CS 277, Data Mining

Cluster Analysis

Padhraic Smyth
Department of Computer Science
Bren School of Information and Computer Sciences
University of California, Irvine

Announcements

- Assignment 1
  - Questions?
  - Due Wednesday, hardcopy in class, code to EEE
    - Be sure to document your code
    - Be sure to clearly explain graphs/plots in your report

- Office Hours
  - Grader (Maryam): Mondays, 4 to 5
  - Professor: Tuesdays, 10 to 11:30

- Projects
  - Will discuss on Wednesday

- EEE Message Board
  - Now open for discussion – intended for inter-student communication

Outline of Today's Lecture

- Clustering Algorithms
  - K-means algorithm
  - Distance measures and multidimensional geometry
  - Hierarchical clustering algorithms
  - Probabilistic model-based clustering
  - Graph-based/spectral clustering

Notation

- N objects, each with d measurements/variables/attributes
  - data vector for ith object,

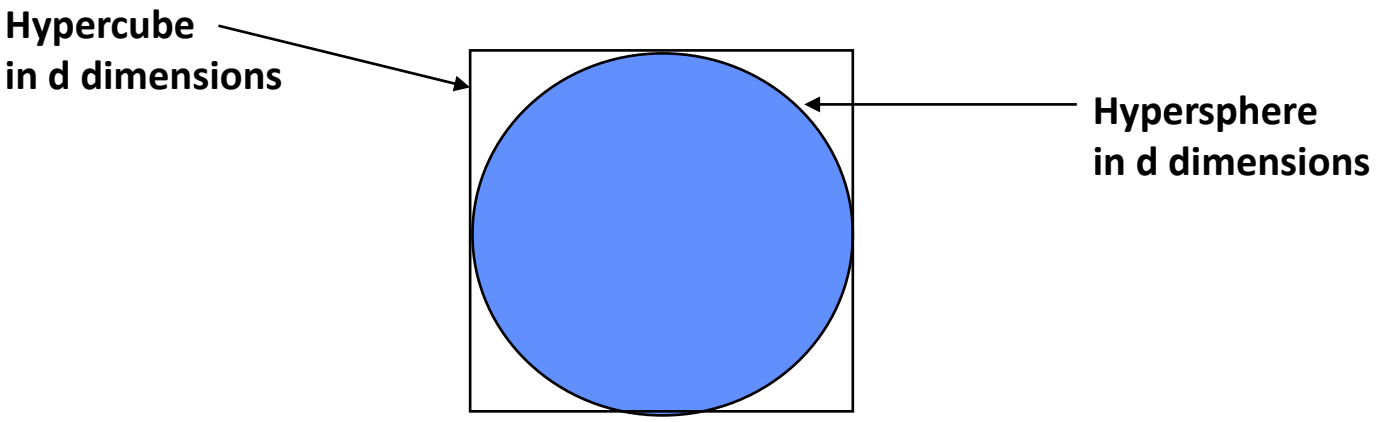
$$x_i = (x_{i1}, x_{i2}, \ldots, x_{id})$$

- Data matrix
  - x_{ij} is the ith row, jth column
  - columns correspond to variables
  - rows correspond to objects or data vectors

• For real-valued data we can think of our points as being in a d-dimensional space, e.g., clusters as “clouds of points”

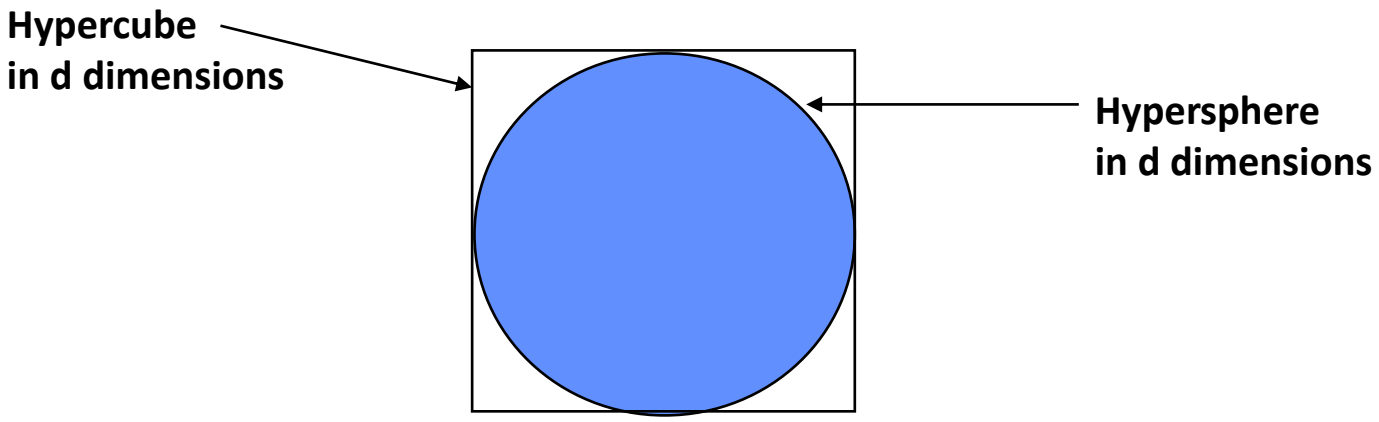
High-Dimensional Data

(From David Scott, *Multivariate Density Estimation,* Wiley, 1992)



High-Dimensional Data

(From David Scott, *Multivariate Density Estimation,* Wiley, 1992)

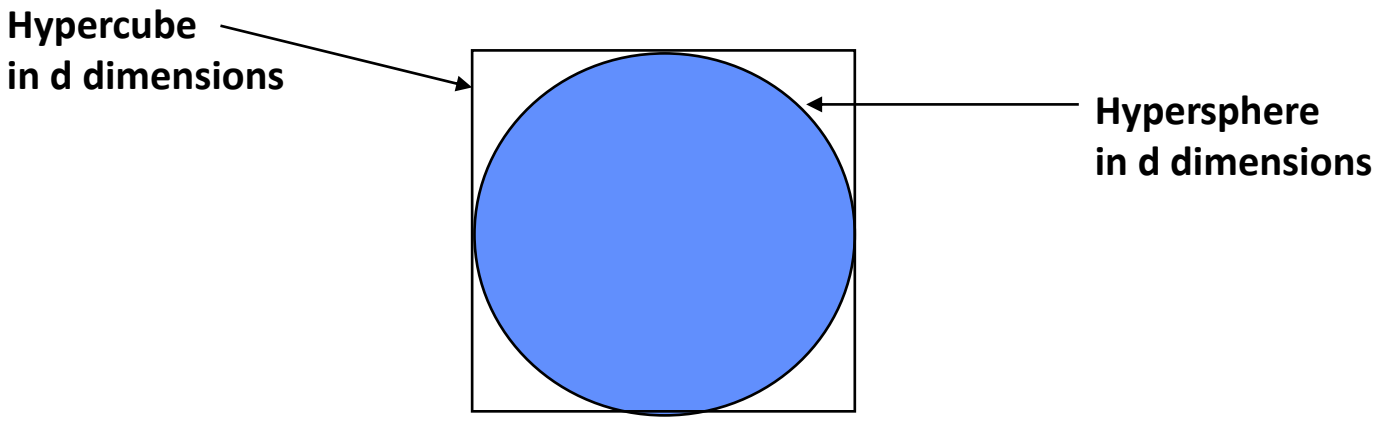


What is the volume of the sphere relative to the cube in d dimensions?

Dimension	2	3	4	5	6	7
Relative Volume	0.79	?	?	?	?	?

High-Dimensional Data

(From David Scott, *Multivariate Density Estimation,* Wiley, 1992)



What is the volume of the sphere relative to the cube in d dimensions?

Dimension	2	3	4	5	6	7
Relative Volume	0.79	0.53	0.31	0.16	0.08	0.04

Euclidean Distance

- n objects each with d real-valued measurements

$$x = (x_1, x_2, \ldots, x_d)$$
$$y = (y_1, y_2, \ldots, y_d)$$

- Most common distance metric is *Euclidean* distance:

$$d_E(x, y) = \left(\sum_{k=1}^{d} (x_k - y_k)^2 \right)^{\frac{1}{2}}$$

- Makes sense in the case where the different measurements are commensurate; each variable on roughly on the same scale

- Recommended to prestandardize the data (divide each variable by its range or variance) before using – otherwise some variables may dominate

Distances between Binary Vectors

	i=1	j=0
i=1	n_{11}	n_{10}
i=0	n_{01}	n_{00}

Number of variables where item j =1 and item i = 0

- matching coefficient

$$\frac{n_{11} + n_{00}}{n_{11} + n_{10} + n_{01} + n_{00}}$$

- Jaccard coefficient (e.g., for sparse vectors, 1's much rarer than 0's)

$$\frac{n_{11}}{n_{11} + n_{10} + n_{01}}$$

Other Types of Distances

- Categorical variables
  - d(x, y) = number of matching values divided by number of dimensions

- Distances between strings of different lengths
  - e.g., “Joe Brown” and “Joseph F. Brown”
  - Can use edit distance or semantic distance ,e.g., d(“Joe” and “Joseph”) is small

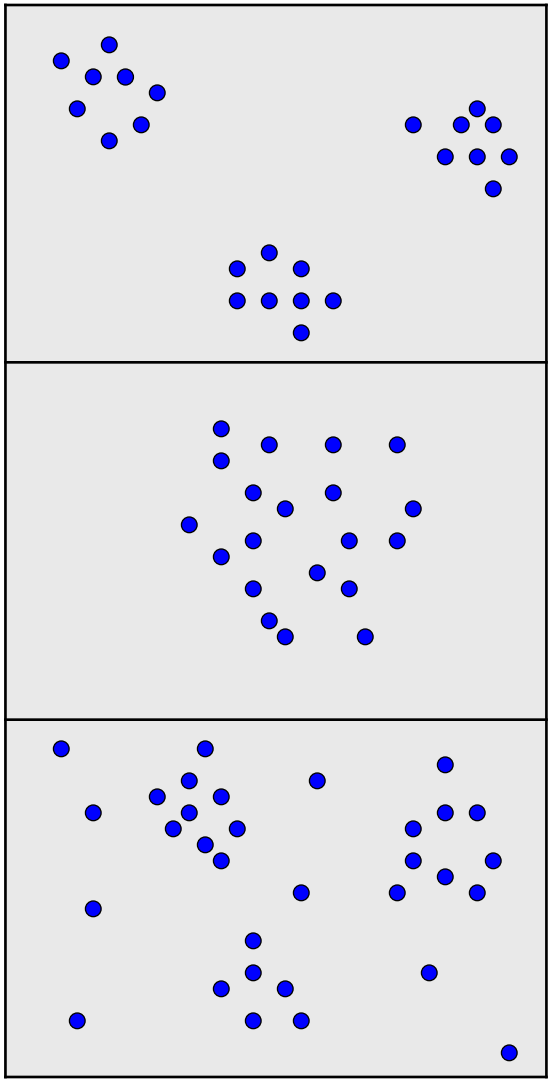
- Distances between images and waveforms
  - Shift-invariant, scale invariant
    e.g., d(x,y) = min_{a,b} { (ax+b) – y }
  - This is in effect another form of edit distance
  - See also dynamic time-warping

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 d-dimensional space

- Important points to keep in mind
  - There is often "no best clustering" for a data set
  - Can think of clustering as a potentially useful way to group data points
  - Different clustering algorithms provide different groupings

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
  - Clustering results produced by a search engine
  - Segmentation of patients in a medical study
  - Discovery of new types of galaxies in astronomical data
  - Clustering of genes with similar expression profiles

  - …. many more

Other Issues in Clustering

- Distance function, d(x,y) is often a critical aspect of clustering, both
  - distance of individual pairs of objects
  - distance of individual objects from clusters

- How is K, number of clusters, selected?

- Different types of data
  - Real-valued versus categorical
  - Input data: N vectors or an N^2 distance matrix?

Different Types of Clustering Algorithms

- partition-based clustering
  - Represent points as vectors and partition points into clusters based on distance in d-dimensional space
- probabilistic model-based clustering
  - e.g. mixture models

  [both work with measurement data, e.g., feature vectors]
- hierarchical clustering
  - Builds a tree (dendrogram) starting from an N x N distance matrix between objects
- graph-based/spectral clustering (not discussed)
  - represent inter-point distances via a graph and apply graph algorithms

Different Types of Input to Clustering Algorithms

- Data matrix
  - N rows
  - d columns

$$\begin{bmatrix} x_{11} & \dots & x_{1f} & \dots & x_{1d} \\ \dots & \dots & \dots & \dots & \dots \\ x_{i1} & \dots & x_{if} & \dots & x_{id} \\ \dots & \dots & \dots & \dots & \dots \\ x_{n1} & \dots & x_{nf} & \dots & x_{nd} \end{bmatrix}$$

- Distance matrix
  - N x N distances
  - between objects

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ : & : & : & & \\ d(n,1) & d(n,2) & \dots & \dots & 0 \end{bmatrix}$$

Clustering: The K-Means Algorithm

The K-Means Clustering Algorithm

- Input:
  - N real-valued vectors $\underline{x}_1, \ldots \underline{x}_N$ of dimension d
  - K = number of clusters required (K > 1)

- Output:
  - K cluster centers, $\underline{c}_1, \ldots \underline{c}_K$, each center is a vector of dimension d
  - A list of cluster assignments (values 1 to K) for each of the N input vectors

Squared Errors and Cluster Centers

- Squared error (distance) between a data point $\underline{x}$ and a cluster center $\underline{c}$:

$$d[\underline{x}, \underline{c}] = \sum_j (x_j - c_j)^2$$

Sum is over the d components/dimensions of the vectors

Squared Errors and Cluster Centers

- Squared error (distance) between a data point $\underline{x}$ and a cluster center $\underline{c}$:

$$d[\underline{x}, \underline{c}] = \sum_j (x_j - c_j)^2$$

Sum is over the d components/dimensions of the vectors

- Total squared error between a cluster center $\underline{c}(k)$ and all N_k points assigned to that cluster:

$$S_k = \sum_i d[\underline{x}_i, \underline{c}_k]$$

Distance is usually defined to be Euclidean distance

Sum is over the N_k points assigned to cluster k

Squared Errors and Cluster Centers

- Squared error (distance) between a data point $\underline{x}$ and a cluster center $\underline{c}$:

$$d[\underline{x}, \underline{c}] = \sum_j (x_j - c_j)^2$$

Sum is over the d components/dimensions of the vectors

- Total squared error between a cluster center $\underline{c}(k)$ and all N_k points assigned to that cluster:

$$S_k = \sum_i d[\underline{x}_i, \underline{c}_k]$$

Distance is usually defined to be Euclidean distance

Sum is over the N_k points assigned to cluster k

- Total squared error summed across K clusters

$$SSE = \sum_k S_k$$

Sum is over the K clusters

K-means Objective Function

- K-means: minimize the total squared error, i.e., find the K clusters centers m(k), and assignments, that minimize

$$SSE = \sum_k S_k = \sum_k \left(\sum_i d[\underline{x}_i, \underline{c}_k] \right)$$

- K-means seeks to minimize SSE, i.e., find the cluster centers such that the sum-squared-error is smallest
  - will place cluster centers strategically to "cover" data
  - similar to data compression (in fact used in data compression algorithms)

K-Means Algorithm

- Random initialization
  - Select the initial K centers randomly from N input vectors randomly
  - Or, assign each of the N vectors randomly to one of the K clusters

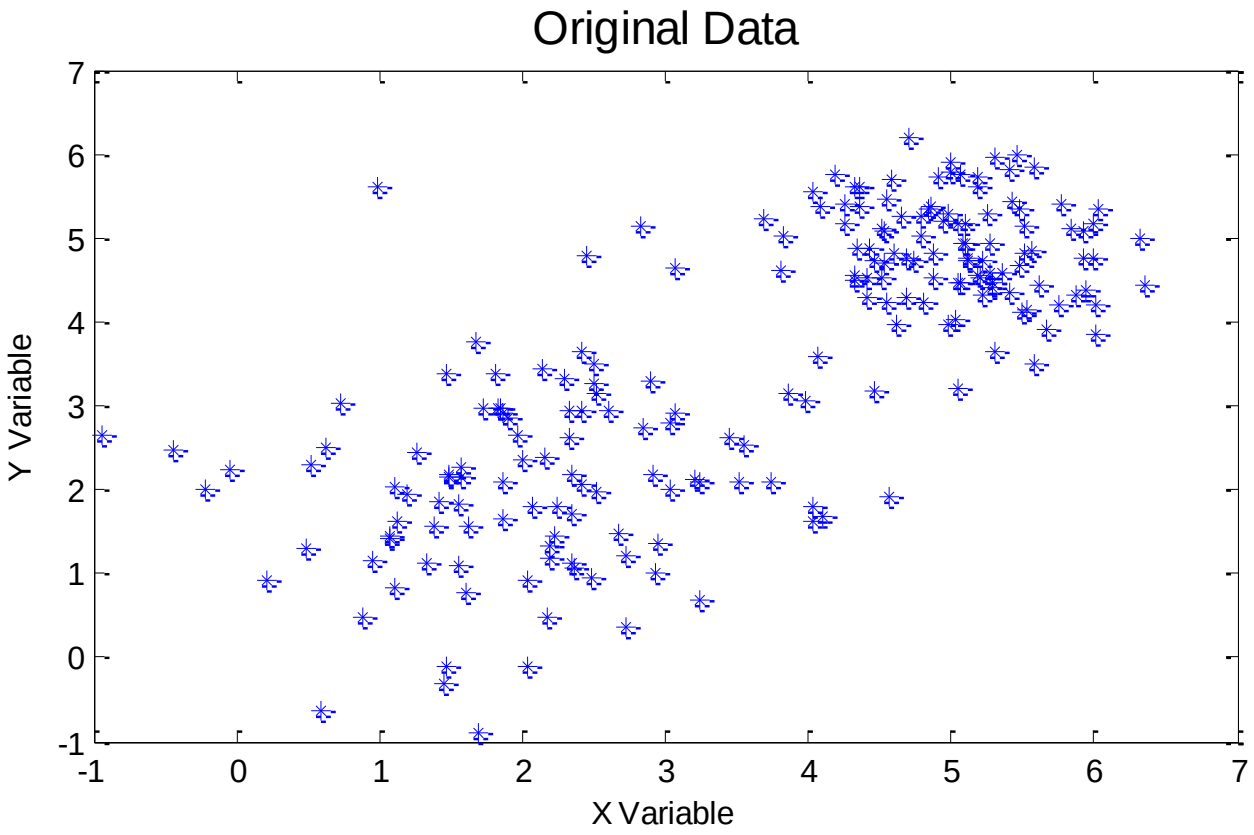
- Iterate:
  - Assignment Step:
    - Assign each of the N input vectors to their closest mean
  - Update the Mean-Vectors (K of them)
    - Compute updated centers: the average value of the vectors assigned to k

$$\text{New } \underline{c}_k = 1/N_k \sum_i \underline{x}_i$$

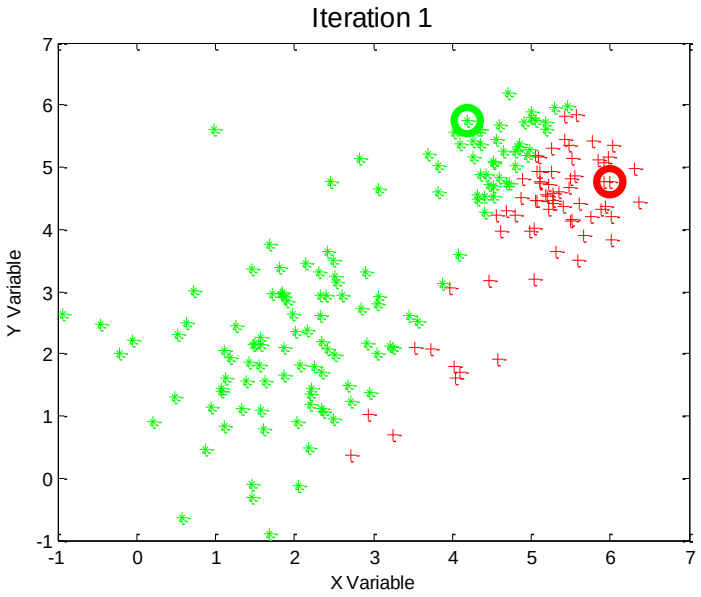
Sum is over the N_k points assigned to cluster k

- Convergence:
  - Did any points get reassigned?
    - Yes: terminate
    - No: return to Iterate step

Example of Running Kmeans

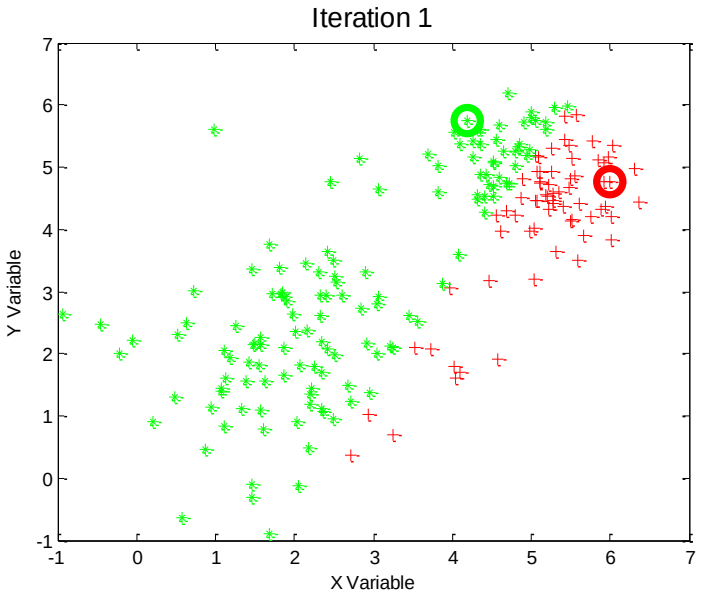


Example

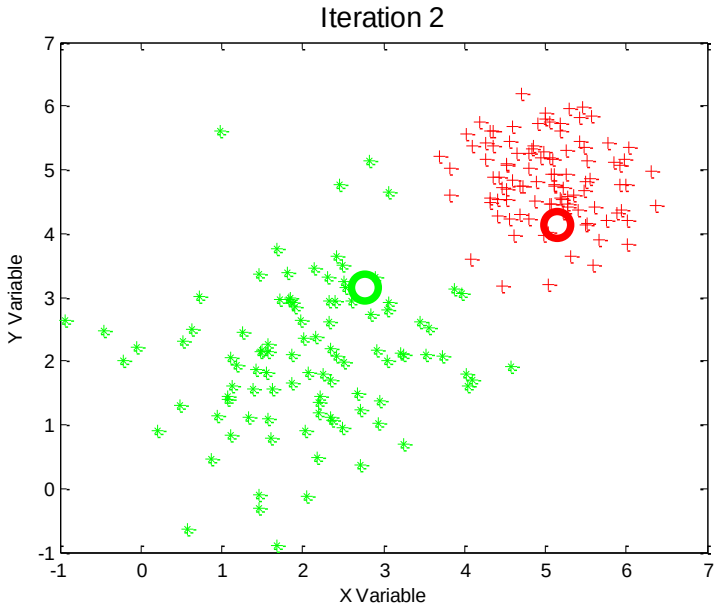


MSE Cluster 1 = 1.31
MSE Cluster 2 = 3.21
Overall MSE = 2.57

Example

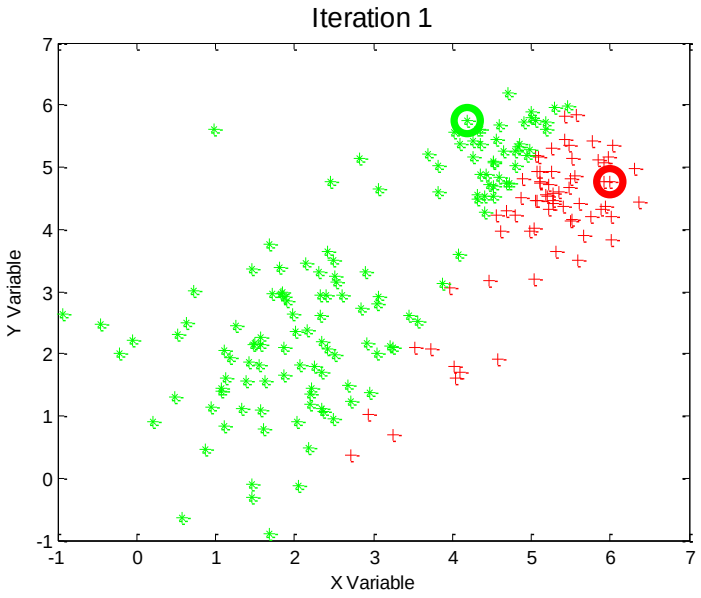


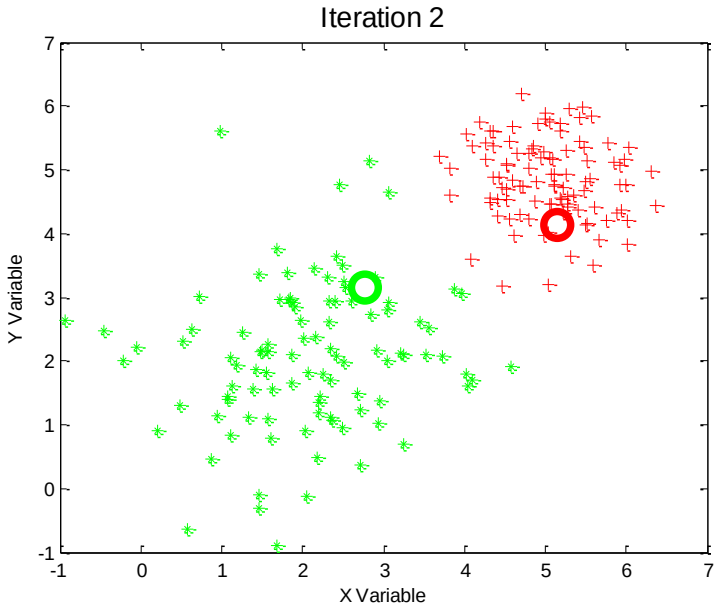
MSE Cluster 1 = 1.31
MSE Cluster 2 = 3.21
Overall MSE = 2.57

