

# **Data Mining**

## **- Unsupervised Learning**

---

**Michalis Vazirgiannis**  
**INF 553 – Data Bases**

# Introduction to Data Mining

---

Based on slides from the book: "Principles of Data Mining", D. Hand, H.

Mannila, P. Smyth, Cambridge press

# Introduction to Data Mining

---

- What is data mining?
- Data sets
  - The “data matrix”
  - Other data formats
- Data mining tasks
  - Prediction and description
- Data mining algorithms
  - Score functions, models, and optimization methods
- The dark side of data mining

---

# What is data mining?

---

## What is data mining?

“Data-driven discovery of models and patterns from massive observational data sets”

---

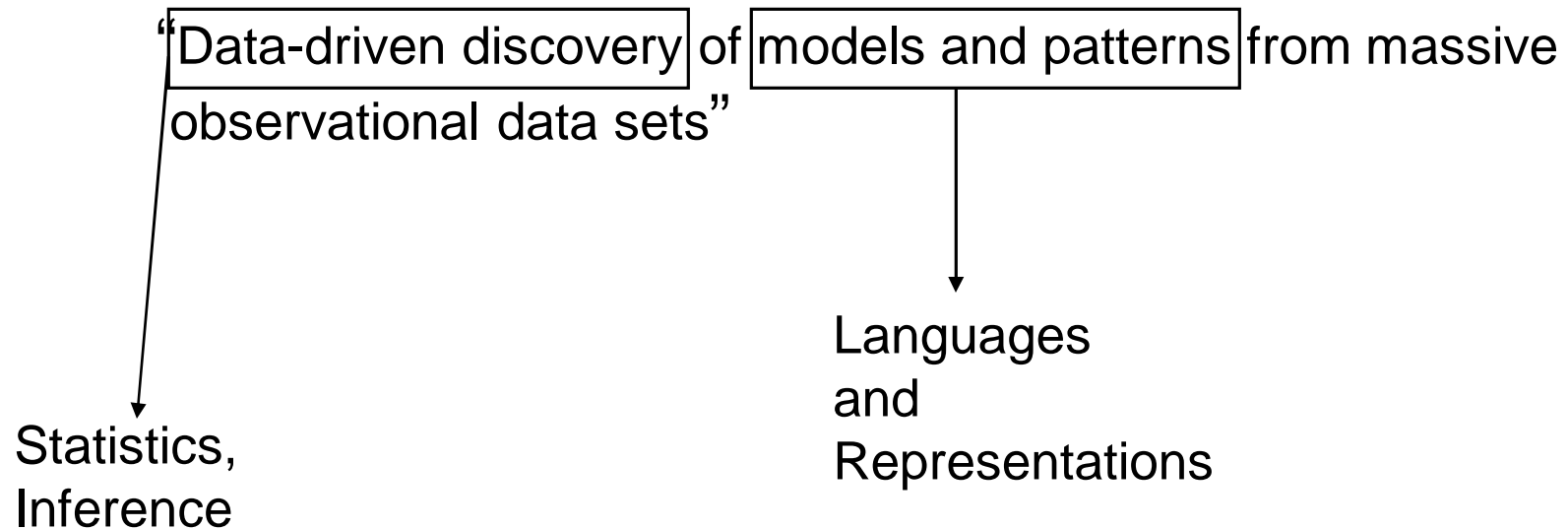
## What is data mining?

“Data-driven discovery of models and patterns from massive observational data sets”

Statistics,  
Inference

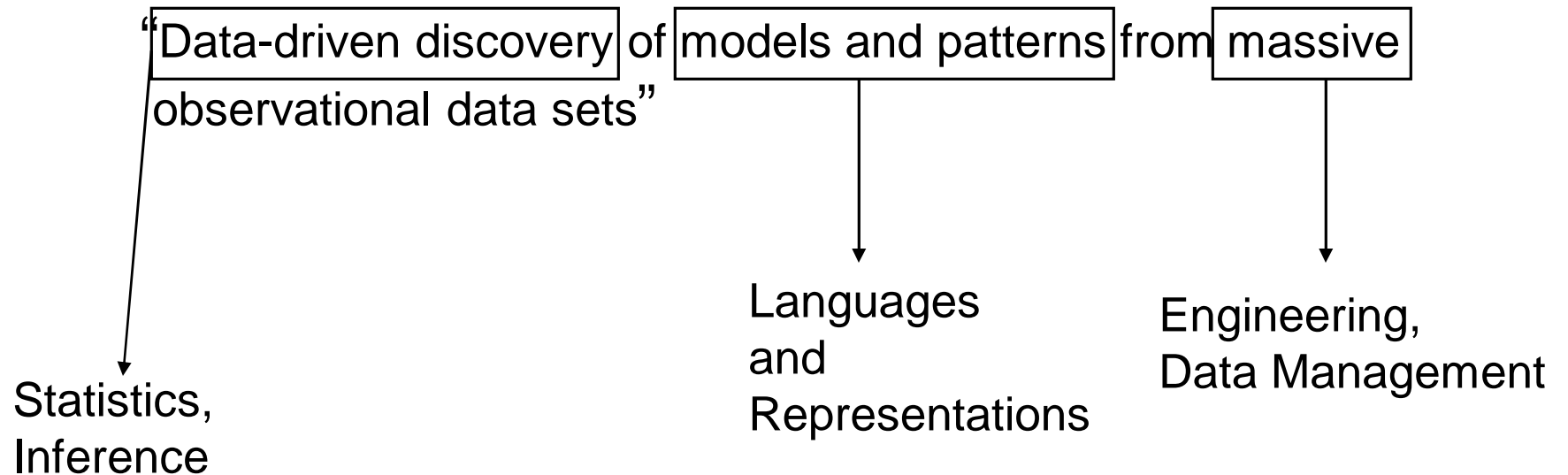
---

## What is data mining?



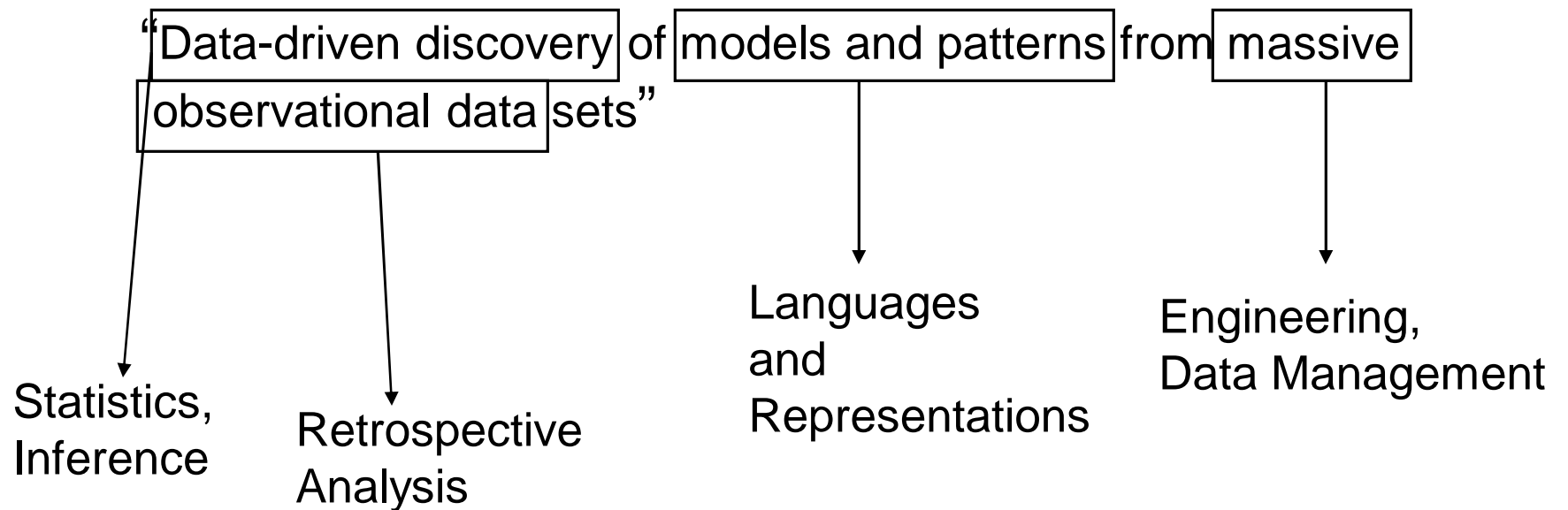
---

## What is data mining?



---

## What is data mining?



---

## Technological Driving Factors

- Larger, cheaper memory
  - Moore's law for magnetic disk density  
"capacity doubles every 18 months"
  - storage cost per byte falling rapidly
  
- Faster, cheaper processors
  - the CRAY of 15 years ago is now on your desk
  
- Success of Relational Database technology
  - everybody is a "data owner"
  
- New ideas in machine learning/statistics
  - Boosting, SVMs, decision trees, etc

# Examples of data volumes

---

- MEDLINE text database
  - 12 million published articles
  
- Google
  - 15.2 billion Web pages indexed
  - 80 million site visitors per day
  
- CALTRANS loop sensor data
  - Every 30 seconds, thousands of sensors, 2Gbytes per day
  
- NASA MODIS satellite
  - Coverage at 250m resolution, 37 bands, whole earth, every day
  
- Walmart transaction data
  - Order of 100 million transactions per day

# Two Types of Data

---

- Experimental Data
  - Hypothesis H
  - design an experiment to test H
  - collect data, infer how likely it is that H is true
  - e.g., clinical trials in medicine
  
- Observational or Retrospective or Secondary Data
  - massive non-experimental data sets
    - e.g., human genome, atmospheric simulations, etc
  - assumptions of experimental design no longer valid
  - how can we use such data to do science?
    - data must support model exploration, hypothesis testing

# Data-Driven Discovery

---

- Observation data
  - cheap relative to experimental data
    - Examples:
      - Transaction data archives for retail stores, airlines, etc
      - Web logs for Amazon, Google, etc
      - The human/mouse/rat genome
      - Etc., etc
    - ⇒ makes sense to leverage available data
    - ⇒ useful (?) information may be hidden in vast archives of data
  
- Contrast data mining with traditional statistics
  - traditional stats: first hypothesize, then collect data, then analyze
  - data mining:
    - few if any a priori hypotheses,
    - data is usually already there
    - analysis is typically data-driven not hypothesis driven
  
  - Nonetheless, statistical ideas are very useful in data mining, e.g., in validating whether discovered knowledge is useful

---

Let the data speak...



---

Let the data speak...

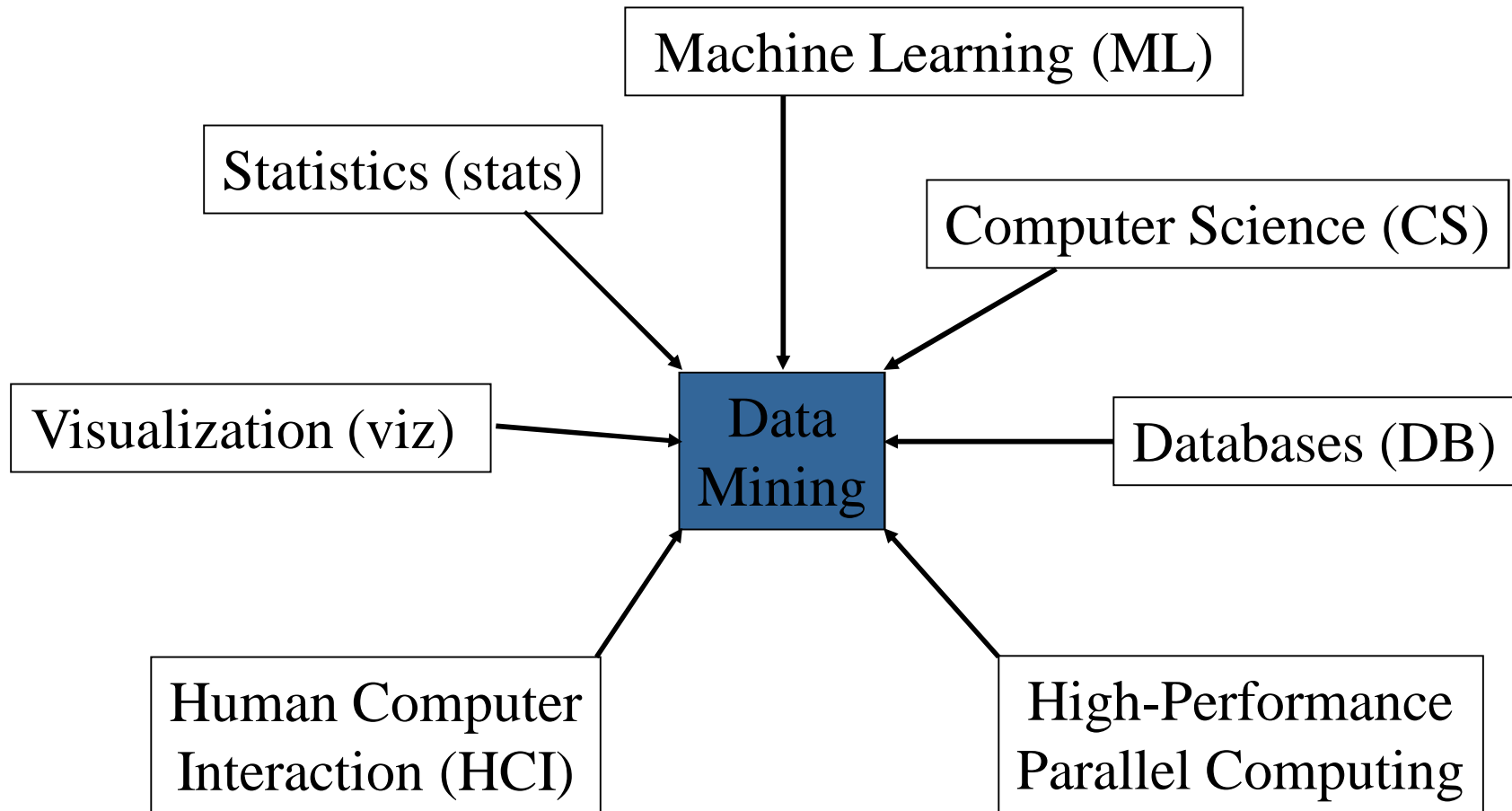


The data may have quite a lot to say..... but it may just be noise!



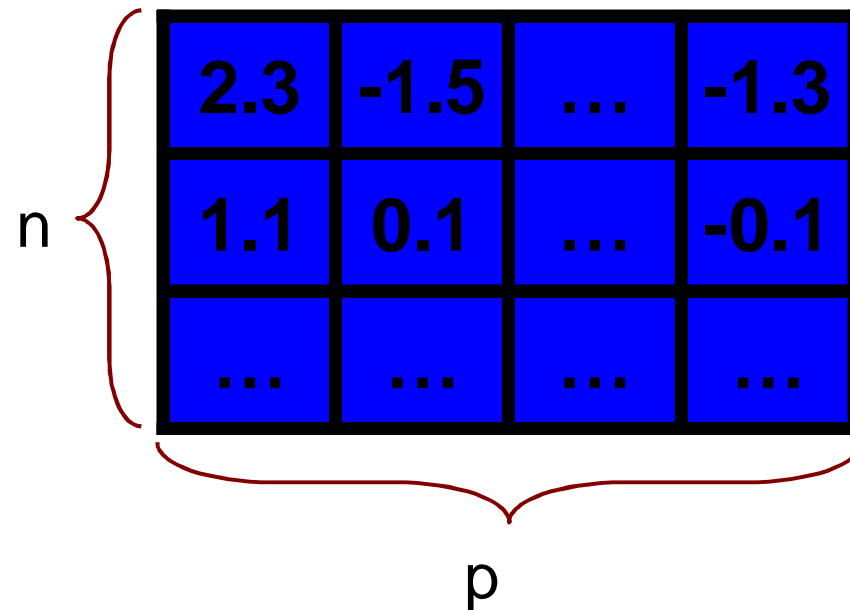
# DM: Intersection of Many Fields

---



## Flat File or Vector Data

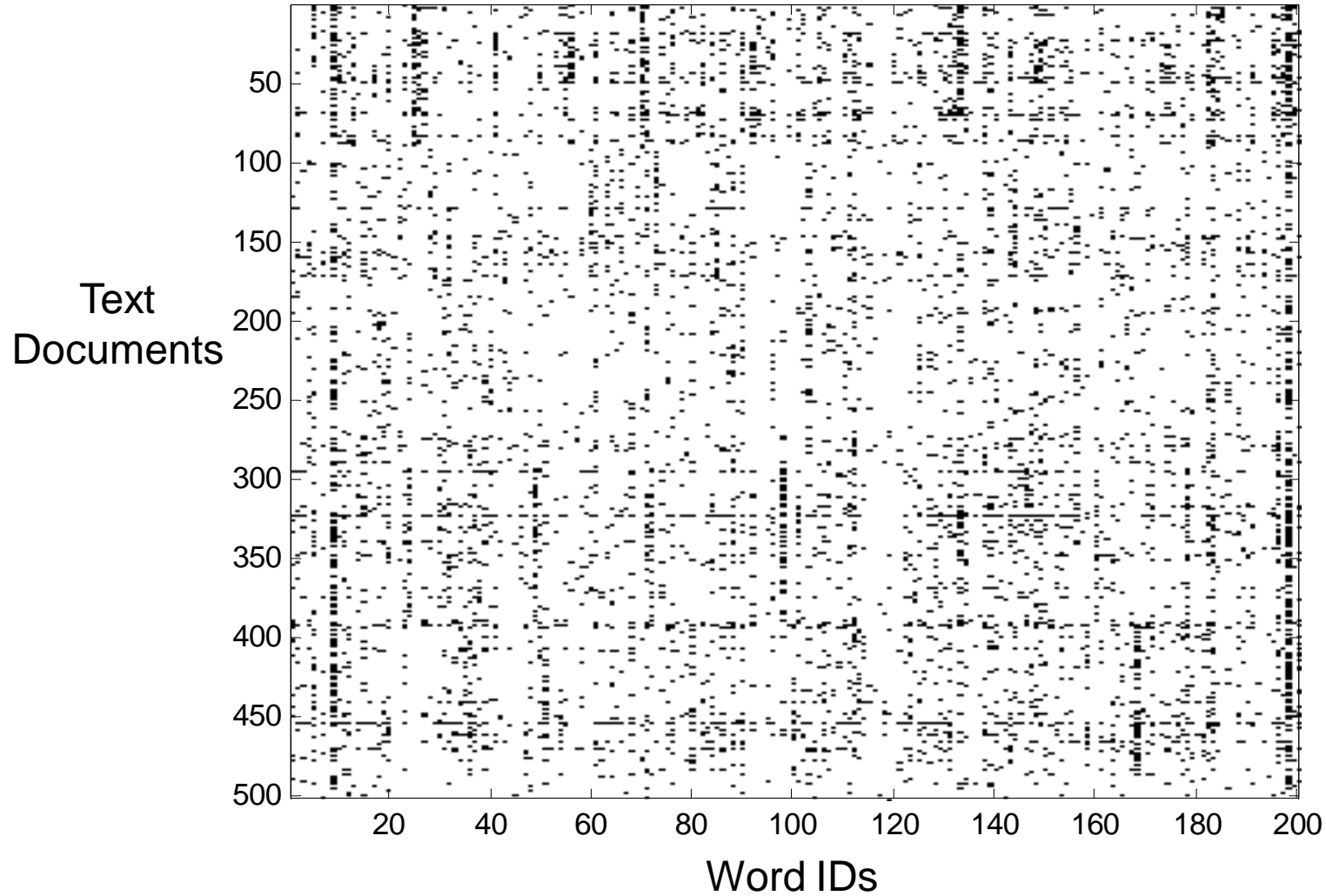
---



- Rows = objects
- Columns = measurements on objects
  - Represent each row as a  $p$ -dimensional vector, where  $p$  is the dimensionality
    - In effect, embed our objects in a  $p$ -dimensional vector space
    - Often useful, but always appropriate
- Both  $n$  and  $p$  can be very large in certain data mining applications

