights (Shi and Malik, 2000) :

$$Ncut(A_1, \dots Ak) = \sum_{i=1}^{k} \frac{cut(Ai, \overline{A_i})}{v(A_i)}$$

Optimizing these functions is NP-hard

Spectral Clustering provides solution to a relaxed version of the above

From Graph Cuts to Spectral Clustering

For simplicity assume k=2:

Define $f: V \to \mathbb{R}$ for Graph G :

$$f_i = \begin{cases} 1 & v_i \in A \\ -1 & v_i \in \overline{A} \end{cases}$$

Optimizing the original cut is equivalent to an optimization of:

$$\sum_{i,j=1}^{n} w_{ij}(f_i - fj)^2$$

$$= \sum_{v_i \in A, vj \in \overline{A}} w_{ij}(1+1)^2 + \sum_{v_i \in \overline{A}, vj \in A} w_{ij}(-1-1)^2$$

$$= \mathbf{8} * \boldsymbol{cut}\left(\boldsymbol{A}, \overline{\boldsymbol{A}}\right)$$

Graph Laplacian

How is the previous useful in Spectral clustering?

$$\sum_{i,j=1}^{n} w_{ij}(f_i - fj)^2$$

$$= \sum_{i,j=1}^{n} w_{ij} {f_i}^2 - 2\sum_{i,j=1}^{n} w_{ij} f_i f_j + \sum_{i,j=1}^{n} w_{ij} {f_j}^2$$

$$= \sum_{i,j=1}^{n} d_i {f_i}^2 - 2\sum_{i,j=1}^{n} w_{ij} f_i f_j + \sum_{i,j=1}^{n} d_j {f_j}^2$$

$$= 2\left(\sum_{i,j=1}^{n} d_{ii} {f_i}^2 - \sum_{i,j=1}^{n} w_{ij} f_i f_j\right)$$

$$= 2(\boldsymbol{f^T D f} - \boldsymbol{f T W f}) = 2\boldsymbol{f^T}(\boldsymbol{D} - \boldsymbol{W})\boldsymbol{f} = 2\boldsymbol{f^T L f}$$

f: a single vector with the cluster assignments of the vertices

$\boldsymbol{L=D-W}$ the Laplacian of a graph

Properties of L

L is

- Symmetric
- Positive
- Semi-definite ($x^T Lx, x$ is a non negative vector)

The smallest eigenvalue of L is 0

- The corresponding eigenvector is $\mathbb{1}$

L has n non-negative, real valued eigenvalues

$$0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$$

Two Way Cut from the Laplacian

We could solve $min_f f^T L f$ where $f \in \{-1,1\}^n$

NP-Hard for discrete cluster assignments

Relax the constraint $to\ f \in R^n$:

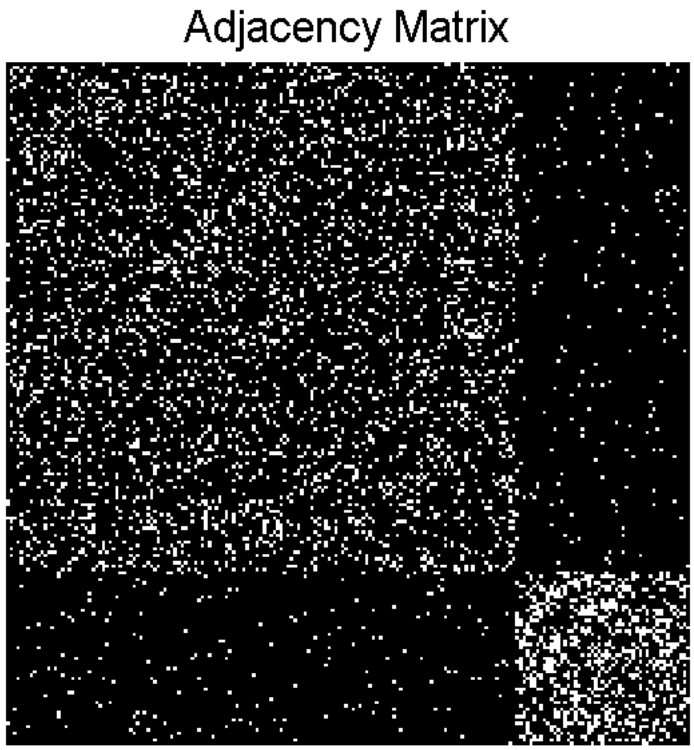
$min_f f^T L f$ subject to $f^T f = n$

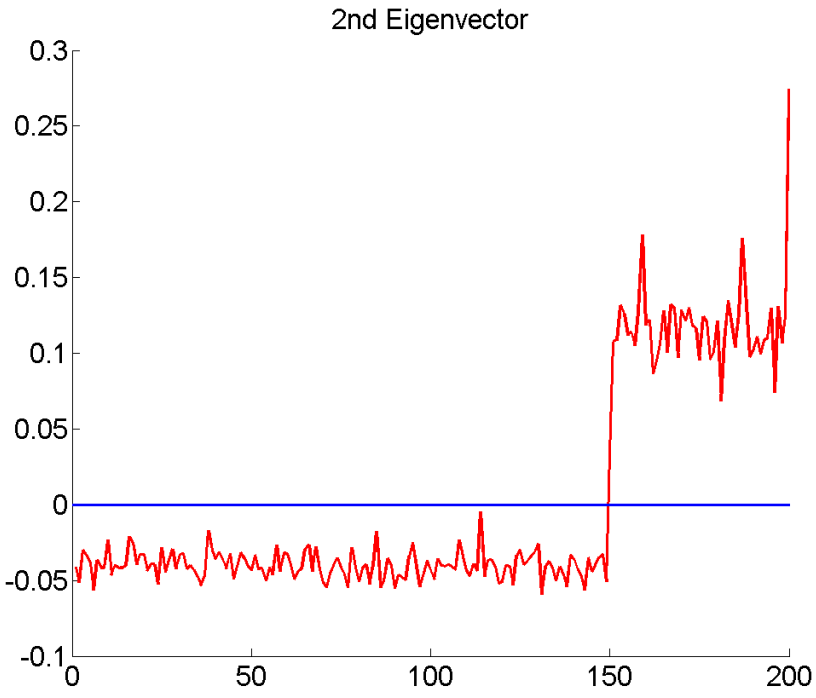
Solution: (**Rayleigh-Ritz Theorem**)

- the eigenvector corresponding to smallest eigenvalue: 0 TRIVIA as it offers no information

- we use the second eigenvector as an approximation (fi>0 vertex belongs to cluster A, fi<0 to the other cluster V-A)

Example


Adjacency Matrix


2nd Eigenvector
0.3
0.25
0.2
0.15
0.1
0.05
0
-0.05
-0.1
0
50
100
150
200

Ratio Cut

$$Ratiocut(A_1, \dots Ak) = \sum_{i=1}^{k} \frac{cut(Ai, \overline{A_i})}{|Ai|}$$

Define $f: V \to \mathbb{R}$ for Graph G :

$$f_i = \begin{cases} \sqrt{\frac{|\overline{A}|}{|A|}} & vi \in A \\ -\sqrt{\frac{|A|}{|\overline{A}|}} & v_i \in \overline{A} \end{cases}$$

$$\sum_{i,j=1}^{n} w_{ij}(f_i - fj)^2 = 2cut(\mathrm{A}, \overline{A}) \left(\sqrt{\frac{|\overline{A}|}{|A|}} + \sqrt{\frac{|A|}{|\overline{A}|}} + 2 \right)$$

$$= 2|V|Ratiocut(A, \overline{A})$$

Ratio Cut

We have $min_f f^T L f$ subject to

$$f^T 1 = 0, \; fTf = n$$

$$f^T 1 = \sum_i^n f_i = \sum_{v_i \in A} \sqrt{\frac{|\overline{A}|}{|A|}} + \sum_{v_i \in \overline{A}} - \sqrt{\frac{|A|}{|\overline{A}|}} = |\mathrm{A}| \sqrt{\frac{|\overline{A}|}{|A|}} - |\overline{A}| \sqrt{\frac{|A|}{|\overline{A}|}} = 0$$

$$f^T f = \sum_i^n f_i^2 = |\overline{A}| + |A| = n$$

The second smallest eigenvalue of $Lf = \lambda f$ approximates the solution

Normalized Cut

- $Ncut(A_1, \dots Ak) = \sum_{i=1}^{k} \frac{cut(Ai, \overline{A_i})}{v(A_i)}$
- Define $f: V \rightarrow \mathbb{R}$ for Graph G :
-

$$f_i = \begin{cases} \sqrt{\frac{v(\overline{A})}{v(A)}} & vi \in A \\ -\sqrt{\frac{v(A)}{v(\overline{A})}} & vi \in \overline{A} \end{cases}$$