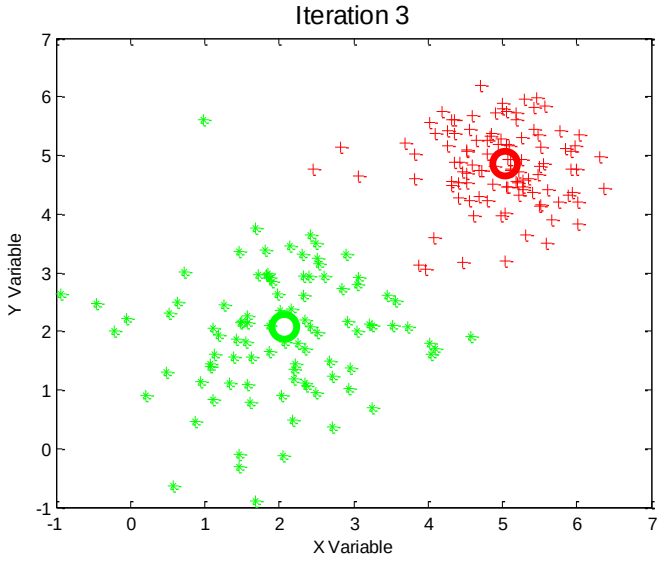


MSE Cluster 1 = 1.01
MSE Cluster 2 = 1.76
Overall MSE = 1.38

Example

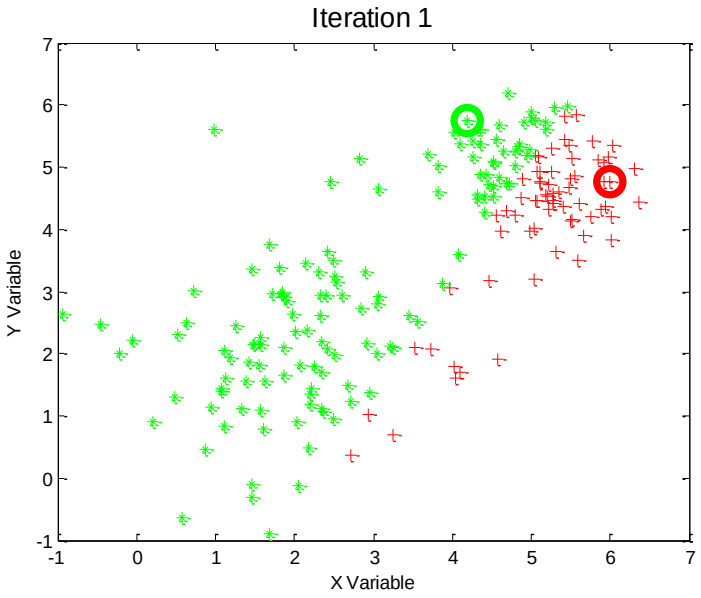


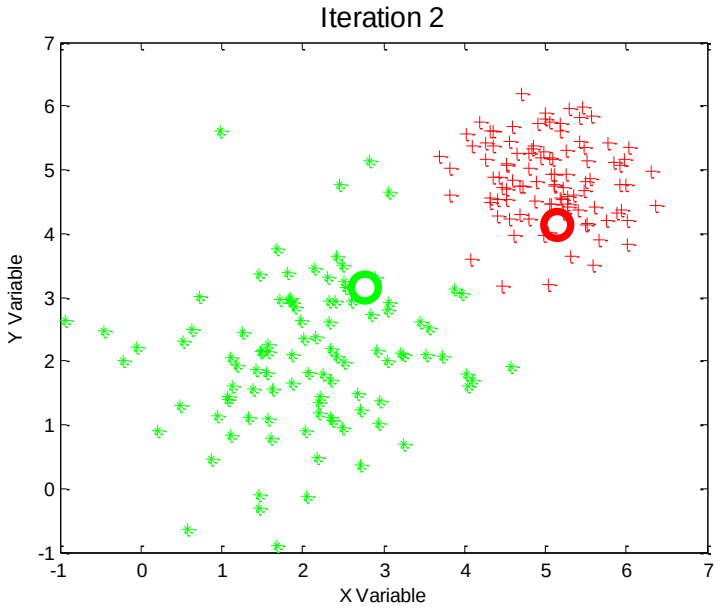


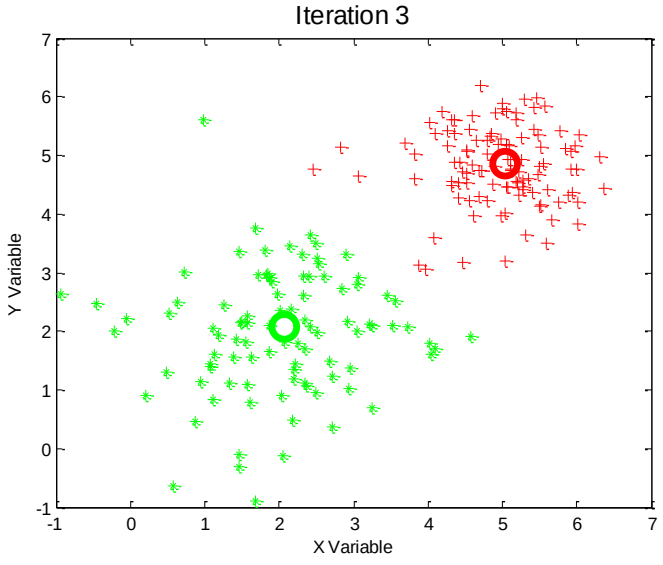


MSE Cluster 1 = 0.84
MSE Cluster 2 = 1.28
Overall MSE = 1.05

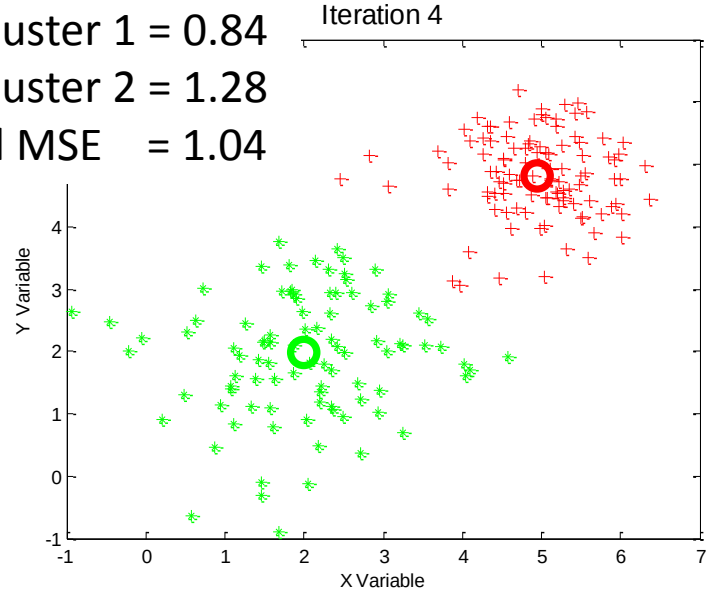
Example







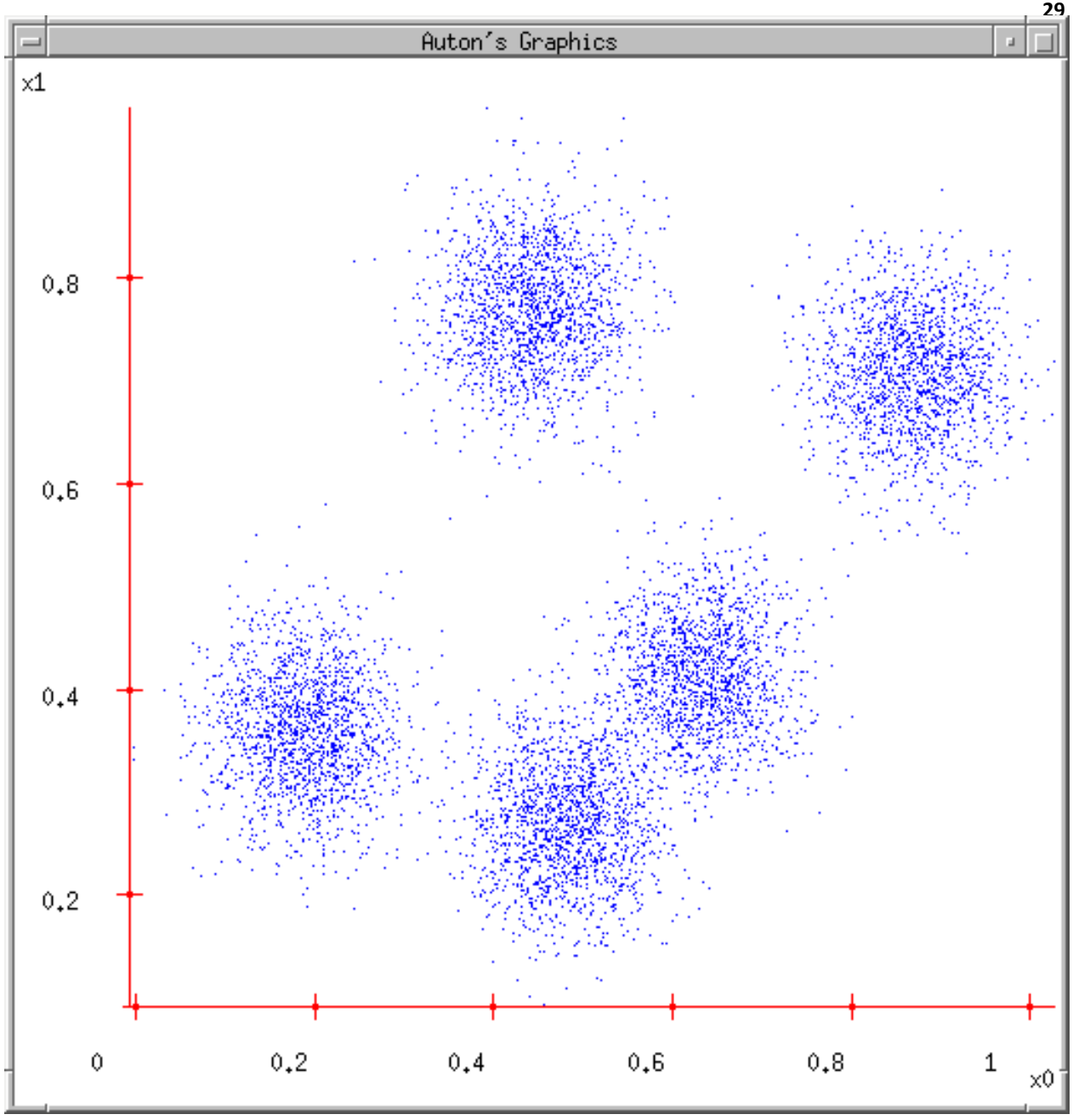
MSE Cluster 1 = 0.84
MSE Cluster 2 = 1.28
Overall MSE = 1.04



K-means

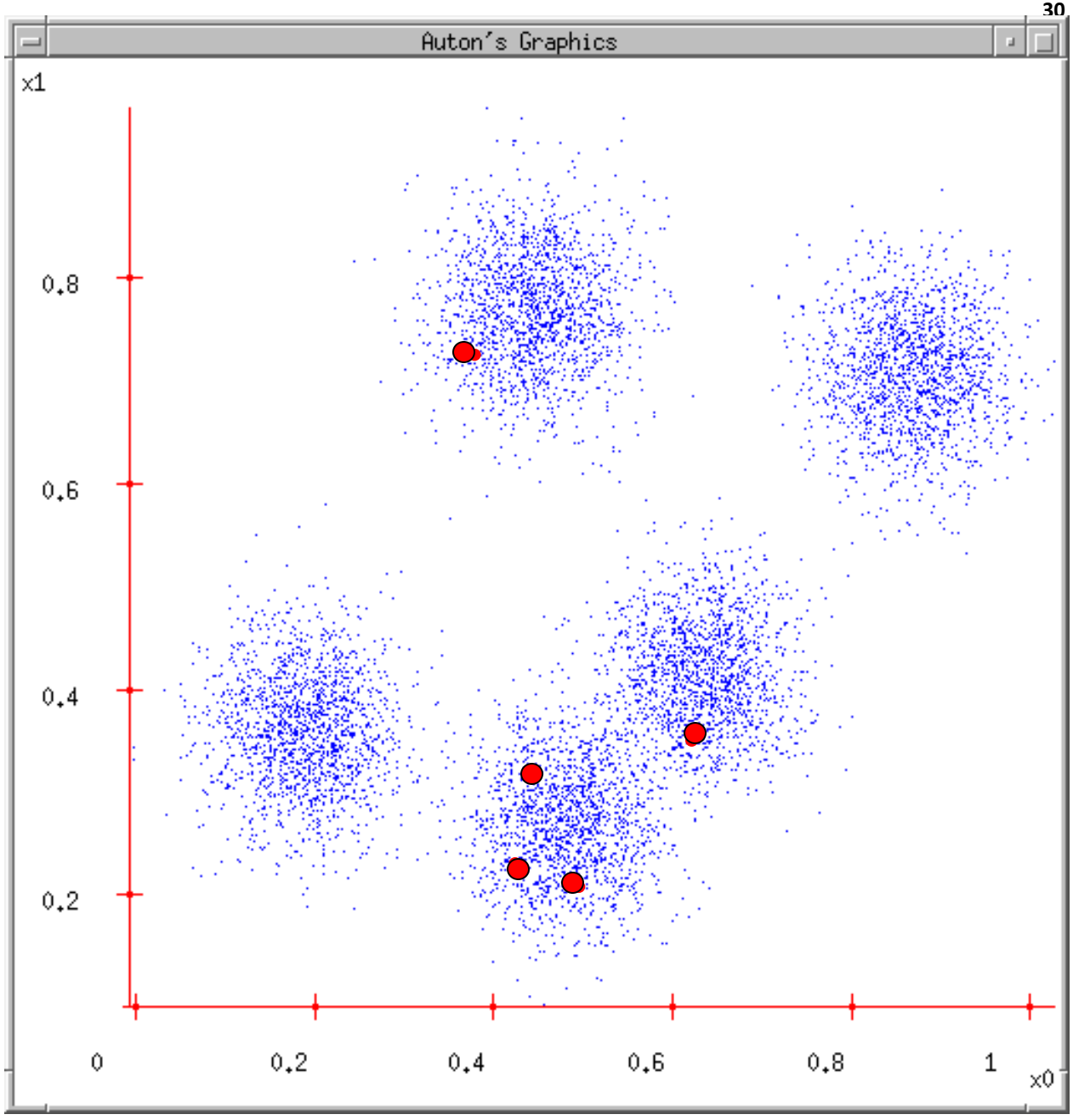
1. Ask user how many clusters they'd like. *(e.g. K=5)*

(Example is courtesy of Andrew Moore, CMU)



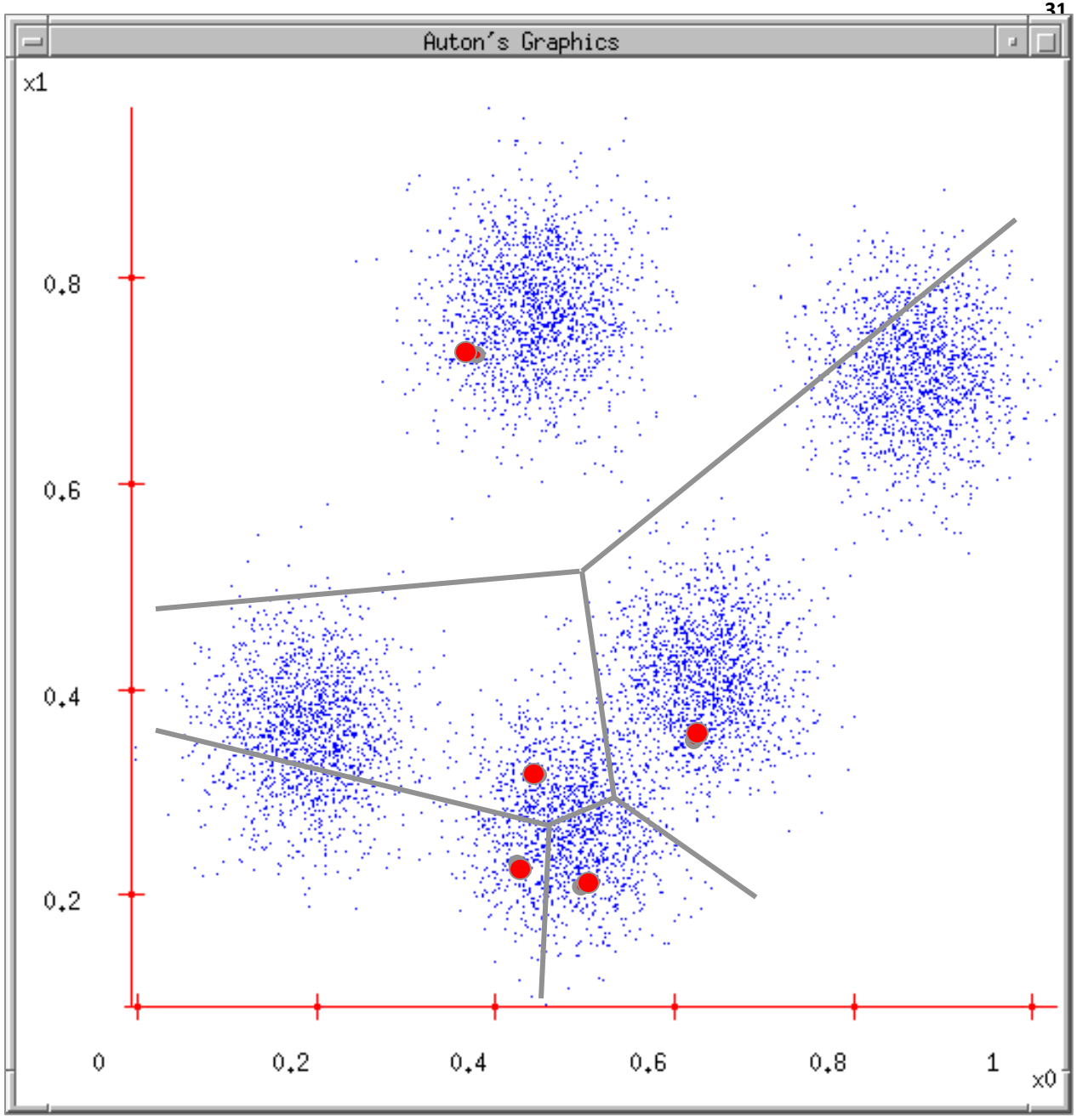
K-means

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



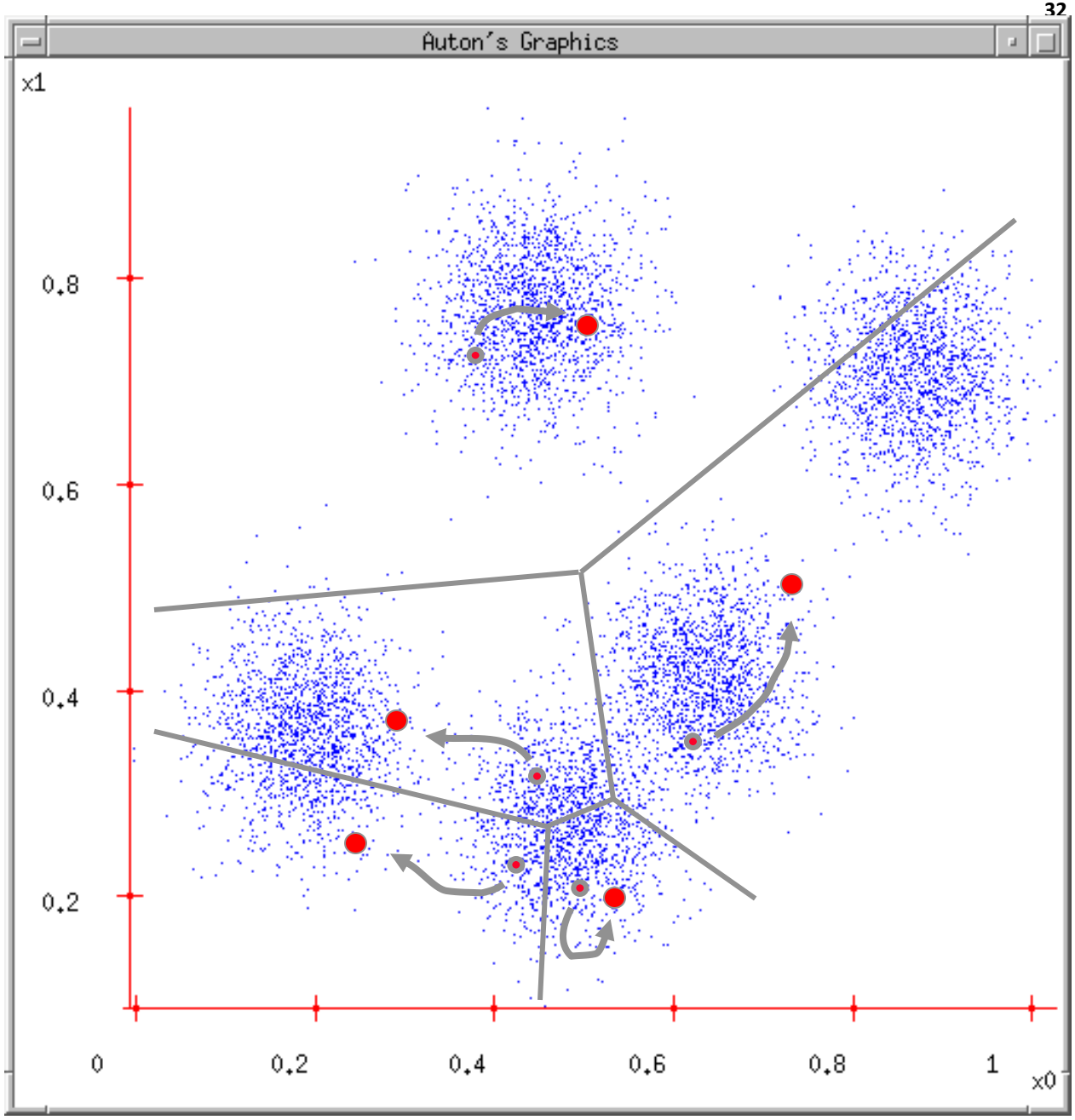
K-means

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.