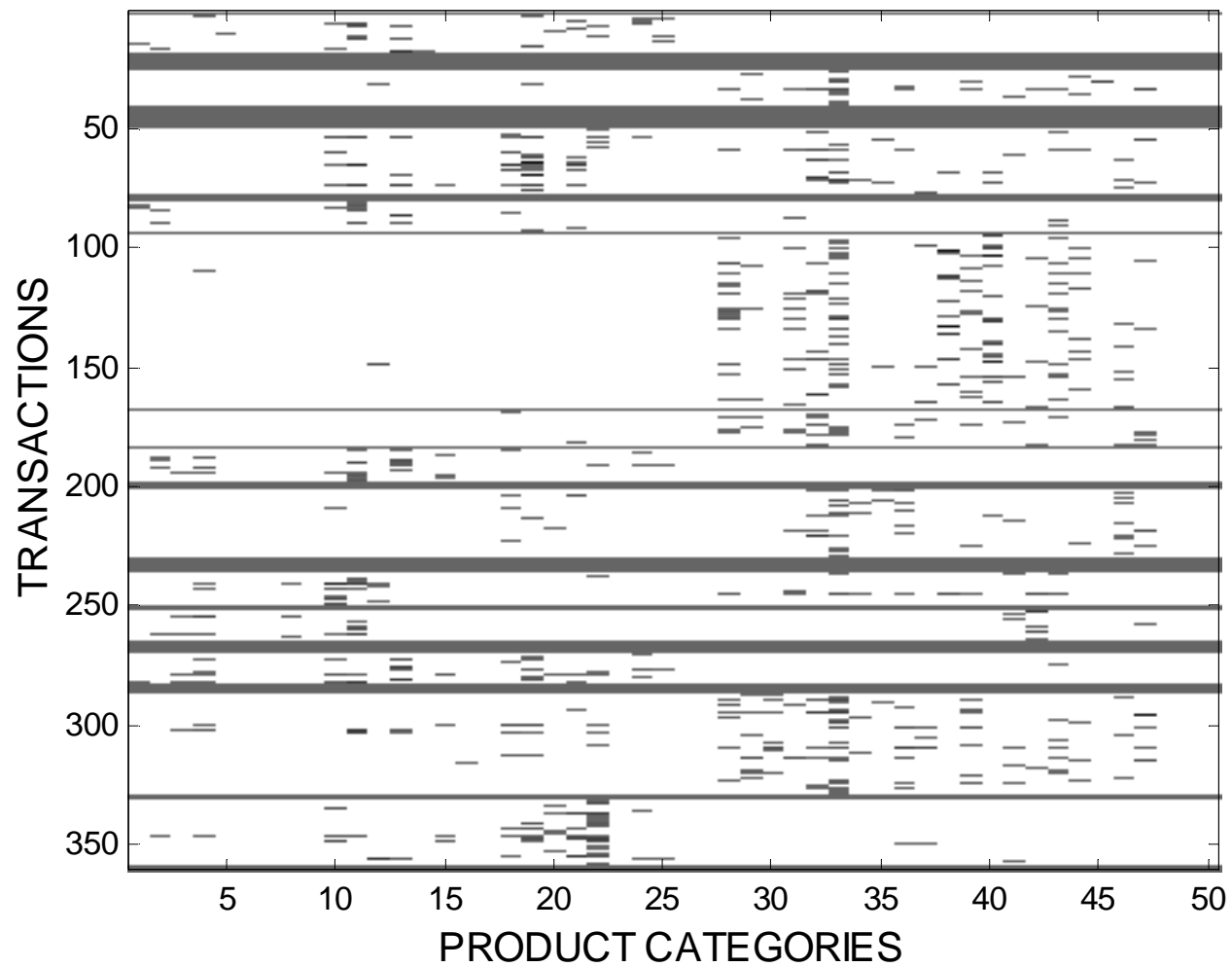
# Sparse Matrix (Text) Data

---



# "Market Basket" Data

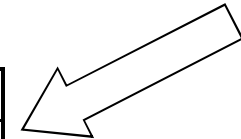
---



# Sequence (Web) Data

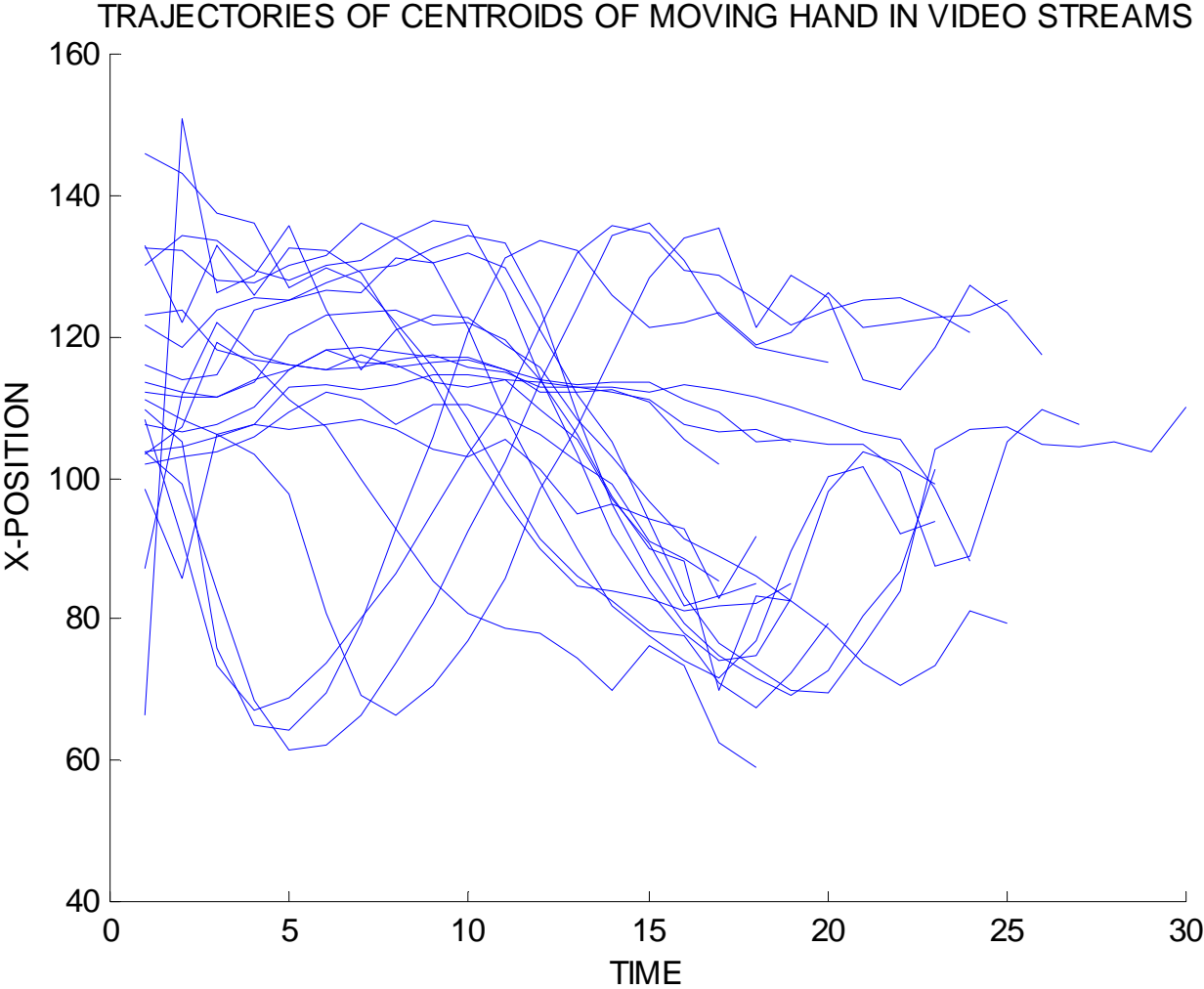
128.195.36.195, -, 3/22/00, 10:35:11, W3SVC, SRVR1, 128.200.39.181, 781, 363, 875, 200, 0, GET, /top.html, -,  
 128.195.36.195, -, 3/22/00, 10:35:16, W3SVC, SRVR1, 128.200.39.181, 5288, 524, 414, 200, 0, POST, /spt/main.html, -,  
 128.195.36.195, -, 3/22/00, 10:35:17, W3SVC, SRVR1, 128.200.39.181, 30, 280, 111, 404, 3, GET, /spt/images/bk1.jpg, -,  
 128.195.36.101, -, 3/22/00, 16:18:50, W3SVC, SRVR1, 128.200.39.181, 60, 425, 72, 304, 0, GET, /top.html, -,  
 128.195.36.101, -, 3/22/00, 16:18:58, W3SVC, SRVR1, 128.200.39.181, 8322, 527, 414, 200, 0, POST, /spt/main.html, -,  
 128.195.36.101, -, 3/22/00, 16:18:59, W3SVC, SRVR1, 128.200.39.181, 0, 280, 111, 404, 3, GET, /spt/images/bk1.jpg, -,  
 128.200.39.17, -, 3/22/00, 20:54:37, W3SVC, SRVR1, 128.200.39.181, 140, 199, 875, 200, 0, GET, /top.html, -,  
 128.200.39.17, -, 3/22/00, 20:54:55, W3SVC, SRVR1, 128.200.39.181, 17766, 365, 414, 200, 0, POST, /spt/main.html, -,  
 128.200.39.17, -, 3/22/00, 20:54:55, W3SVC, SRVR1, 128.200.39.181, 0, 258, 111, 404, 3, GET, /spt/images/bk1.jpg, -,  
 128.200.39.17, -, 3/22/00, 20:55:07, W3SVC, SRVR1, 128.200.39.181, 0, 258, 111, 404, 3, GET, /spt/images/bk1.jpg, -,  
 128.200.39.17, -, 3/22/00, 20:55:36, W3SVC, SRVR1, 128.200.39.181, 1061, 382, 414, 200, 0, POST, /spt/main.html, -,  
 128.200.39.17, -, 3/22/00, 20:55:36, W3SVC, SRVR1, 128.200.39.181, 0, 258, 111, 404, 3, GET, /spt/images/bk1.jpg, -,  
 128.200.39.17, -, 3/22/00, 20:55:39, W3SVC, SRVR1, 128.200.39.181, 0, 258, 111, 404, 3, GET, /spt/images/bk1.jpg, -,  
 128.200.39.17, -, 3/22/00, 20:56:03, W3SVC, SRVR1, 128.200.39.181, 1081, 382, 414, 200, 0, POST, /spt/main.html, -,  
 128.200.39.17, -, 3/22/00, 20:56:04, W3SVC, SRVR1, 128.200.39.181, 0, 258, 111, 404, 3, GET, /spt/images/bk1.jpg, -,  
 128.200.39.17, -, 3/22/00, 20:56:33, W3SVC, SRVR1, 128.200.39.181, 0, 262, 72, 304, 0, GET, /top.html, -,  
 128.200.39.17, -, 3/22/00, 20:56:52, W3SVC, SRVR1, 128.200.39.181, 19598, 382, 414, 200, 0, POST, /spt/main.html, -,

User 1	2	3	2	2	3	3	3	1	1	1	3	1	3	3	3
User 2	3	3	3	1	1	1									
User 3	7	7	7	7	7	7	7								
User 4	1	5	1	1	1	5	1	5	1	1	1	1	1	1	
User 5	5	1	1	5											
...															



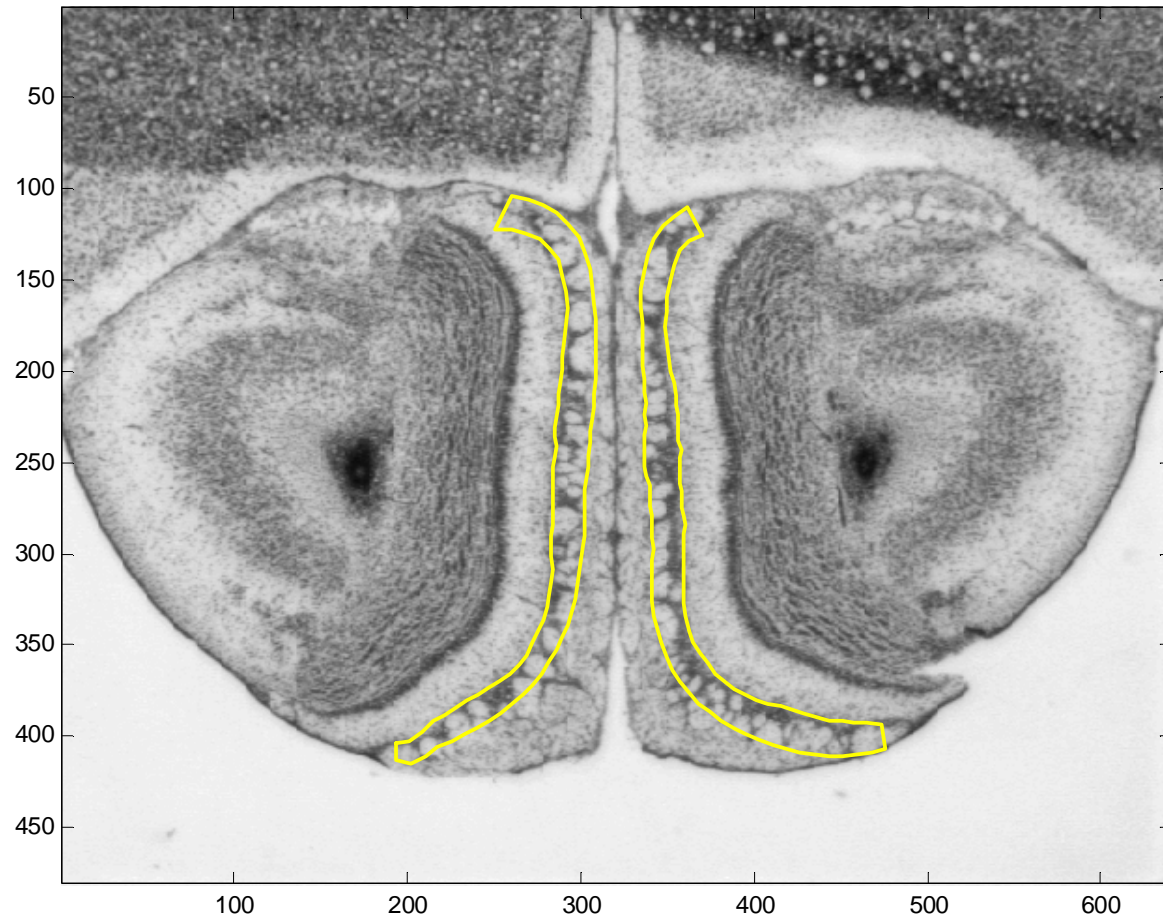
# Time Series Data

---



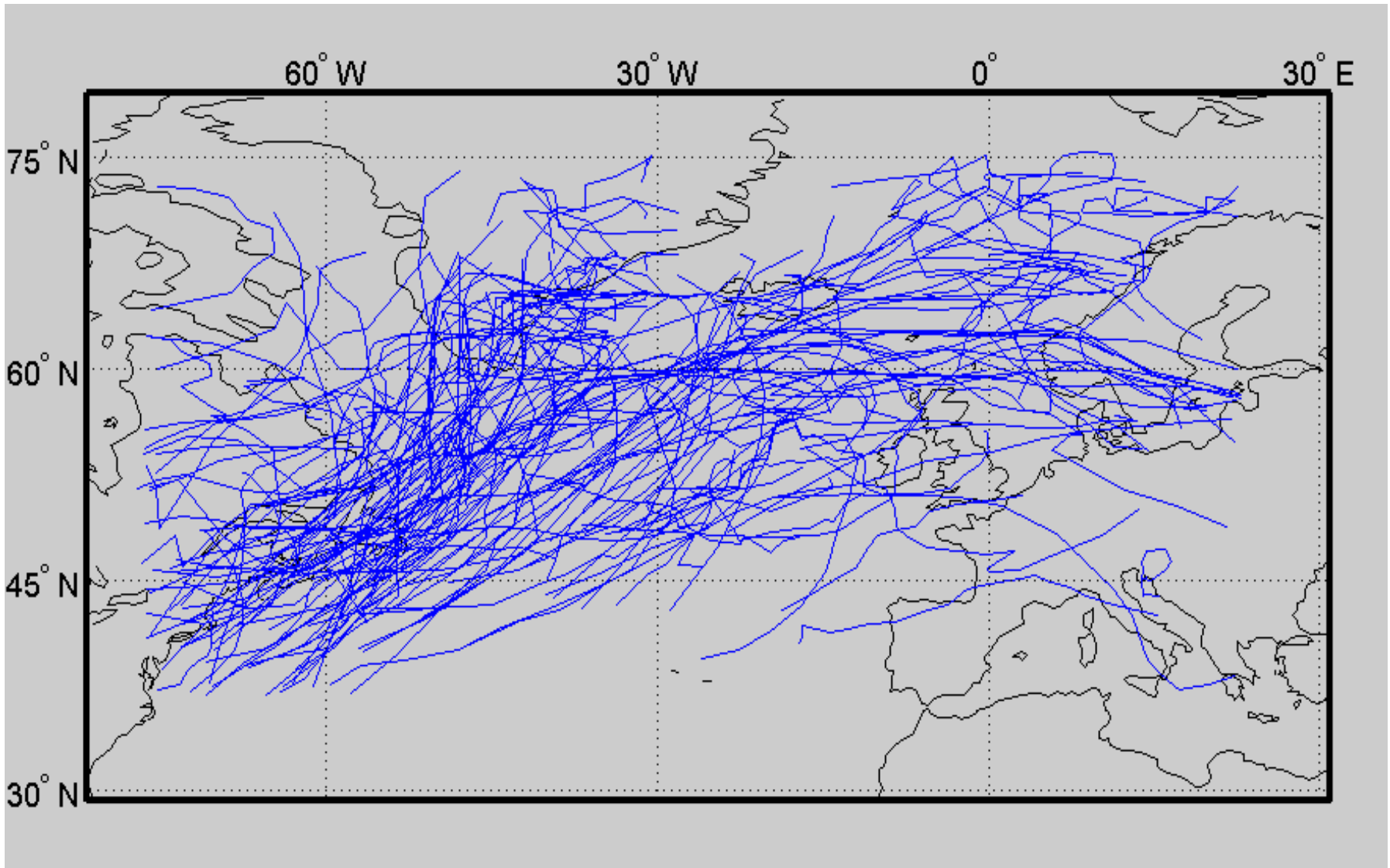
# Image Data

---

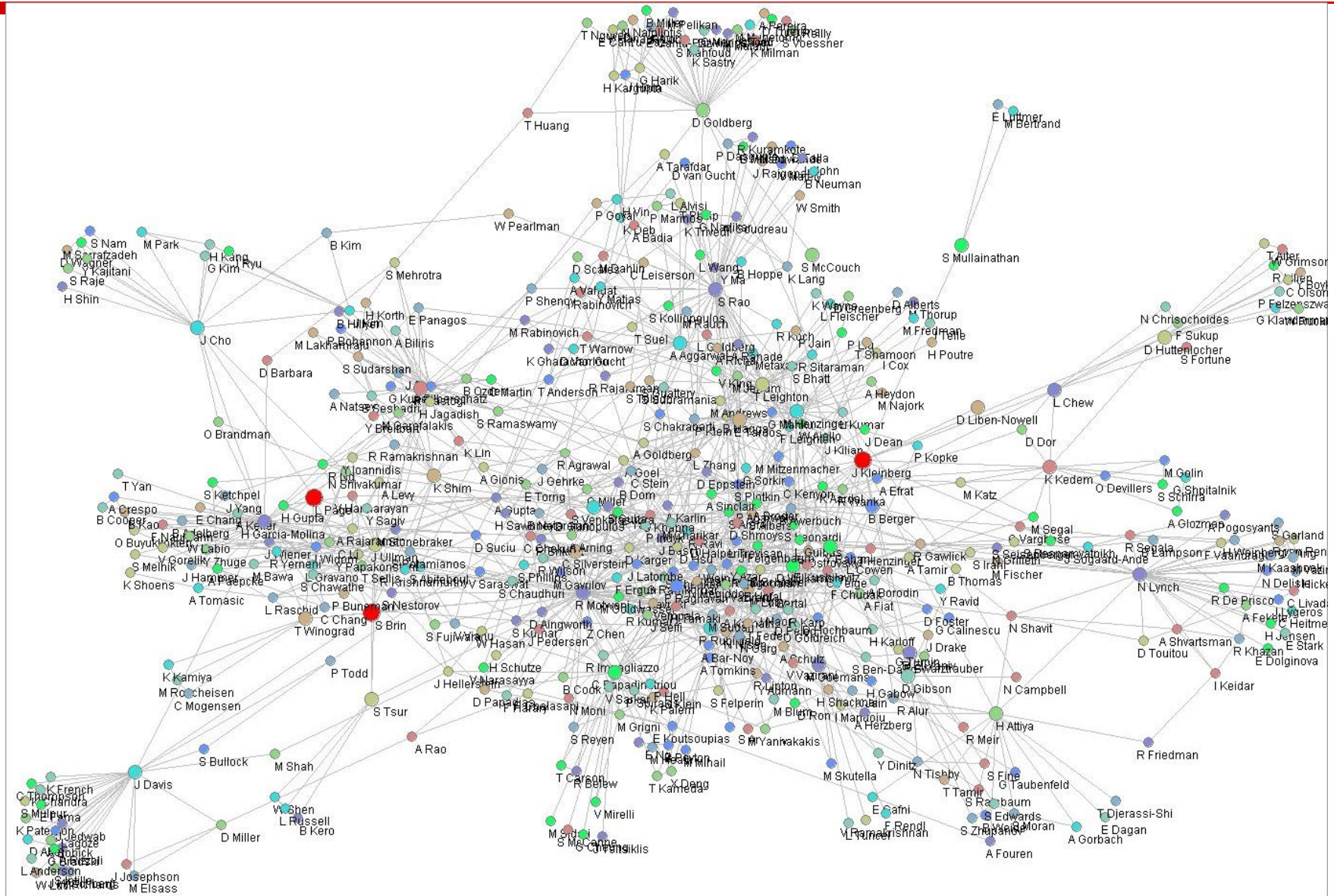


# Spatio-temporal data

---



# Relational Data



# Different Data Mining Tasks

---

- Exploratory Data Analysis
- Descriptive Modeling
- Predictive Modeling
- Discovering Patterns and Rules
- + others....

# Exploratory Data Analysis

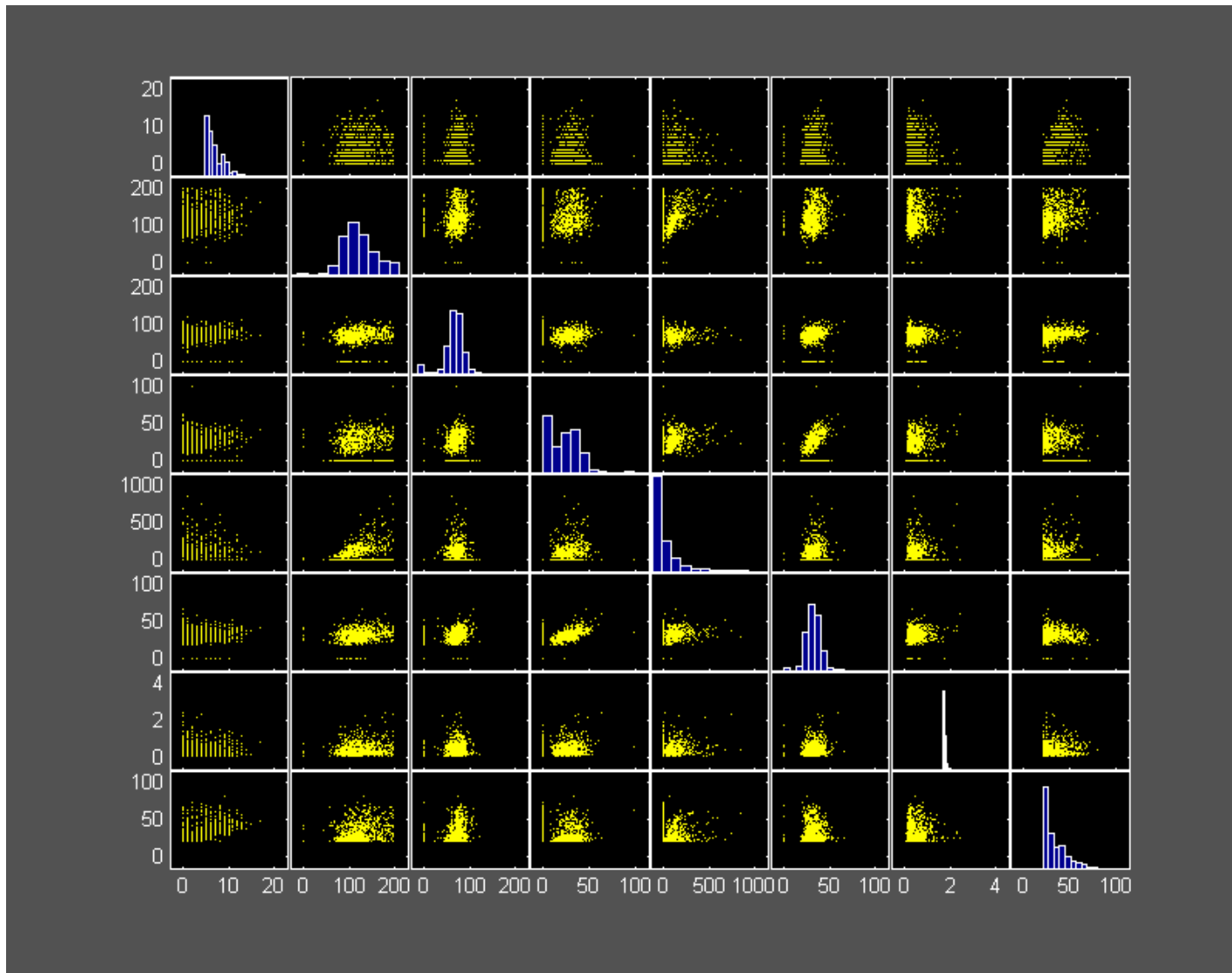
---

- Getting an overall sense of the data set
  - Computing summary statistics:
    - Number of distinct values, max, min, mean, median, variance, skewness,...
  
- Visualization is widely used
  - 1d histograms
  - 2d scatter plots
  - Higher-dimensional methods
  
- Useful for data checking
  - E.g., finding that a variable is always integer valued or positive
  - Finding the some variables are highly skewed
  
- Simple exploratory analysis can be extremely valuable
  - You should always “look” at your data before applying any data mining algorithms

# Example of Exploratory Data Analysis

(Pima Indians data, scatter plot matrix)

---



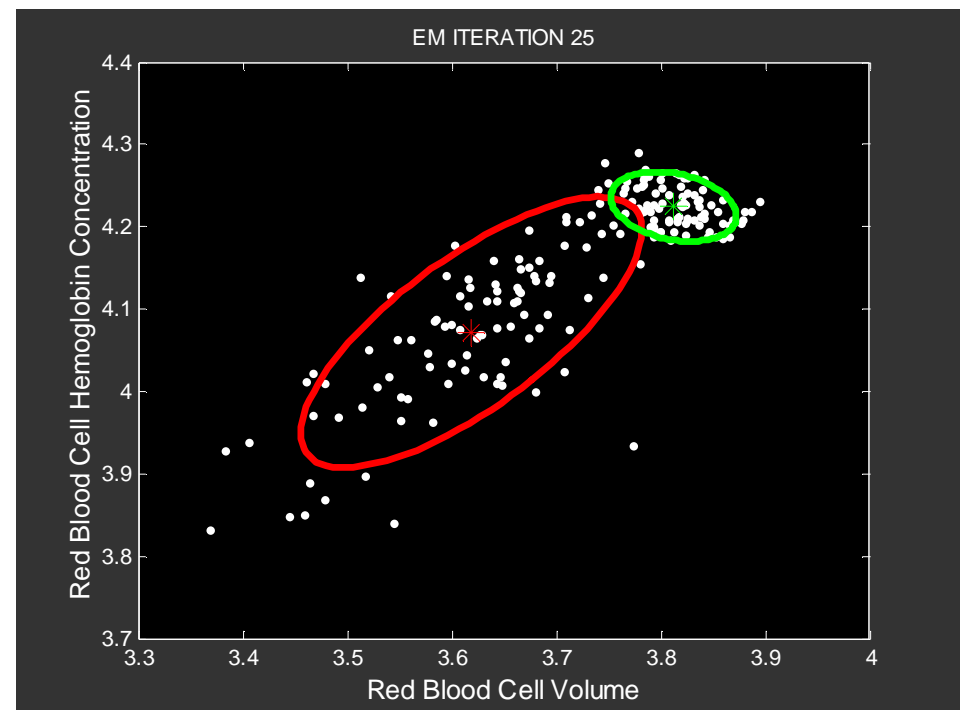
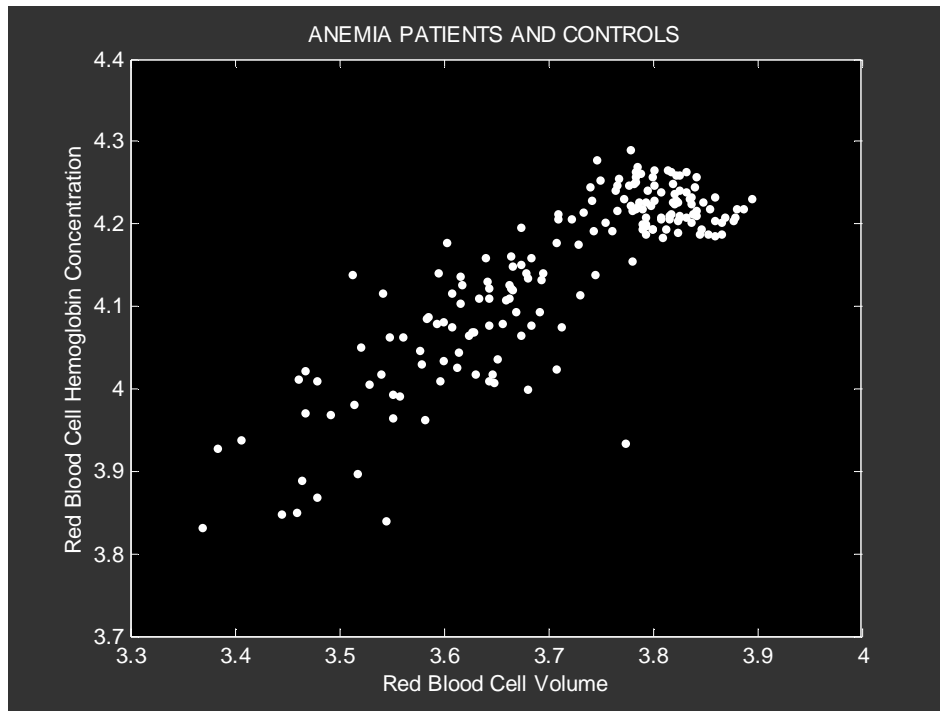
# Descriptive Modeling

---

- Goal is to build a “generative” or “descriptive” model,
  - E.g., a model that could simulate the data if needed
  - models the underlying process
  
- Examples:
  - Density estimation:
    - estimate the joint distribution  $P(x_1, \dots, x_p)$
  - Cluster analysis:
    - Find natural groups in the data
  - Dependency models among the  $p$  variables
    - Learning a Bayesian network for the data

# Example of Descriptive Modeling

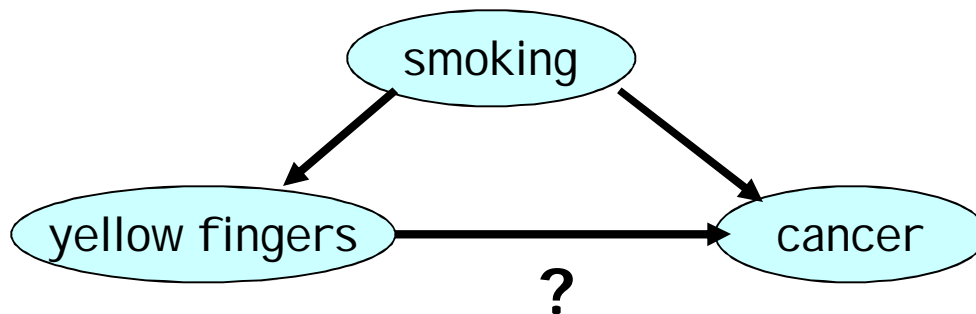
---



# Another Example of Descriptive Modeling

---

- Learning Directed Graphical Models (aka Bayes Nets)
  - goal: learn directed relationships among  $p$  variables
  - techniques: directed (causal) graphs
  - challenge: distinguishing between correlation and causation
    - example: Do yellow fingers cause lung cancer?  
hidden cause: smoking



# Predictive Modeling

---

- Predict one variable  $Y$  given a set of other variables  $\underline{X}$ 
  - Here  $\underline{X}$  could be a  $p$ -dimensional vector
  
- Classification:  $Y$  is categorical
- Regression:  $Y$  is real-valued
  
- In effect this is function approximation, learning the relationship between  $Y$  and  $\underline{X}$
  
- Many, many algorithms for predictive modeling in statistics and machine learning
  
- Often the emphasis is on predictive accuracy, less emphasis on understanding the model

# Example of Predictive Modeling

---

- Background
  - AT&T has about 100 million customers
  - It logs 200 million calls per day, 40 attributes each
  - 250 million unique telephone numbers
  - Which are business and which are residential?
  