- $\sum_{i,j=1}^{n} w_{ij}(f_i - fj)^2 =$

$$2cut(\mathrm{A}, \overline{A}) \left(\sqrt{\frac{v(\overline{A})}{v(A)}} + \sqrt{\frac{v(A)}{v(\overline{A})}} + 2 \right)$$

$$= 2v(\mathrm{V})\mathrm{Ncut}(\mathrm{A}, \overline{A})$$

Normalized Cut

Similarly we come to : $min_f f^T L f$

subject to $f^T D1 = 0,\ fTDf = v(V)$

Assume $h = D^{1/2} f$

$min_h h^T D^{-1/2} \mathrm{L} D^{-1/2} \mathrm{h}$ subject to

$$h^T D^{1/2} 1 = 0, \qquad h^T h = v(V)$$

The answer is in the eigenvector of the second smallest eigenvalue of $L_{sym} = D^{-1/2} \mathrm{L} D^{-1/2}$

Shi and Malik (2000)

L_{sym} is the normalized Laplacian

has n non-negative, real valued eigenvalues

$$0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$$

Multi-Way Graph Partition

Define $f_{ij} = \begin{cases} \frac{1}{\sqrt{|Aj|}} & vi \in Aj \\ 0 & ot\ \ ewi\ h\ \ s \end{cases}$

e

we have a vector indicating the cluster a vertex belongs to

Similarly to the other equations we can deduce:

$$f_i^T L f_i = cu\ (Ai, \overline{A_i}) / |tAi|$$

$$\sum_{i=1}^{k} f_i^T L f_i = \sum_{i=1}^{k} (F^T LF)_{ii} = Tr(F^T LF)$$

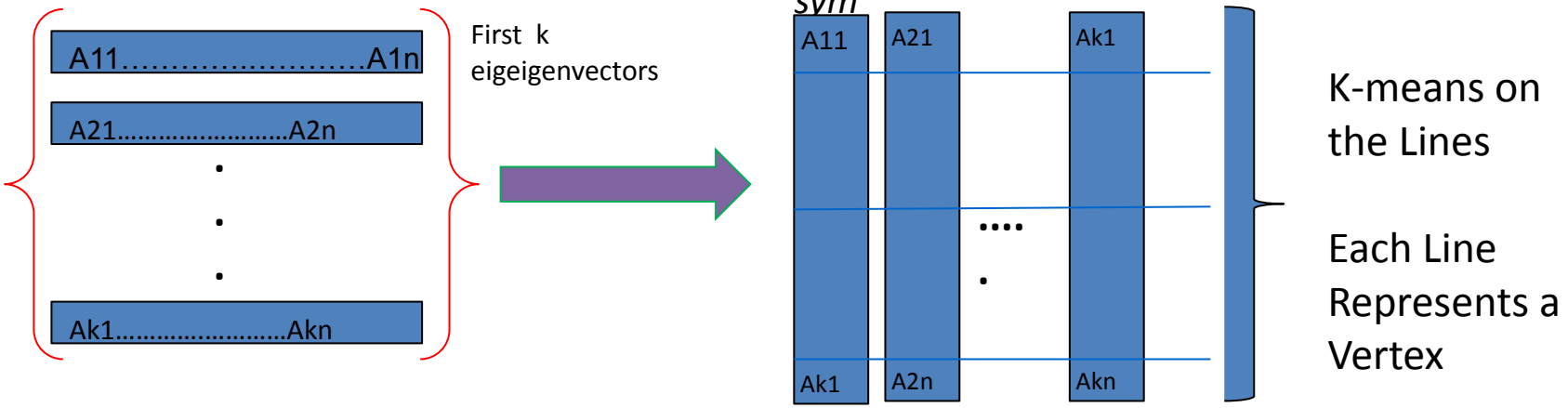
Where Tr is the Trace of a Matrix

So now the RatioCut becomes:

$$mi\ \ (F^T LF)\ sunje\ \ c\ \ to\ FTF = I$$

Multi-Way Graph Partition

- The solution can now be given by the first k eigenvectors of L as columns
- The real values need to be converted to cluster assignments
  - We use k-means to cluster the rows
  - We can substitute *L* with L_{sym}