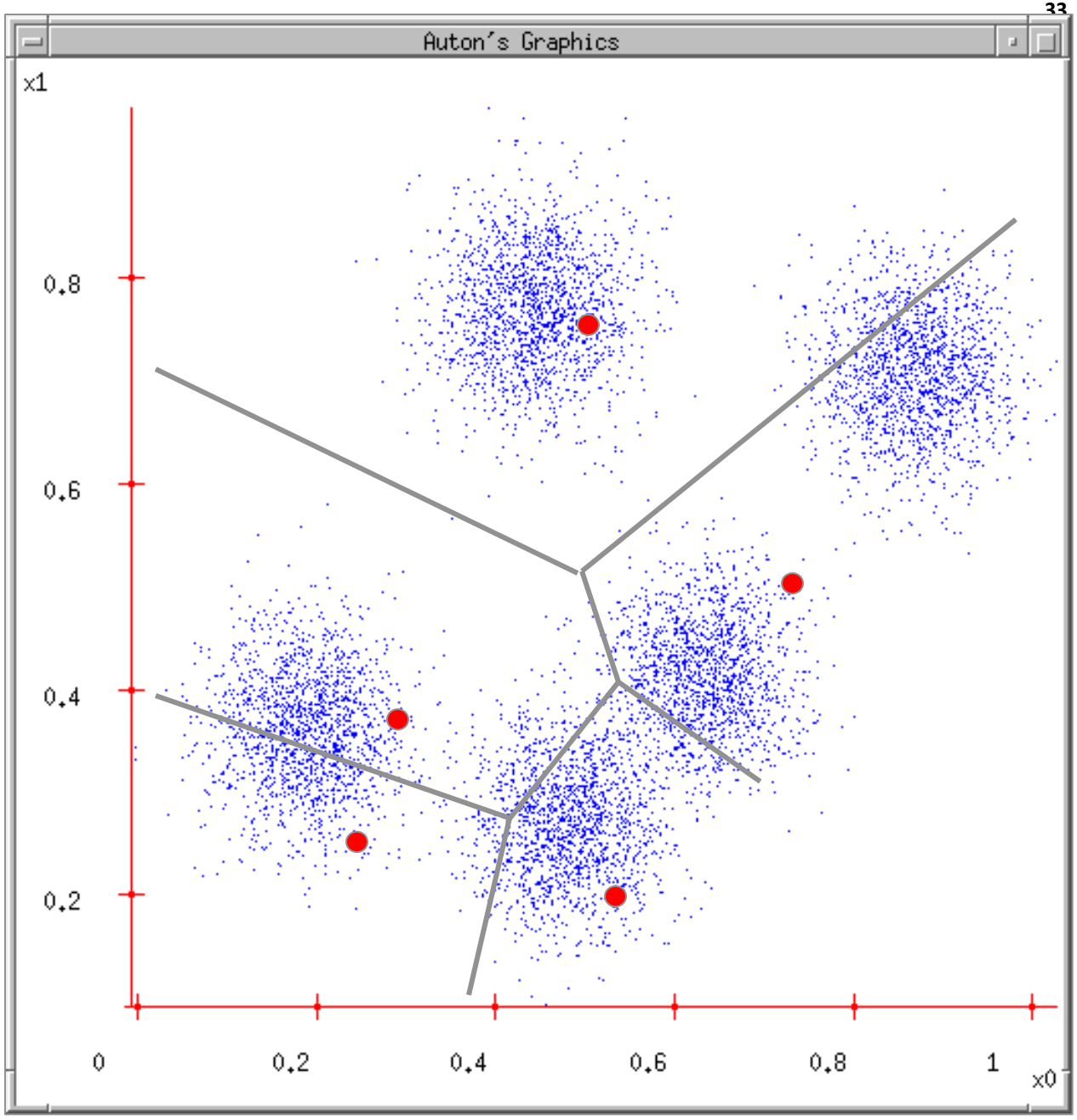
K-means

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

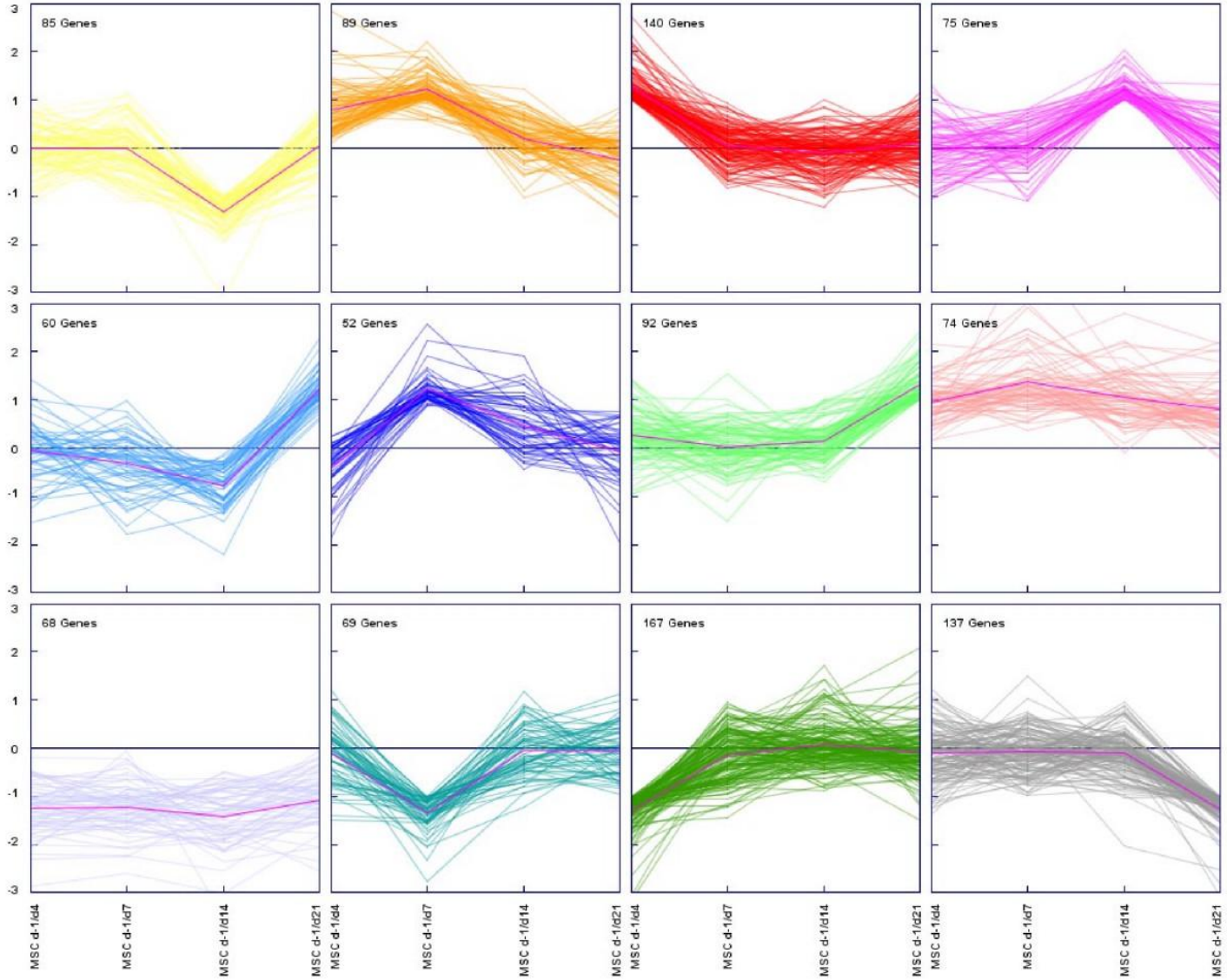
1. Ask user how many clusters they'd like. *(e.g. K=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns
5. New Centers => new boundaries
6. Repeat until no change



K-Means Clustering Applied to Time-Series Data (Gene Expression over Time)

1108 genes

From
Kulterer et al.,
BMC Genomics
2007



Clustering RGB Vectors for Image Compression

Original Image

Image after clustering
8-bit RGB vectors
into 11 clusters

Figures from David Forsyth, UC Berkeley

Properties of the K-Means Algorithm

- Time complexity??

  O(N K d) in time per iteration

  e = cost of distance computation, e.g., e = d for Euclidean distance in d dimensions

  This is good: linear time in each input parameter

- Convergence to Global Minimum?

  Does K-means always converge to the best possible solution?

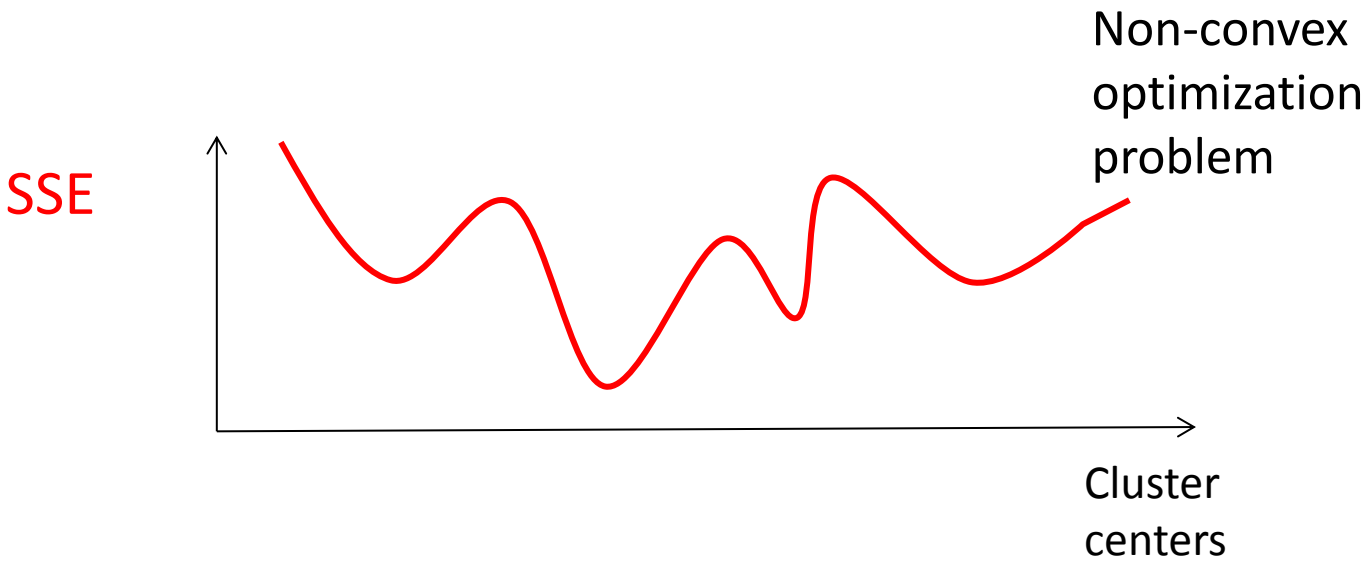
  i.e., the set of K centers that minimize the SSE?

  No: always converges to *some* solution, but not necessarily the best

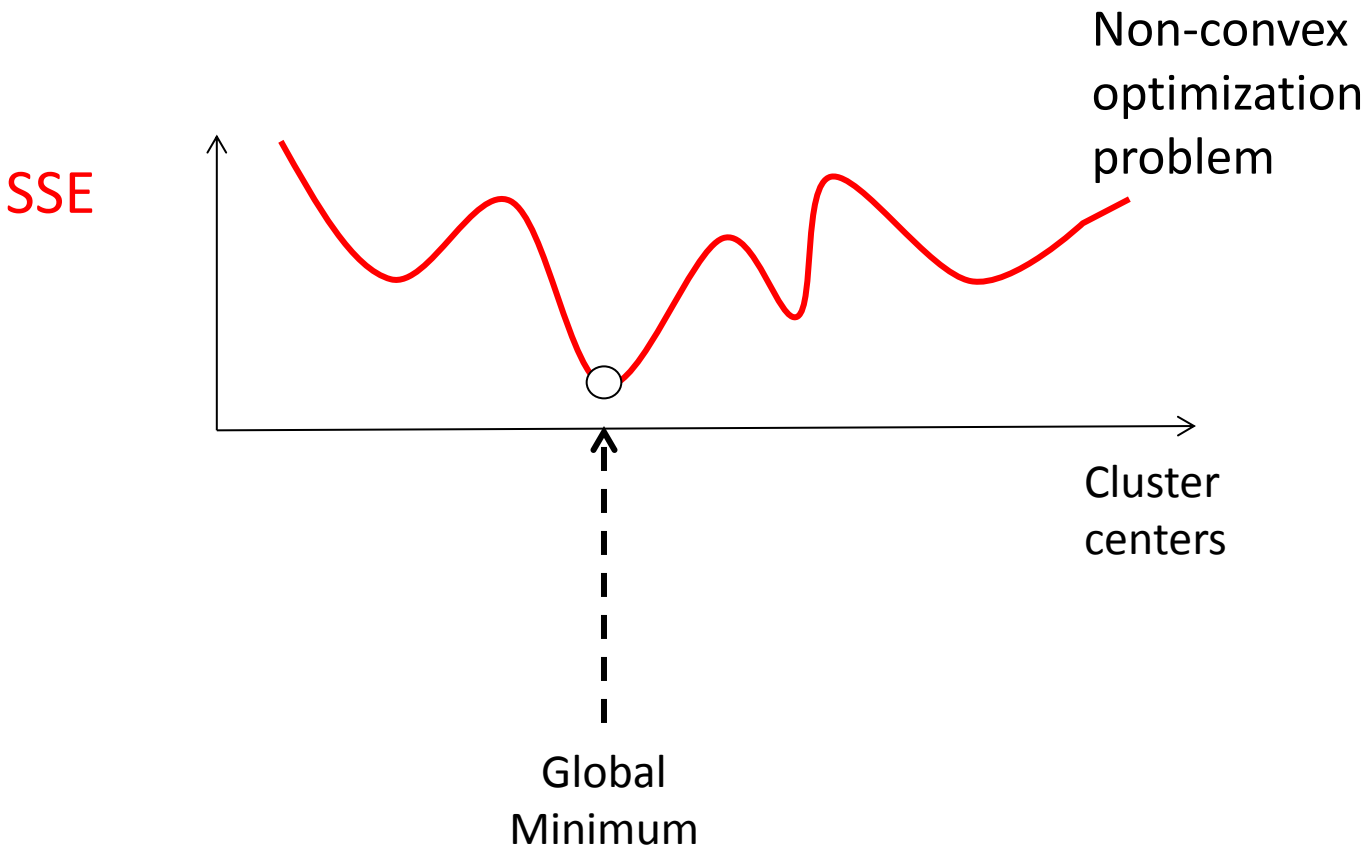
  - Depends on the starting point chosen

  To think about: prove that SSE always decreases after every iteration of the K-means algorithm, until convergence. (hint: need to prove that assignment step and computation of cluster centers both decrease the SSE)

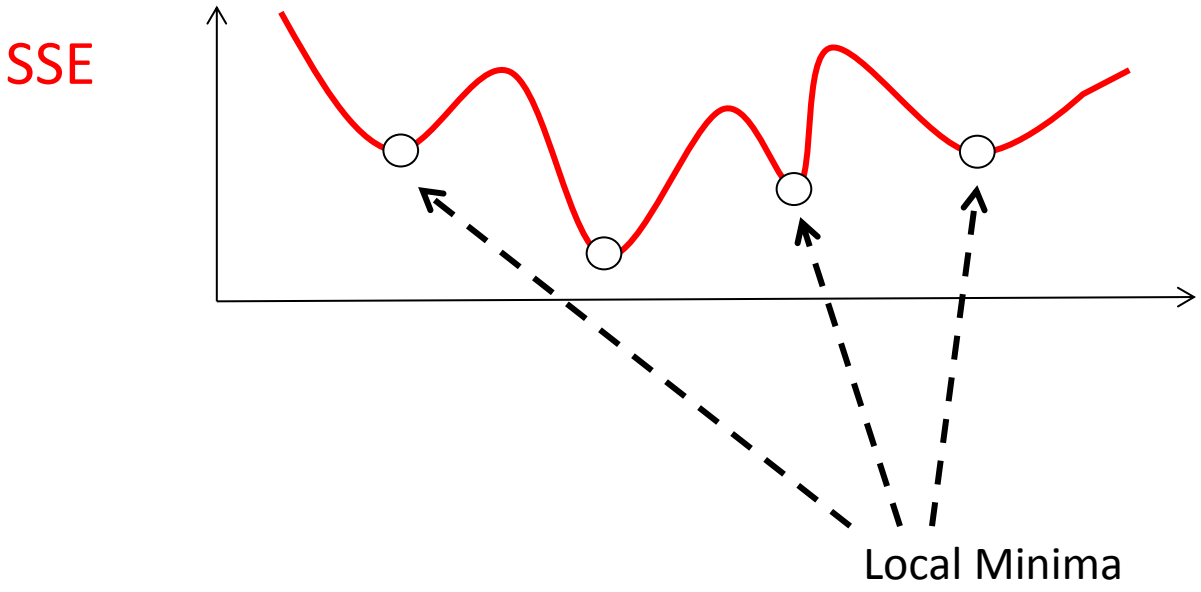
Local Search and Local Minima



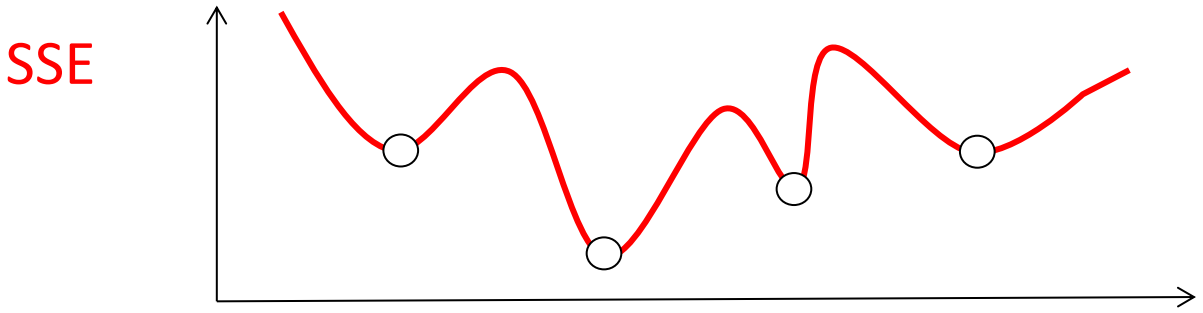
Local Search and Local Minima

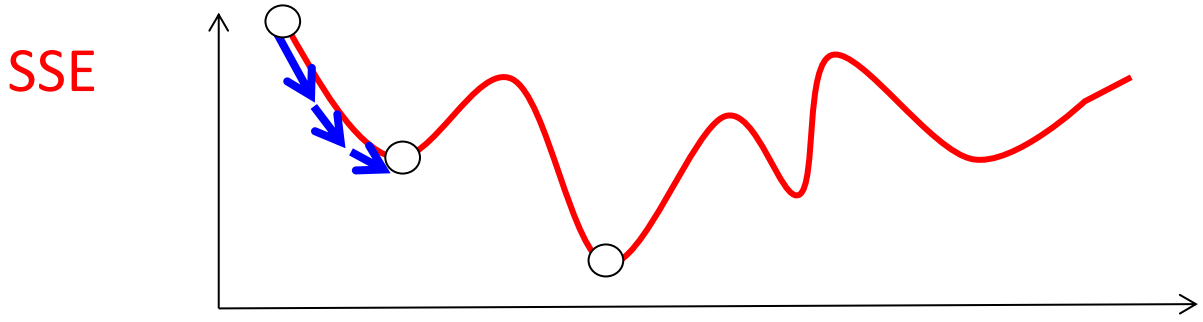


Local Search and Local Minima

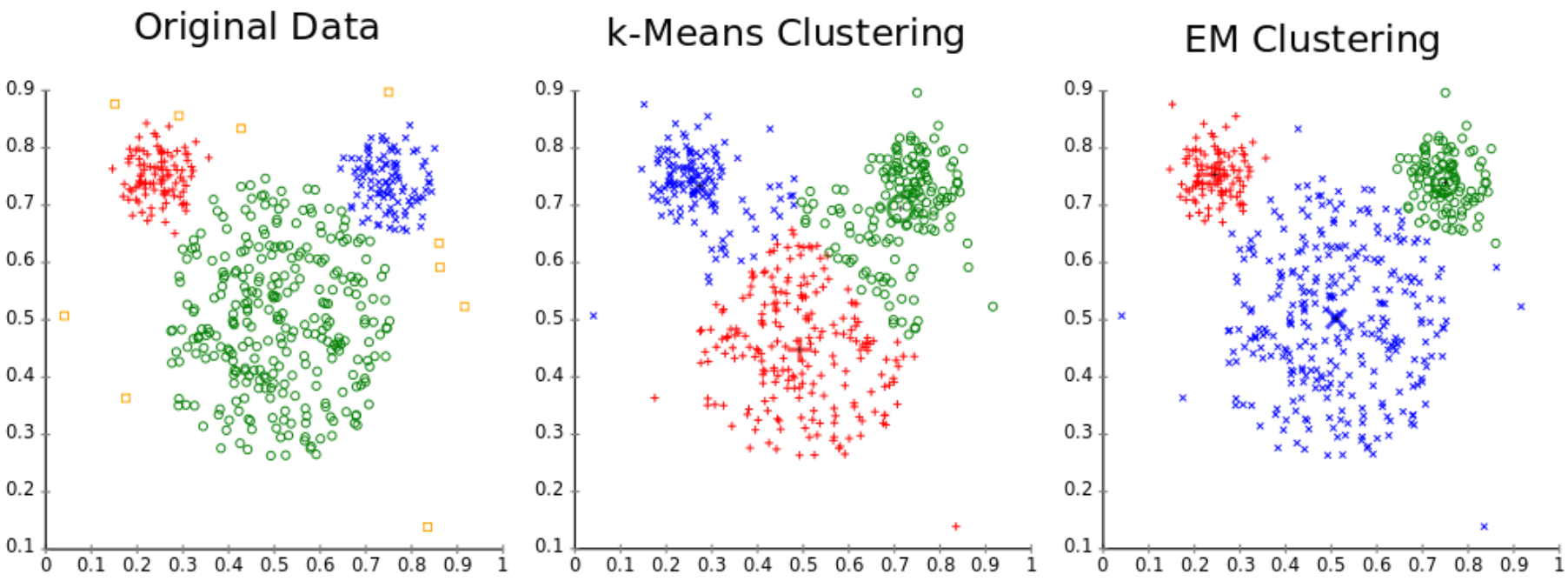


Local Search and Local Minima





Suboptimal Results from K-means on Simulated Data



Why does k-means not perform so well on this example?

Figure from http://en.wikipedia.org/wiki/K-means_clustering

Issues with 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
    - e.g., in signal processing/data compression is similar to vector-quantization

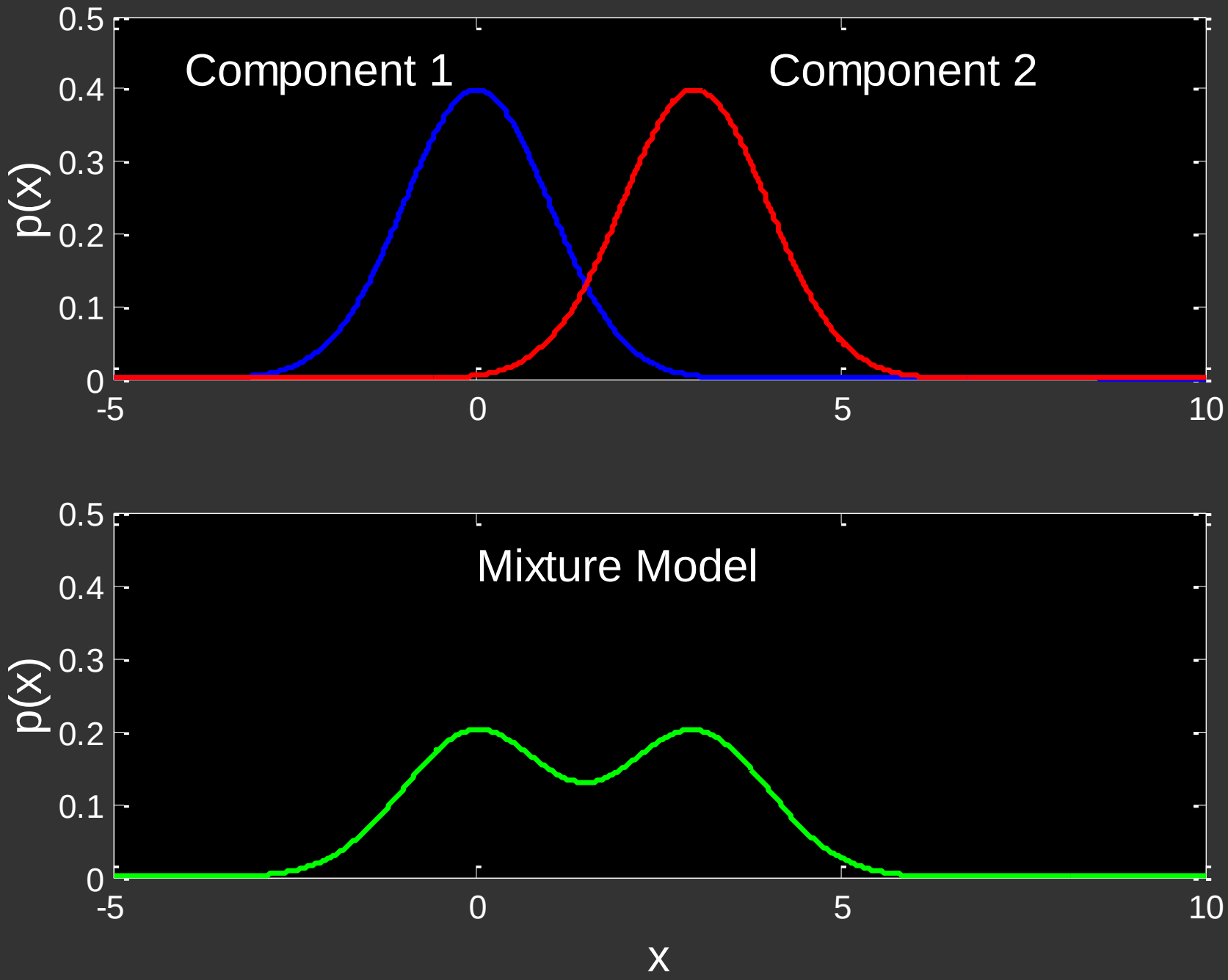
- Choice of distance measure
  - Euclidean distance
  - Weighted Euclidean distance
  - Many others possible

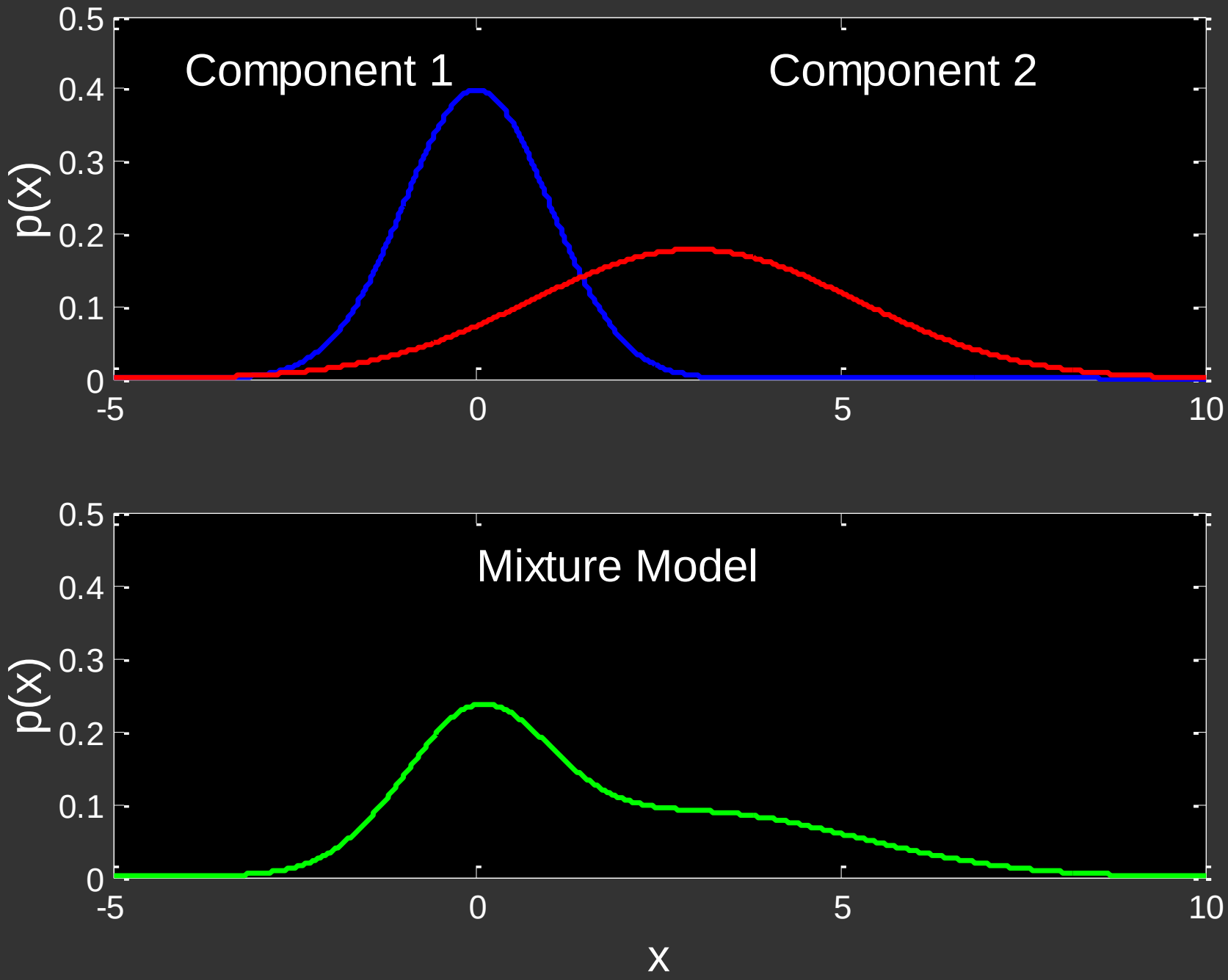
- Selection of K
  - "scree diagram" - plot SSE versus K, look for "knee" of curve
    - Limitation: may not be any clear K value