- Solution (Pregibon and Cortes, AT&T, 1997)
  - Proprietary model, using a few attributes, trained on known business customers to adaptively track  $p(\text{business}|\text{data})$
  - Significant systems engineering: data are downloaded nightly, model updated (20 processors, 6Gb RAM, terabyte disk farm)
  
- Status:
  - running daily at AT&T
  - HTML interface used by AT&T marketing

# Pattern Discovery

---

- Goal is to discover interesting “local” patterns in the data rather than to characterize the data globally
  
- given market basket data we might discover that
  - If customers buy wine and bread then they buy cheese with probability 0.9
  - These are known as “association rules”
  
- Given multivariate data on astronomical objects
  - We might find a small group of previously undiscovered objects that are very self-similar in our feature space, but are very far away in feature space from all other objects

# Example of Pattern Discovery

---

ADACABDABAABBDDBCADDDDBCDDBC**CBBC**CDADADAADABDBBDAB  
ABBCDDDCDDABDCBBDBDBCBBABBBCBBABCBBACBBDBAACCADDA  
DBDBBCBCCBBBDCABDDBBADD BBBCCACDABBABDDCDDBBABDB  
DDBDDBCACDBBCCBBACDCADCBAACCADCCACCDDADCBCADADBAA  
CCDDDCBDBDCCCCACACACCDABDDBCADADBCBDDADABCCABDAAC  
ABCABACBDDDCBADCBADDDDCDDCADCCBBADABBAAADAAABCCB  
CABDBAADCBCDACBCABABCCBACBDABDDDADAABADCDCDBBCDB  
DADDCCBBCDBAADADBCAAAADBDCADBDBBBCDCBCCCDCCADAAD  
ACABDABAABBDDBCADDDDBCDDBC**CBBC**CDADADACCCDABAABBCB  
DBDBADBBBBCDADABABBDACDCDDDBBCDBBCBBCCDABCADDADBA  
CBBBCDBAAADDDBDDCABACBCADCDCBAAADCADDADAABBACCBB

# Example of Pattern Discovery

---

ADACABDABAABBDDBCADDDDBCDDBC**CBBCC**DADADAADABDBBDAB  
ABBCDDDCDDABDCBBDBDBCBBABBBCBBABCBBACBBDBAACCADDA  
DBDBB**CBBCC**BBBDCABDDBBADDBBBBCCACDABBABDDCDDBBABDB  
DDBDDBCACDBBCCBBACDCADCBAACCADCCACCDDADCBCADADBAA  
CCDDDCBDBDCCCCACACACCCDABDDBCADADBCBDDADABCCABDAAC  
ABCABACBDDDCBADCBADDDDCDDCADCCBBADABBAAADAAABCCB  
CABDBAADCBCDACBCABABCCBACBDABDDDADAABADCDCDBBCDB  
DADD**CBBCCD**BAADADBCAAAADBDCADBDBBBCCD**CCBCC**CDCCADAAD  
ACABDABAABBDDBCADDDDBCDDBC**CBBCC**DADADACCCDABAABBCB  
DBDBADBBBBBCDADABABBDACDCDDDBBCDBBCBBCCDABCADDADBA  
**CBBBC**CDBAAADDDBDDCABACBCADCDCBAAADCADDADAABBACCBB

# Example of Pattern Discovery

---

- IBM “Advanced Scout” System
  - Bhandari et al. (1997)
  - Every NBA basketball game is annotated,
    - e.g., time = 6 mins, 32 seconds
    - event = 3 point basket
    - player = Michael Jordan
  - This creates a huge untapped database of information
- IBM algorithms search for rules of the form  
    “If player A is in the game, player B’s scoring rate increases from 3.2 points per quarter to 8.7 points per quarter”
- IBM claimed around 1998 that all NBA teams except 1 were using this software..... the “other team” was Chicago.

# Structure: Models and Patterns

---

- Model = abstract representation of a process

e.g., very simple linear model structure

$$Y = a X + b$$

- a and b are parameters determined from the data
  - $Y = aX + b$  is the model structure
  - $Y = 0.9X + 0.3$  is a particular model
  - “All models are wrong, some are useful” (G.E. Box)
- 
- Pattern represents “local structure” in a data set
  - E.g., if  $X > x$  then  $Y > y$  with probability p
  - or a pattern might be a small cluster of outliers in multi-dimensional space

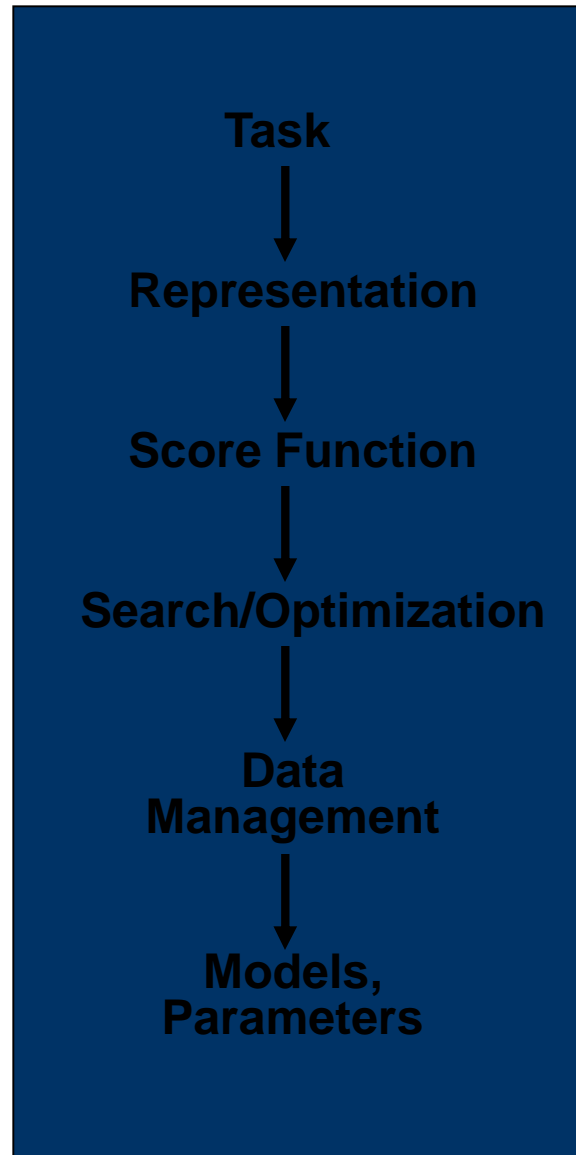
# Components of Data Mining Algorithms

---

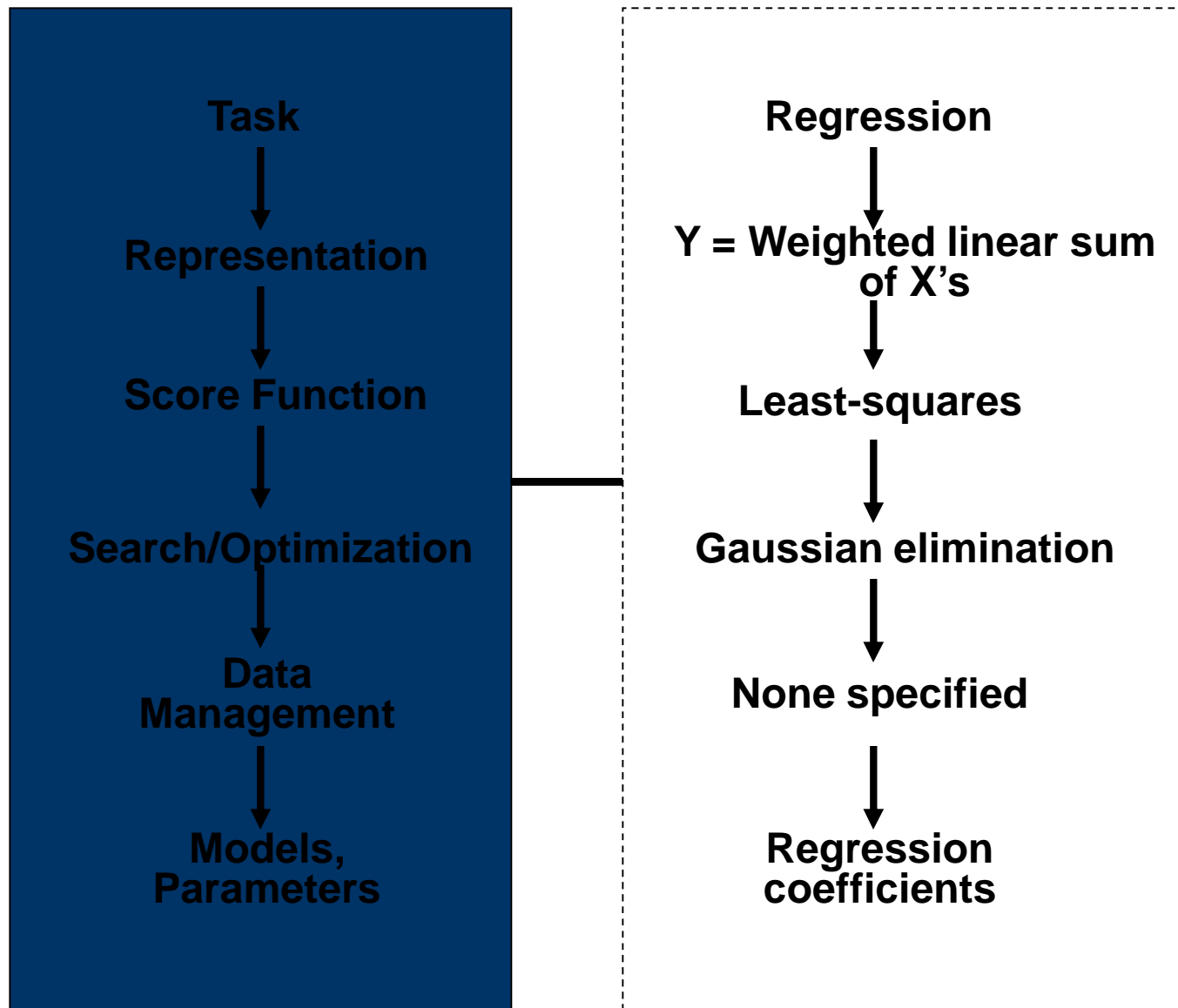
- Representation:
  - Determining the nature and structure of the representation to be used;
- Score function
  - quantifying and comparing how well different representations fit the data
- Search/Optimization method
  - Choosing an algorithmic process to optimize the score function; and
- Data Management
  - Deciding what principles of data management are required to implement the algorithms efficiently.

# What's in a Data Mining Algorithm?

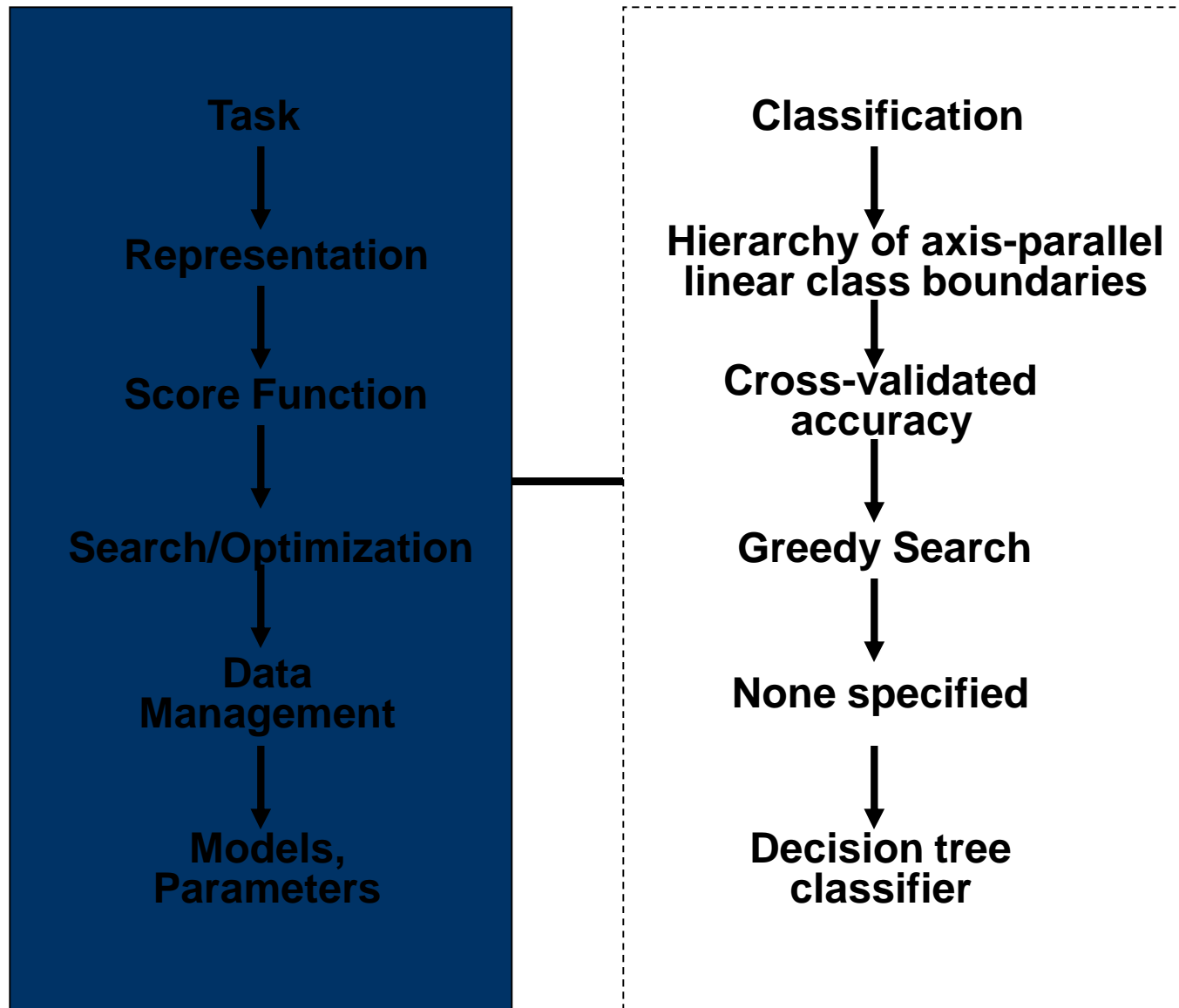
---



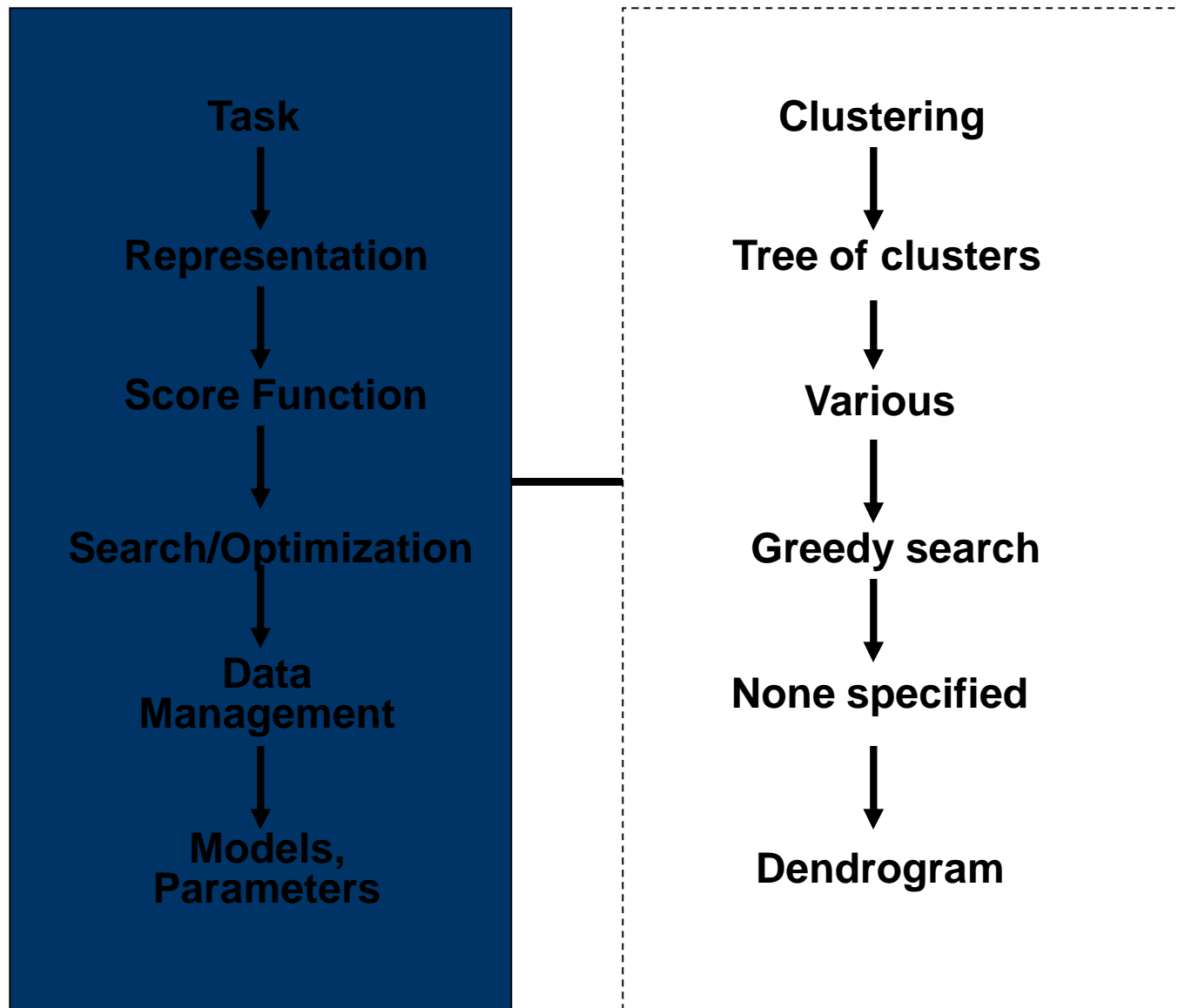
# An Example: Multivariate Linear Regression



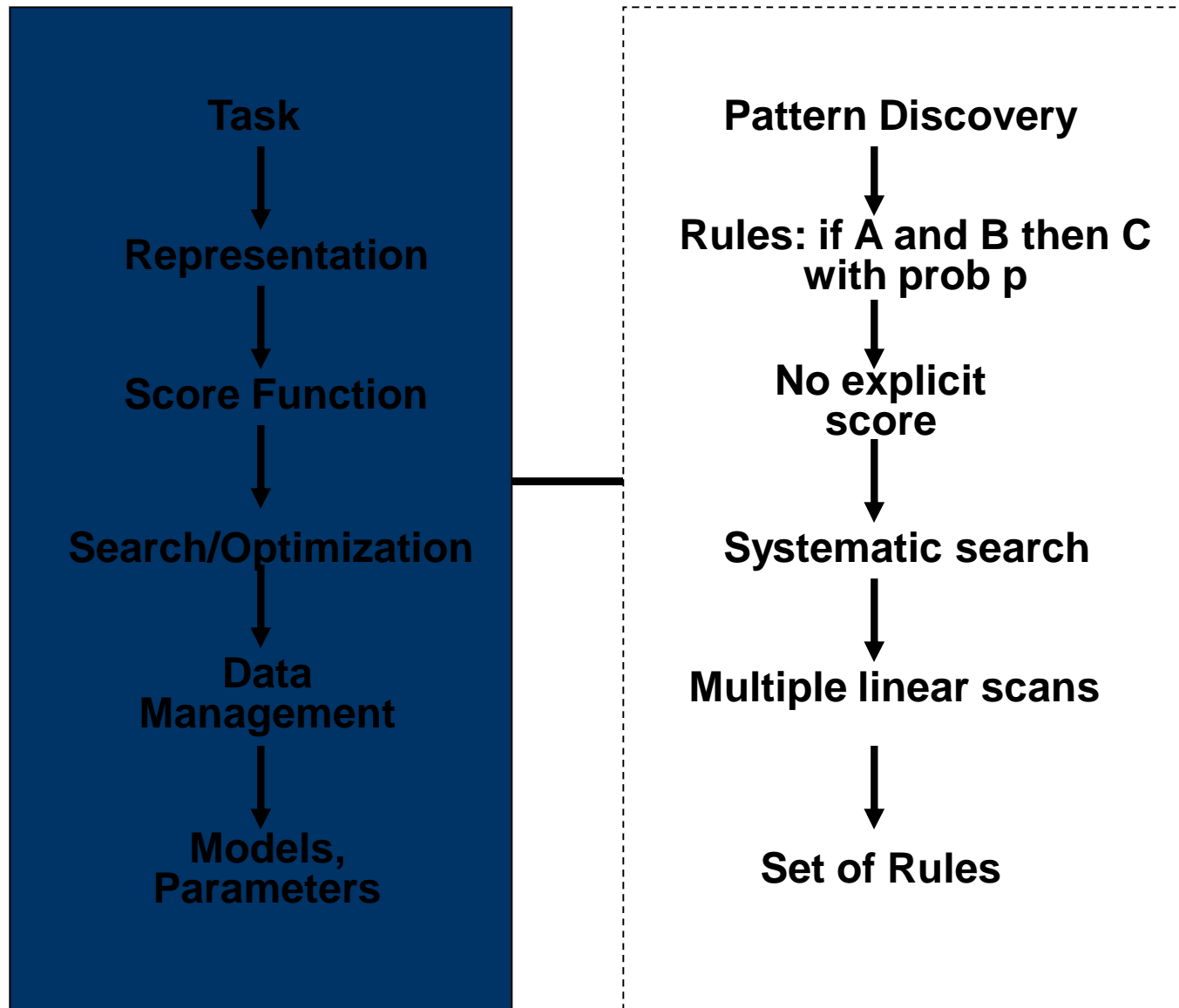
# An Example: Decision Trees (C4.5 or CART)



# An Example: Hierarchical Clustering



# An Example: Association Rules



# Unsupervised Learning (Clustering)

---

Using partially slides from Smyth, et. al, "Data Mining" book

# 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

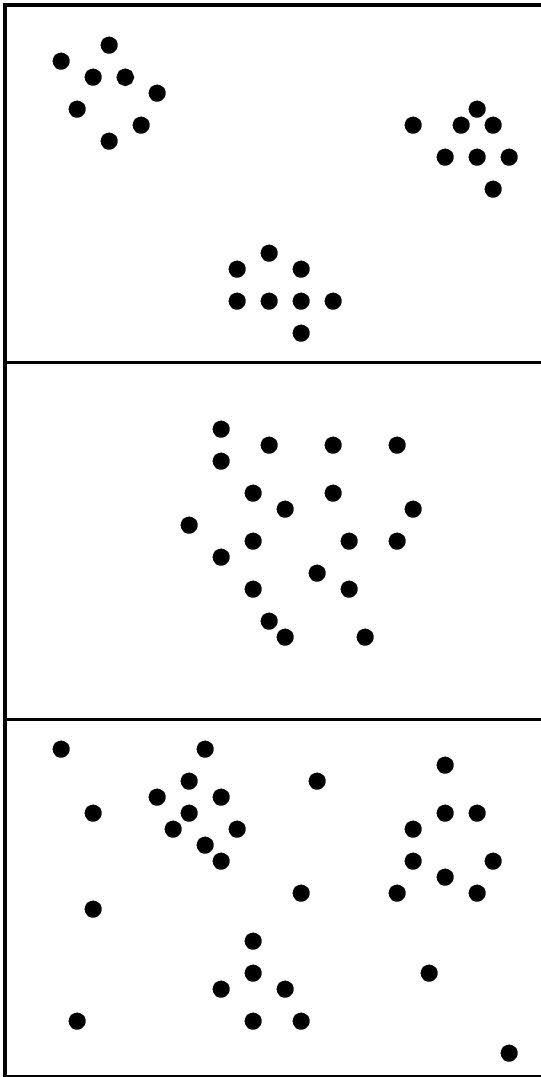
# Clustering

---

- “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

# K-means Clustering

---

- basic idea:
  - Score =  $wc(C)$  = sum-of-squares within cluster distance
  - start with randomly chosen cluster centers  $c_1 \dots c_k$
  - 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  $\underline{c}_1 \dots \underline{c}_k$
    - recompute cluster centers over data assigned to them
      - $\underline{c}_i = 1/(n_i) \sum_{x \in C_i} \underline{x}$
- algorithm terminates (finite number of steps)
  - decreases  $\text{Score}(X, C)$  each iteration membership changes
- converges to local maxima of  $\text{Score}(X, C)$ 
  - not necessarily the global maxima ...
  - different initial centers (seeds) can lead to diff local maxs