Algorithm for k>2

Compute Laplacian (L, L_{sym}).
Compute the first k eigenvectors $u_1, \ldots, u_k$ of L.
Let $U \in \mathbb{R}^{nxk}$ the matrix containing the vectors u1, . . . , uk as columns.

For i = 1, . . . , n,

let $y_i \in \mathbb{R}^k$ the vector corresponding to the i-th row of U.
Cluster the points $y_i = 1, \ldots . n \in \mathbb{R}^k$ with the k-means algorithm into clusters C1, . . . ,Ck.

Output: Clusters $A_1, \ldots, A_k$ with $Ai = \{j | v_j \in Ci\}$

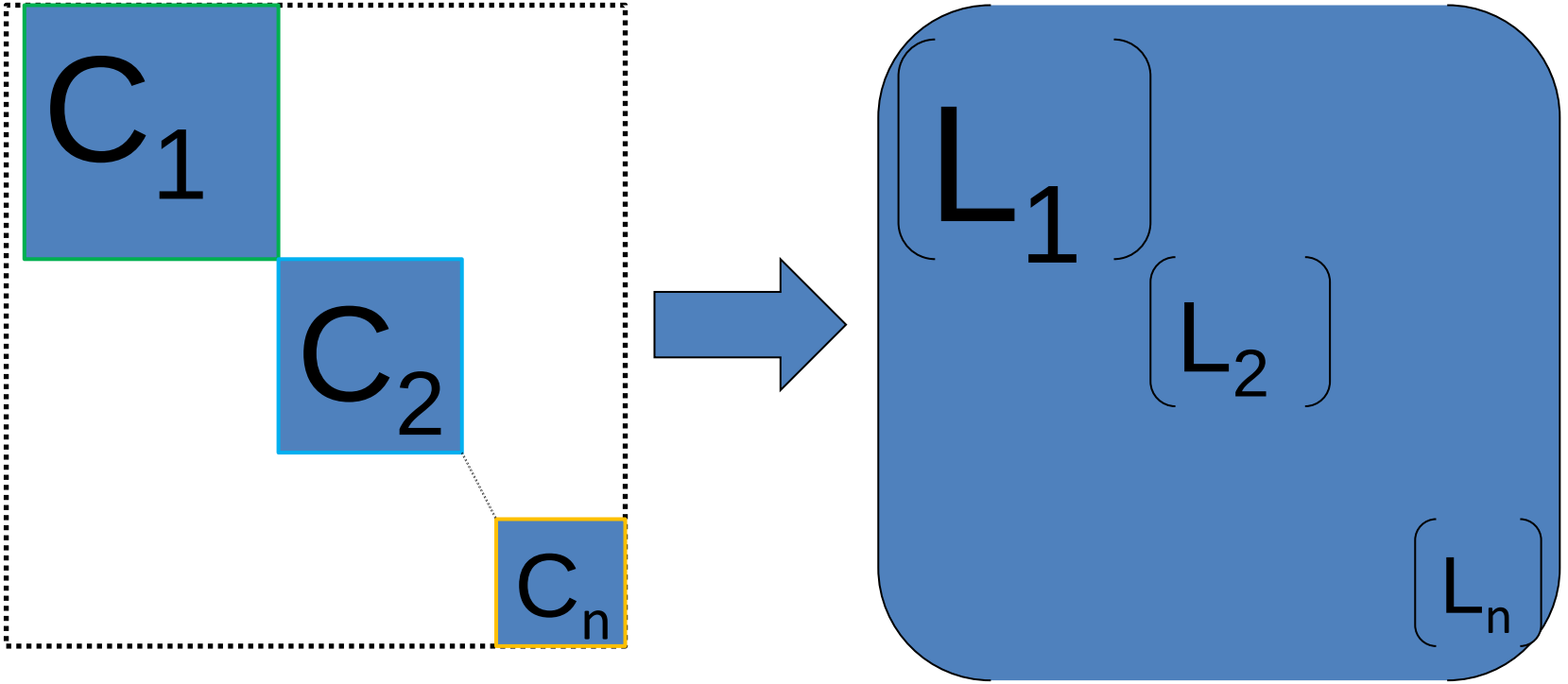
HOW DO WE CHOOSE k?