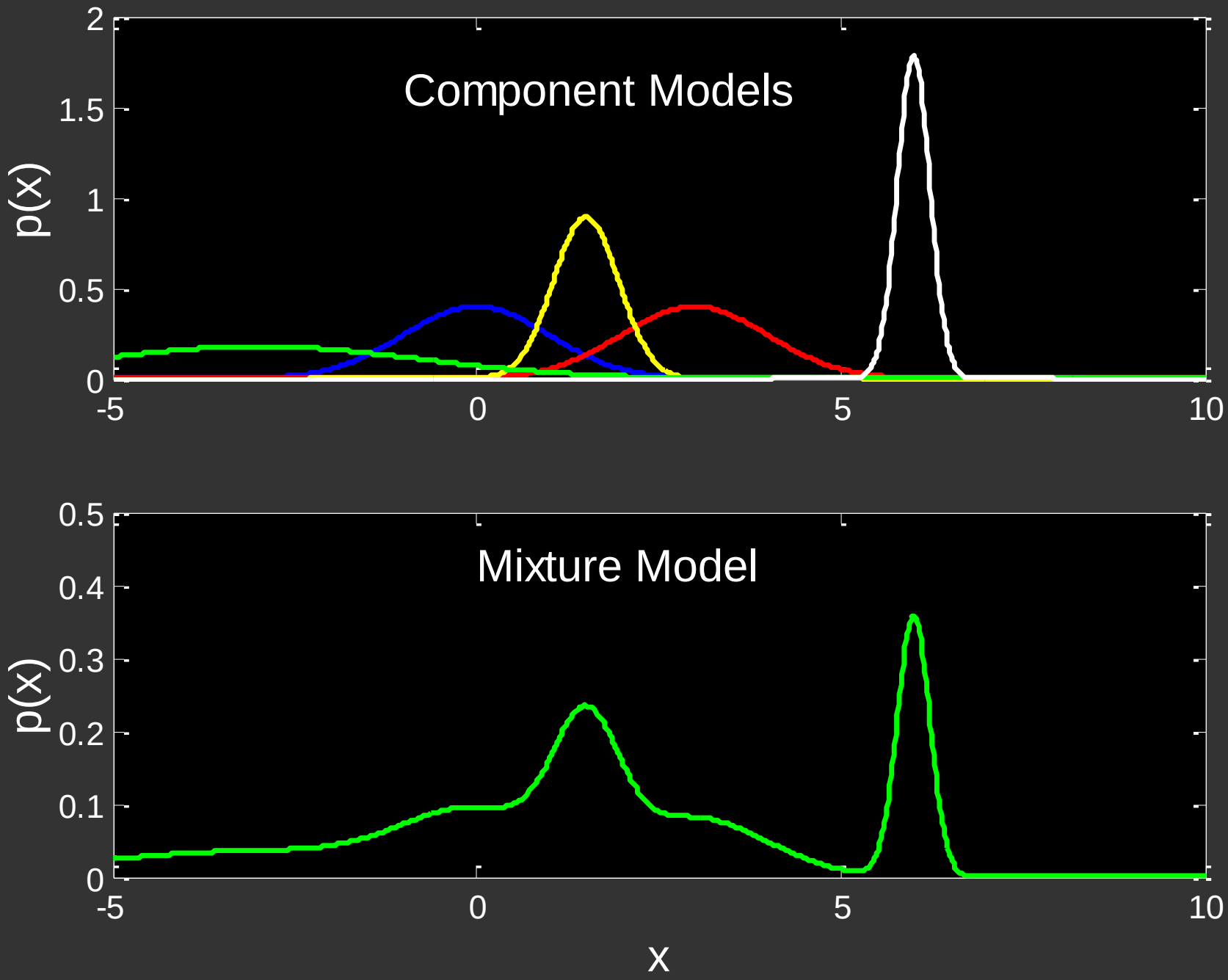
Clustering: Probabilistic Model-Based Clustering

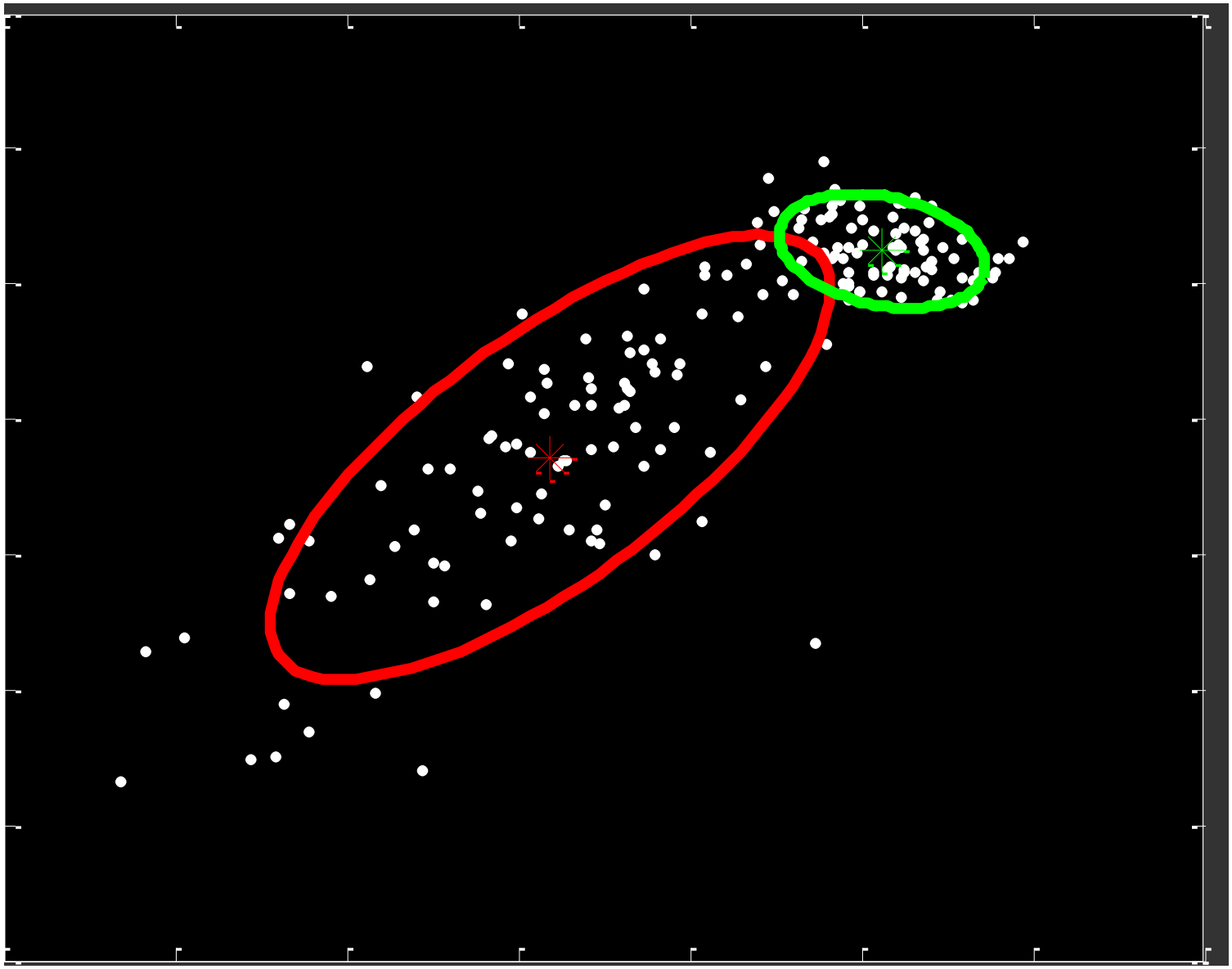
Probabilistic Clustering

- Hypothesize that the data are being generated by a mixture of K multivariate probability density functions (e.g., Gaussians)
  - Each density function is a cluster
  - Data vectors have a probability of belonging to a cluster rather than 0-1 membership
- Clustering algorithm
  - Learn the parameters (mean, covariance for Gaussians) of the K densities
  - Learn the probability memberships for each input vector
- Can be solved with the Expectation-Maximization algorithm
- Can be thought of as a probabilistic version of K-means


Component 1
Component 2
Mixture Model
p(x)
p(x)
0.5
0.4
0.3
0.2
0.1
0
-5
0
5
10
x

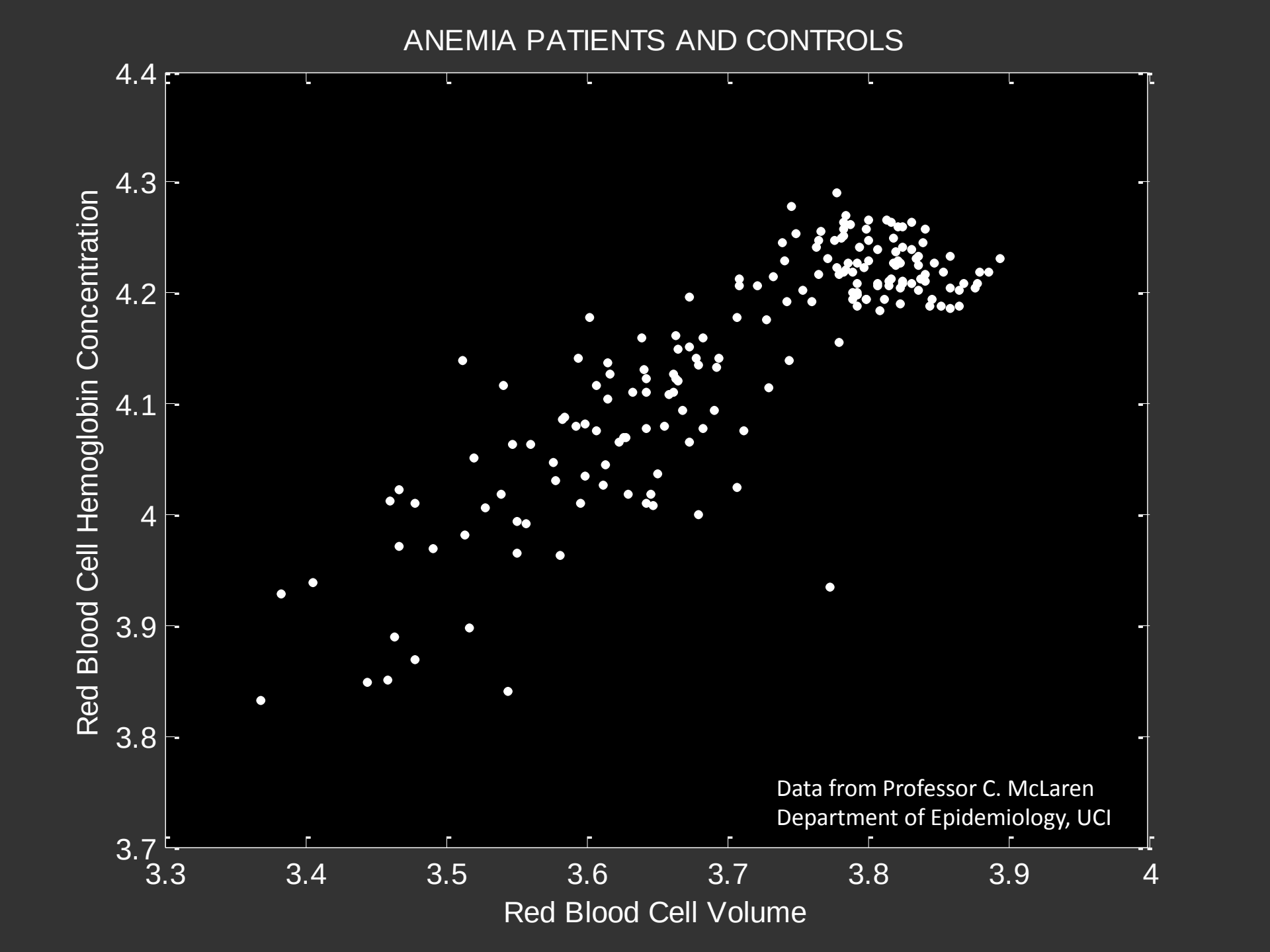

Component 1
Component 2
p(x)
0.5
0.4
0.3
0.2
0.1
0
-5
0
5
10
Mixture Model
p(x)
0.5
0.4
0.3
0.2
0.1
0
-5
0
5
10
x

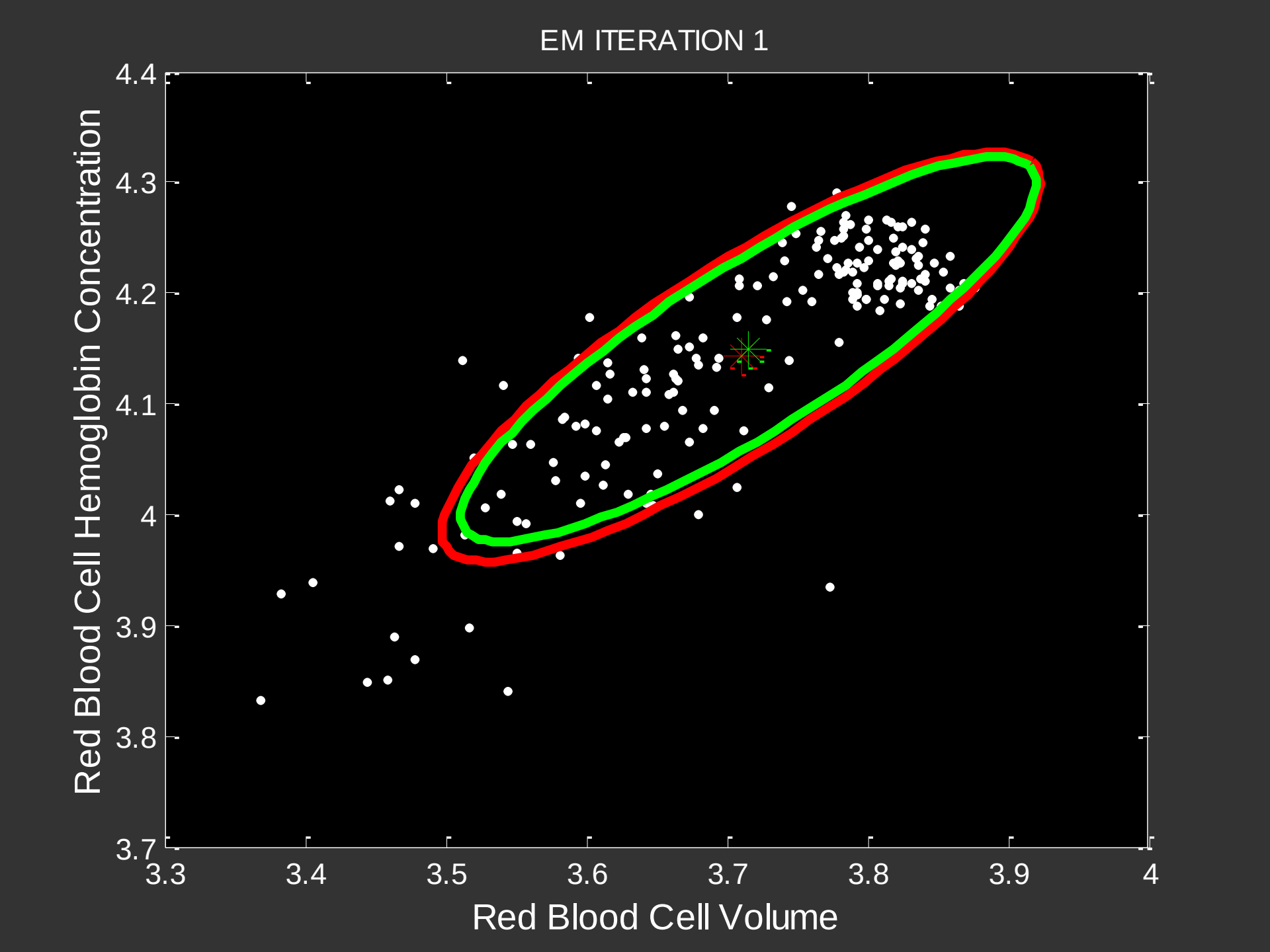

Component Models
p(x)
2
1.5
1
0.5
0
-5
0
5
10
Mixture Model
p(x)
0.5
0.4
0.3
0.2
0.1
0
-5
0
5
10
x

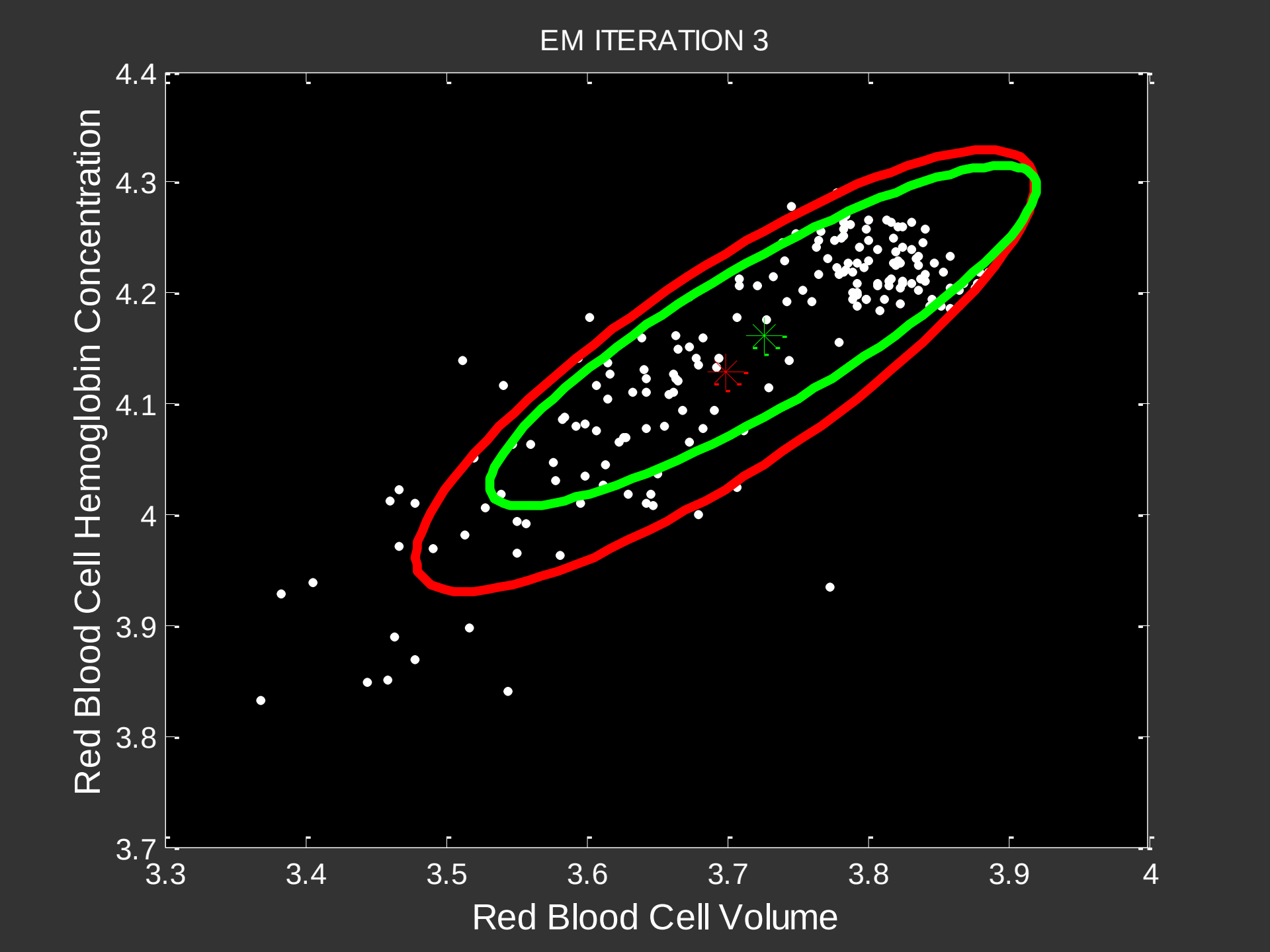


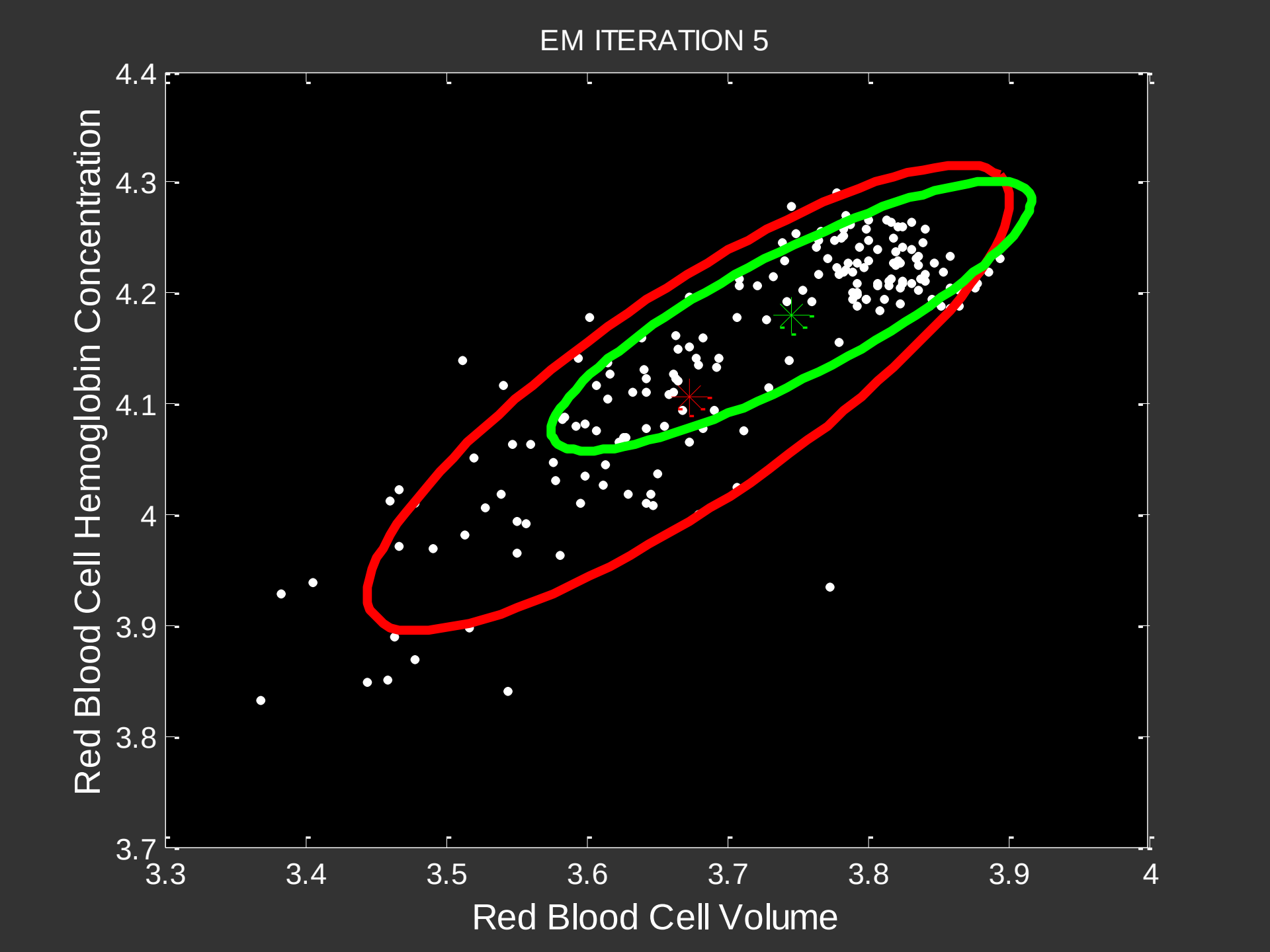
Expectation-Maximization (EM) for Gaussian Clustering