# K-means Complexity

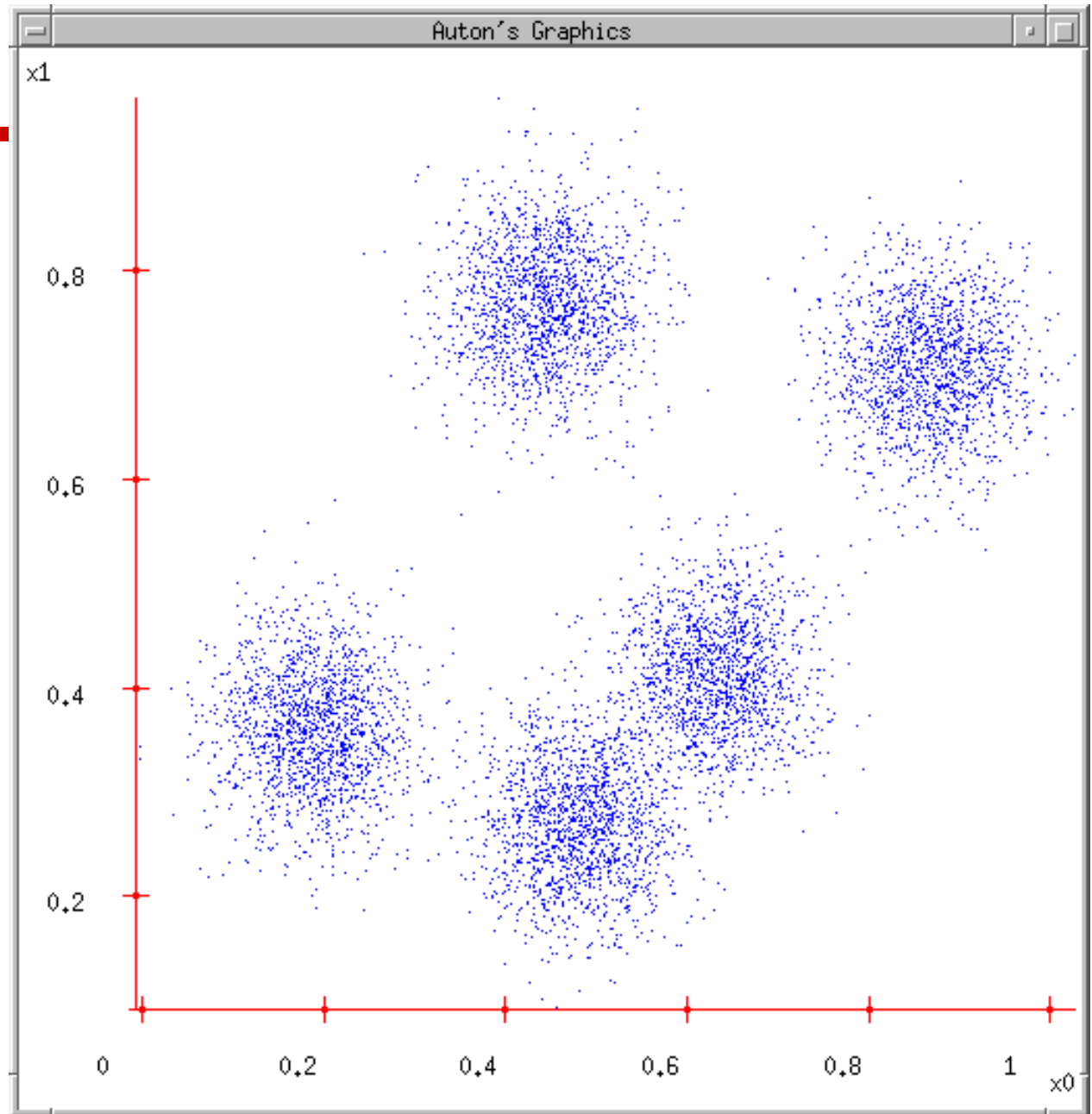
---

- time complexity =  $O(I e n k)$   $\ll$  exhaustive's  $n^k$ 
  - $I$  = number of iterations (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)

---

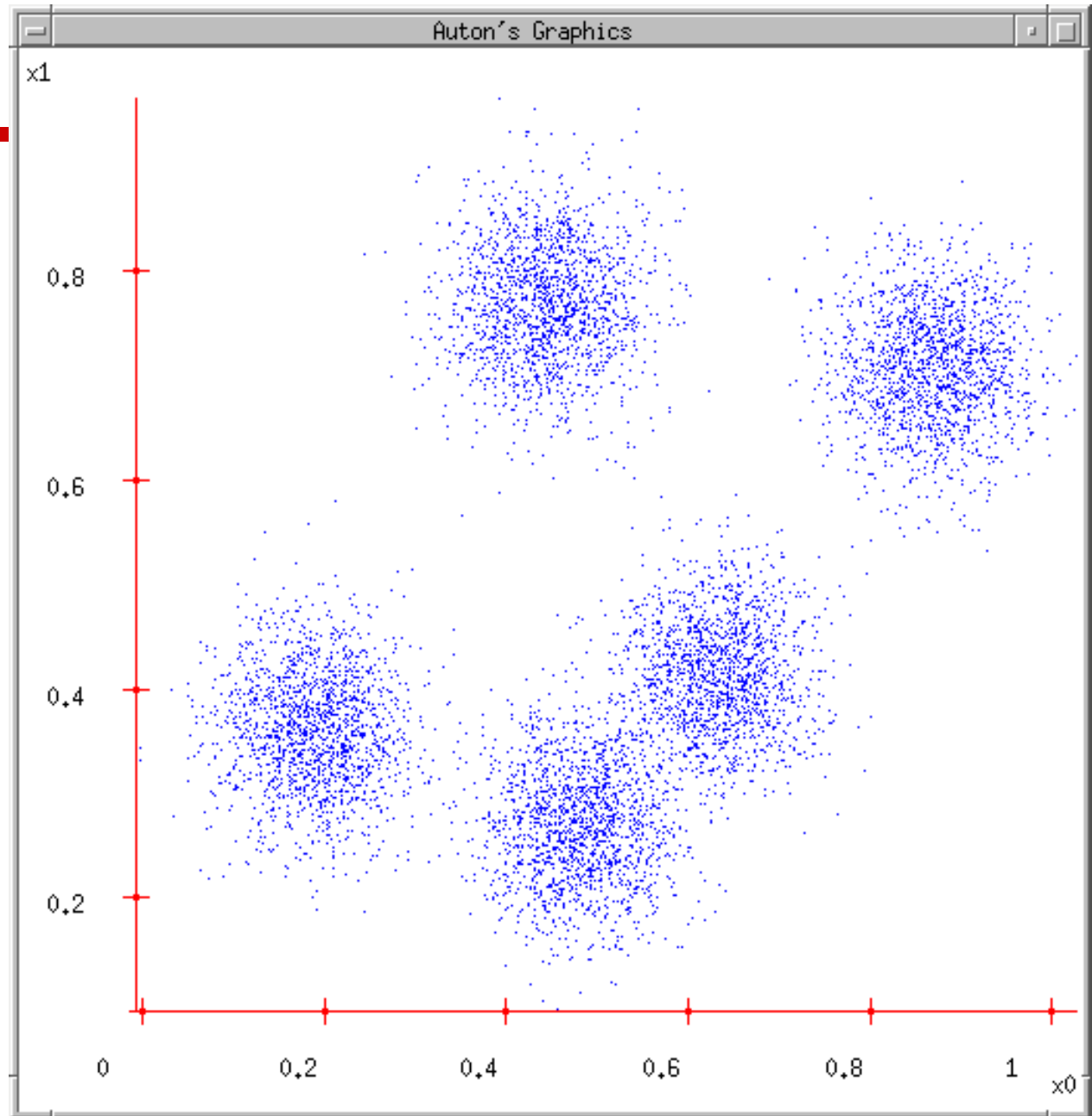
# K-means example

(courtesy of  
Andrew Moore,  
CMU)



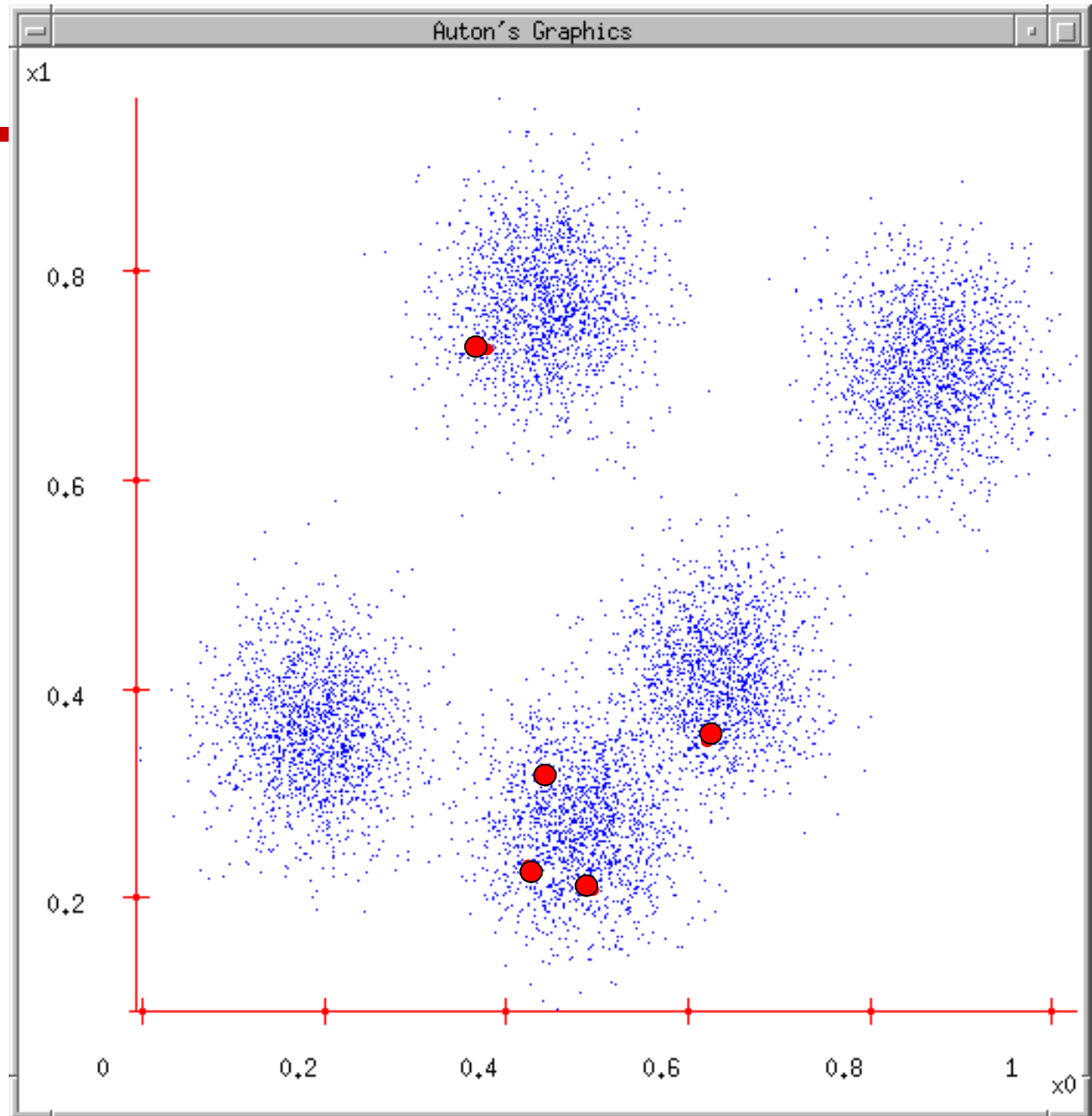
# K-means

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



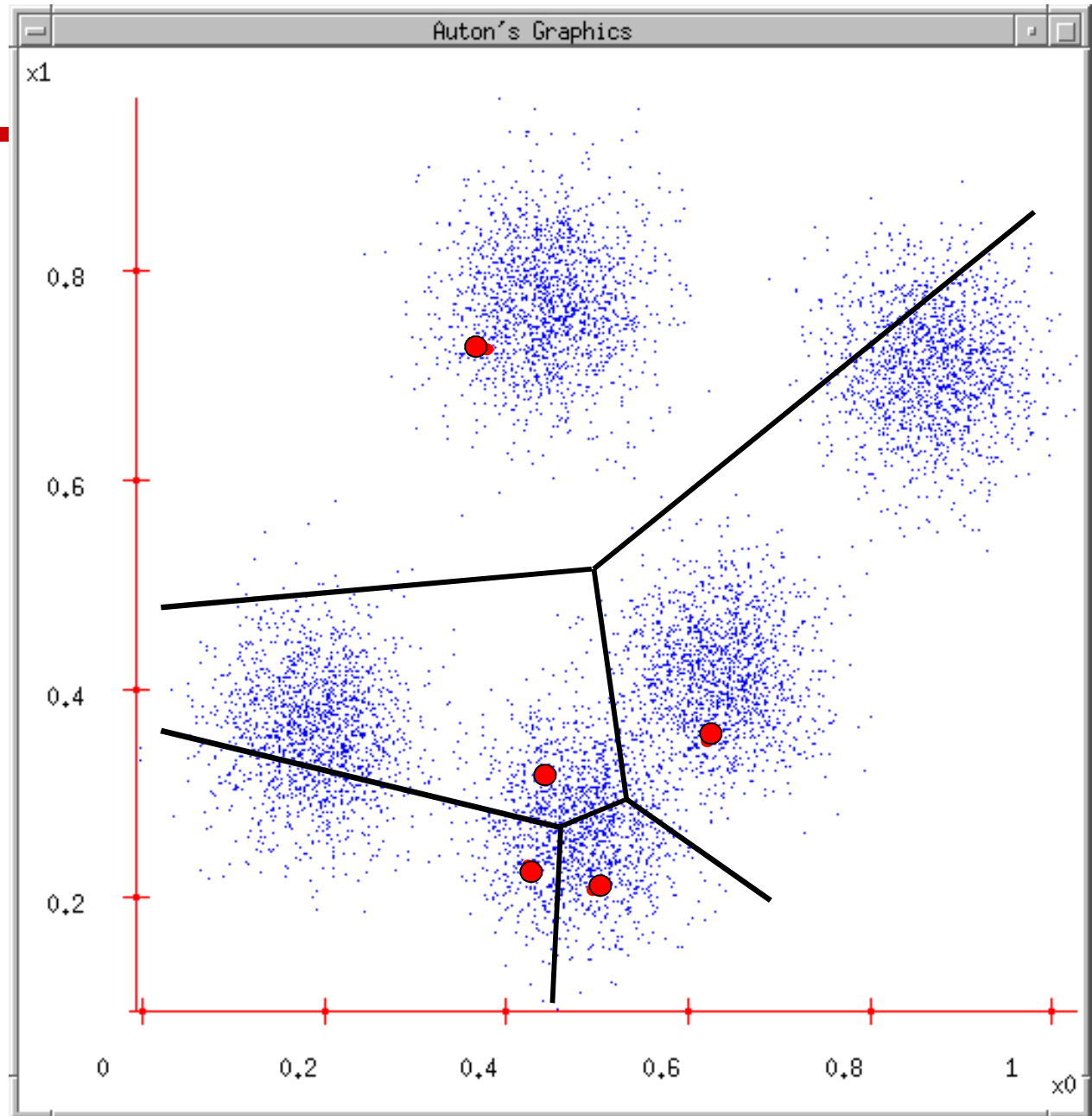
# K-means

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



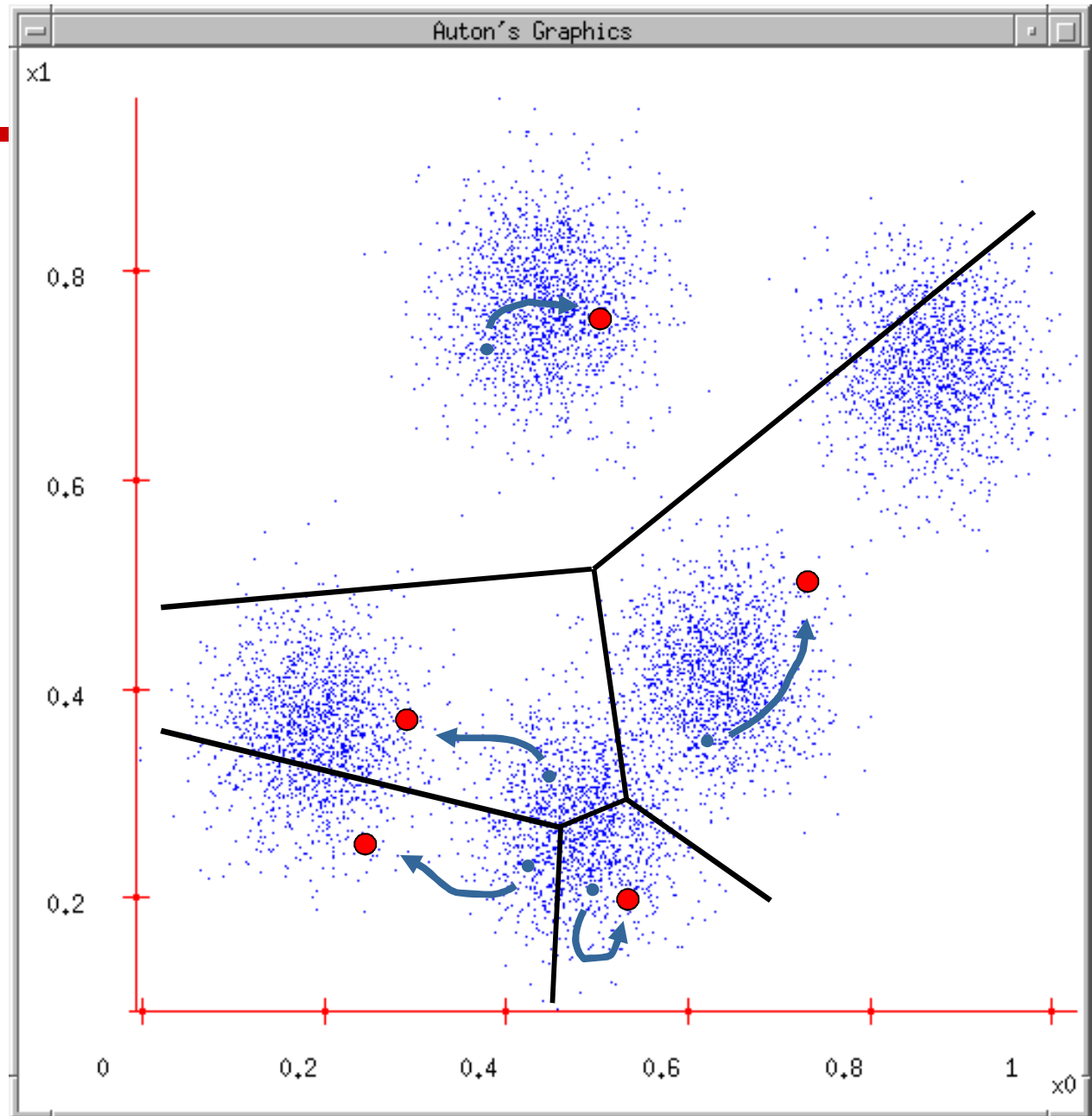
# K-means

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)