We choose the k that maximizes the eigengap:

$$\Delta_k = |\lambda_k - \lambda_{k-1}|$$ (**Davis-Kahan Theorem**)

Ideally: for k connected components the Laplacian has k 0-eigenvalues

Laplacian-Eigenvectors-EigenValues



Everything sorted according to cluster : block diagonal form Matrix
L follows the same form composed on $L_1 \ldots L_n$
Each L_i has the same properties as L: nx0 min eigenvalues etc..
Each "Second" eigenvector is a cut of C_i from the rest of the graph and holds a mapping (distance) of a vertex to the cluster i

Simple example

$$\begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

2 Eigenvectors

(1100) and
(0011)

Mapping vertices in their clusters

Permutation does not change the result

The cut remains the same regardless of the ordering

$$\begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}$$

2 Eigenvectors

(1010) and
(0101)

Mapping vertices to the same clusters

References

1. UIrike von Luxburg, A Tutorial on Spectral Clustering, Statistics and Computing, 2007
2. Davis, C., W. M. Kahan (March 1970). The rotation of eigenvectors by a perturbation. III. SIAM J. Numerical Analysis 7
3. Shi, Jianbo, and Jitendra Malik. "Normalized cuts and image segmentation."*Pattern Analysis and Machine Intelligence, IEEE Transactions on* (2000).
4. Mechthild Stoer and Frank Wagner. 1997. A simple min-cut algorithm. *J. ACM*
5. Ng, Jordan & Weiss, K-means algorithm on the embeded eigen-space, NIPS 2001
6. Hagen, L. Kahng, , "New spectral methods for ratio cut partitioning and clustering," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* , 1992

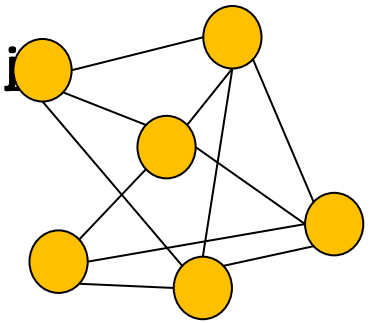
Modularity Based Methods

- Most of the community evaluation measures (e.g., conductance, cut-based measures), quantify the quality of a community based on
  - **Internal connectivity** (intra-community edges)
  - **External connectivity** (inter-community edges)
- **Question:** Is there any other way to distinguish groups of nodes with good community structure?
- **Random graphs** are not expected to present inherent community structure
- **Idea:** Compare the number of edges that lie **within a cluster** with the expected one in case of **random graphs** with the same degree distribution – **modularity measure**

Main idea

- **Modularity** function **[Newman and Girvan '04], [Newman '06]**
- Initially introduced as a measure for assessing the strength of communities
  - **Q = (fraction of edges within communities) –**

    (expected number of edges within communities)
- What is the **expected** number of edges?
- Consider a configuration model
  - **Random graph** model with the same degree distribution
  - Let $\mathbf{P_{ij}}$ = probability of an edge between nodes **i** and **j** with degrees $\mathbf{k_i}$ and $\mathbf{k_j}$ respectively
  - Then $\mathbf{P_{ij} = k_i k_j / 2m}$, where $\mathbf{m = |E| = \frac{1}{2} \sum_i k_i}$



Formal definition of modularity

$$Q = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(C_i, C_j)$$

where

- **A** is the adjacency matrix
- **ki**, **kj** the degrees of nodes **i** and **j** respectively
- **m** is the number of edges
- **Ci** is the community of node **i**
- **δ(.)** is the Kronecker function: 1 if both nodes **i** and **j** belong on the same community (**Ci = Cj**), 0 otherwise