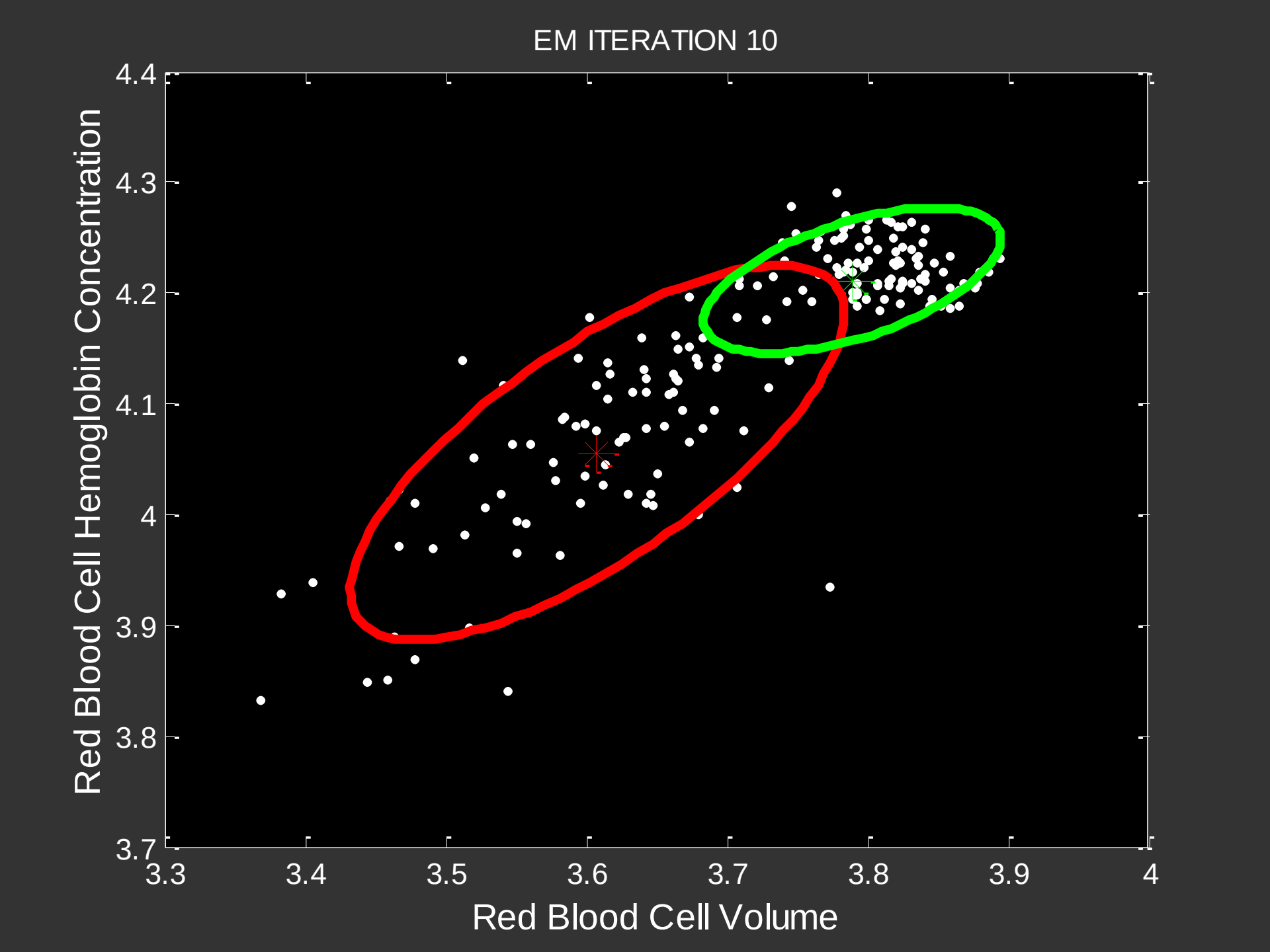
- E-Step
  - Given K means and K covariance shapes of K Gaussian clusters….
    ..compute probability that each of N data points to belong to each of K Gaussian clusters
- M-Step
  - Given probability that each of N data points to belong to each of K Gaussian clusters….
    …compute K new means and K new covariance shapes for the K Gaussian clusters
- Initialize randomly (the means/covariances or the memberships)
- One iteration is an E step and an M step
- Continue to iterate until the K means/covariances and/or the memberships are not changing (or hardly changing)
- Increases the P(data | clusters) at each step, converges to (local) optimum

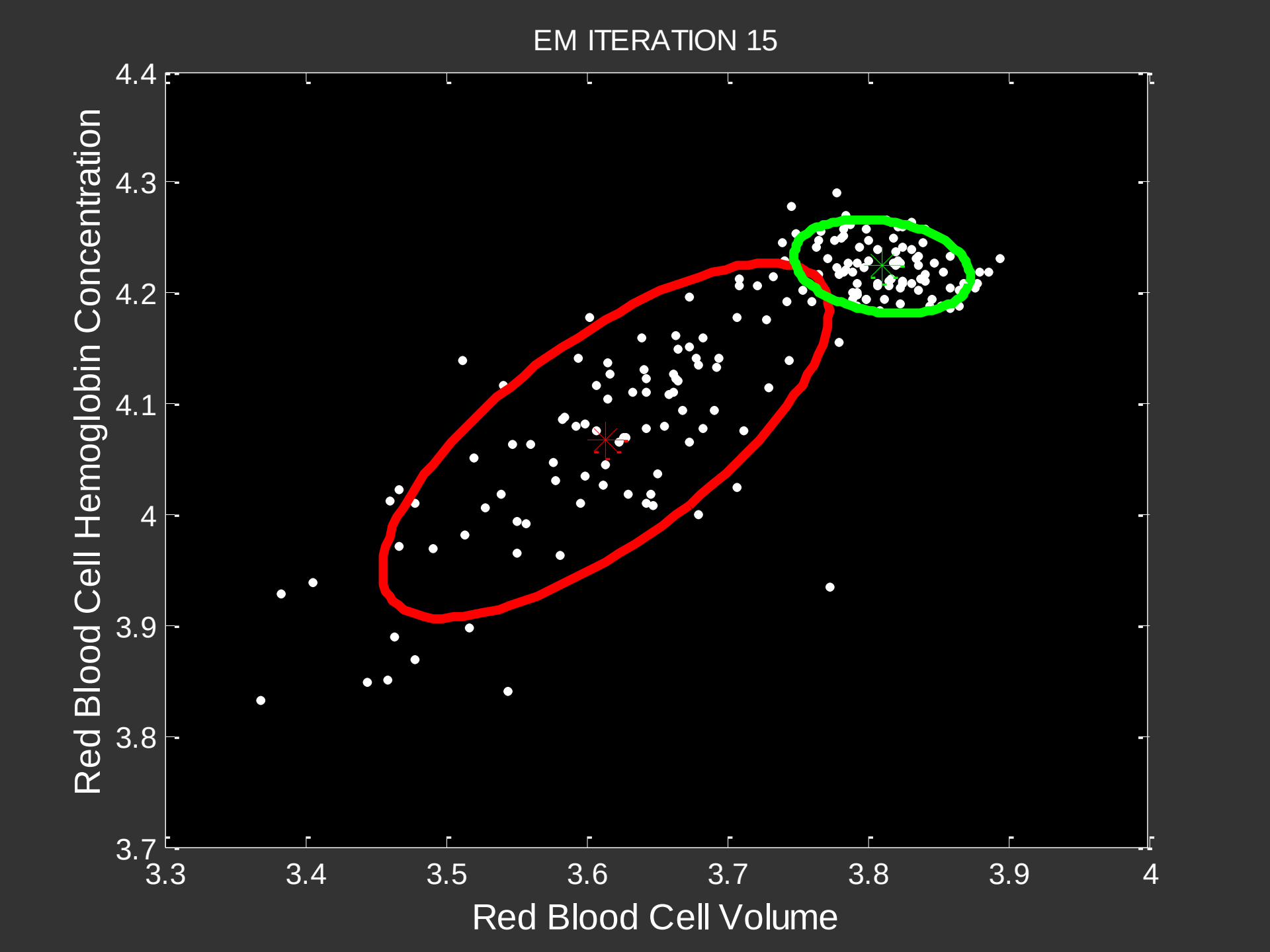

ANEMIA PATIENTS AND CONTROLS
Red Blood Cell Hemoglobin Concentration
Red Blood Cell Volume
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4
Data from Professor C. McLaren
Department of Epidemiology, UCI

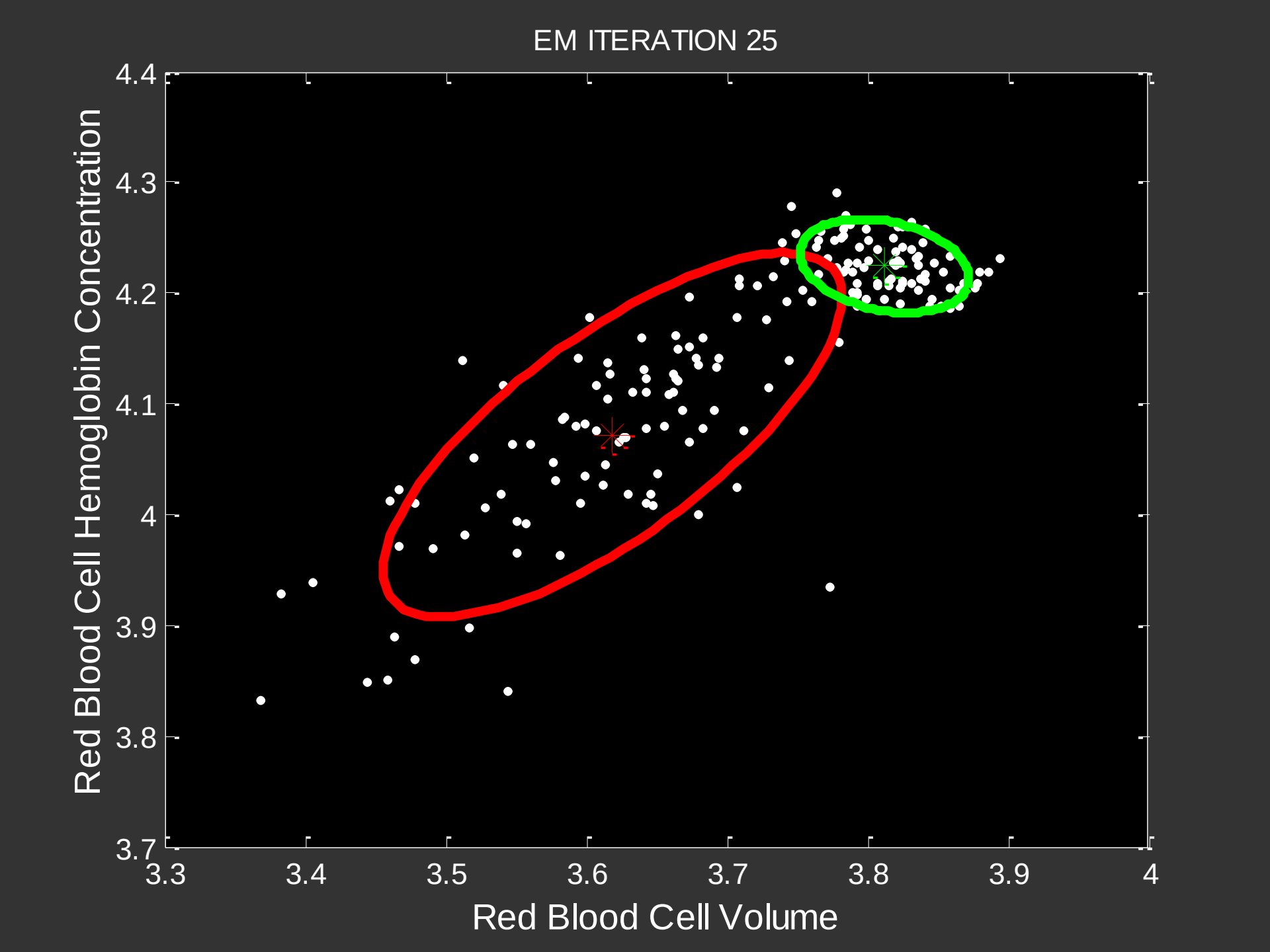

EM ITERATION 1
Red Blood Cell Hemoglobin Concentration
Red Blood Cell Volume
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4

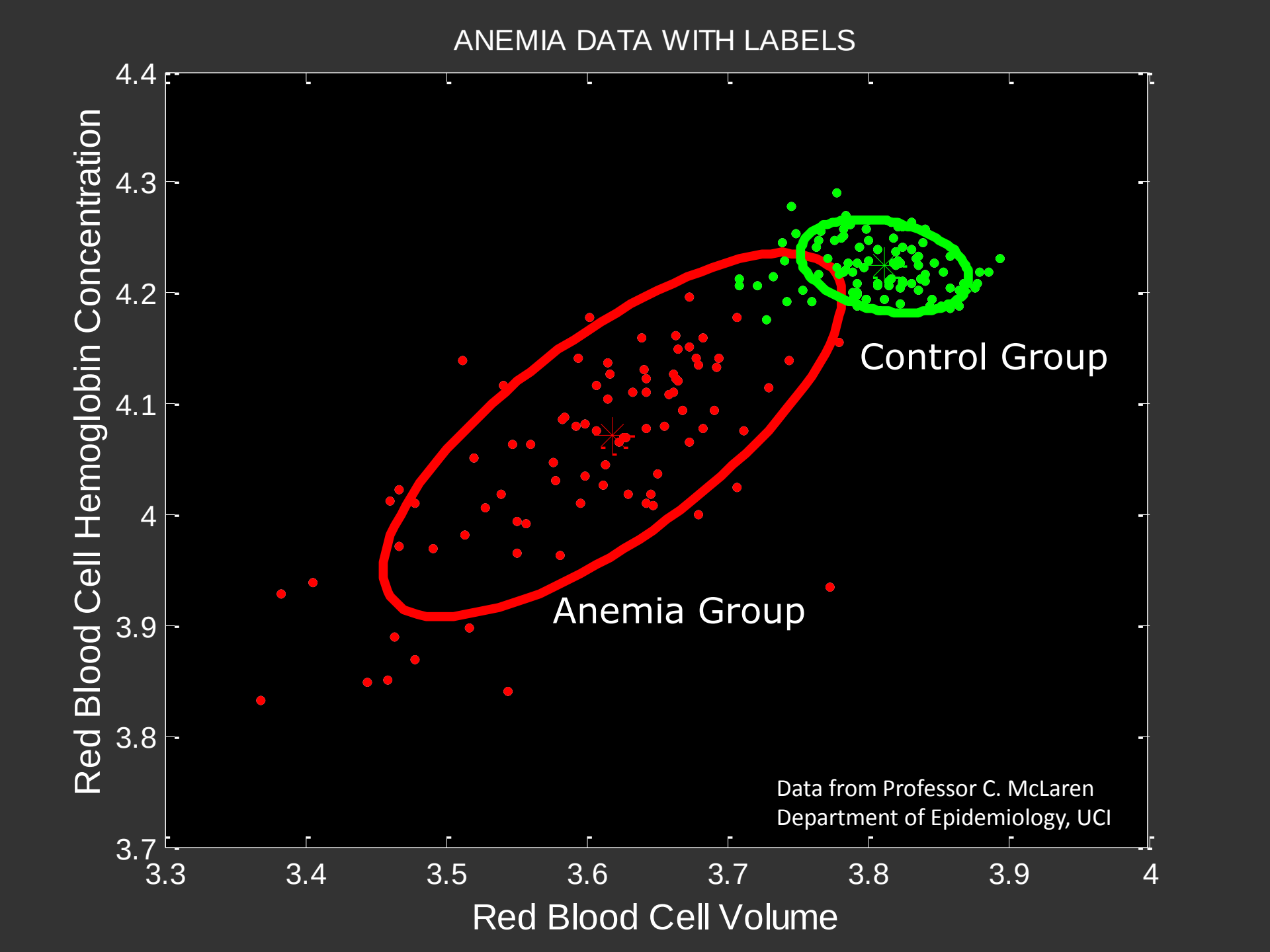

EM ITERATION 3
Red Blood Cell Hemoglobin Concentration
Red Blood Cell Volume
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4


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

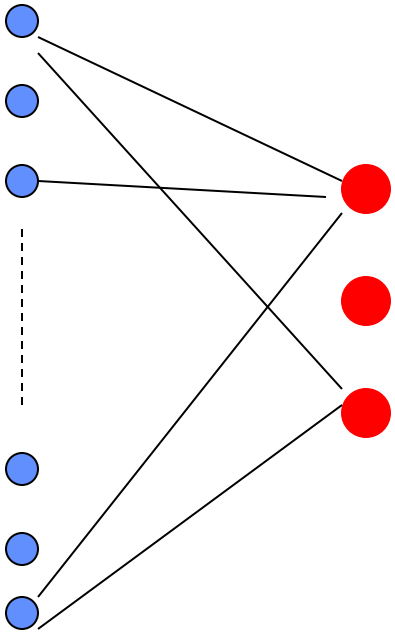

EM ITERATION 15
Red Blood Cell Hemoglobin Concentration
Red Blood Cell Volume
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4


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


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
Control Group
Anemia Group
Data from Professor C. McLaren
Department of Epidemiology, UCI

The EM Algorithm: putting N objects into K groups

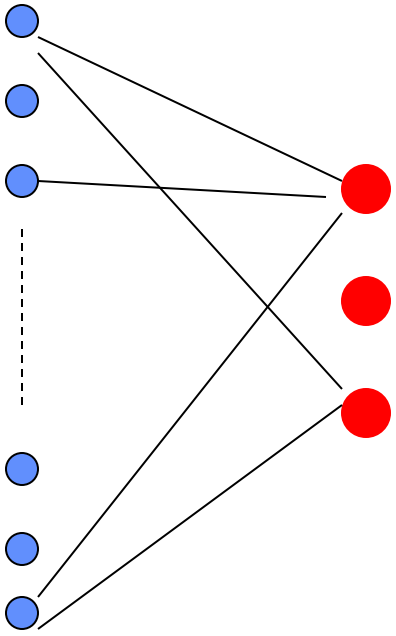
N objects
(observed data)



K clusters , parameters θ
(unknown)

The EM Algorithm: putting N objects into K groups

N objects
(observed data)



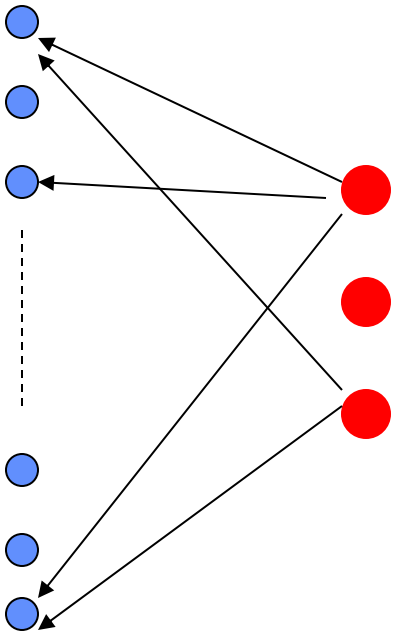
K clusters , parameters θ
(unknown)

Given
P(group k|object i)
we can estimate θ

Given θ we can compute
P(group k | object i)

A chicken and egg problem

The E (Expectation) Step

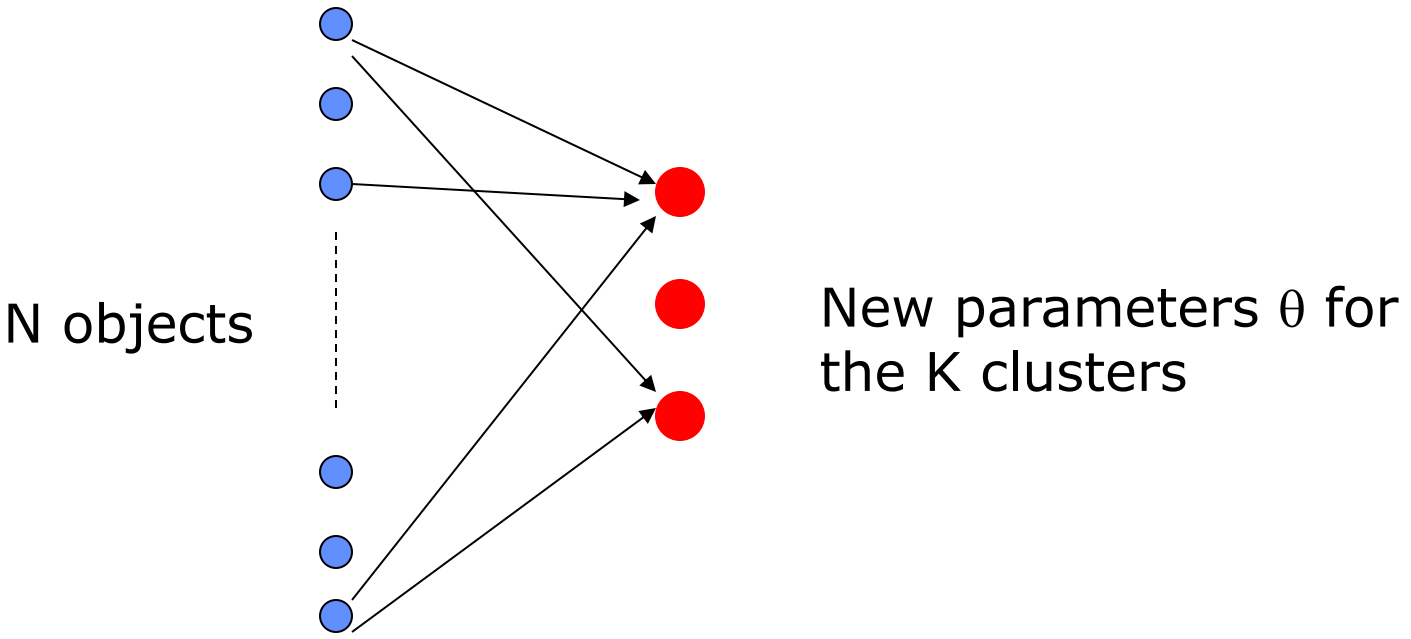


N objects

Current K clusters, parameters θ

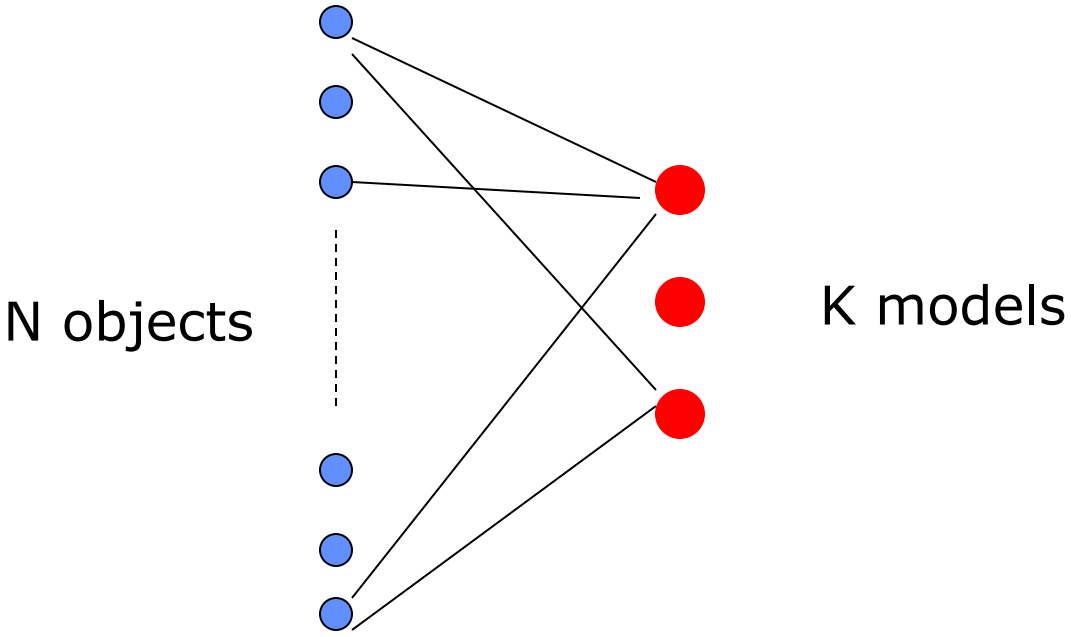
E step: Compute p(group k | object i)

The M (Maximization) Step

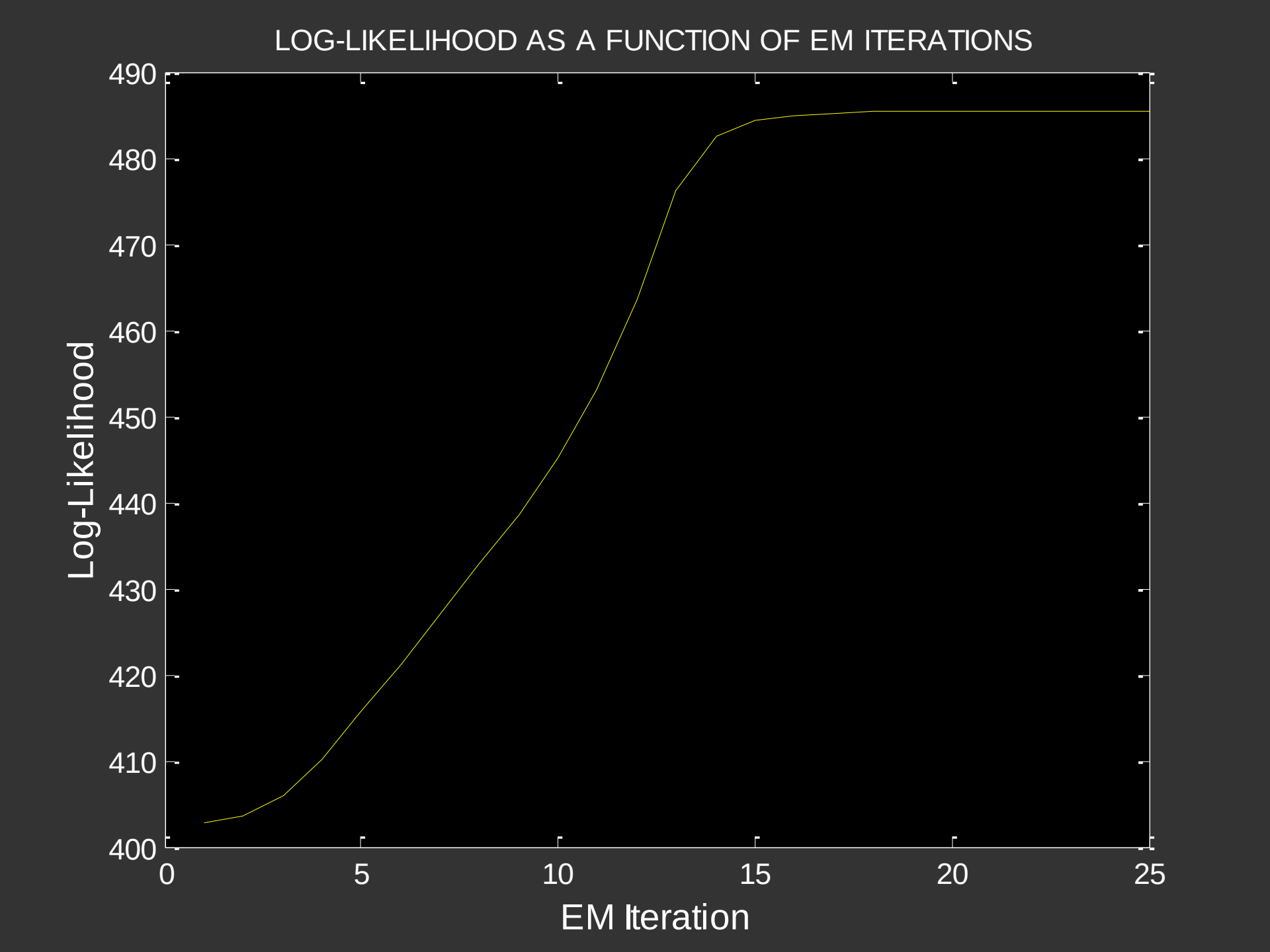


M step: Compute θ, given N objects and memberships

Complexity of EM (for mixture models)