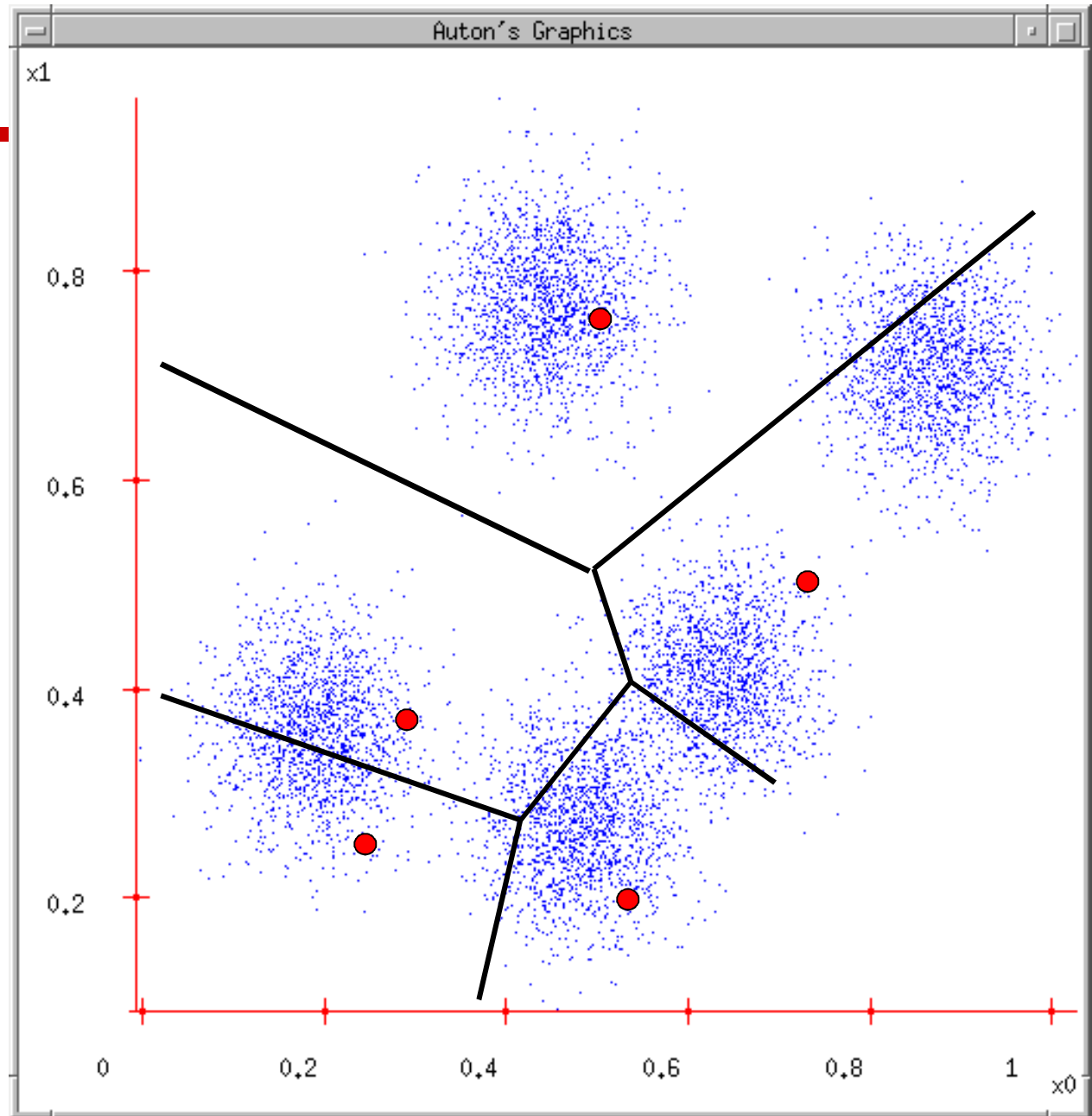
# K-means

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



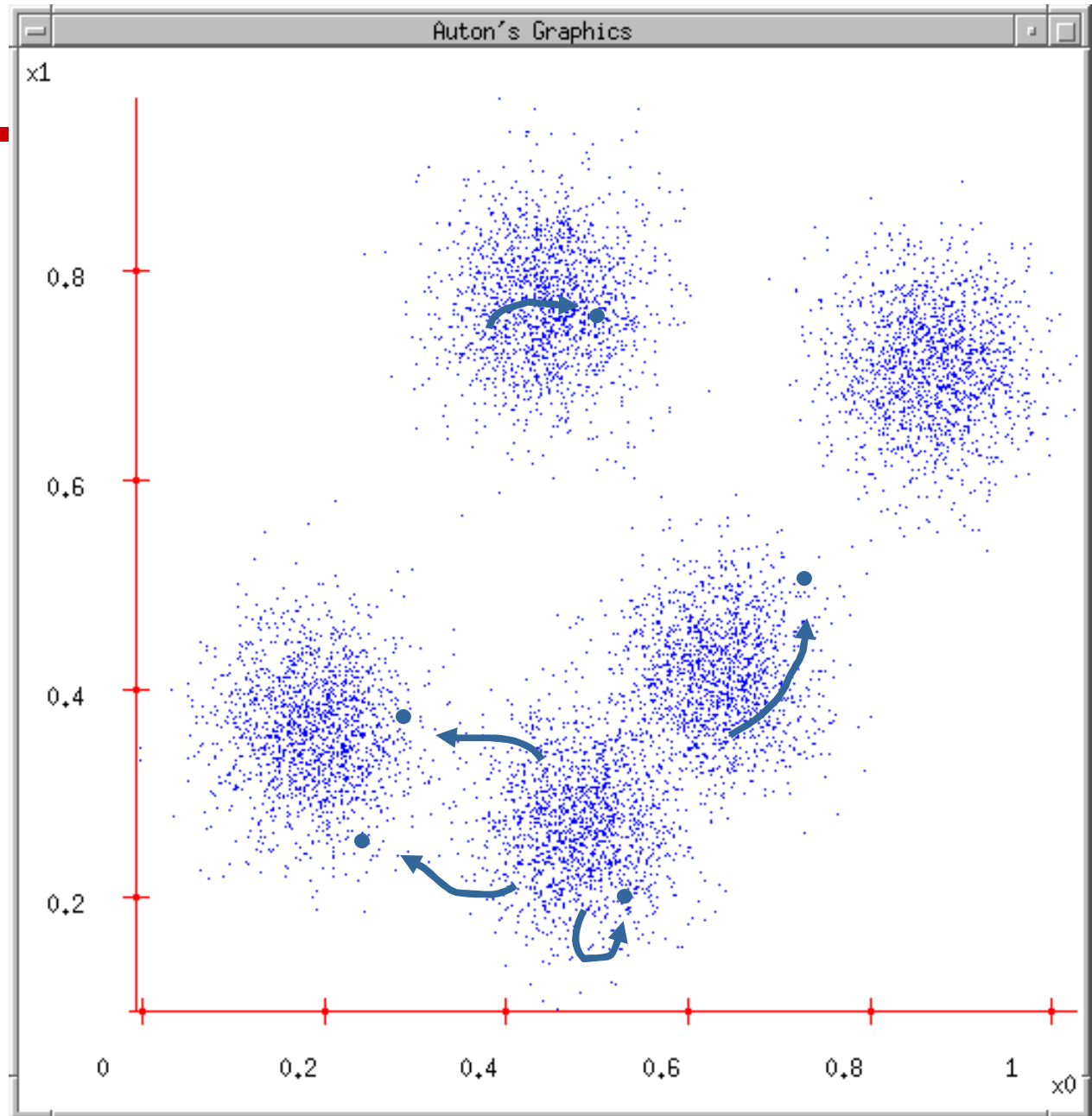
# K-means

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

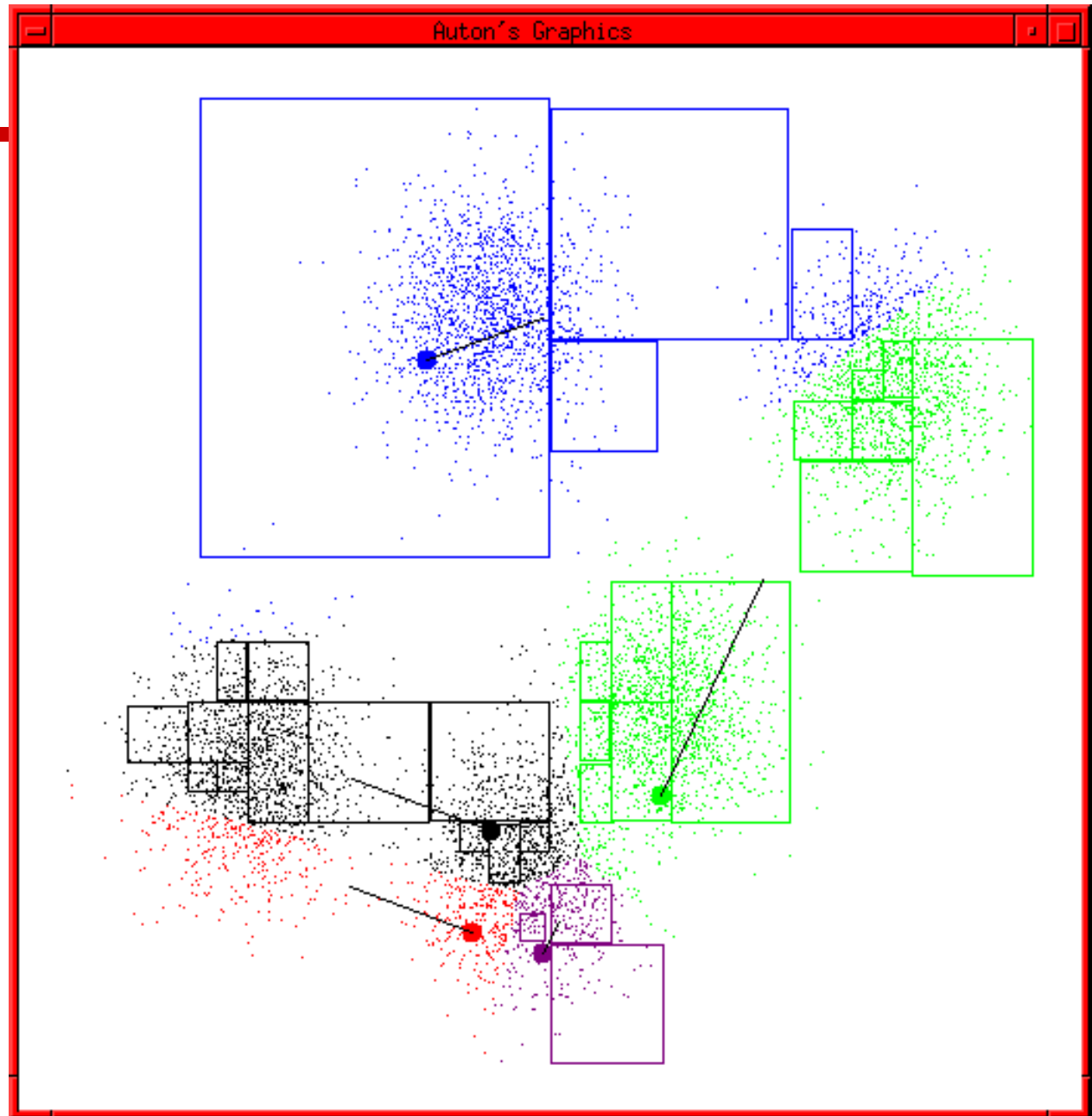
1. Ask user how many clusters they'd like.  
(e.g.  $k=5$ )
2. Randomly guess  $k$  cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns...
5. ...and jumps there
6. ...Repeat until terminated!



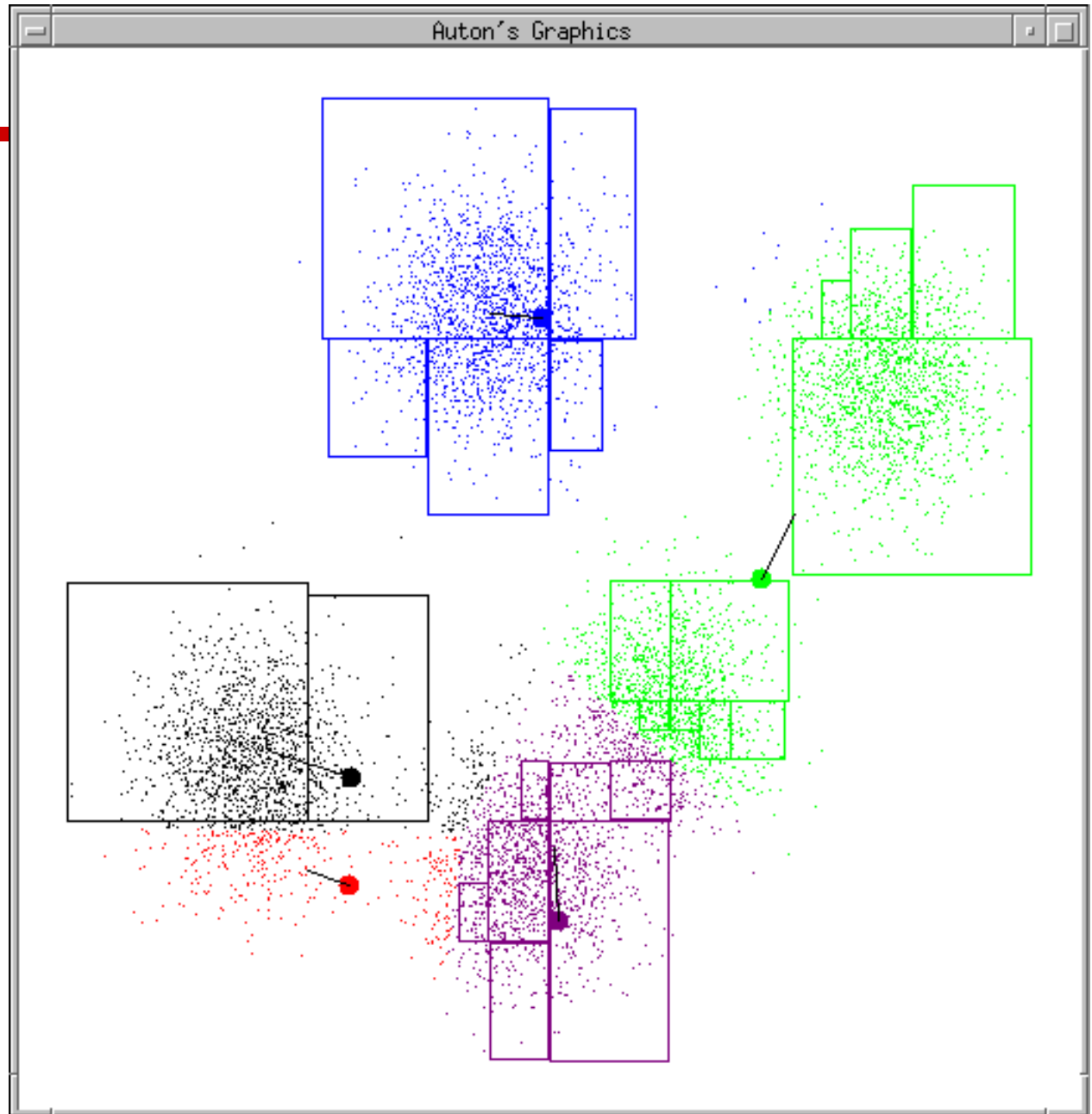
# Accelerated Computations

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

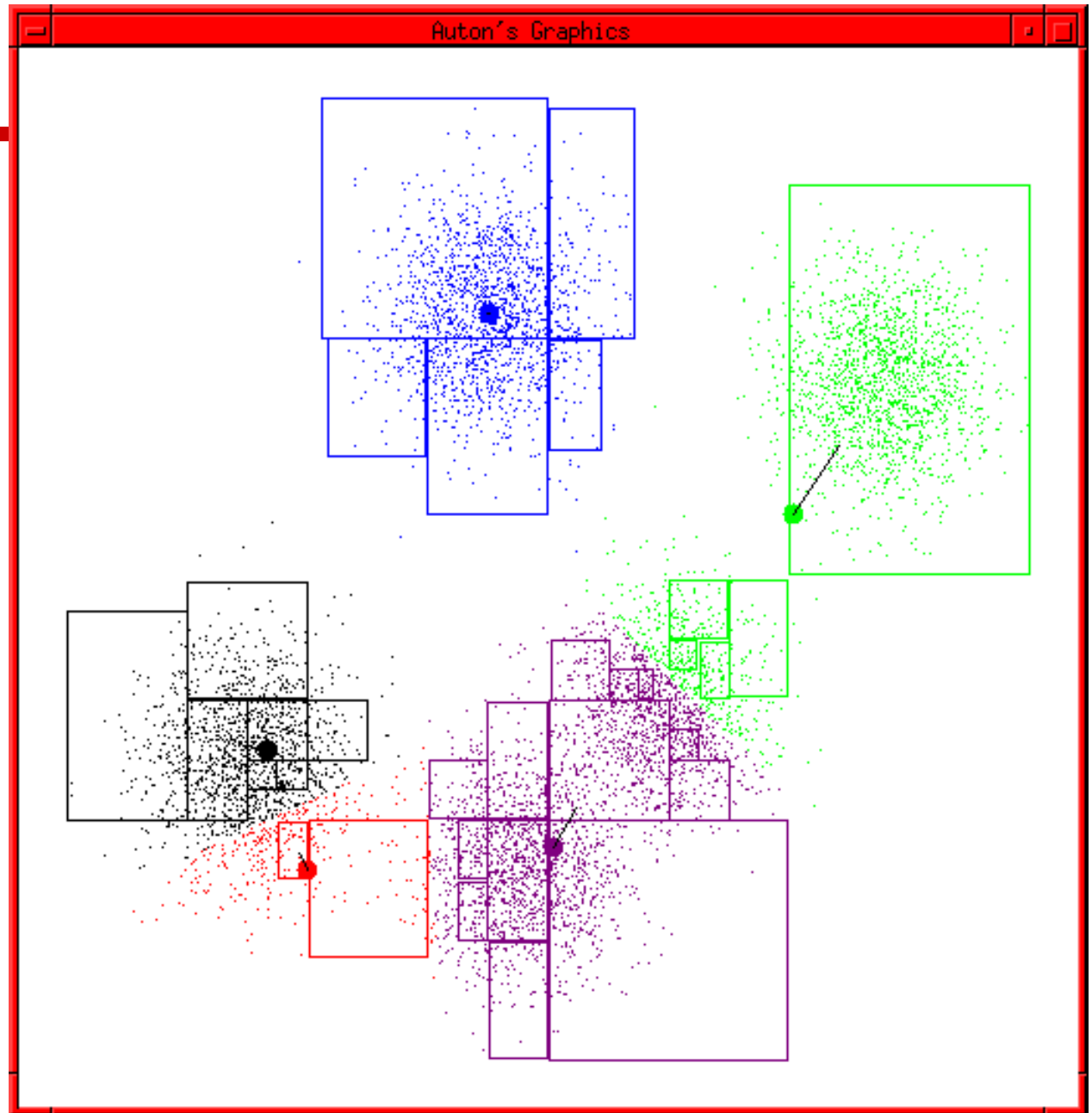
*Dan Pelleg and Andrew Moore. Accelerating Exact k-means Algorithms with Geometric Reasoning. Proc. Conference on Knowledge Discovery in Databases 1999, (KDD99) (available on [www.autonlab.org/pap.html](http://www.autonlab.org/pap.html))*



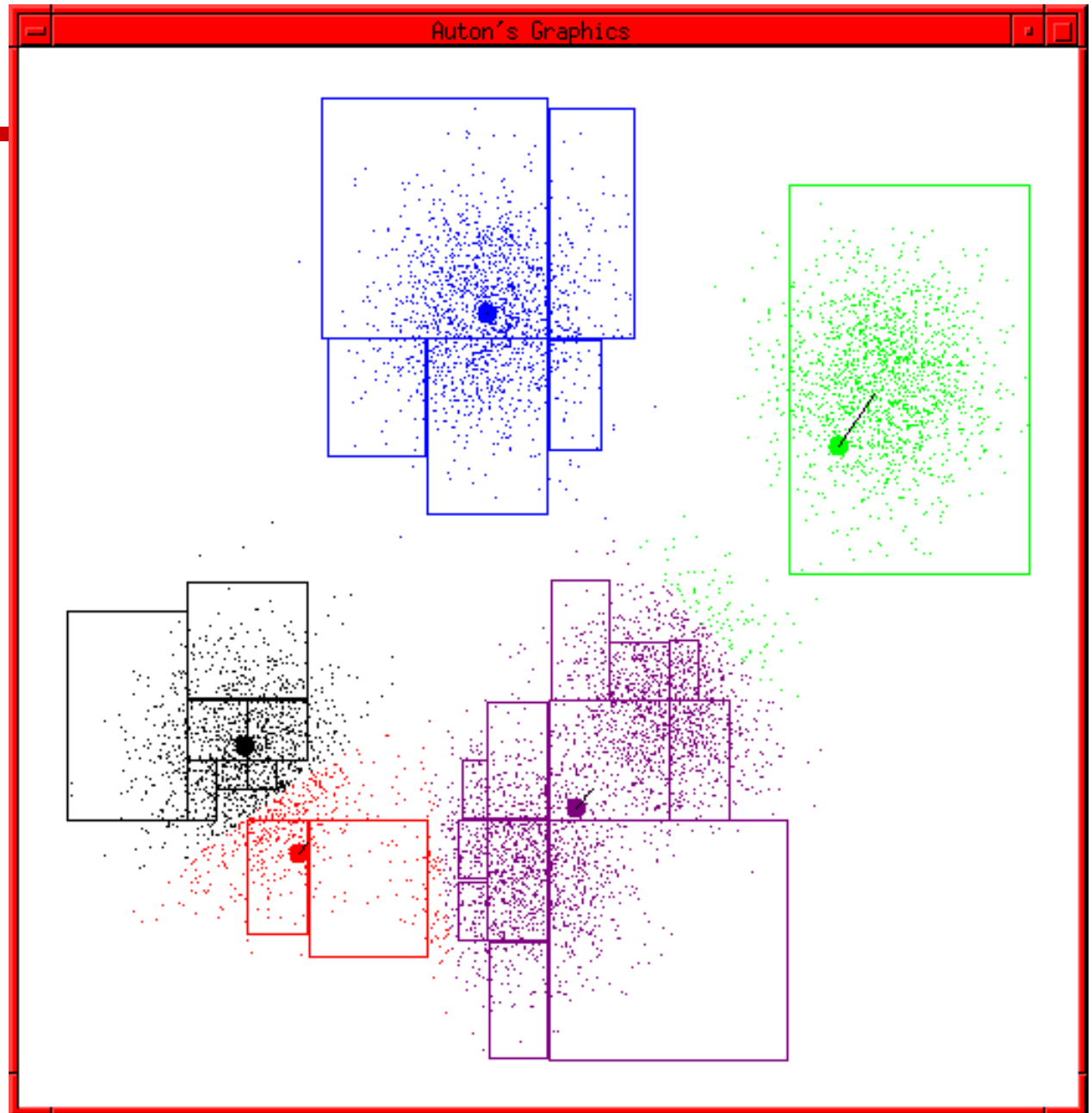
K-means  
continues...



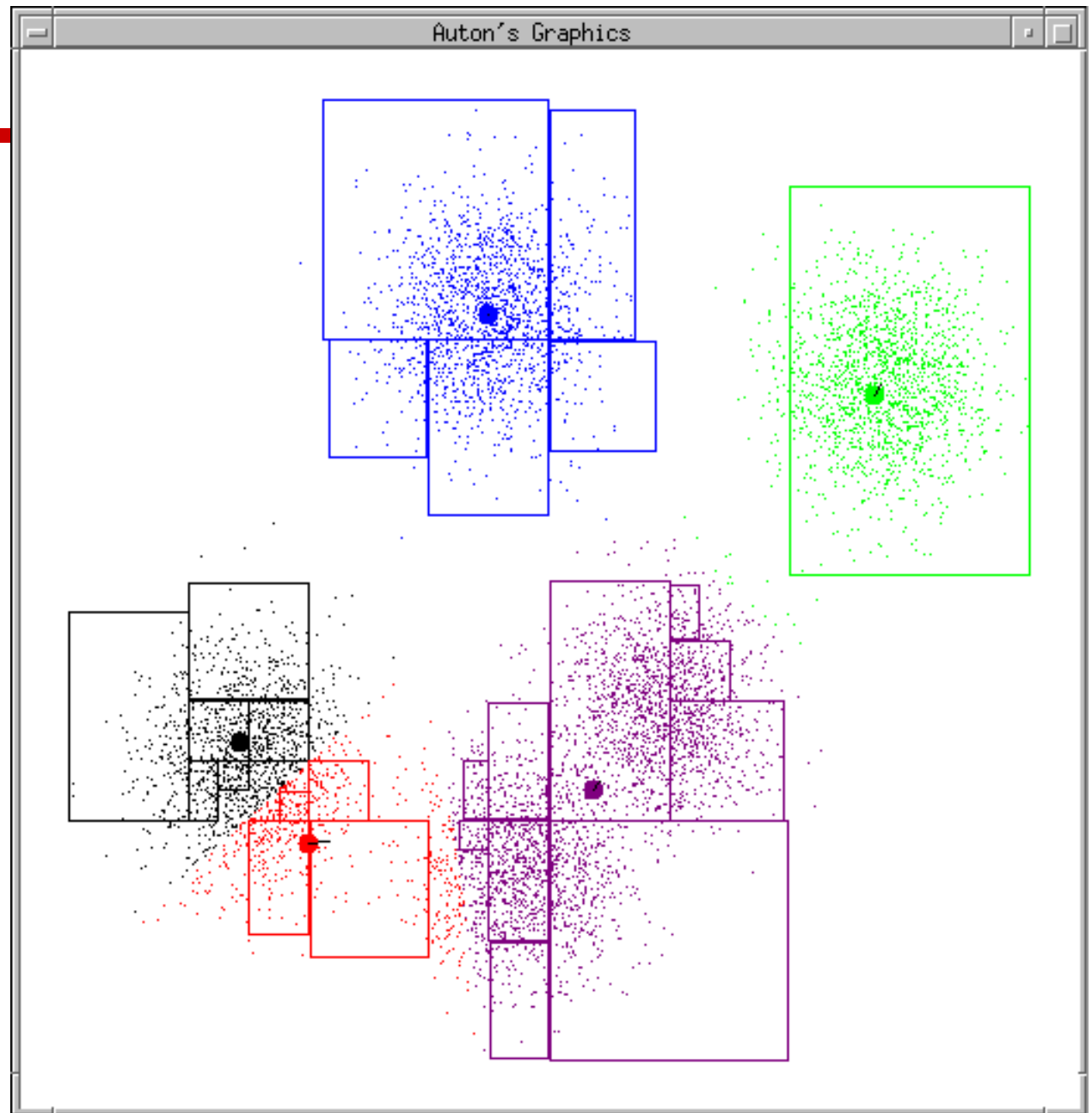
K-means  
continues...



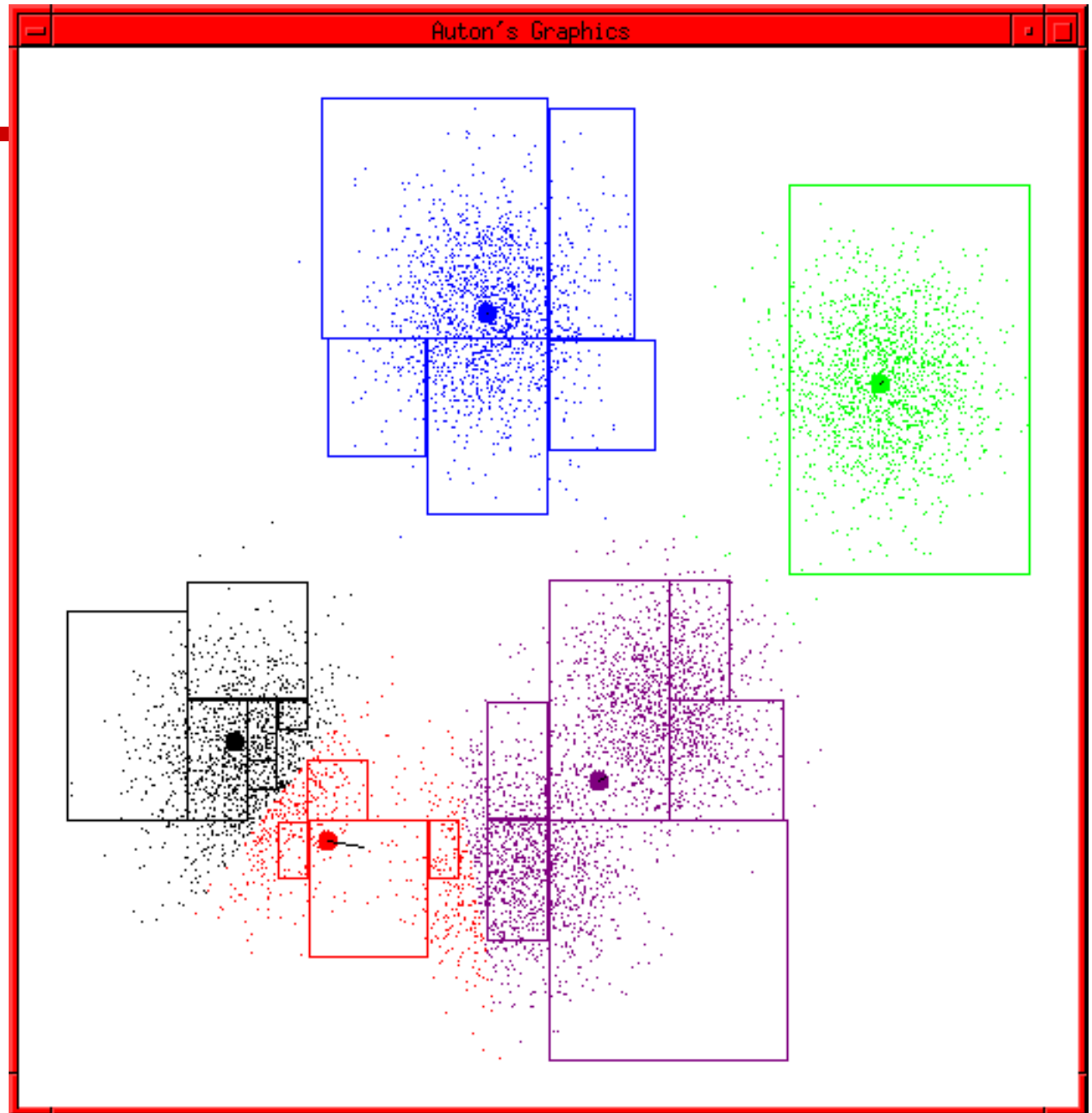
K-means  
continues...