[Newman and Girvan '04], [Newman '06]

Properties of modularity

$$Q = \frac{1}{2m}\sum_{ij}\left(A_{ij} - \frac{k_i k_j}{2m}\right)\delta(c_i, c_j)$$

- **Larger** modularity **Q** indicates **better** communities (more than random intra-cluster density)
  - The community structure would be better if the number of internal edges exceed the expected number
- Modularity value is always **smaller than 1**
- It can also take **negative values**
  - E.g., if each node is a community itself
  - No partitions with positive modularity → No community structure
  - Partitions with large negative modularity → Existence of subgraphs with small internal number of edges and large number of inter-community edges

[Newman and Girvan '04], [Newman '06], [Fortunato '10]

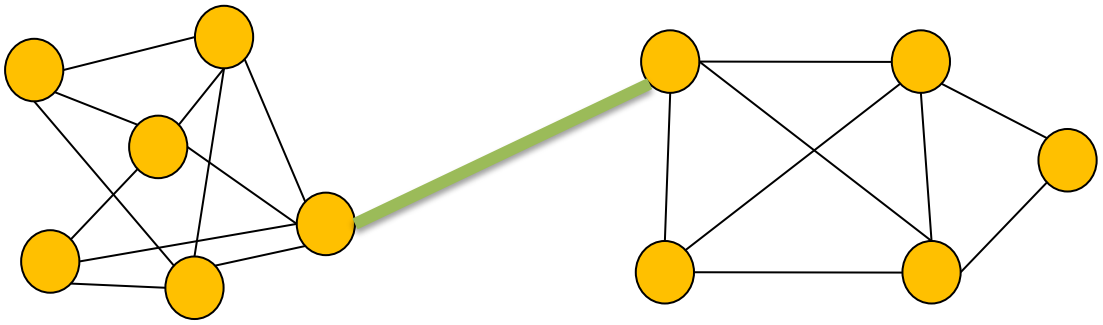
Applications of modularity

- Modularity can be applied:
  - As **quality function** in clustering algorithms
  - As **evaluation measure** for comparison of different partitions or algorithms
  - As a community detection tool itself
    - **Modularity optimization**
  - As criterion for reducing the size of a graph
    - Size reduction preserving modularity **[Arenas et al. '07]**

[Newman and Girvan '04], [Newman '06], [Fortunato '10]

Modularity-based clustering

- Modularity was first applied as a **stopping criterion** in the Newman-Girvan algorithm
- Newman-Girvan algorithm **[Newman and Girvan '04]**
  - A **divisive** algorithm (detect and remove edges that connect vertices of different communities)
  - **Idea:** try to identify the edges of the graph that are most between other vertices → responsible for connecting many node pairs
  - Select and remove edges based to the value of **betweenness centrality**
  - **Betweenness centrality:** number of **shortest paths** between every pair of nodes, that pass through an edge



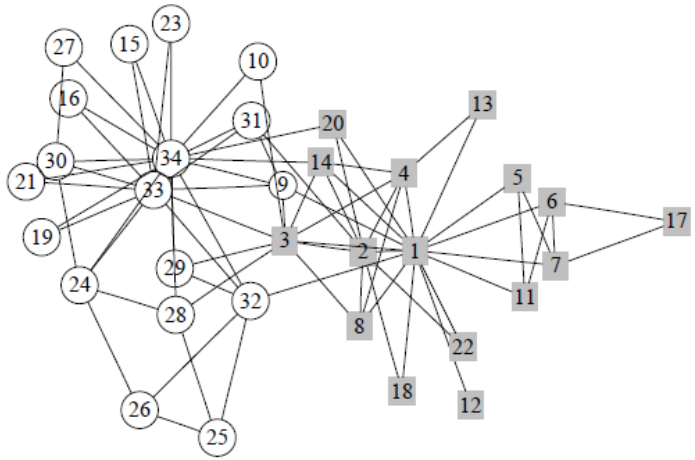
Edge betweenness is higher for edges that connect different communities

Newman-Girvan algorithm (1)