Complexity per iteration scales as O(N K f(d))

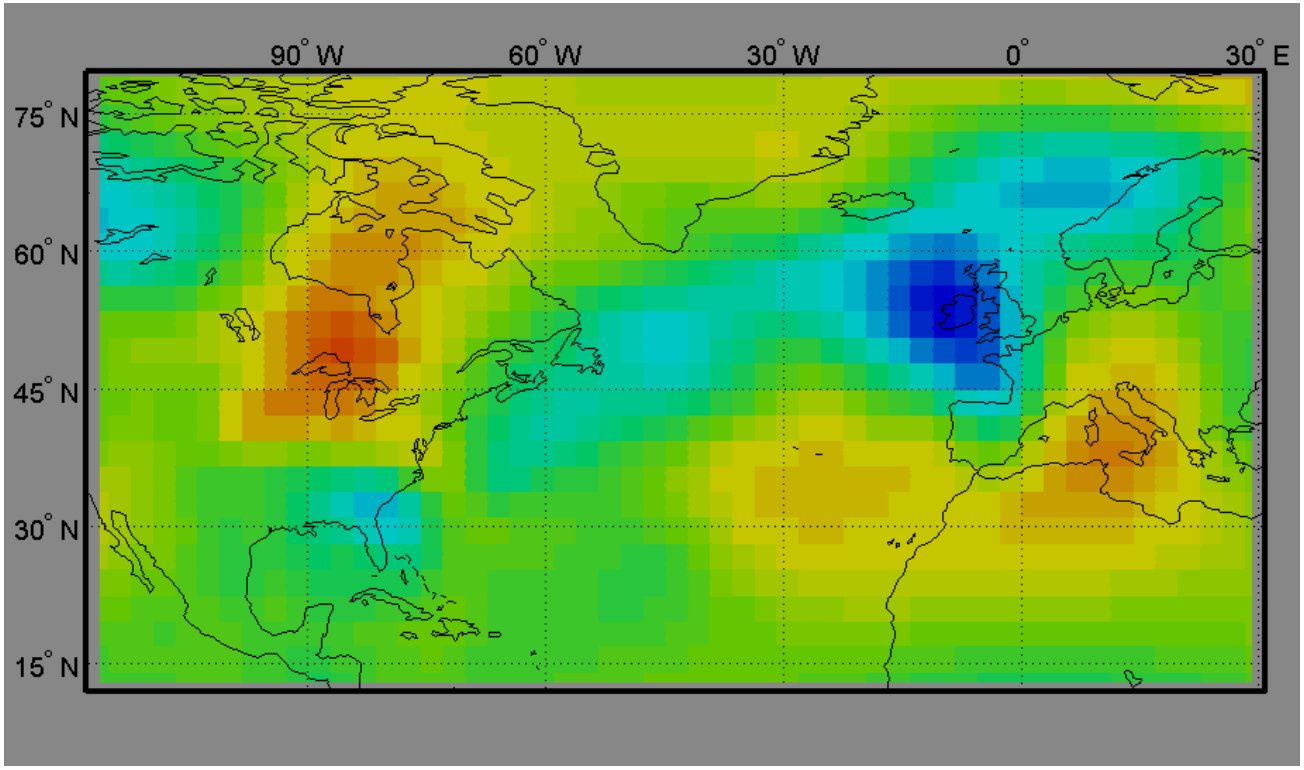

LOG-LIKELIHOOD AS A FUNCTION OF EM ITERATIONS
Log-Likelihood
EM Iteration
490
480
470
460
450
440
430
420
410
400
0
5
10
15
20
25

Links between K-means and EM

- EM with Gaussian clusters can be seen as a generalization of K-means
  - Each algorithm consists of 2 steps, updating (1) "memberships" and (2) cluster parameters
- Assignment Step versus E-Step
  - K-means: "hard assignment" of each data point to its closest cluster
  - EM: "soft assignment" of data points to clusters, via P(cluster k | data, parameters)
- Update Cluster Centers versus M-Step
  - K-means: compute new cluster centers
  - EM: compute new cluster centers and Gaussian covariance ("shape") matrices
- Both have their advantages
  - EM with Gaussians can be more flexible…if clusters are approximately Gaussian
  - K-means can be more robust with non-Gaussian data

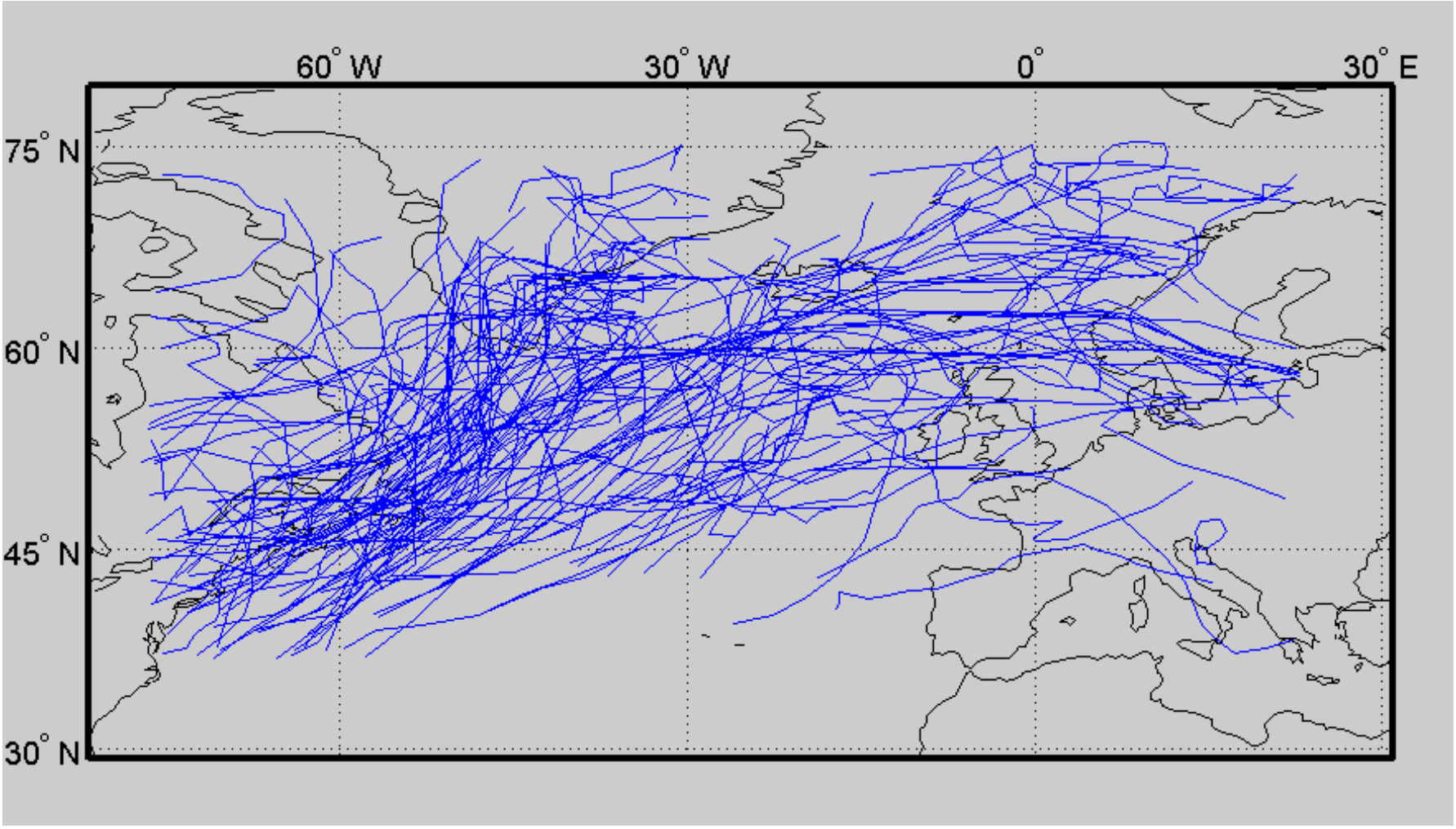
Atlantic Ocean Sea-Level Pressure Data

- Mean sea-level pressure (SLP) on a 2.5° by 2.5° grid
- Four times a day, every 6 hours, over 15 to 20 years



Atlantic Cyclone Trajectories

Goal: how can we cluster these trajectories?

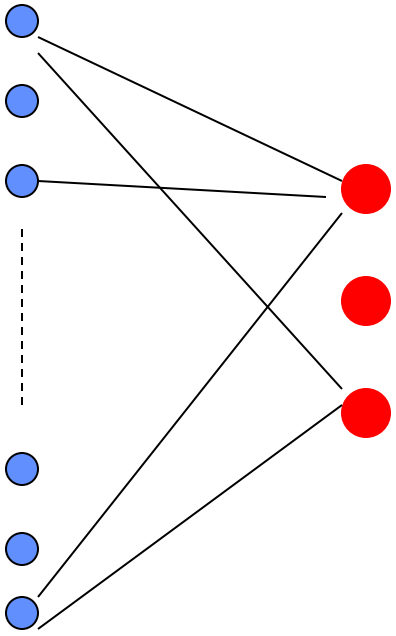


Clustering Methodology

- Clusters of curves
  - model as mixtures of noisy linear/quadratic curves
    - note: true paths are not linear
    - use the model as a first-order approximation for clustering
  - Use EM to learn the mixtures/clusters (generalization of Gaussian case
    - Scott Gaffney's PhD thesis, 2005
- Advantages
  - allows for variable-length trajectories
  - allows coupling of other "features" (e.g., intensity)
  - provides a quantitative (e.g., predictive) model

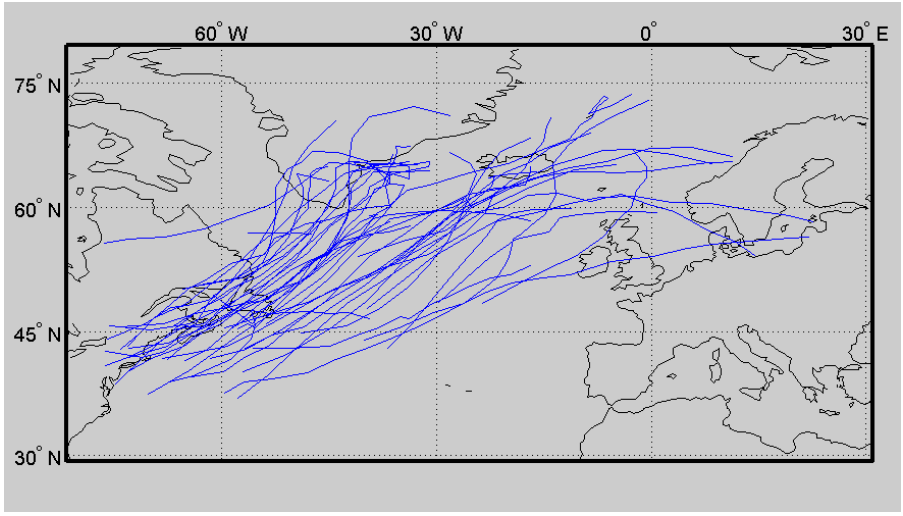
EM Algorithm for Curve Clustering

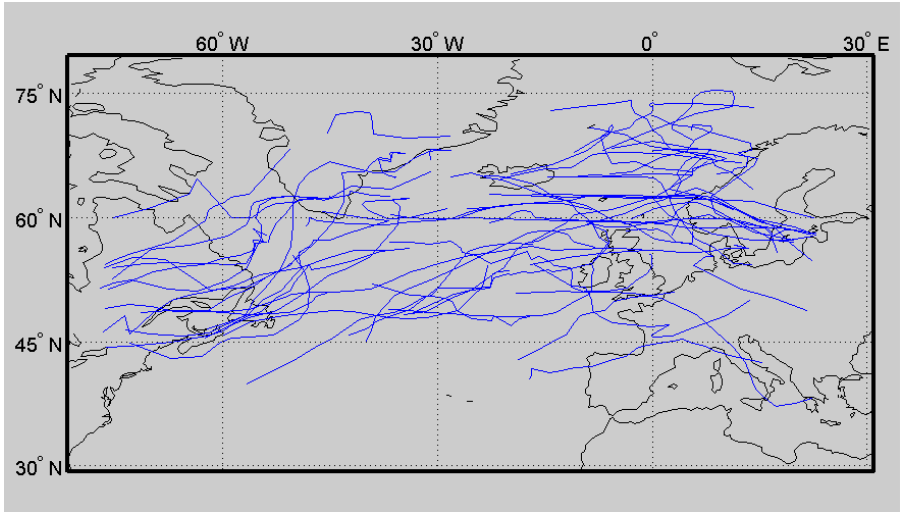
N curves
(different lengths)

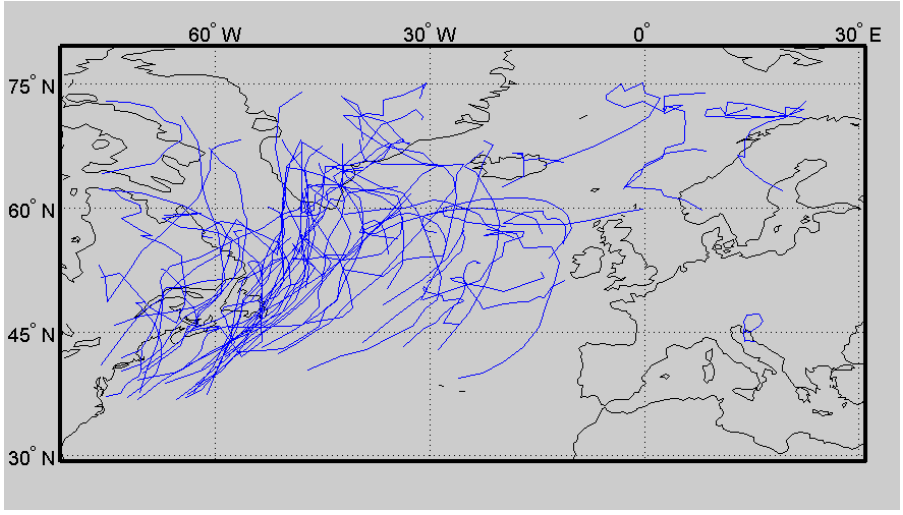


K curve clusters
(parameters unknown)

Clusters of Trajectories







Scientific Value of Clustering

- Visualization and Exploration
  - improved understanding of cyclone dynamics
- Change Detection
  - can quantitatively compare cyclone statistics over different era's or from different models
- Linking cyclones with climate and weather
  - correlation of clusters with NAO index
  - correlation with windspeeds in Northern Europe

Clustering: Hierarchical Clustering Methods

Different Types of Input to Clustering Algorithms

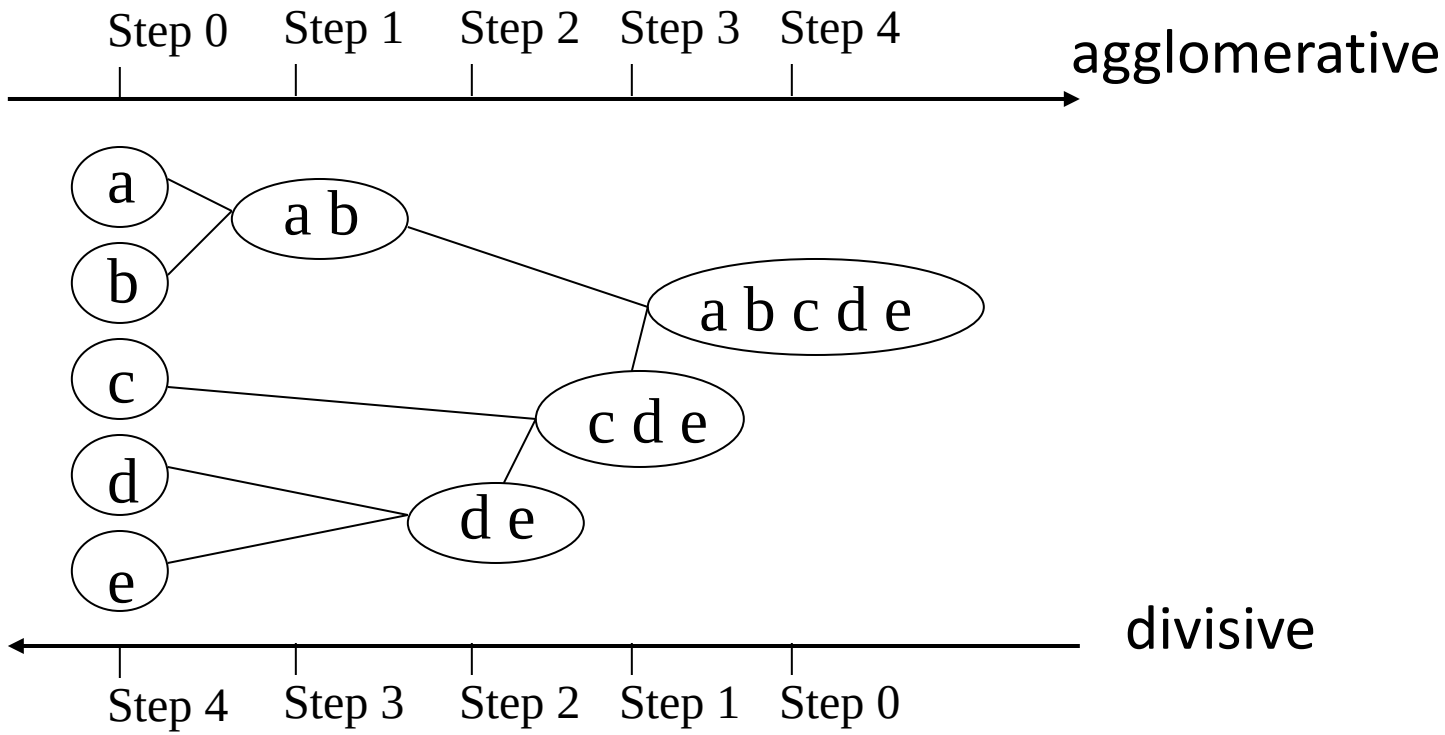
- Data matrix
  - N rows
  - d columns

$$\begin{bmatrix} x_{11} & \dots & x_{1f} & \dots & x_{1d} \\ \dots & \dots & \dots & \dots & \dots \\ x_{i1} & \dots & x_{if} & \dots & x_{id} \\ \dots & \dots & \dots & \dots & \dots \\ x_{n1} & \dots & x_{nf} & \dots & x_{nd} \end{bmatrix}$$

- Distance matrix
  - N x N distances
  - between objects

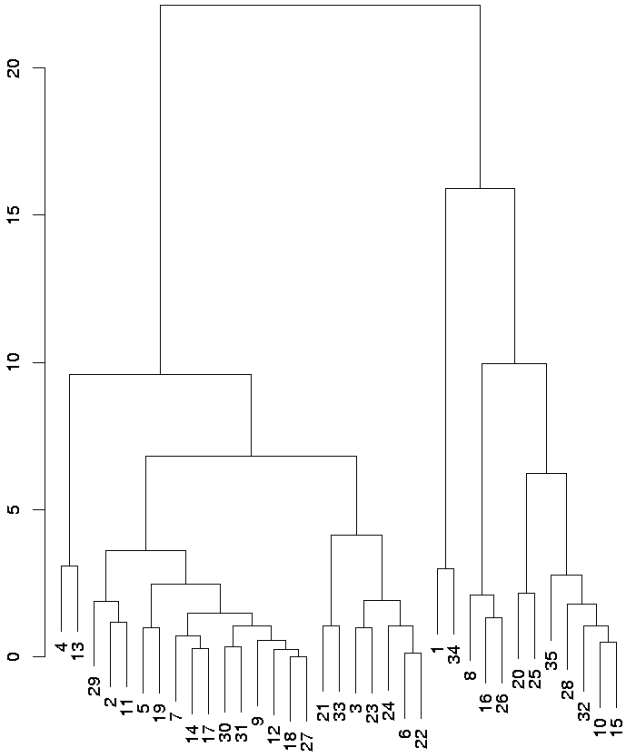
$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ : & : & : & & \\ d(n,1) & d(n,2) & \dots & \dots & 0 \end{bmatrix}$$

Hierarchical Clustering



Hierarchical Clustering

- Produces tree of nested clusters
- Works from a distance matrix
  - advantage: x's can be any type of object
  - disadvantage: computation
- Two basic approaches:
  - merge points (agglomerative)
  - divide superclusters (divisive)
- visualize both via "dendograms"
  - shows nesting structure
  - merges or splits = tree nodes
- Applications
  - e.g., clustering of gene expression data
  - Useful for seeing hierarchical structure, for relatively small data sets



Agglomerative Methods: Bottom-Up

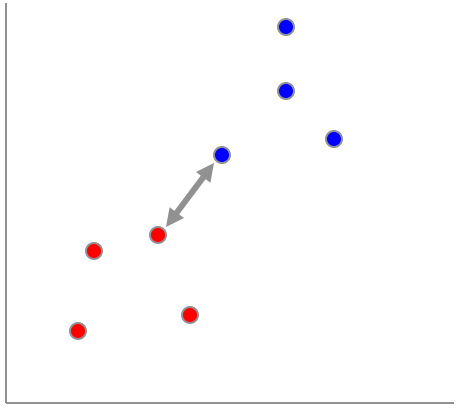
- algorithm based on distance between clusters:
  - for i=1 to n let C_i = { x(i) }, i.e. start with n singletons
  - while more than one cluster left
    - let C_i and C_j be cluster pair with minimum distance, dist[C_i , C_j]
    - merge them, via $C_i = C_i \cup C_j$ and remove C_j

- time complexity = $O(n^2)$ to $O(n^3)$
  - n iterations (start: n clusters; end: 1 cluster)
  - 1st iteration: $O(n^2)$ to find nearest singleton pair

- space complexity = $O(n^2)$
  - accesses all distances between x(i)'s

Distances Between Clusters

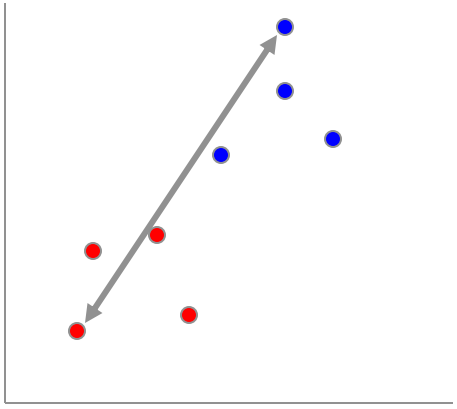
- single link / nearest neighbor measure:
  - $D(Ci,Cj) = \min \{ d(x,y) \mid x \in Ci, y \in Cj \}$
  - can be outlier/noise sensitive



Distances Between Clusters

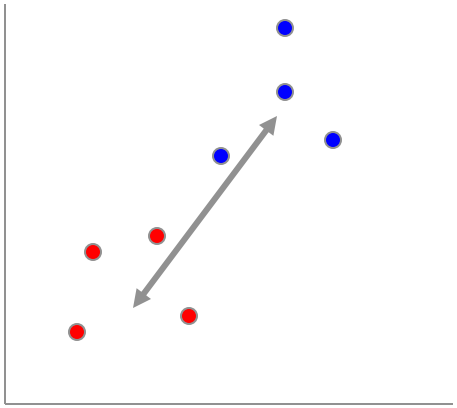
- single link / nearest neighbor measure:
  - $D(C_i,C_j) = \min \{ d(x,y) \mid x \in C_i, y \in C_j \}$
  - can be outlier/noise sensitive

- complete link / furthest neighbor measure:
  - $D(C_i,C_j) = \max \{ d(x,y) \mid x \in C_i, y \in C_j \}$
  - enforces more "compact" clusters



Distances Between Clusters

- single link / nearest neighbor measure:
  - D(Ci,Cj) = min { d(x,y) | x ∈ Ci, y ∈ Cj }
  - can be outlier/noise sensitive