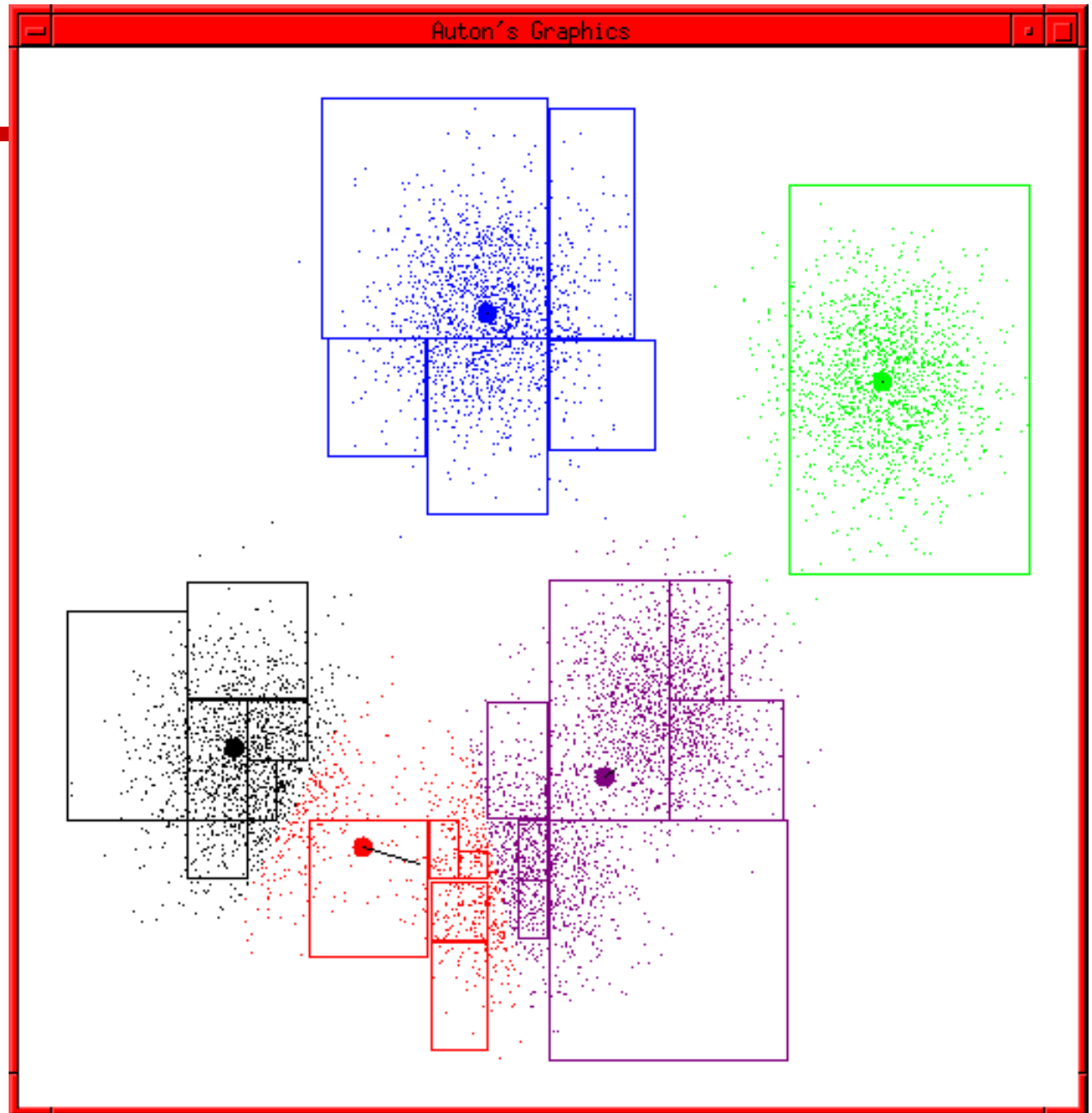
K-means  
continues...



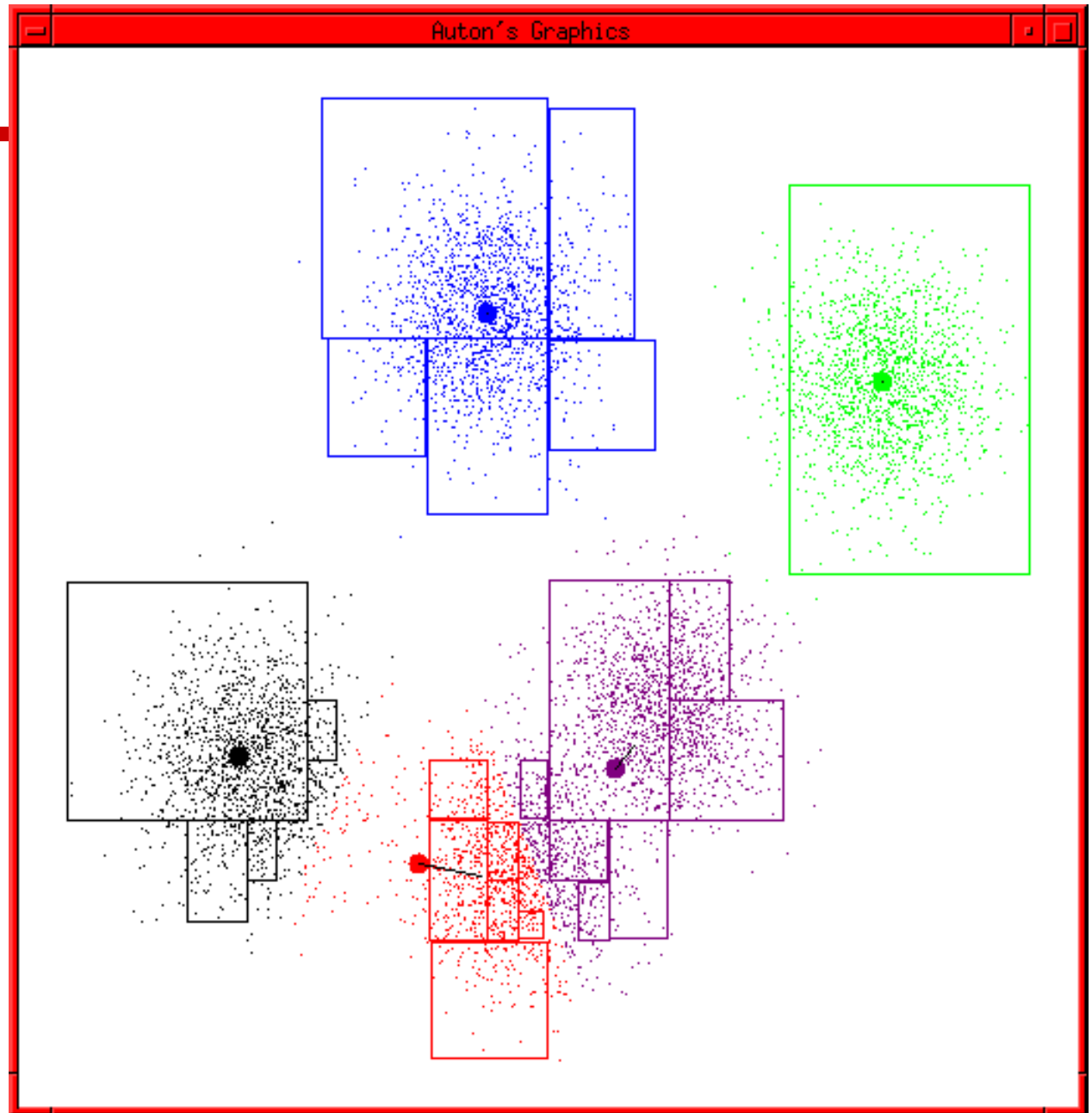
K-means  
continues...



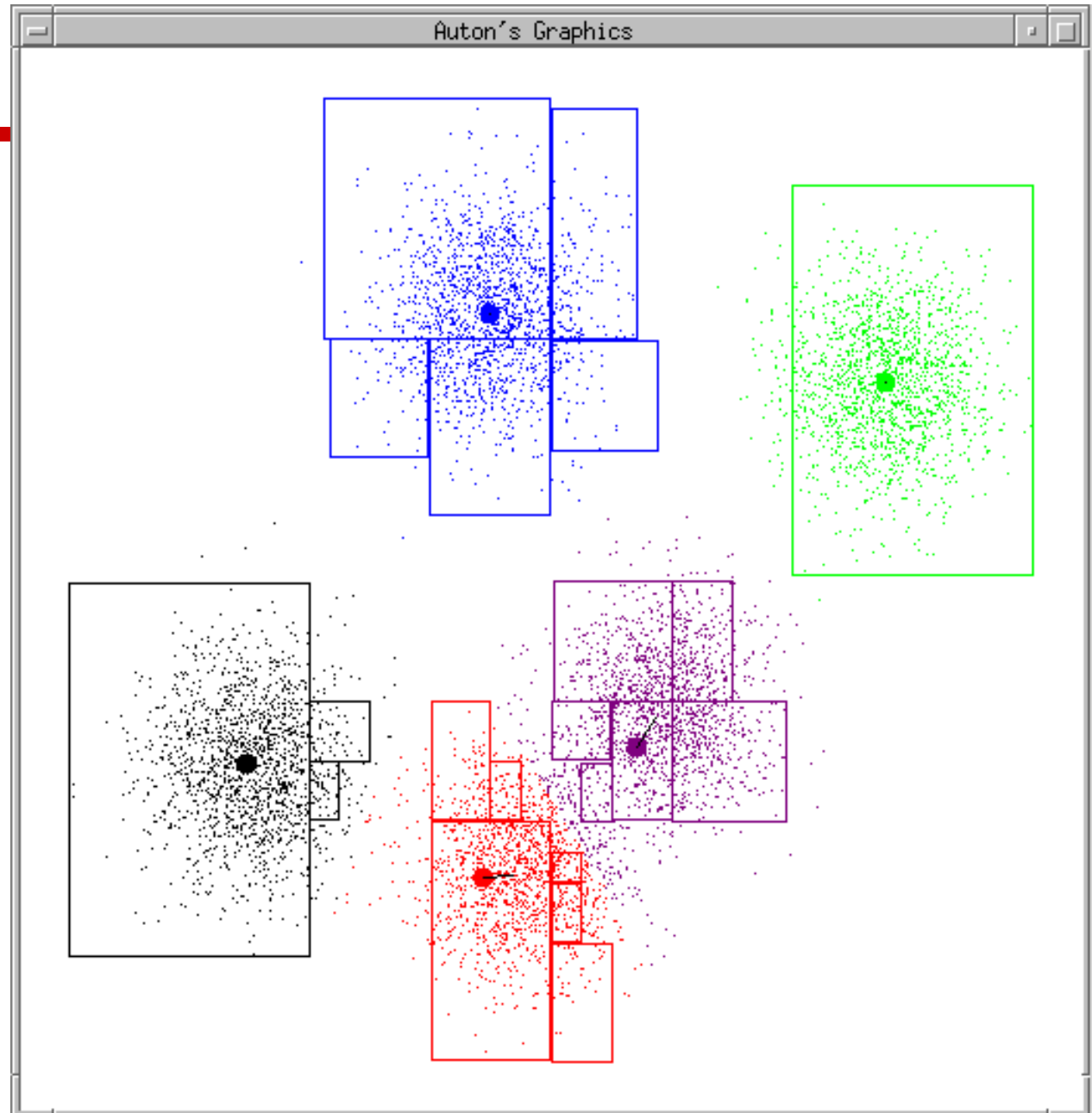
K-means  
continues...



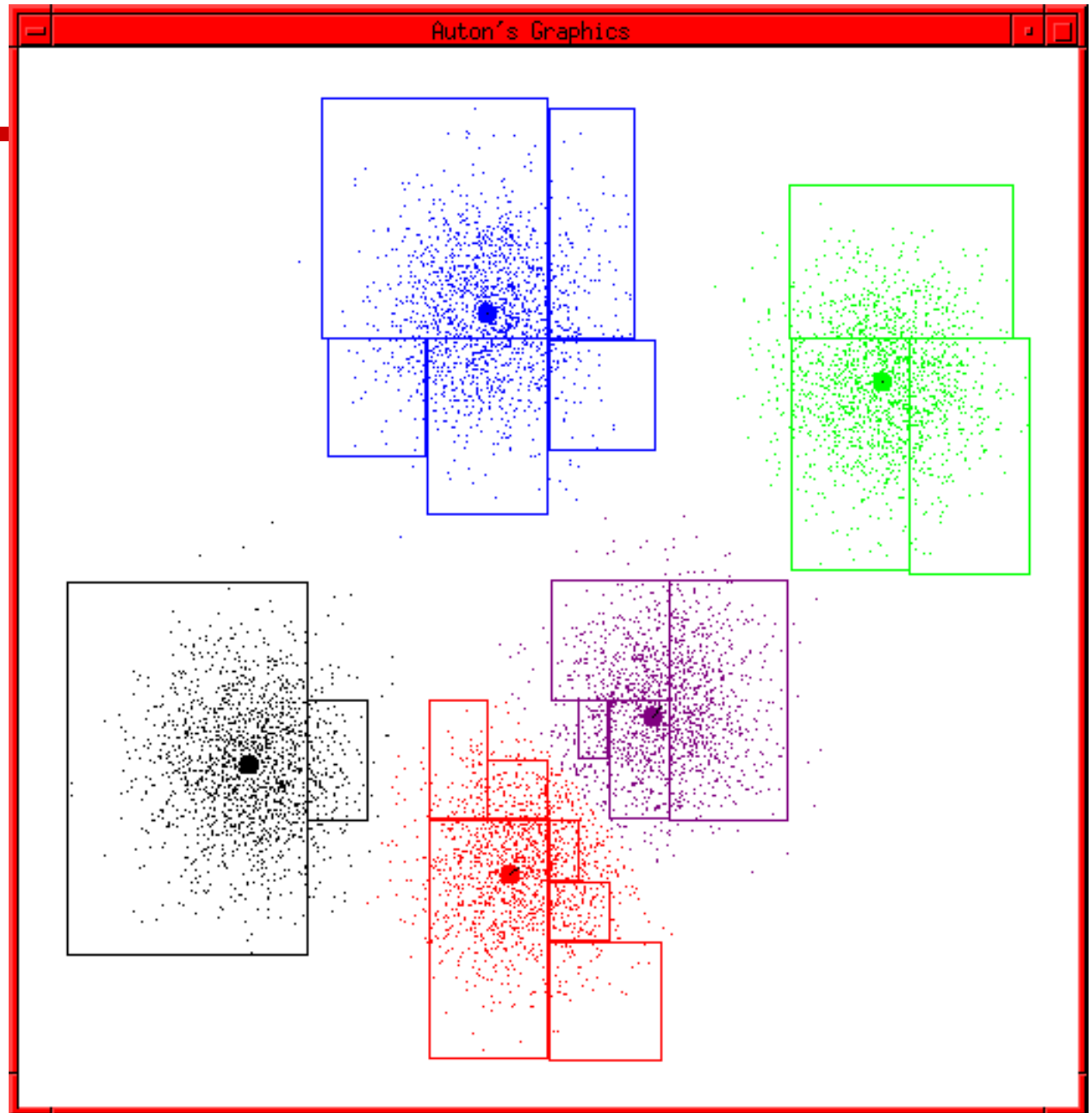
K-means  
continues...



K-means  
continues...

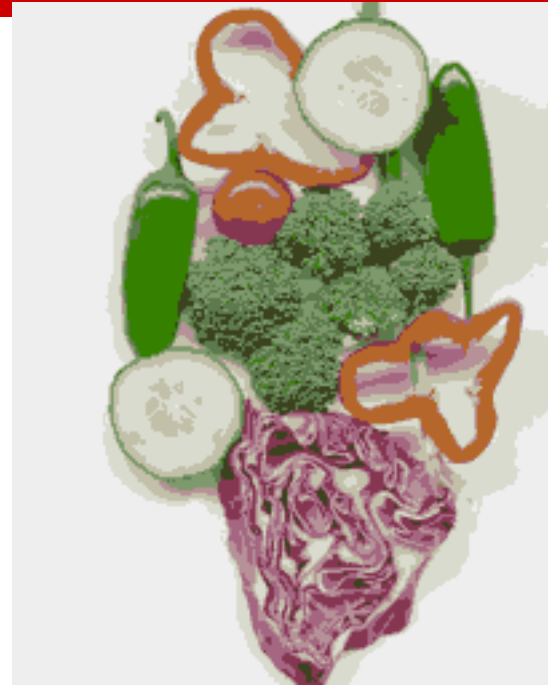


K-means  
terminates





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

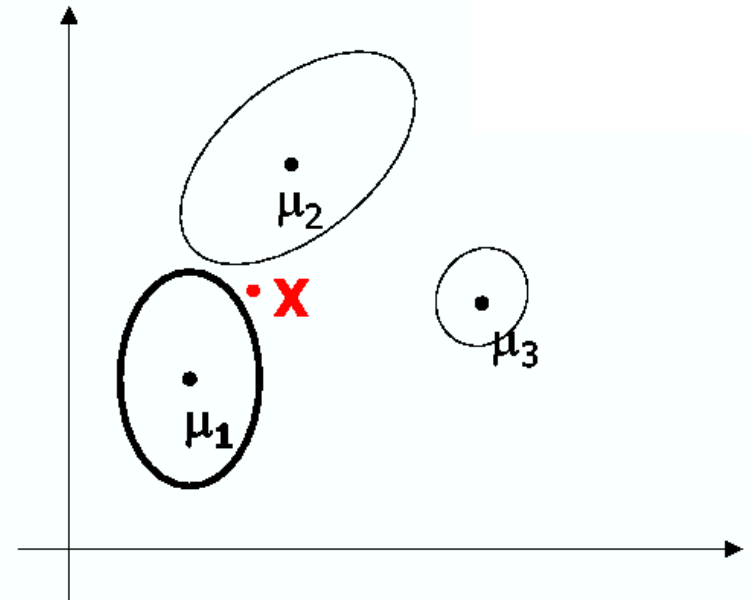
# Probabilistic Clustering: Mixture Models

---

- assume a probabilistic model for each component cluster
  - mixture model:  $f(x) = \sum_{k=1 \dots K} w_k f_k(x; \theta_k)$
  - where  $w_k$  are K mixing weights
    - $\forall w_k : 0 \leq w_k \leq 1$  and  $\sum_{k=1 \dots K} w_k = 1$
  - where K components  $f_k(x; \theta_k)$  can be:
    - Gaussian
    - Poisson
    - exponential
    - ...
- $$P(x) = \frac{\exp\left\{-\frac{1}{2}(x - \mu)' \Sigma^{-1}(x - \mu)\right\}}{\sqrt{(2\pi)^d |\Sigma|}}$$
- Note:
    - Assumes a model for the data (advantages and disadvantages)
    - Results in probabilistic membership:  $p(\text{cluster } k \mid x)$

# Gaussian Mixture Models (GMM)

- model for k-th component is normal  $N(\mu_k, \Sigma_k)$ 
  - often assume diagonal covariance:  $\Sigma_{jj} = \sigma_j^2, \Sigma_{i \neq j} = 0$
  - or sometimes even simpler:  $\Sigma_{jj} = \sigma^2, \Sigma_{i \neq j} = 0$
- $f(x) = \sum_{k=1 \dots K} w_k f_k(x; \theta_k)$  with  $\theta_k = \langle \mu_k, \Sigma_k \rangle$  or  $\langle \mu_k, \sigma_k \rangle$
- generative model:
  - randomly choose a component
    - selected with probability  $w_k$
  - generate  $x \sim N(\mu_k, \sigma_k)$
  - note:  $\mu_k$  &  $\sigma_k$  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”  $p(k | x) = w_k f_k(x; \theta_k) / f(x)$
    - M-step: pick a new  $\theta$  to max expected data log-likelihood
    - Iterate: guaranteed to climb to (local) maximum of  $L(\theta)$

# MIXTURE MODELS - I

---

A mixture model models a probability density function as sum of parameterized functions.

$$p_X(x) = \sum_{k=1}^K a_k h(x|\lambda_k)$$

- $p_X(x)$  is the modeled probability distribution function,
- $K$  is the number of components in the mixture model,
- $a_k$  mixture proportion of component  $k$ .  $0 < a_k < 1, a_1 + \dots + a_K = 1$
- $h(x | \lambda_k)$  is a probability distribution parameterized by  $\lambda_k$ .

## Mixture models – II

---

$$p_X(x) = \sum_{k=1}^K a_k h(x|\lambda_k)$$

**used when we know  $h(x)$  and we can sample from  $p_X(x)$ , but we would like to determine the  $a_k$  and  $\lambda_k$  values.**

i.e. Such situations can arise in studies in which we sample from a population that is composed of several distinct subpopulations.

common to think of mixture modeling as a missing data problem.

- assume data points "membership" in one of the distributions we are using to model the data.
- membership is unknown, or missing.
- estimation to devise model functions parameters to match data membership in individual model distributions.

# Expectation maximization (EM) - 2

---

- The [Expectation-maximization algorithm](#) computes missing memberships of data points in a chosen distribution model.
- start with initial parameters ( $a_k$ 's and  $\lambda_k$ 's). -> Expectation Step, -> Maximization Step.

## The 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.

## Simple example.

collection of data points coming from a sum of two Gaussian distributions.

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

$f$  is the mixing coefficient in  $(0, 1]$ , assume  $\sigma$  is known and constant.

For each data points, compute a membership value for each of the two Gaussians

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

and similarly for  $y_{2,i}$

# Expectation maximization (EM) - 2

---

## □ The maximization step

- With expectation values for group membership,
- recompute estimates of distribution parameters.

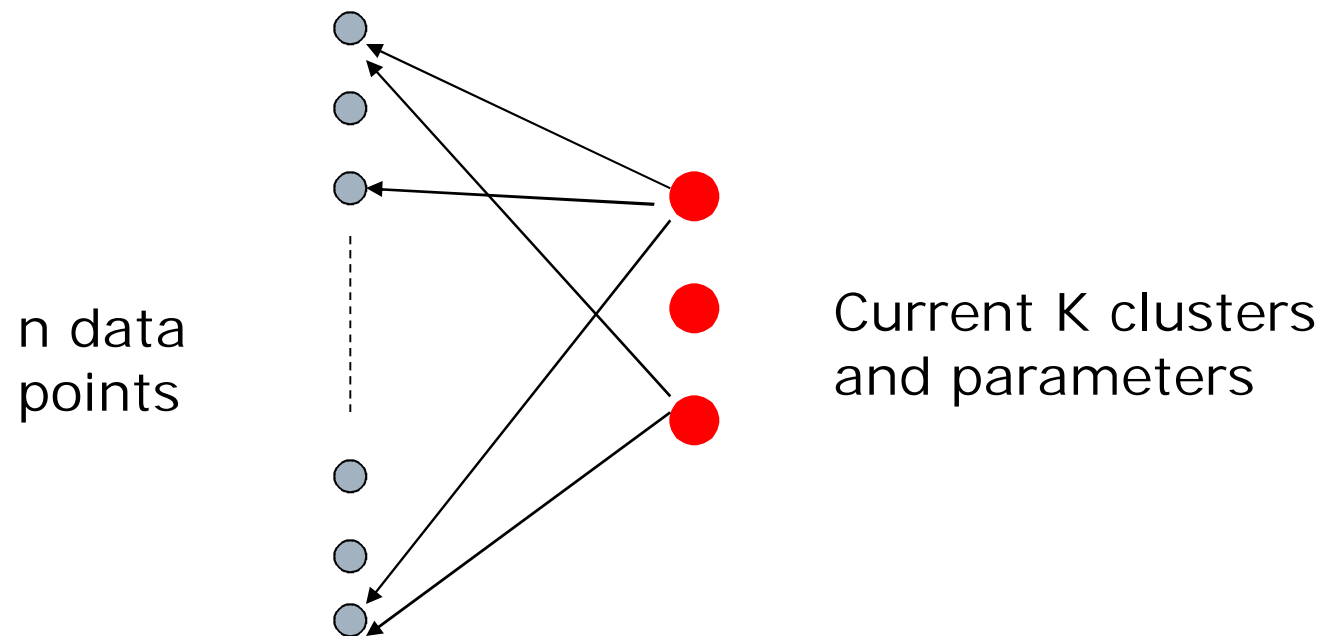
$$\left\{ \begin{array}{l} f = \frac{\sum_i y_{i,2}}{N} \\ \mu_1 = \frac{\sum_i y_{i,1} x_i}{\sum_i y_{i,1}} \end{array} \right.$$

- $N$  is the total number of data points.

- back to the Expectation step,
- recompute new membership values.
- repeated until no further change in the mixture model parameters.