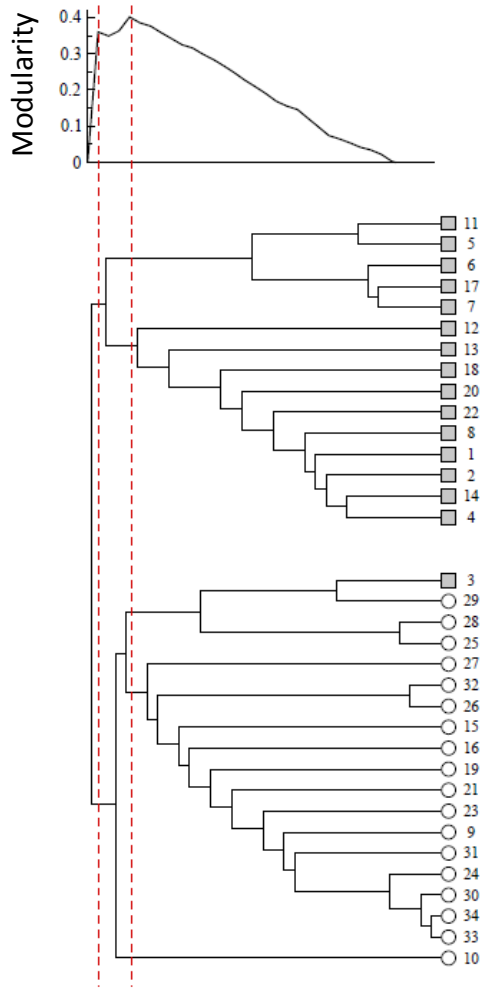
- **Basic steps:**
  1. Compute betweenness centrality for all edges in the graph
  2. Find and remove the edge with the highest score
  3. Recalculate betweenness centrality score for the remaining edges
  4. Go to step 2
- How do we know if the produced communities are **good ones** and stop the algorithm?
  - The output of the algorithm is in the form of a **dendrogram**
  - Use **modularity** as a criterion to cut the dendrogram and terminate the algorithm (Q ~= 0.3-0.7 indicates good partitions)
- Complexity: **$O(m^2n)$** (or **$O(n^3)$** on a sparse graph)

[Newman and Girvan '04], [Girvan and Newman '02]

Newman-Girvan algorithm (2)



Zachary's karate club



Community structure

[Newman and Girvan '04]

Modularity optimization

- High values of modularity indicate good quality of partitions
- **Goal:** find the partition that corresponds to the maximum value of modularity
- **Modularity maximization** problem
  - Computational difficult problem **[Brandes et al. '06]**
  - Appoximation techniques and heuristics
- Four main categories of techniques
  1. Greedy techniques
  2. Spectral optimization
  3. Simulated annealing
  4. Extremal optimization **[Fortunato '10]**

Greedy techniques

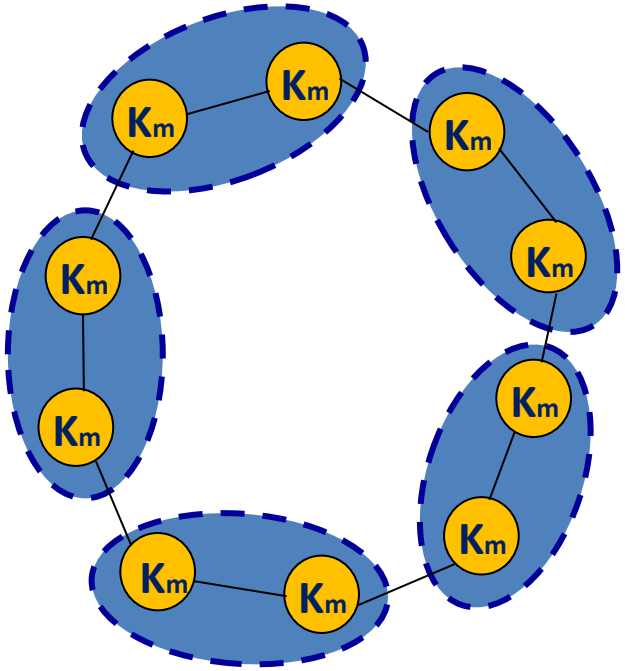
- Newman's algorithm **[Newman '04b]**
  - Agglomerative (bottom-up) hierarchical clustering algorithm
  - **Idea:** Repeatedly join pairs of communities that achieve the greatest increase of modularity (dendrogram representation)