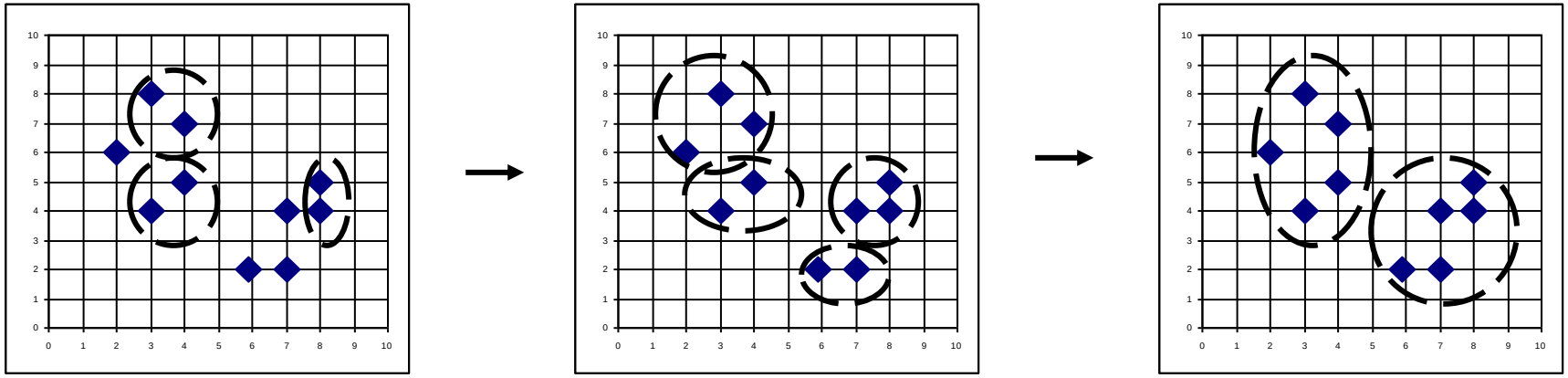


- complete link / furthest neighbor measure:
  - D(Ci,Cj) = max { d(x,y) | x ∈ Ci, y ∈ Cj }
  - enforces more "compact" clusters

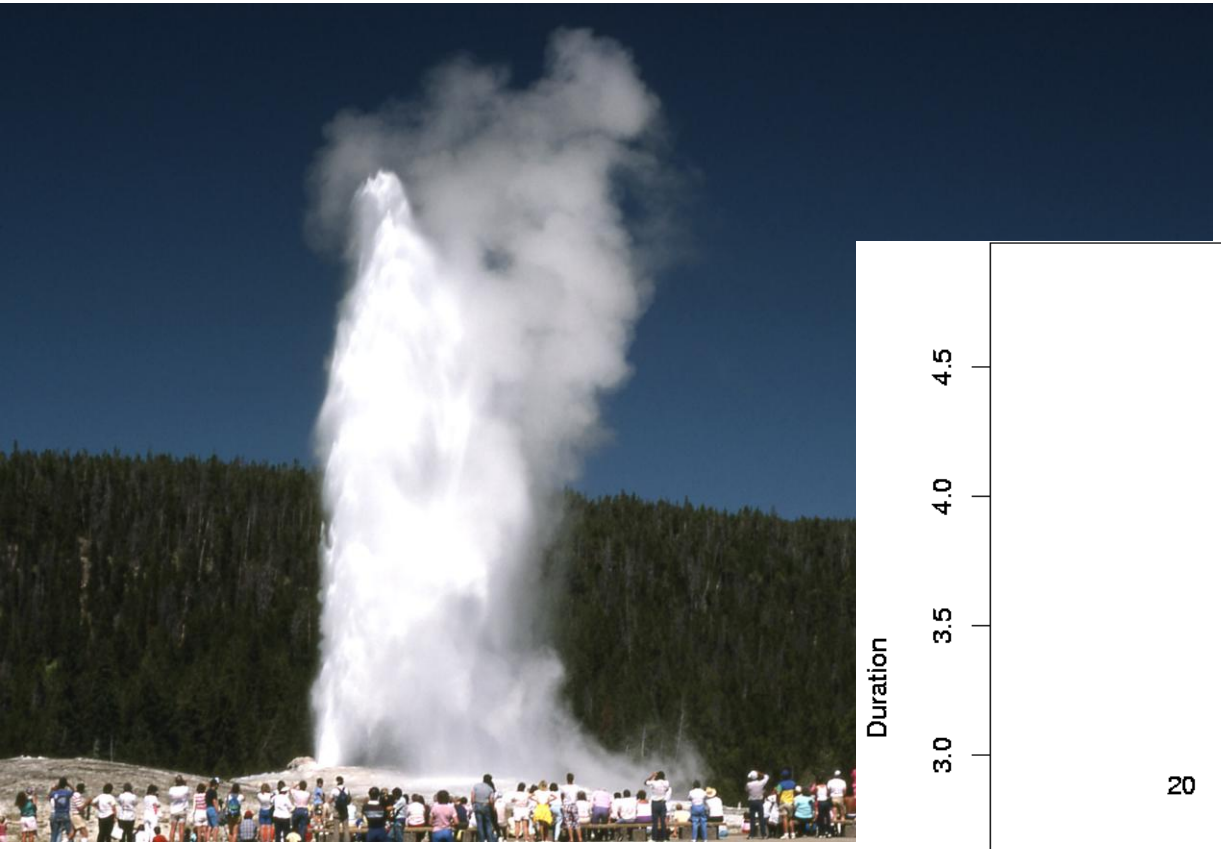
- intermediates between those extremes:
  - average link: D(Ci,Cj) = avg { d(x,y) | x ∈ Ci, y ∈ Cj }
  - centroid: D(Ci,Cj) = d(c_i,c_j) where c_i , c_j are centroids
  - Wards's SSE measure (for vector data):
    - Merge clusters than minimize increase in within-cluster sum-of-squared-dists
  - Note that centroid and Ward require that centroid (vector mean) can be defined

- Which to choose? Different methods may be used for exploratory purposes, depends on goals and application

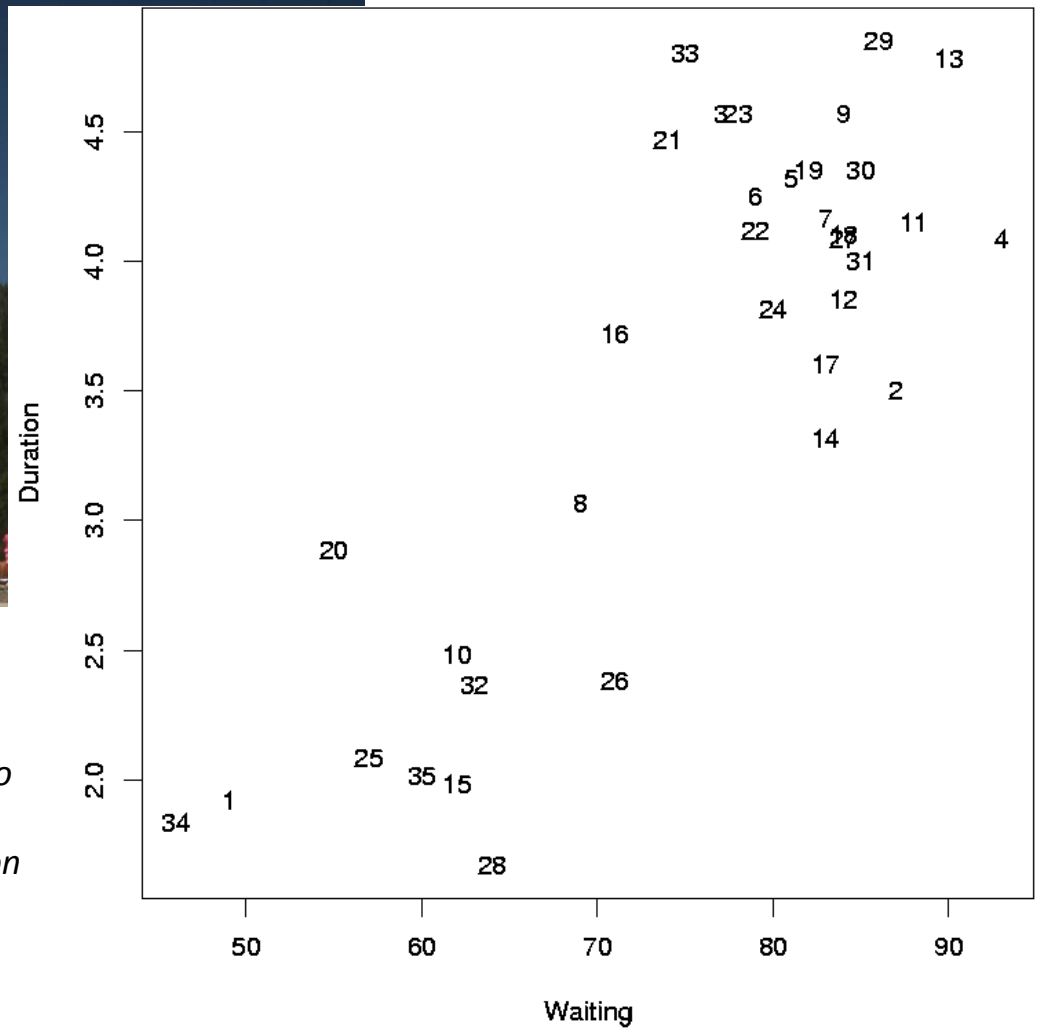
Simple Example of Single-Link Agglomerative Clustering



Old-Faithful Eruption Timing Data Set

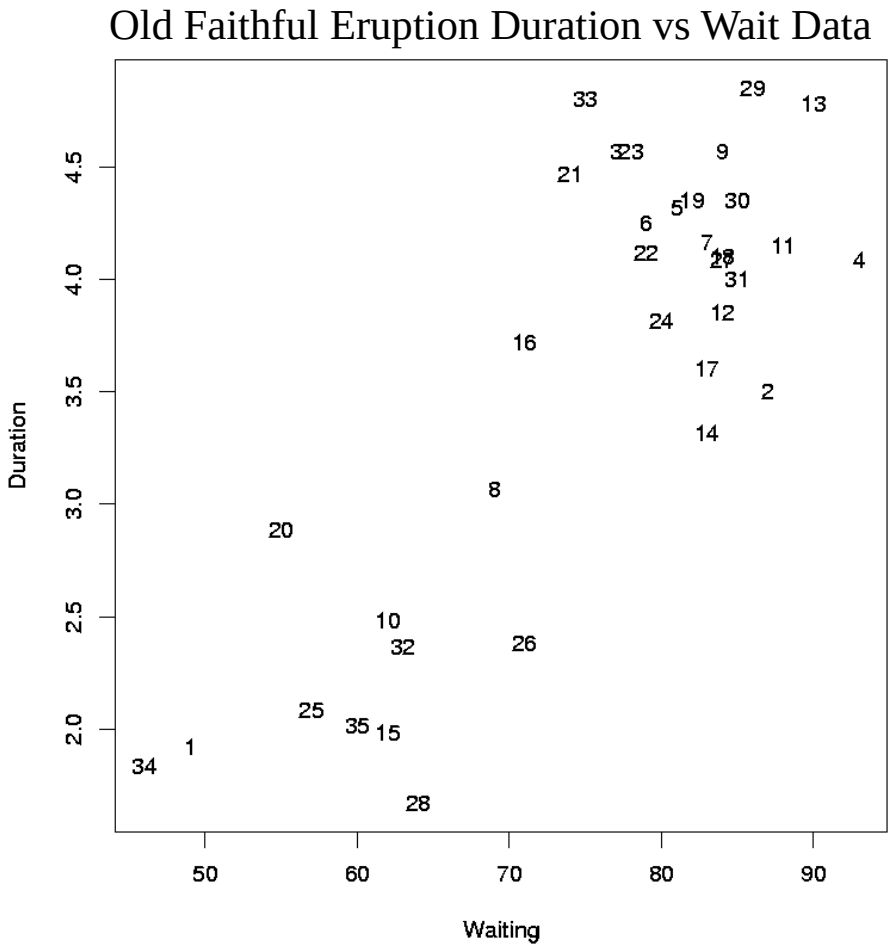


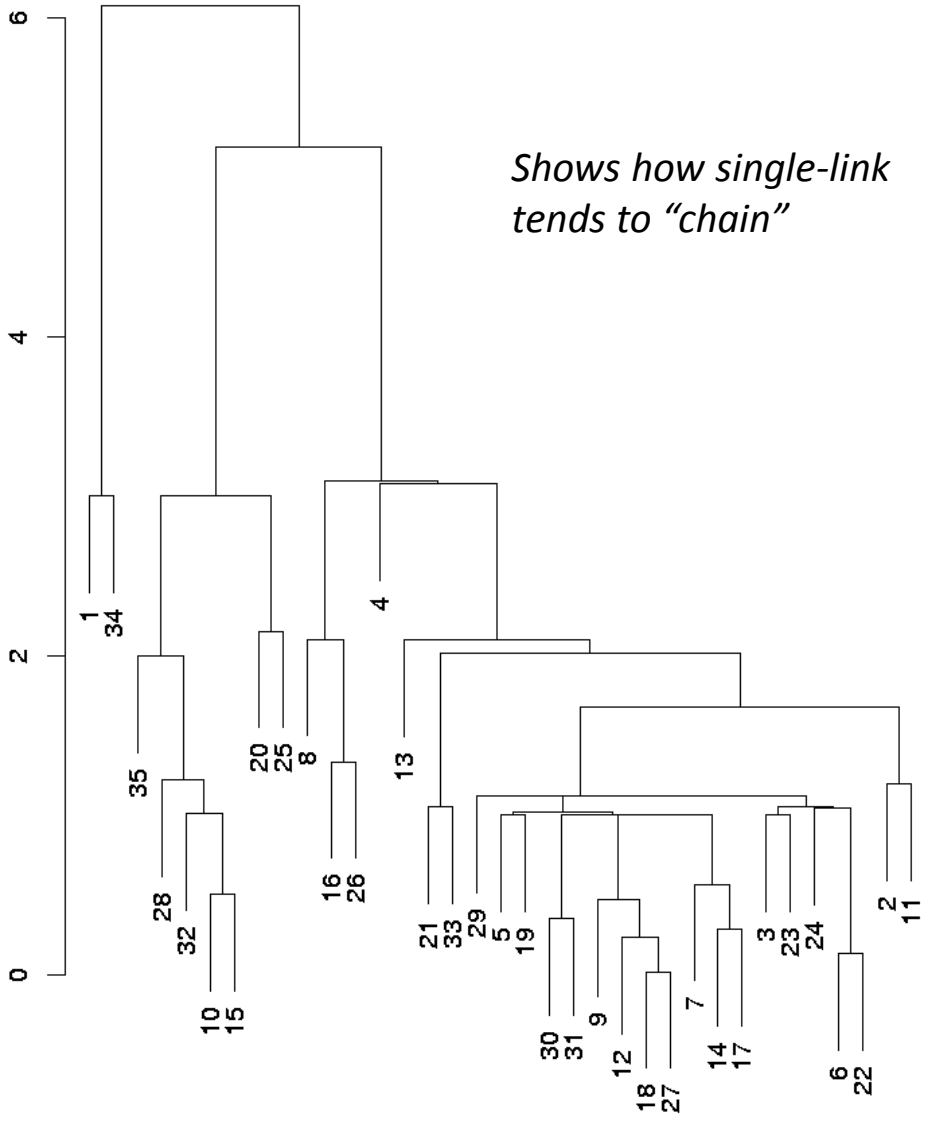
Old Faithful Eruptions
Duration vs Wait Data



Notice that these variables are not scaled: so waiting time will get a lot more emphasis in Euclidean distance calculations than duration

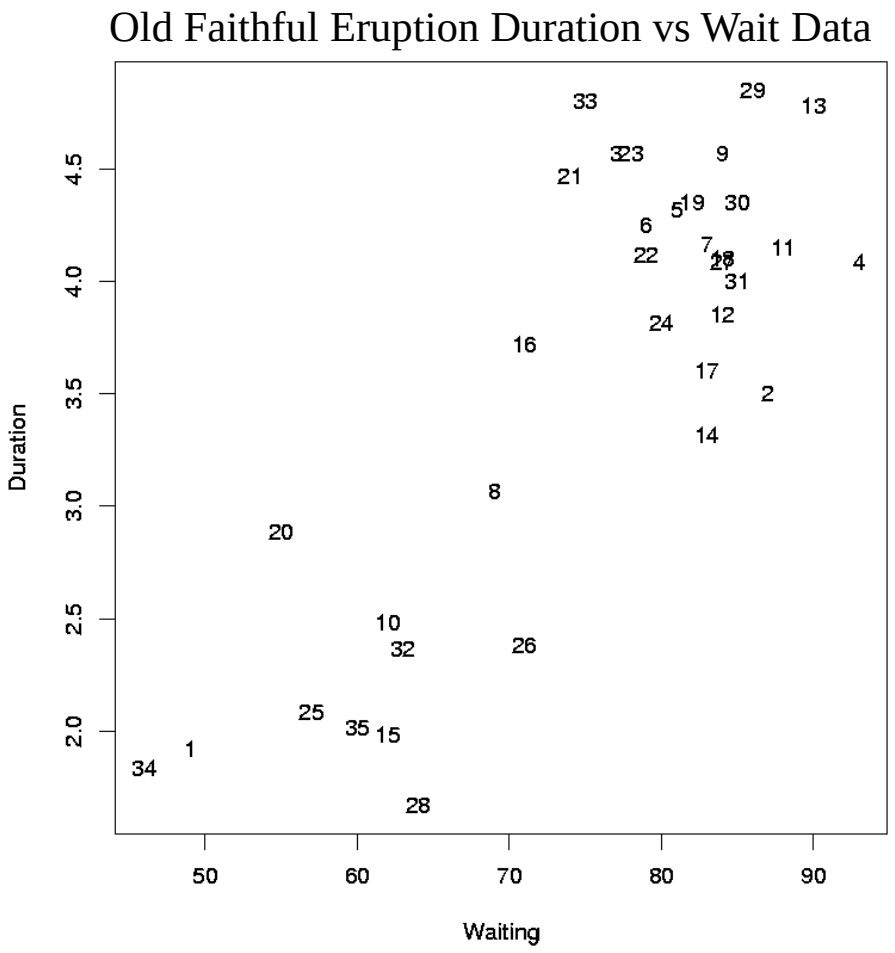
Dendrogram Using Single-Link Method

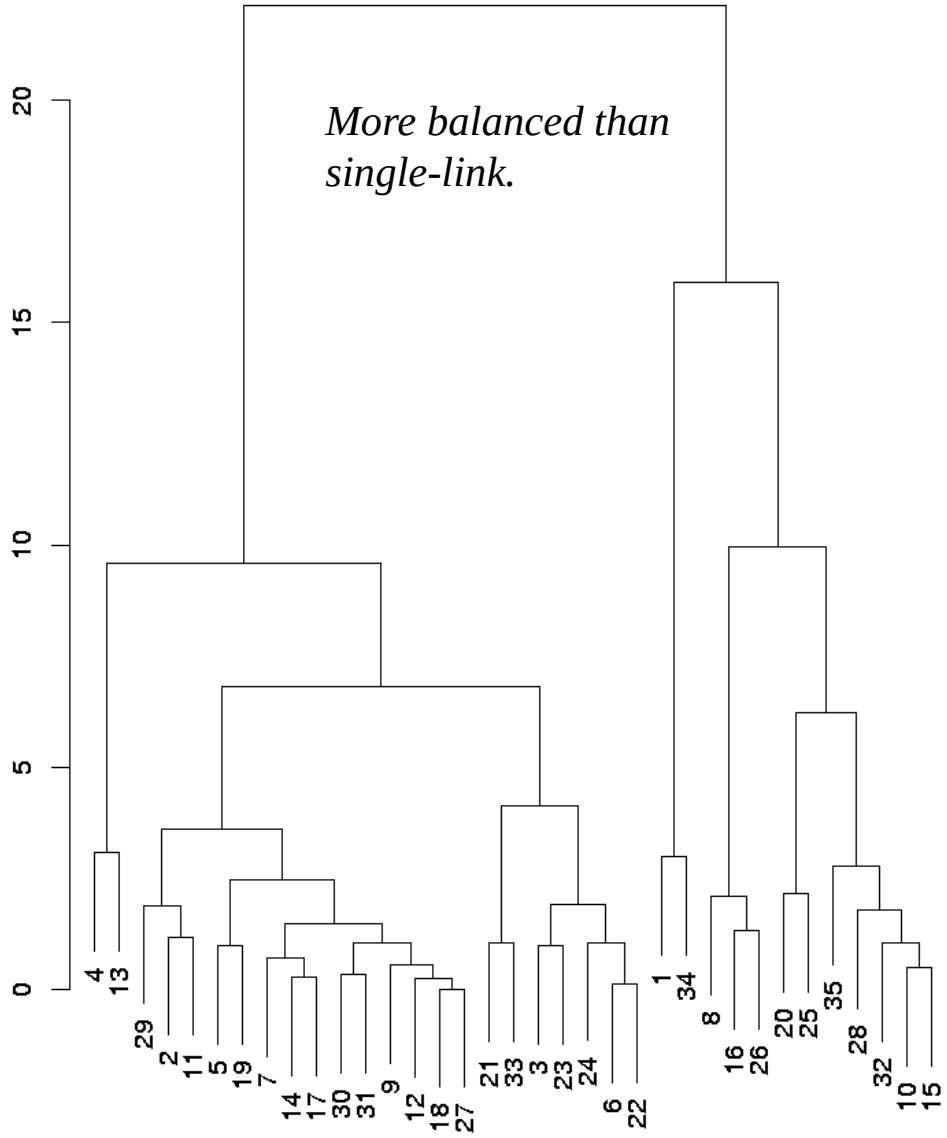


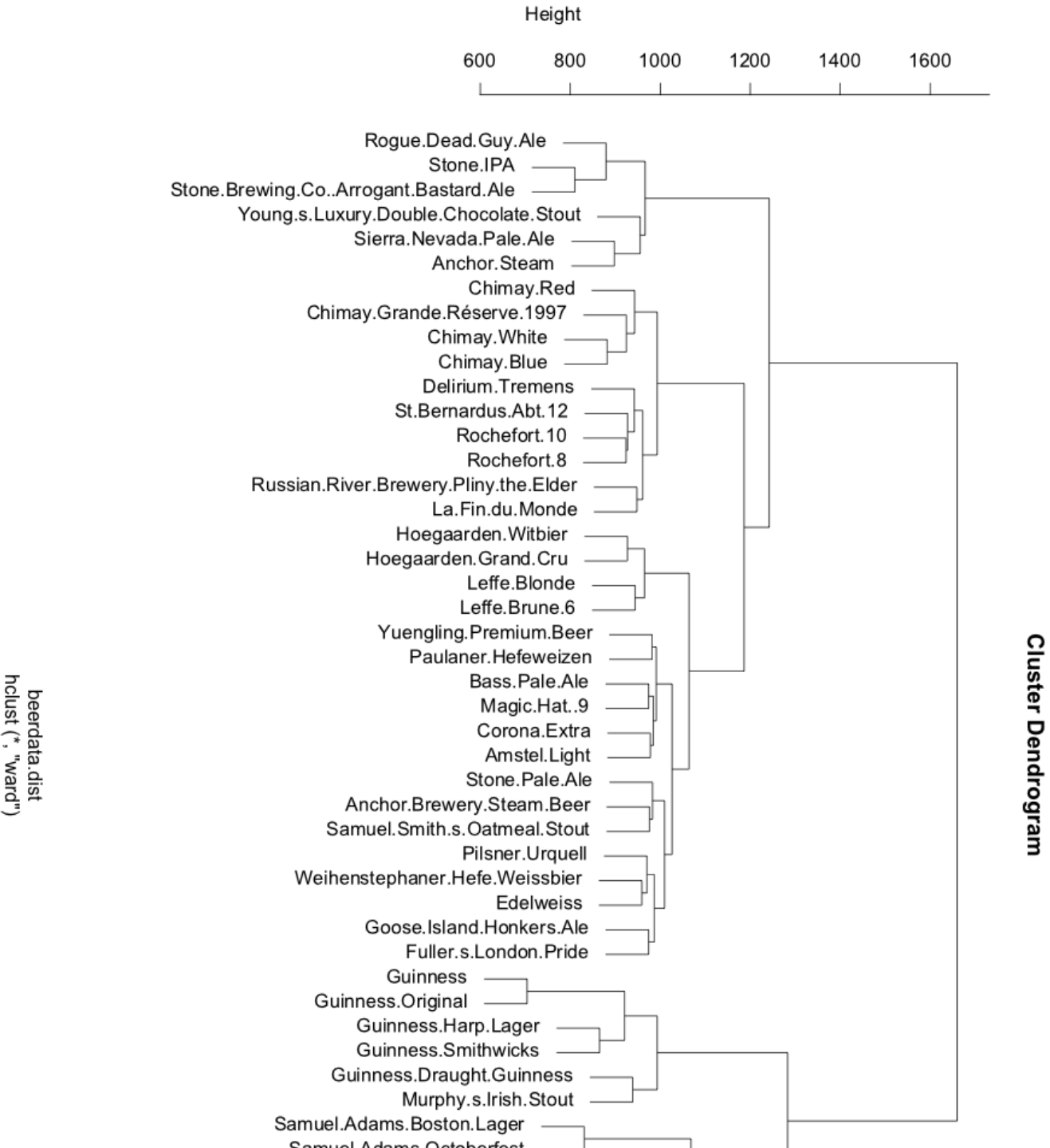


Shows how single-link tends to "chain"

Dendogram Using Ward's SSE Distance



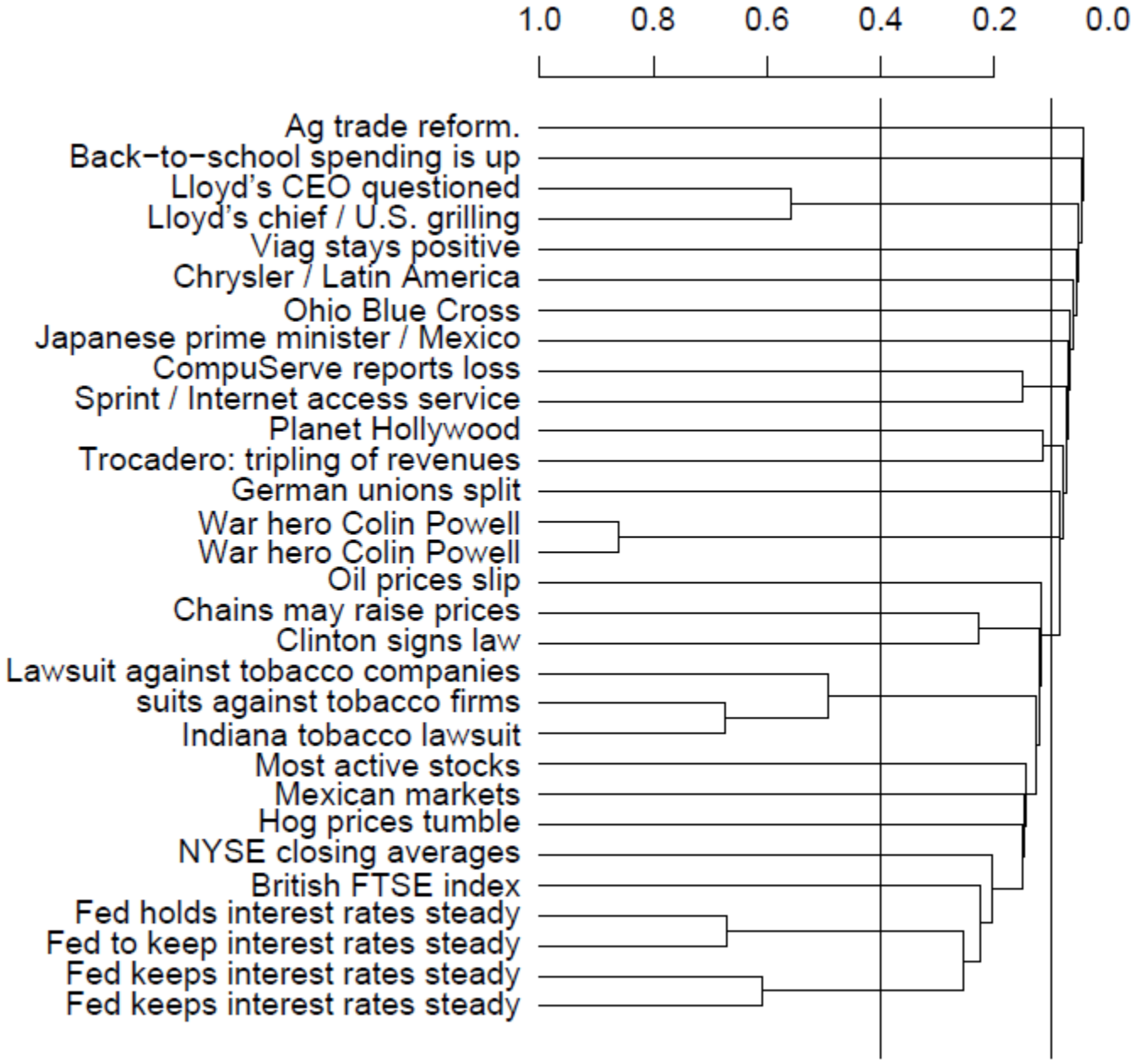




Hierarchical Clustering
Based on Votes of Favorite Beers

Based on Ward's method

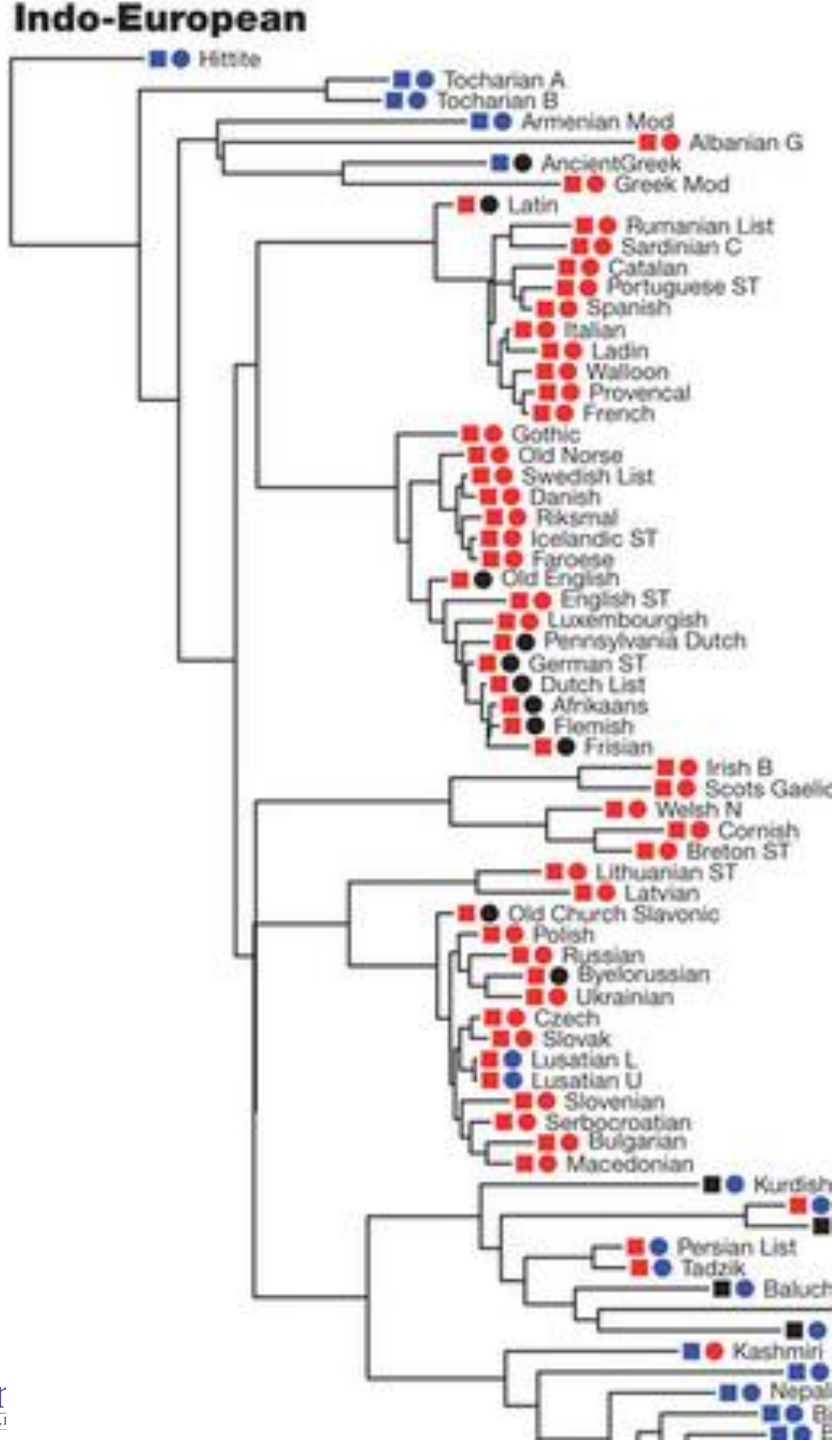
From data.ranker.com



Single-Link clustering of Reuters news stories

Clustering algorithms are widely used in Web search and on sites such as Google news

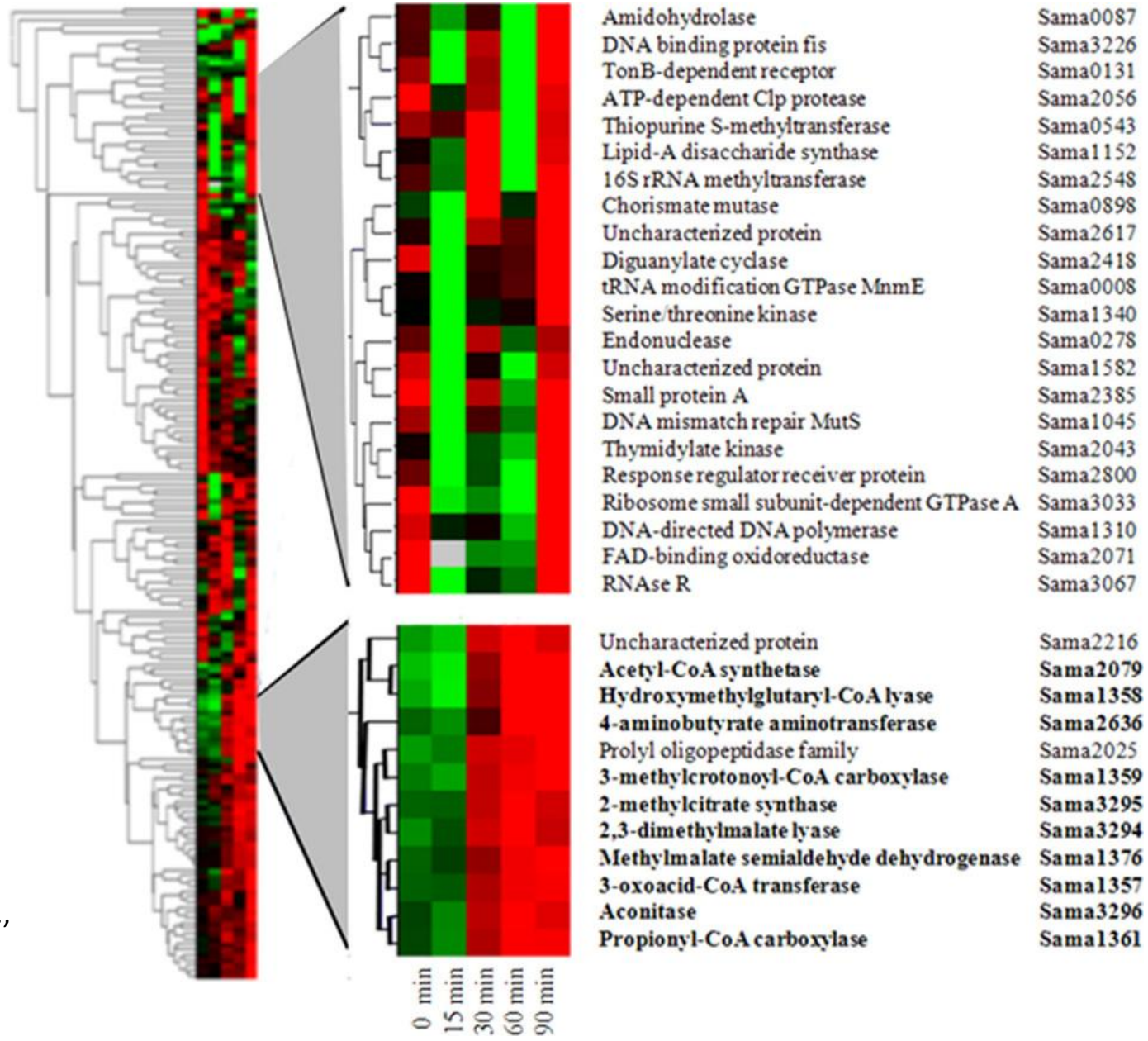
Figure from Chapter 17 of Manning, Raghavan, and Schutze



Hierarchical Cluster Structure in Linguistics

From Dunn et al., Nature 2011

Hierarchical Cluster Structure for Protein Expression Data over Time



From Parnell et al., Nature 2011

Proc. Natl. Acad. Sci. USA
Vol. 95, pp. 14863–14868, December 1998
Genetics

Cluster analysis and display of genome-wide expression patterns

MICHAEL B. EISEN*, PAUL T. SPELLMAN*, PATRICK O. BROWN†, AND DAVID BOTSTEIN*‡

*Department of Genetics and †Department of Biochemistry and Howard Hughes Medical Institute, Stanford University School of Medicine, 300 Pasteur Avenue, Stanford, CA 94305

Contributed by David Botstein, October 13, 1998

ABSTRACT **A system of cluster analysis for genome-wide expression data from DNA microarray hybridization is described that uses standard statistical algorithms to arrange genes according to similarity in pattern of gene expression. The output is displayed graphically, conveying the clustering and the underlying expression data simultaneously in a form intuitive for biologists. We have found in the budding yeast *Saccharomyces cerevisiae* that clustering gene expression data groups together efficiently genes of known similar function, and we find a similar tendency in human data. Thus patterns seen in genome-wide expression experiments can be interpreted as indications of the status of cellular processes. Also, coexpression of genes of known function with poorly characterized or novel genes may provide a simple means of gaining leads to the functions of many genes for which information is not available currently.**

The rapid advance of genome-scale sequencing has driven the

be used, such as the Euclidean distance, angle, or dot products of the two n-dimensional vectors representing a series of n measurements. We have found that the standard correlation coefficient (i.e., the dot product of two normalized vectors) conforms well to the intuitive biological notion of what it means for two genes to be "coexpressed;" this may be because this statistic captures similarity in "shape" but places no emphasis on the magnitude of the two series of measurements.

It is not the purpose of this paper to survey the various methods available to cluster genes on the basis of their expression patterns, but rather to illustrate how such methods can be useful to biologists in the analysis of gene expression data. We aim to use these methods to organize, but not to alter, tables containing primary data; we have thus used methods that can be reduced, in the end, to a reordering of lists of genes. Clustering methods can be divided into two general classes, designated supervised and unsupervised clustering (4). In supervised clustering, vectors are classified with respect to

Approach

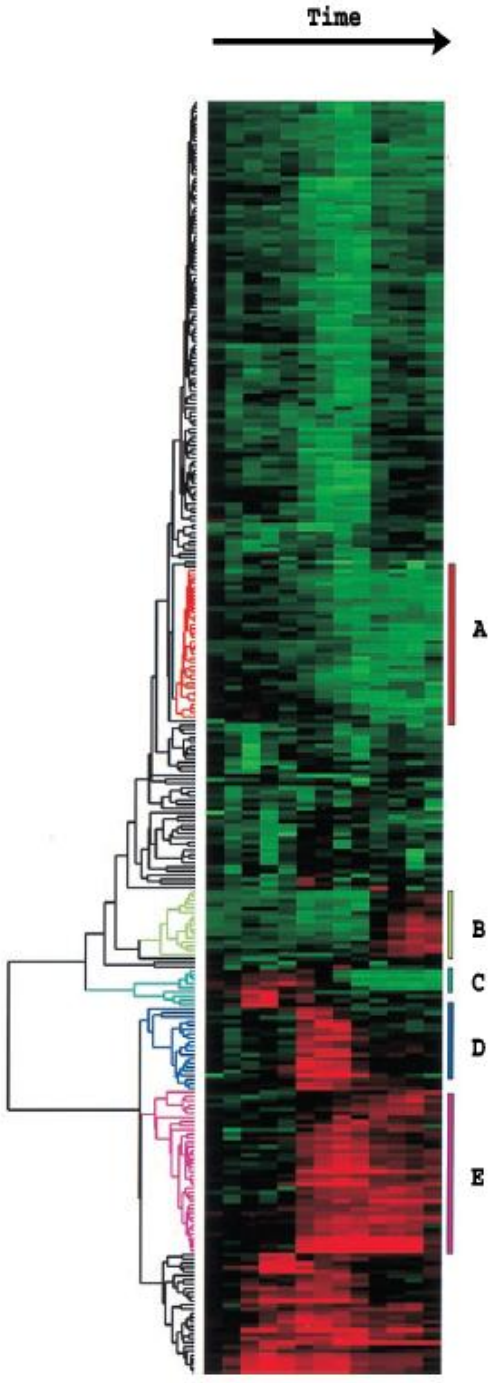
- Hierarchical clustering of genes using average linkage method

Metrics. The gene similarity metric we use is a form of correlation coefficient. Let G_i equal the (log-transformed) primary data for gene G in condition i. For any two genes X and Y observed over a series of N conditions, a similarity score can be computed as follows:

$$S(X, Y) = \frac{1}{N} \sum_{i=1,N} \left(\frac{X_i - X_{offset}}{\Phi_X} \right) \left(\frac{Y_i - Y_{offset}}{\Phi_Y} \right)$$

- Clustered time-course data of 8600 human genes and 2467 genes in budding yeast

- This paper was the first to show that clustering of expression data yields significant biological insights into gene function



"Heat-Map" Representation (human data)

red for log ratios 3.0 and above. Each gene is represented by a single row of colored boxes; each time point is represented by a single column. Five separate clusters are indicated by colored bars and by identical coloring of the corresponding region of the dendrogram. As described in detail in ref. 11, the sequence-verified named genes in these clusters contain multiple genes involved in (*A*) cholesterol biosynthesis, (*B*) the cell cycle, (*C*) the immediate–early response, (*D*) signaling and angiogenesis, and (*E*) wound healing and tissue remodeling. These clusters also contain named genes not involved in these processes and numerous uncharacterized genes. A larger version of this image, with gene names, is available at http://rana.stanford.edu/clustering/serum.html.

Evaluation

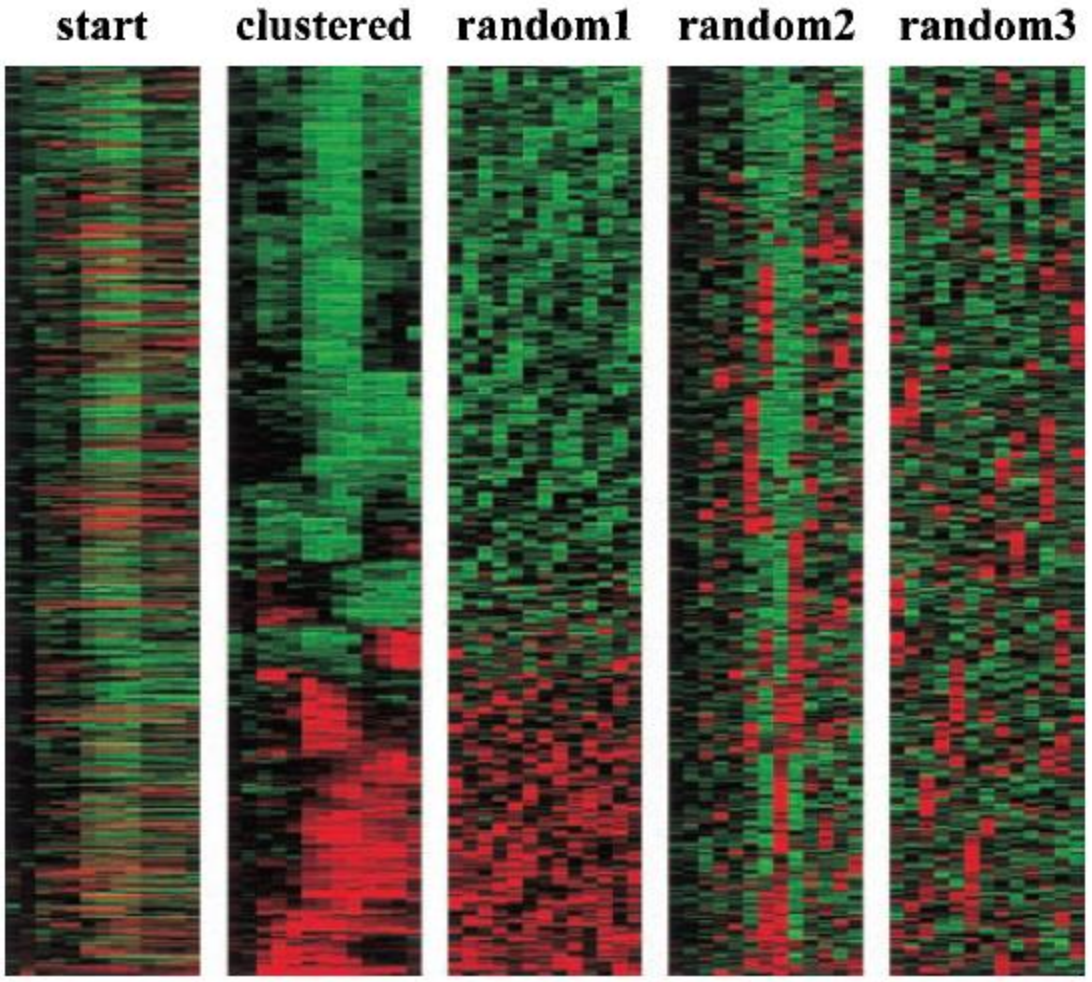


FIG. 3. To demonstrate the biological origins of patterns seen in Figs. 1 and 2, data from Fig. 1 were clustered by using methods described here before and after random permutation within rows (random 1), within columns (random 2), and both (random 3).

Divisive Methods: Top-Down

- General algorithm:
  - begin with single cluster containing all data
  - split into components, repeat until clusters = single points

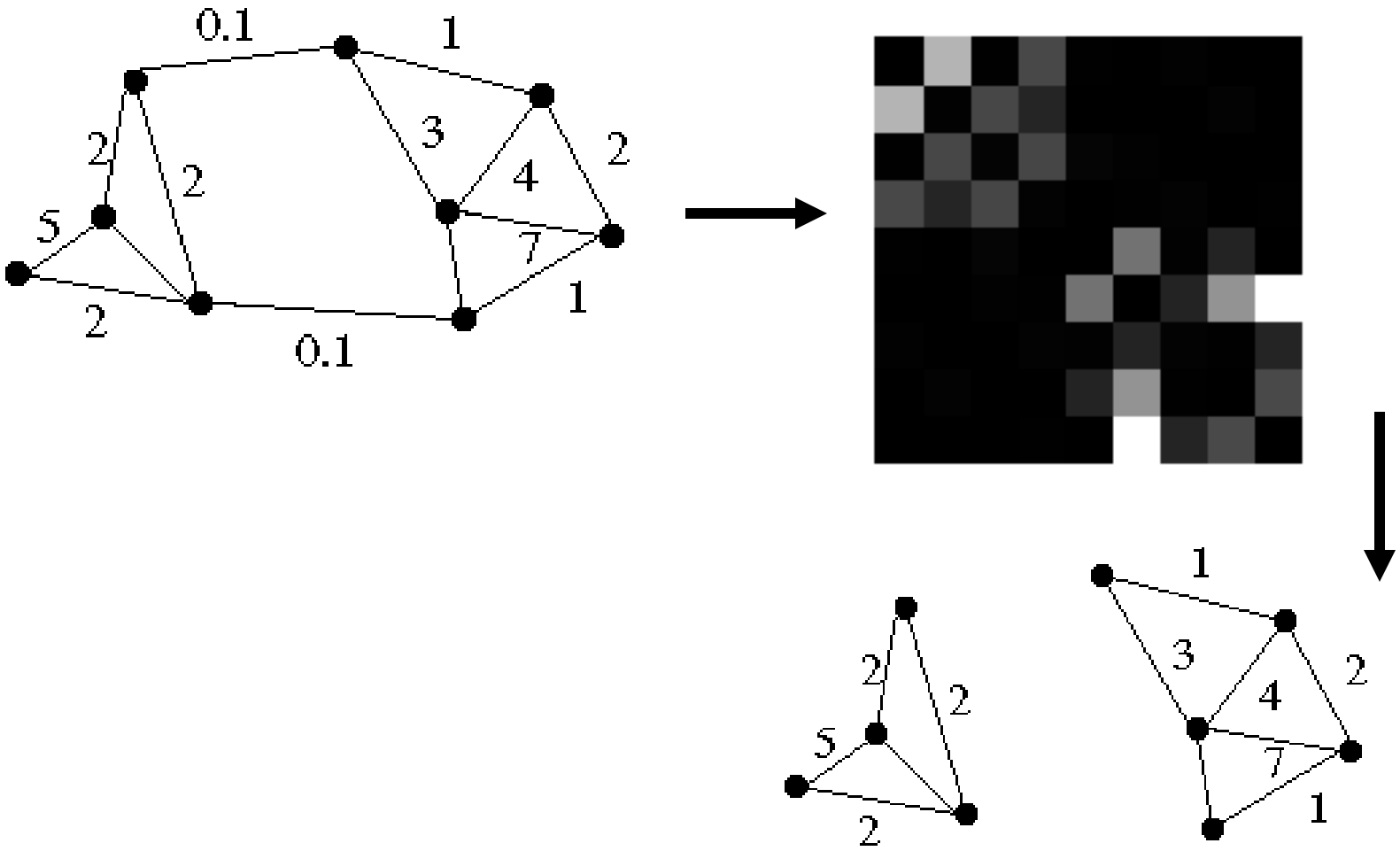
- Two major types:
  - monothetic:
    - split by one variable at a time -- restricts choice search space
    - analogous to DTs
  - polythetic
    - splits by all variables at once -- many choices makes difficult

- Less commonly used than agglomerative methods
  - generally more computationally intensive
    - more choices in search space

Scalability of Hierarchical Clustering

- N objects to cluster
- Hierarchical clustering algorithms scale as $O(N^2)$ to $O(N^3)$
  - Why?
- This is problematic for large N……
- Solutions?
  - Use K-means (or a similar algorithm) to create an initial set of K clusters and then use hierarchical clustering from there
  - Use approximate fast algorithms

Spectral/Graph-based Clustering, e.g., Min-Cut Methods



Summary

- Many different approaches and algorithms to clustering
- No “optimal” or “best” approach
- Computational complexity may be an issue for large N
- Data dimensionality can also be an issue
- Validation/selection of K is often an ill-posed problem
  - Often no “right answer” on what the optimal number of clusters is

General References on Clustering

- *Cluster Analysis* (5th ed), B. S. Everitt, S. Landau, M. Leese, and D. Stahl, Wiley, 2011 (comprehensive overview of clustering methods and algorithms)

- *Algorithms for Clustering Data*, A. K. Jain and R. C. Dubes, 1988, Prentice Hall. (a bit outdated but has many useful ideas and references on clustering)

- How many clusters? which clustering method? answers via model-based cluster analysis, C. Fraley and A. E. Raftery, the *Computer Journal*, 1998. (good overview article on probabilistic model-based clustering)

- Chapters 16 and 17 in the Manning/Raghavan/Schutze text, focused on clustering of text documents