# The E (Expectation) Step

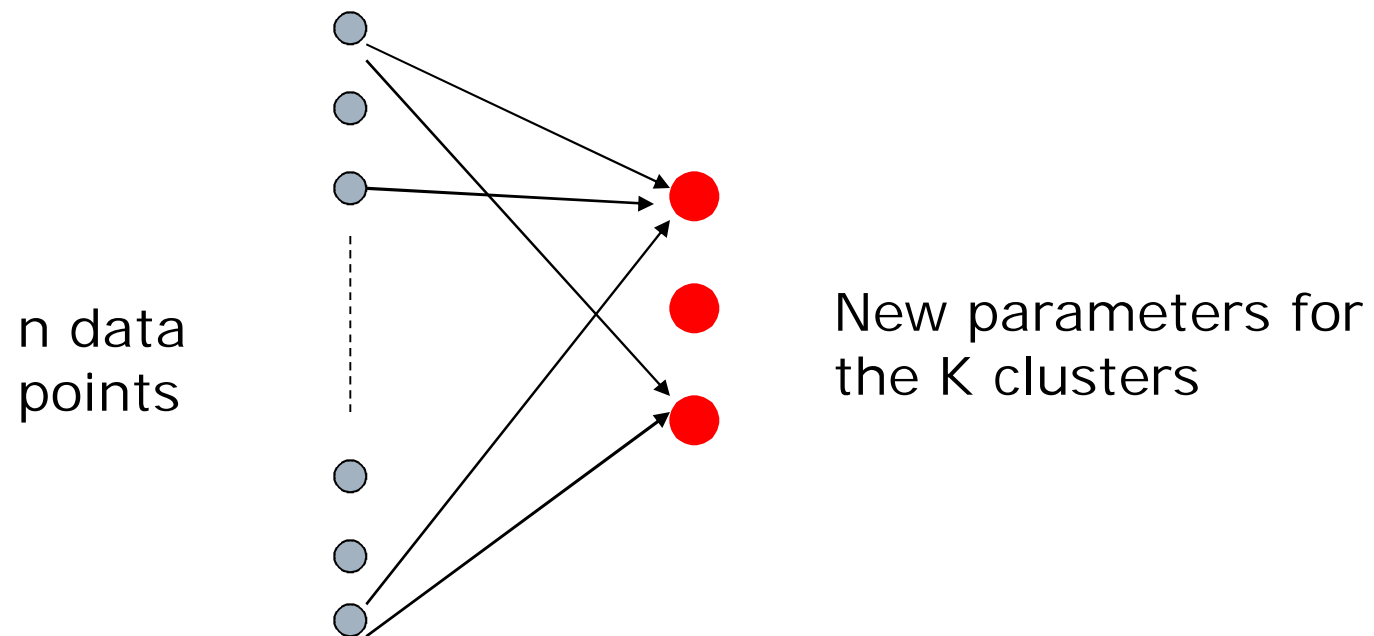
---



E step: Compute  $p(\text{data point } i \text{ is in group } k)$

# The M (Maximization) Step

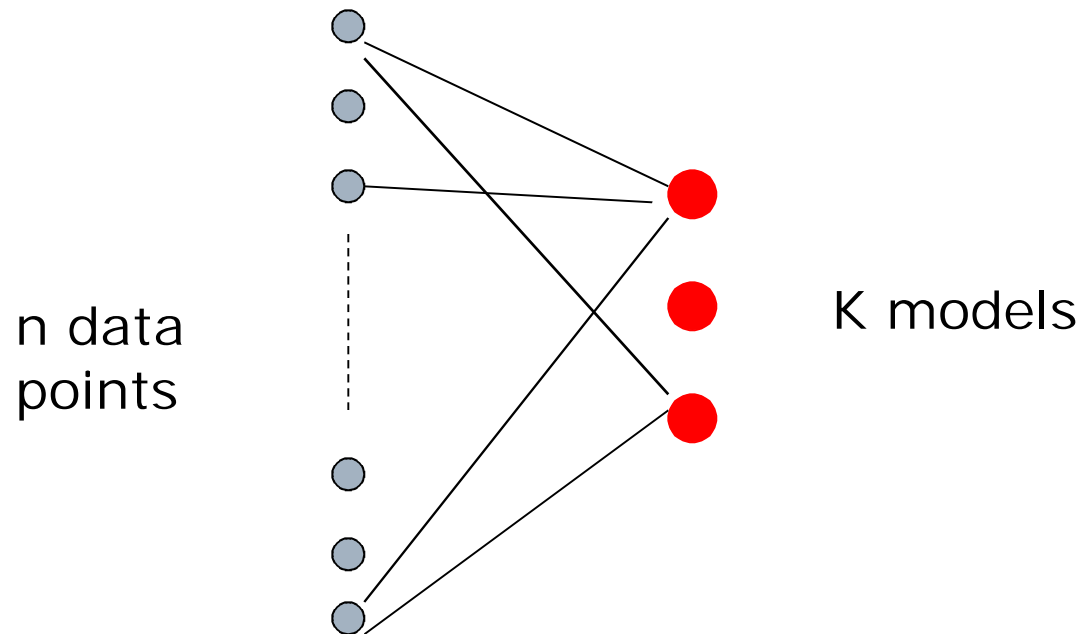
---



M step: Compute  $\theta$ , given  $n$  data points and memberships

# Complexity of EM for mixtures

---



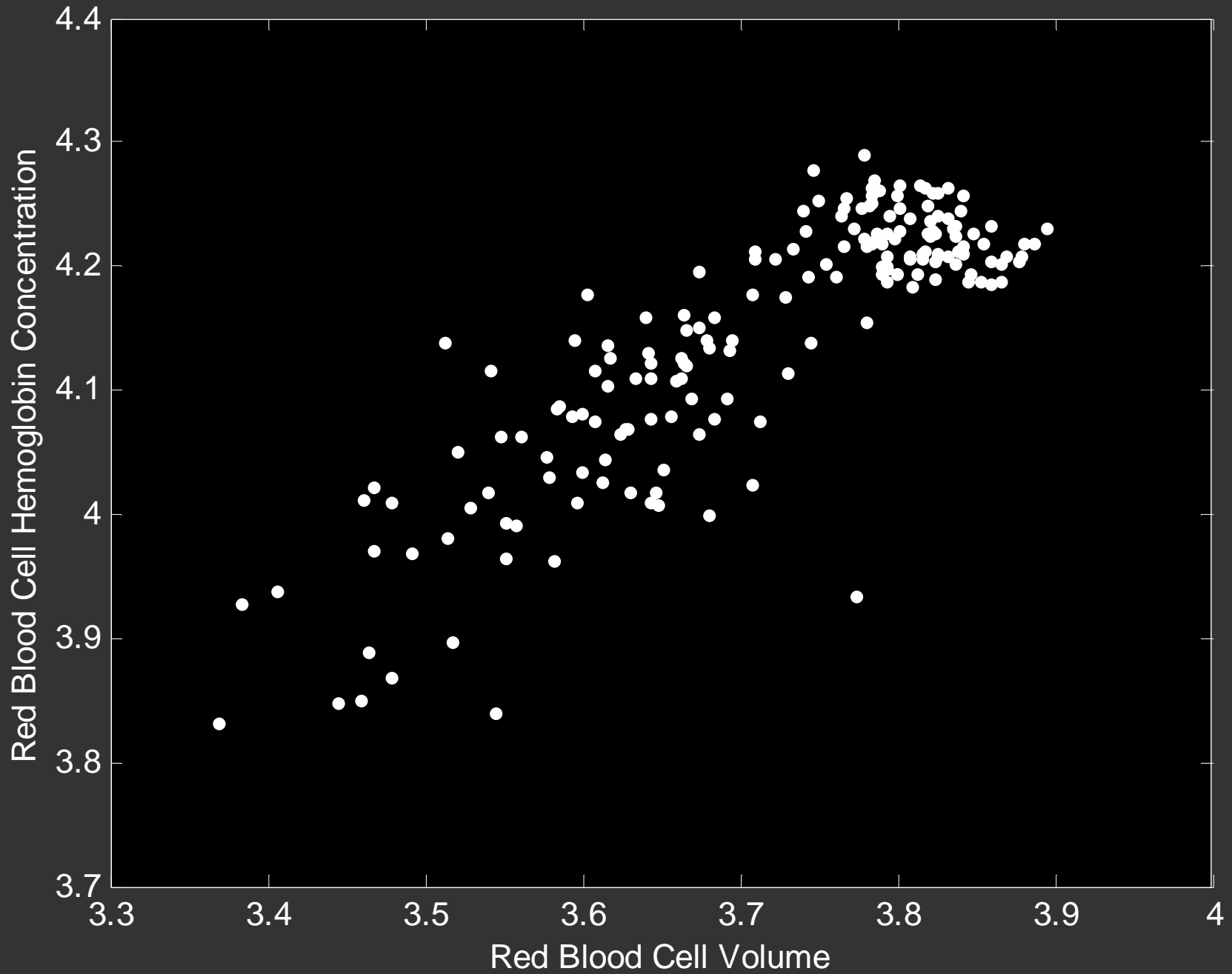
Complexity per iteration scales as  $O( n K f(p) )$

# Comments on Mixtures and EM Learning

---

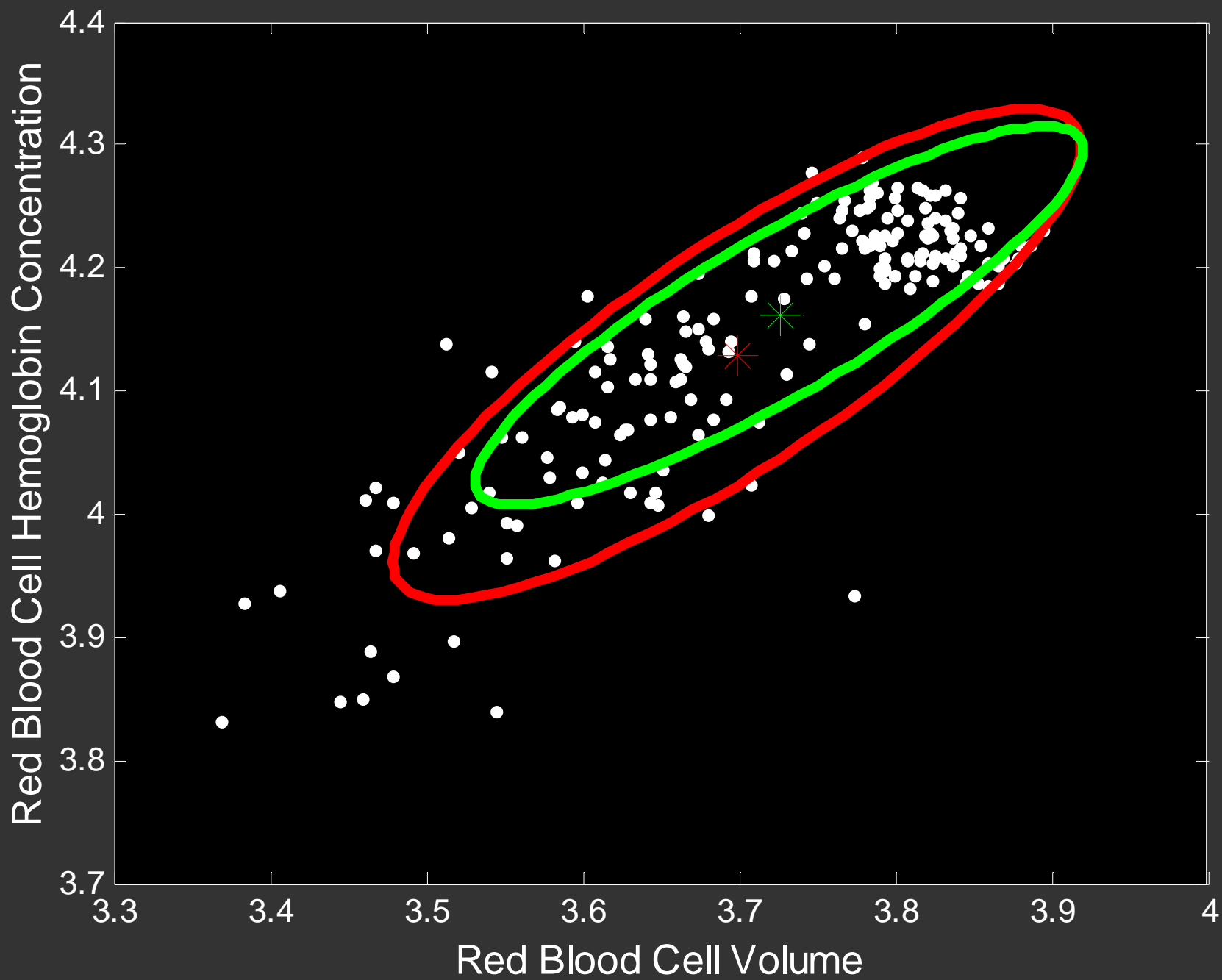
- Complexity of each EM iteration
  - Depends on the probabilistic model being used
    - e.g., for Gaussians, Estep is  $O(nK)$ , Mstep is  $O(Knp^2)$
  - Sometimes E or M-step is not closed form
    - => can requires numerical methods at each iteration
  
- K-means interpretation
  - Gaussian mixtures with isotropic (diagonal, equi-variance)  $\Sigma_k$  's
  - Approximate the E-step by choosing most likely cluster (instead of using membership probabilities)
  
- Generalizations...
  - Mixtures of multinomials for text data
  - Mixtures of Markov chains for Web sequences
  - etc

# ANEMIA PATIENTS AND CONTROLS

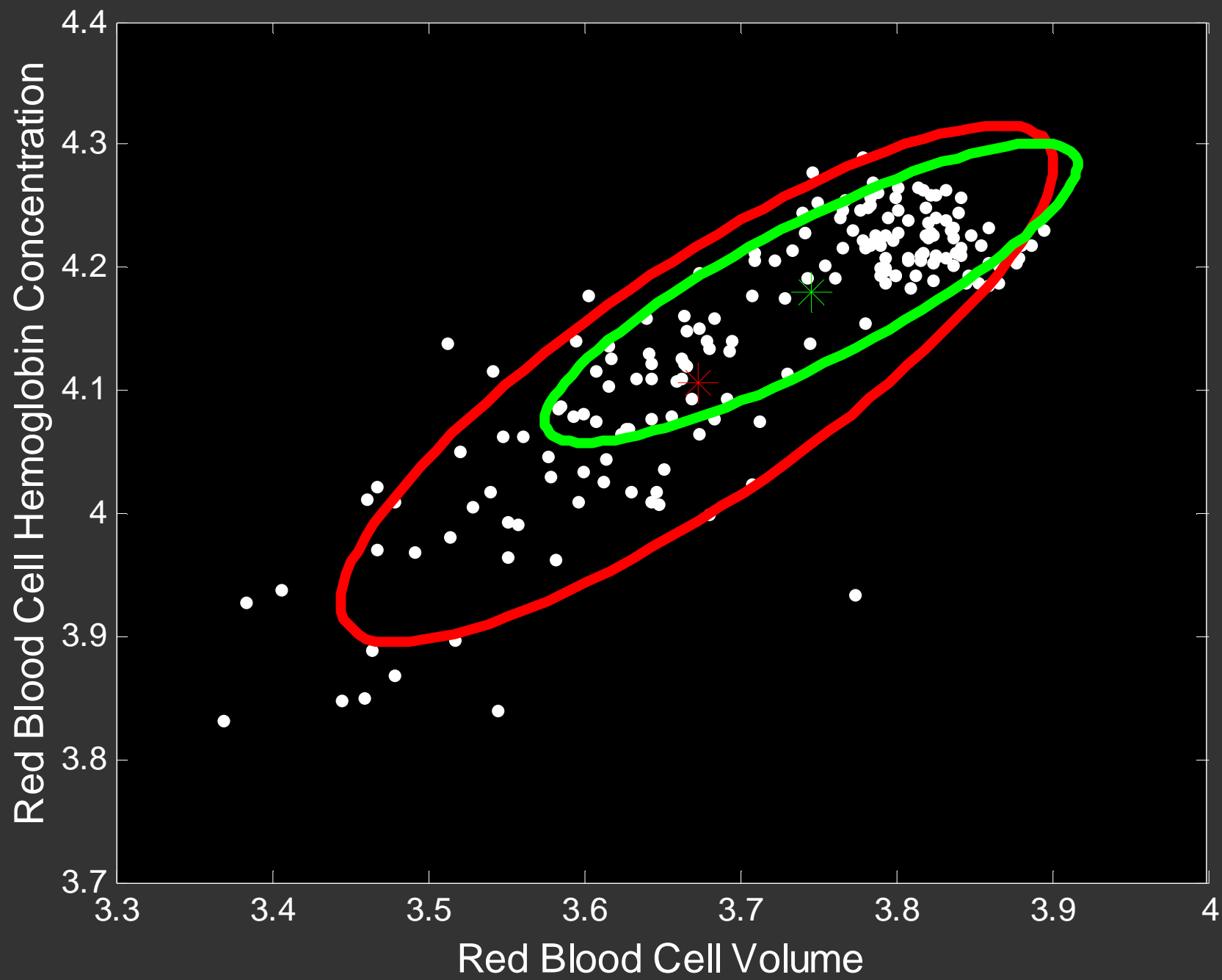




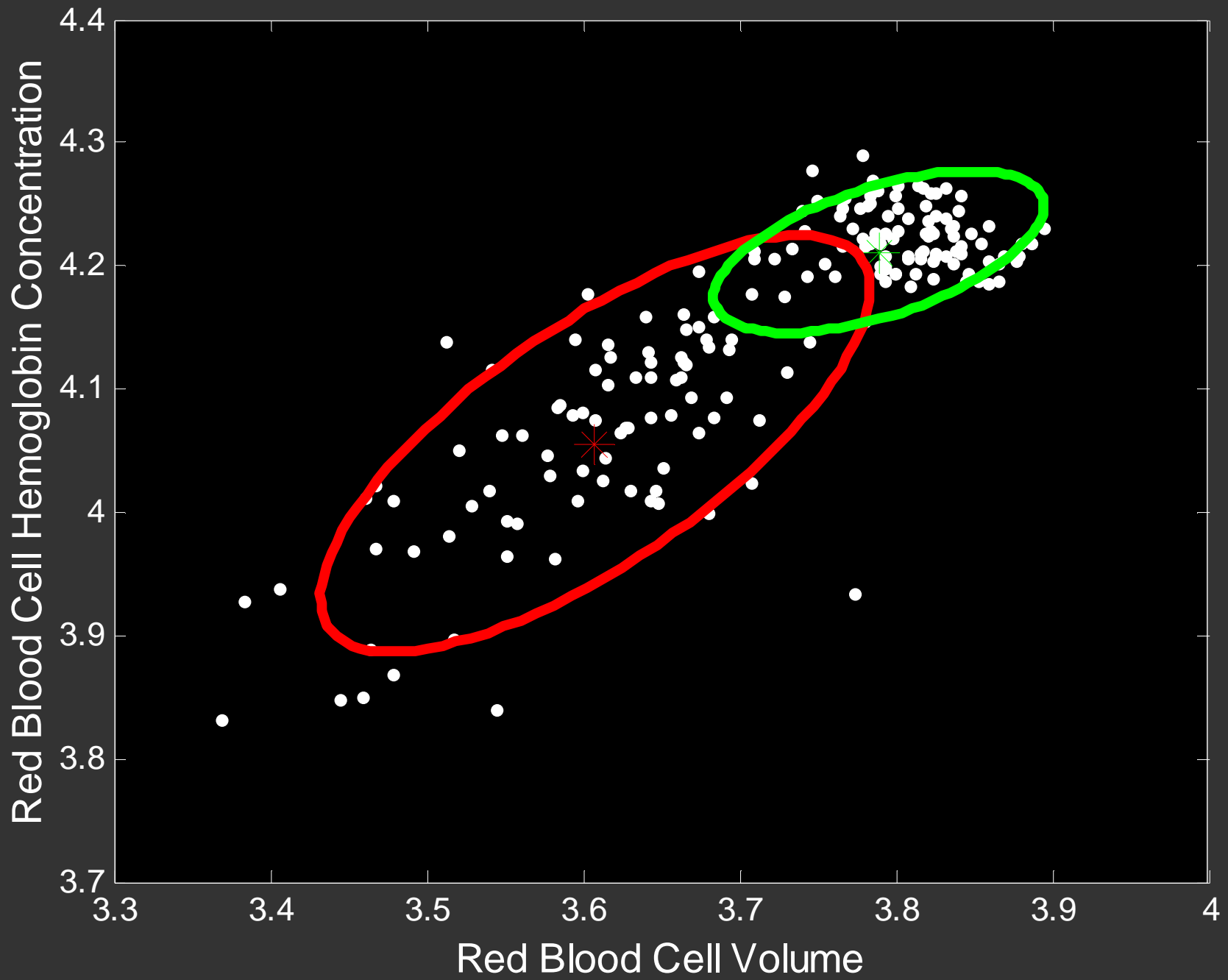
EM ITERATION 3



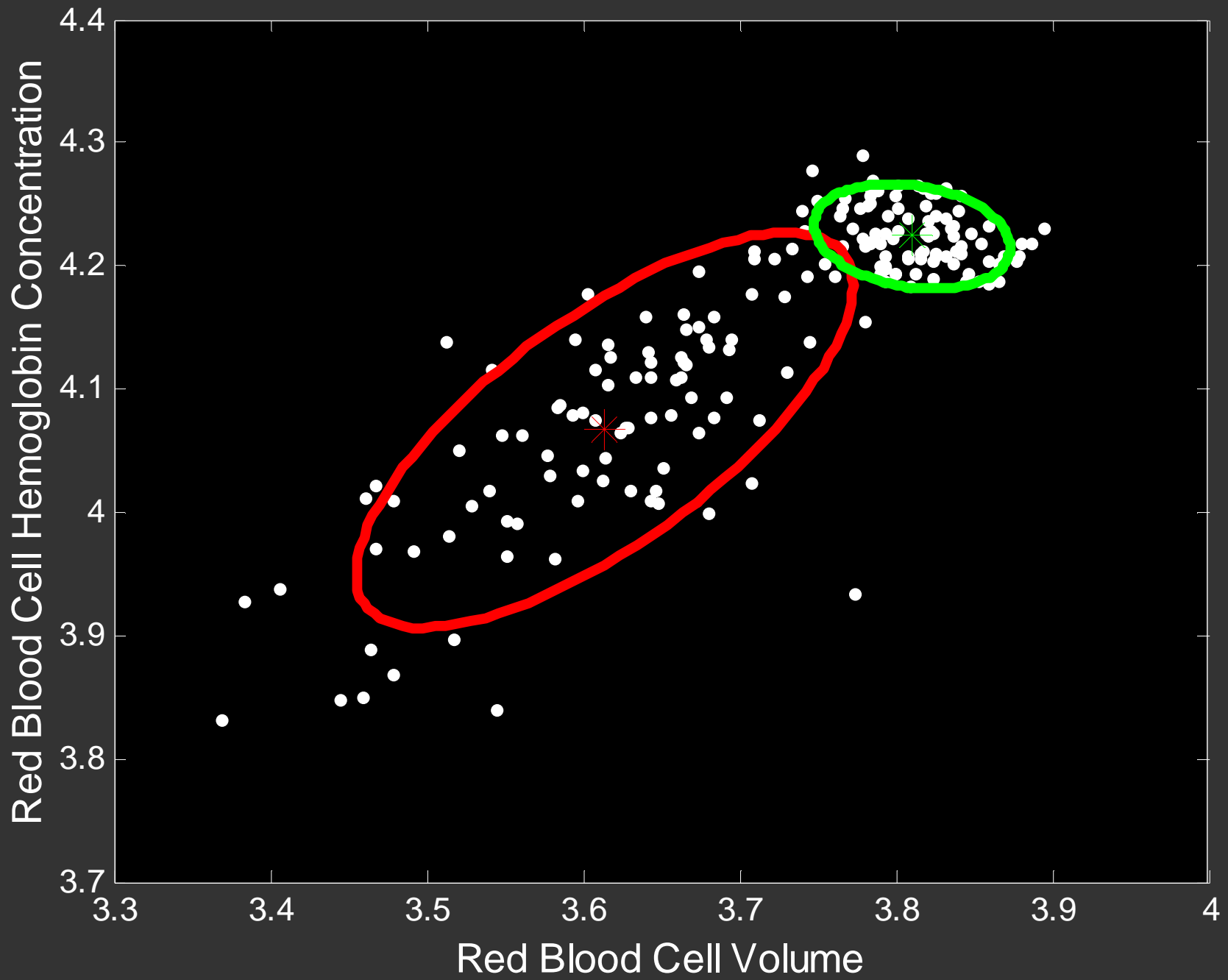
EM ITERATION 5



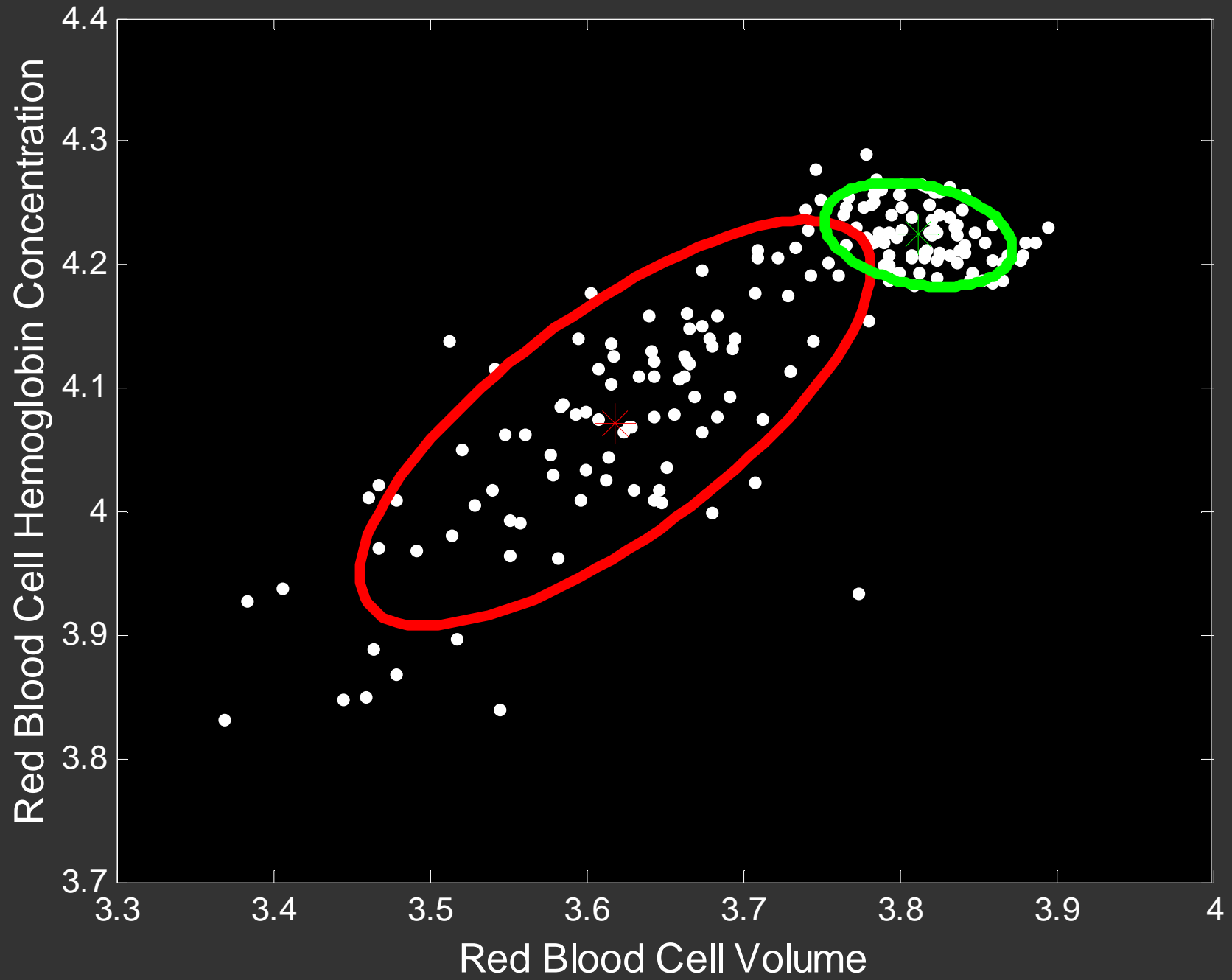
EM ITERATION 10



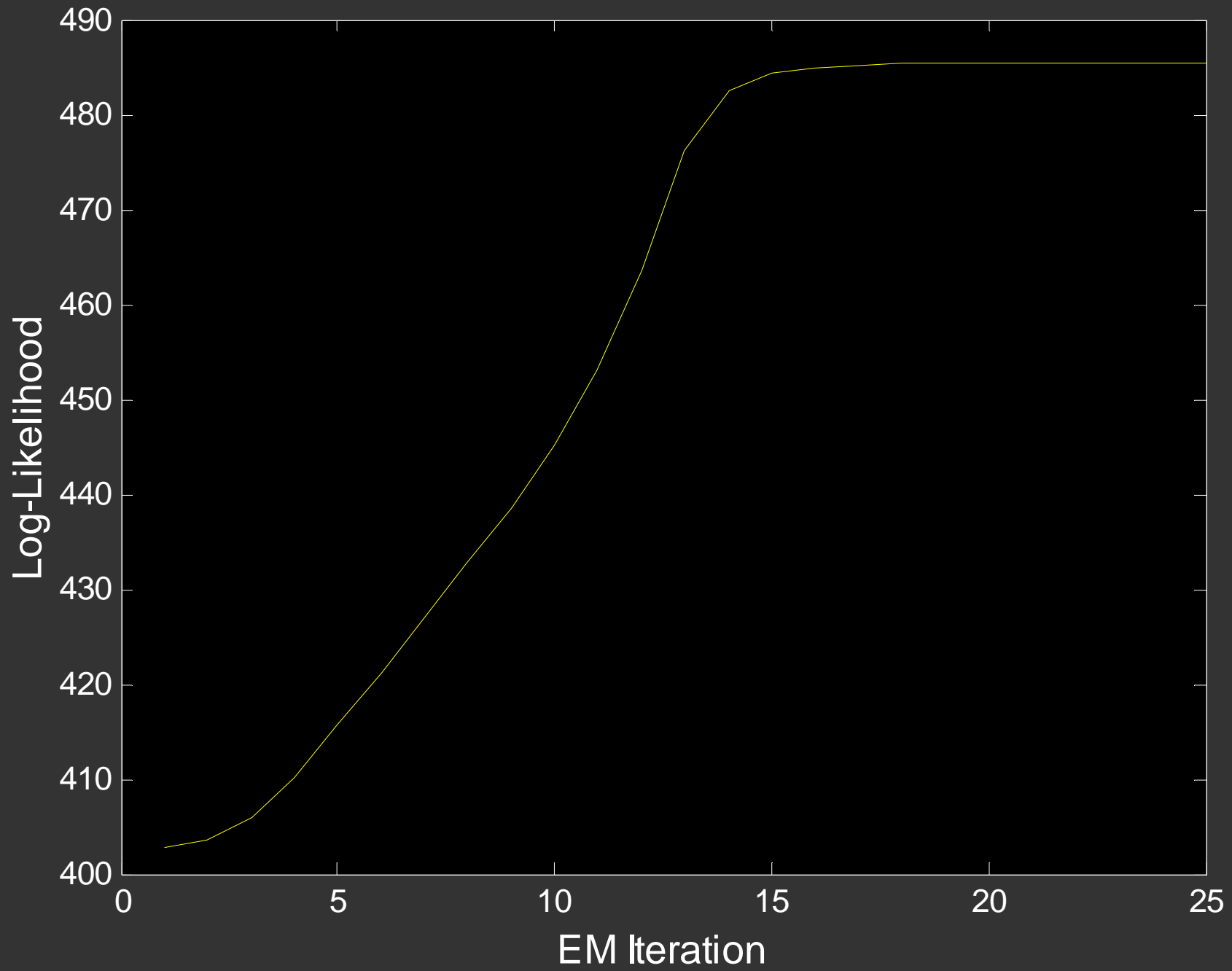
EM ITERATION 15



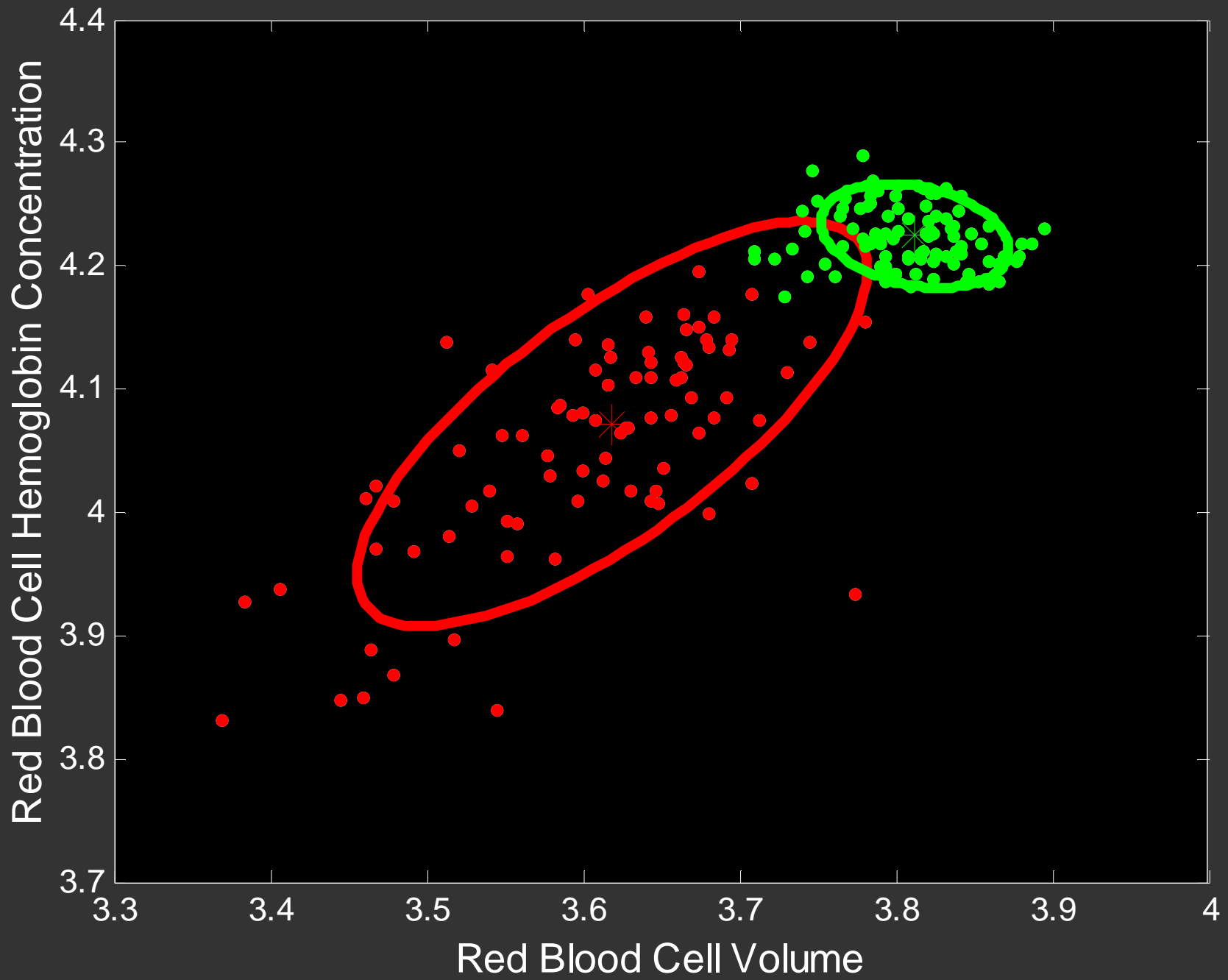
EM ITERATION 25



LOG-LIKELIHOOD AS A FUNCTION OF EM ITERATIONS



ANEMIA DATA WITH LABELS



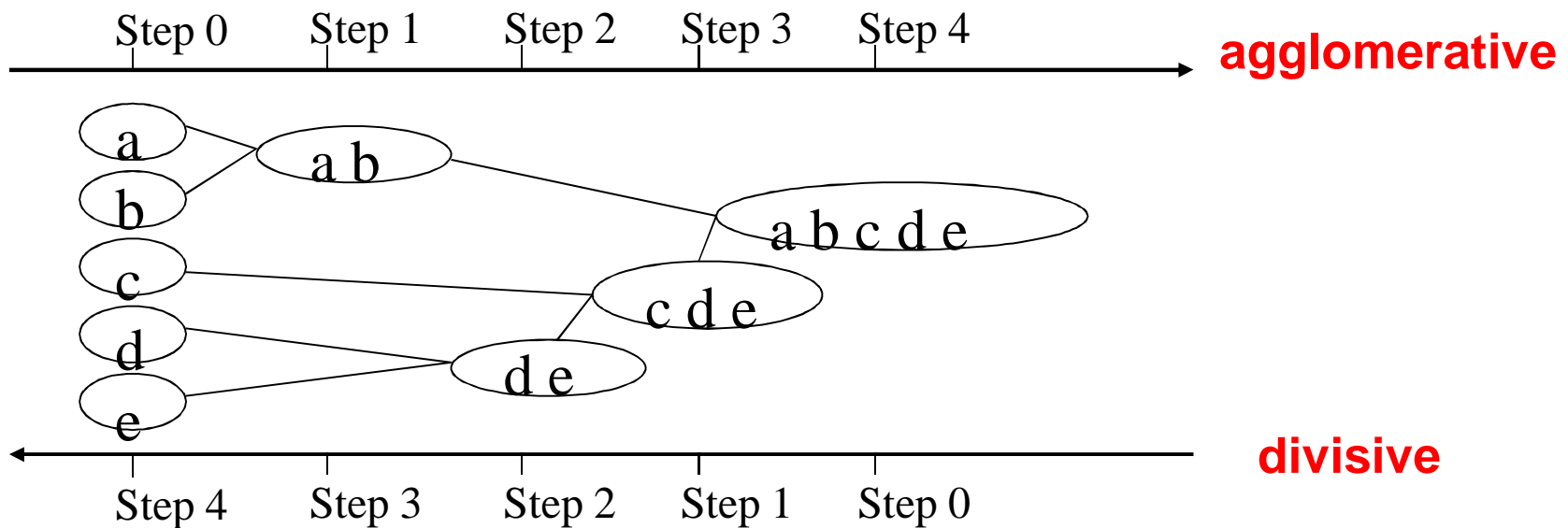
# Hierarchical Clustering

## □ Two basic approaches:

- merging smaller clusters into larger ones (agglomerative),
- splitting larger clusters (divisive)

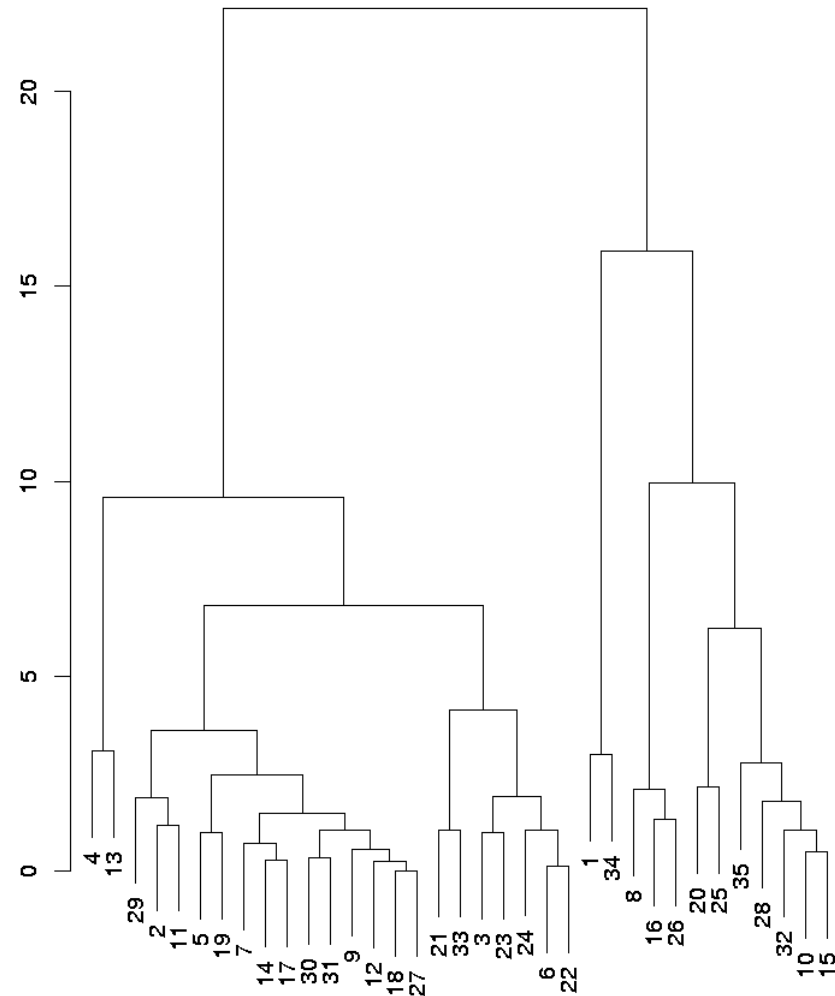
## □ visualize both via "dendograms"

- shows nesting structure
- merges or splits = tree nodes



# Hierarchical Clustering: Complexity

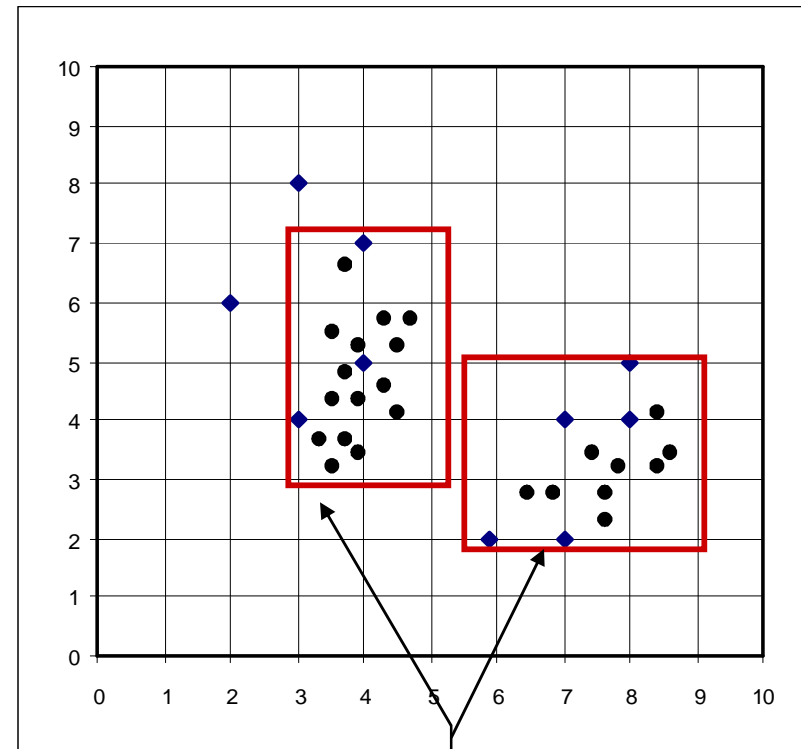
- **Quadratic algorithms**
- **Running time** can be improved using
  - sampling
    - [Guha et al, SIGMOD 1998]
    - [Kollios et al, ICDE 2001]
  - or using the triangle inequality (when it holds)



\*based on slides by Padhraic Smyth UC, Irvine

# Density-based Algorithms

- **Clusters** are **regions of space which have a high density** of points
- Clusters can have **arbitrary shapes**



**Regions of  
high density**

# Density-based Clustering Algorithms

---

- **Clustering based on density** (local cluster criterion), such as density-connected points
- **Major features:**
  - ✓ Discover clusters of arbitrary shape
  - ✓ Handle noise
  - ✓ Need density parameters as termination condition
  - ✓ Work for low dimensional spaces
- **Representative algorithms:**
  - ✓ **DBSCAN:** Ester, et al. (KDD'96)
  - ✓ **DENCLUE:** Hinneburg & D. Keim (KDD'98)

# Clustering High Dimensional Data

---

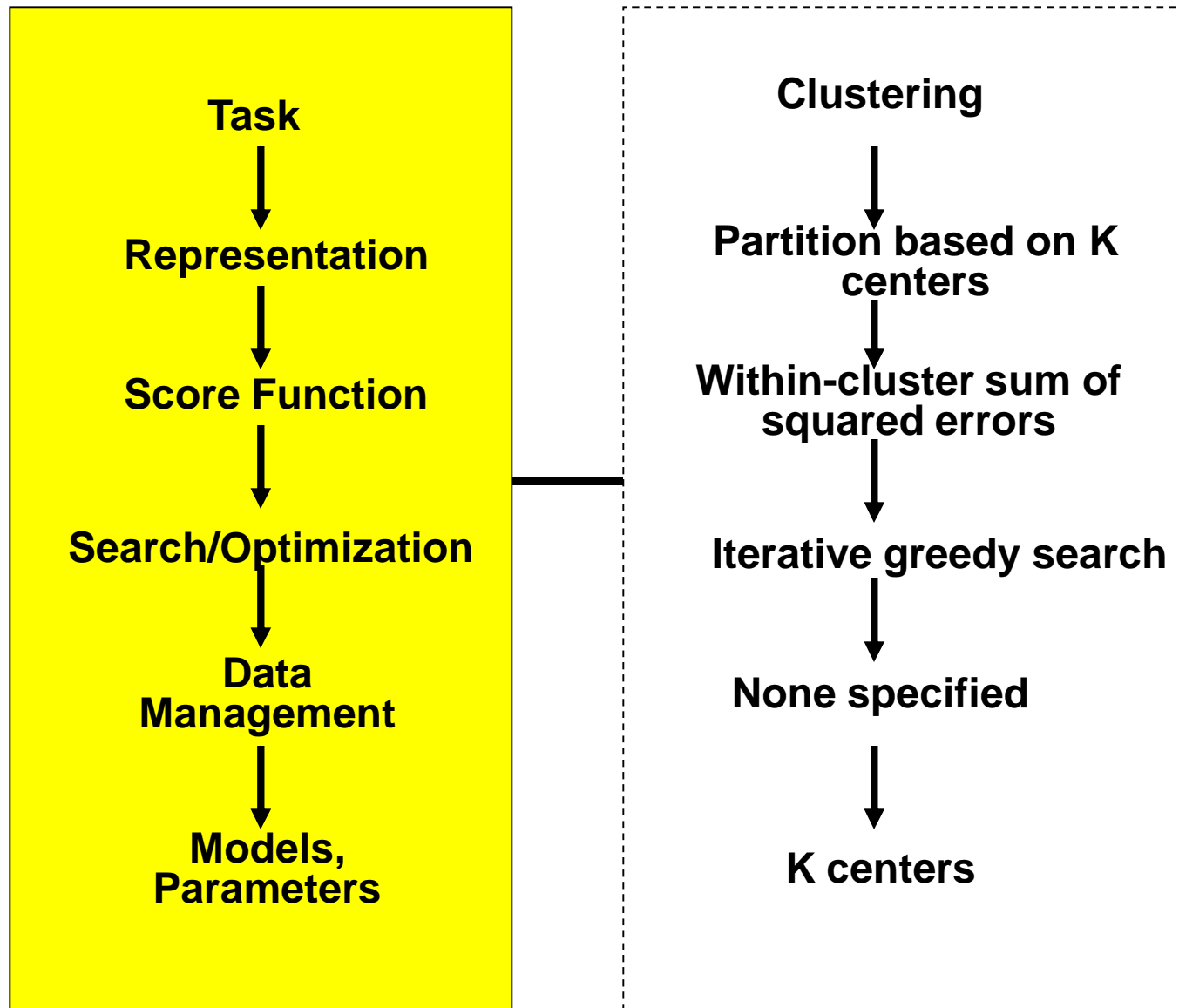
- Fundamental to all clustering techniques is the choice of distance measure between data points

$$D(\mathbf{x}_i, \mathbf{x}_j) = \sum_{k=1}^q (x_{ik} - x_{jk})^2$$

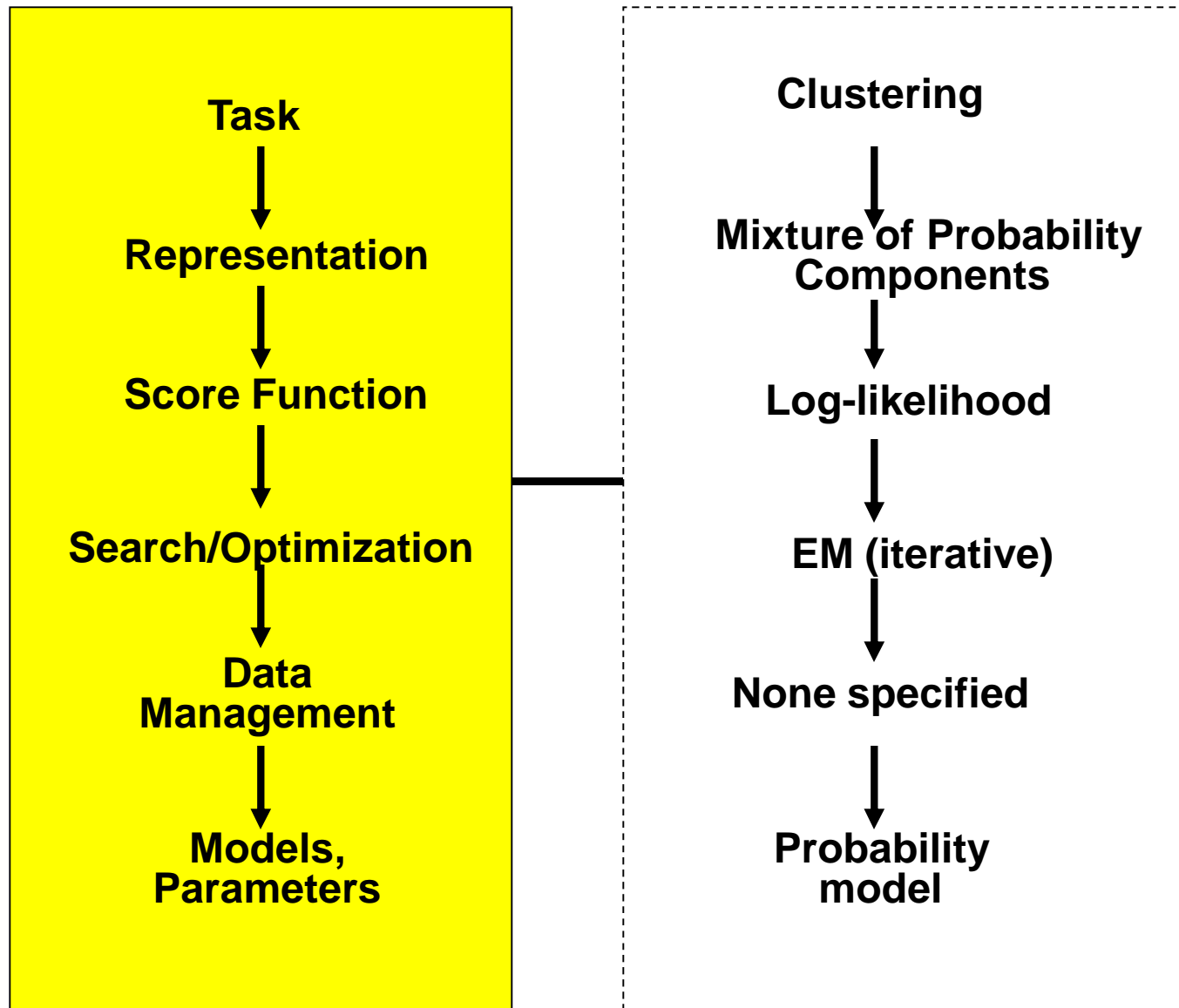
**Squared Euclidean distance**

- **Assumption:** All features are **equally important**
  - Such approaches fail in high dimensional spaces
- Feature selection (Dy and Brodley, 2000)
- Dimensionality Reduction

# K-Means Clustering



# Probabilistic Model-Based Clustering



# Single-Link Hierarchical Clustering