  1. Initially, each node of the graph belongs on its own cluster (**n**)
  2. Repeatedly, join communities in pairs by adding edges
     a. At each step, choose the pairs that achieve the **greatest increase** (or minimum decrease) of modularity
     b. Consider only pairs of communities **between which there exist edges** (merging communities that do not share edges, it can never improve modularity)

  - Complexity: **O((m+n) n)** (or **$O(n^2)$** on a sparse graph)

Resolution limit of modularity

- **Resolution Limit** of modularity **[Fortunato and Barthelemy '07]**
- The method of modularity optimization may not detect communities with relatively small size, which depends on the total number of edges in the graph



- **Km** are cliques with **m** edges (**m ≤ sqrt(|E|)**)
- **Km** represent well-defined clusters
- However, the maximum modularity corresponds to clusters formed by two or more cliques
- It is difficult to know if the community returned by modularity optimization corresponds to a **single community** or a **union of smaller communities**

References (modularity)

- M.E.J. Newman and M. Girvan. Finding and evaluating community structure in networks. Physical Review E 69(02), 2004.
- M.E.J. Newman. Modularity and community structure in networks. PNAS, 103(23), 2006.
- S.E. Schaeffer. Graph clustering. Computer Science Review 1(1), 2007.
- S. Fortunato. Community detection in graphs. Physics Reports 486 (3-5), 2010.
- M. Coscia, F. Giannotti, and D. Pedreschi. A classification for community discovery methods in complex networks. Statistical Analysis and Data Mining 4 (5), 2011.
- A. Arenas, J. Duch, A. Fernandez, and S. Gomez. Size reduction of complex networks preserving modularity. New J. Phys., 9(176), 2007.
- M. Girvan and M.E.J. Newman. Community structure in social and biological networks. PNAS 99(12), 2002.
- U. Brandes, D. Delling, M. Gaertler, R. Gorke, M. Hoefer, Z. Nikoloski, and D. Wagner. On Modularity Clustering. IEEE TKDE 20(2), 2008.
- M.E.J. Newman. Fast algorithm for detecting community structure in networks. Phys. Rev. E 69, 2004.
- A. Clauset, M.E.J. Newman, and C. Moore. Finding community structure in very large networks. Phys. Rev. E 70, 2004.

References (modularity)

- M.E.J. Newman. Finding community structure in networks using the eigenvectors of matrices. Phys. Rev. E 74, 2006.
- R. Guimera, M. Sales-Pardo, L.A.N. Amaral. Modularity from Fluctuations in Random Graphs and Complex Networks. Phys. Rev. E 70, 2004.
- J. Duch and A. Arenas. Community detection in complex networks using Extremal Optimization. Phys. Rev. E 72, 2005.
- A. Arenas, J. Duch, A. Fernandez, and S. Gomez. Size reduction of complex networks preserving modularity. New Journal of Physics 9(6), 2007.
- E.A. Leicht and M.E.J. Newman. Community structure in directed networks. Phys. Rev. Lett. 100, 2008.
- V. Nicosia, G. Mangioni, V. Carchiolo, and M. Malgeri. Extending the definition of modularity to directed graphs with overlapping communities. J. Stat. Mech. 03, 2009.
- S. Muff, F. Rao, A. Caflisch. Local modularity measure for network clusterizations. Phys. Rev. E, 72, 2005.
- S. Fortunato and M. Barthelemy. Resolution limit in community detection. PNAS 104(1), 2007.

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Τέλος Ενότητας # 4

Μάθημα: Εξόρυξη γνώσης από Βάσεις Δεδομένων και τον Παγκόσμιο Ιστό, **Ενότητα # 4:** Unsupervised Learning (Clustering)

Διδάσκων: Μιχάλης Βαζιργιάννης**, Τμήμα:** Προπτυχιακό Πρόγραμμα Σπουδών "Πληροφορικής"

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο

ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης