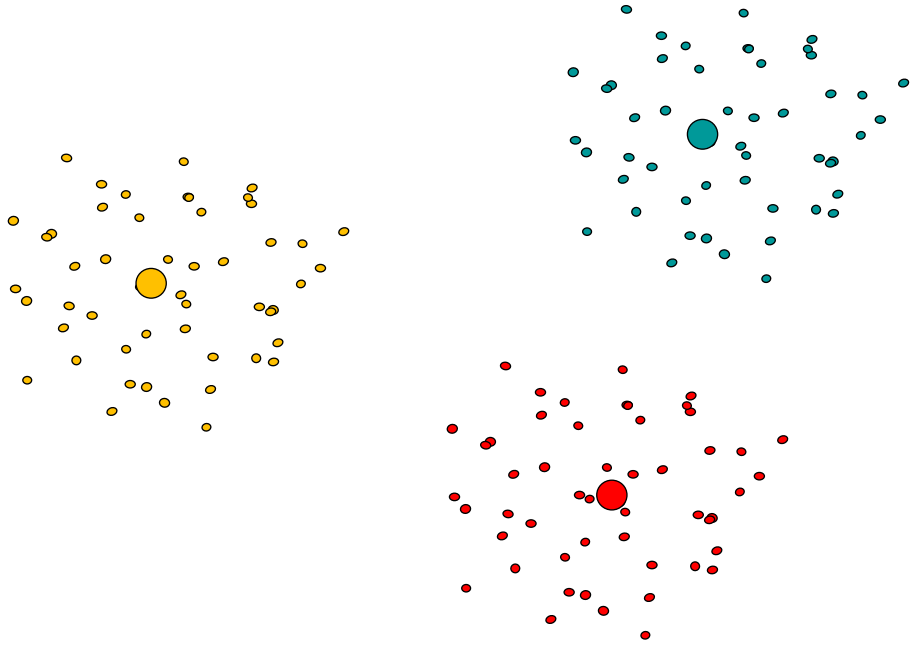
Machine Learning

Clustering and Dimensionality Reduction

Recap from previous lecture

In the previous lecture, we saw how SVM (Support Vector Machine) could be used for classification, this lecture will discuss the **unsupervised learning** (clustering).

Today we will take a step back to better understand clustering- what it is and why it is useful- and examine a variety of clustering methods.



Clustering data

Main goal: Given a dataset of N records, we wish to partition the dataset into k << N groups such that records in the same group are similar to each other, and records in different groups are dissimilar.

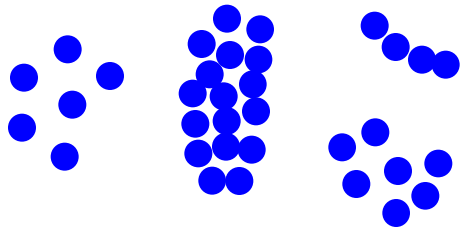
Given a population of individuals, we want to **identify** and **characterize** the underlying subgroups (or "clusters") of individuals, and possibly use these subgroups for **prediction**.

We want to explain the data in terms of its natural groupings.

How many groups are there?
Who belongs to which group?
Characteristics of each group?

Example: identify congressional voting blocs.

How well do blocs correspond to party affiliation?
Are there relevant blocs within a party?
Are there "mavericks" that vote across party lines?
How much do blocs vary with proposed legislation?



2-D example

Clustering data

Main goal: Given a dataset of N records, we wish to partition the dataset into k << N groups such that records in the same group are similar to each other, and records in different groups are dissimilar.

Given a population of individuals, we want to **identify** and **characterize** the underlying subgroups (or "clusters") of individuals, and possibly use these subgroups for **prediction**.

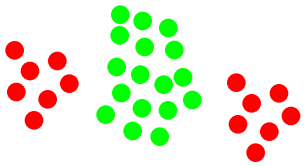
How does clustering differ from prediction?

There may not be any single output that we are trying to predict.

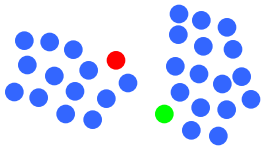
We are interested in an underlying **structure** that explains or predicts many characteristics of the population.

Clustering can sometimes improve prediction performance.

Improve model-based classification by learning multiple models per class.



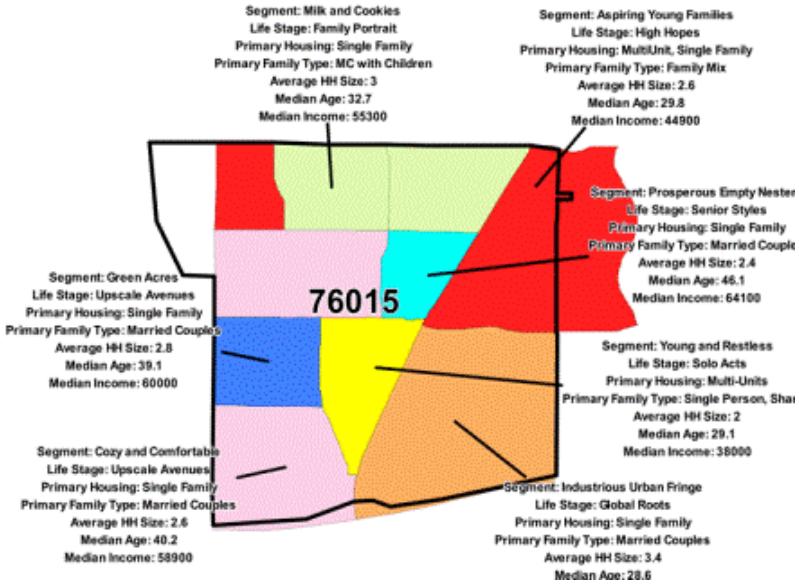
We may have little or no labeled training data for learning class models, but lots of unlabeled data.



Applications of clustering

Main goal: Given a dataset of N records, we wish to partition the dataset into k << N groups such that records in the same group are similar to each other, and records in different groups are dissimilar.

An important application of clustering is to improve **prediction** of an individual's characteristics or behavior, using information obtained from other members of their inferred group.



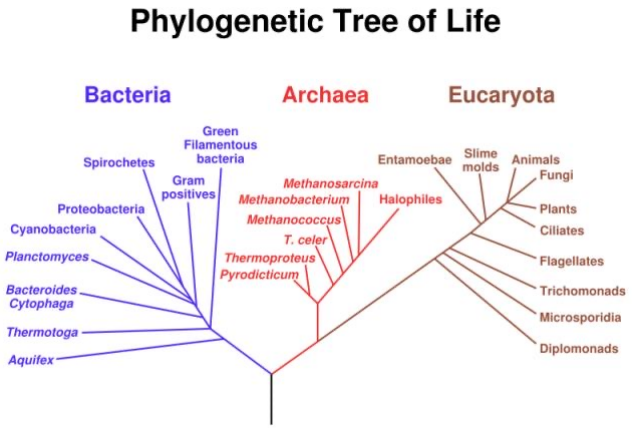
Customer database segmentation: To which subgroups should I market my product, and how should I target them? (Similarly, voter database segmentation)

Patient database segmentation: Different subgroups of patients may benefit from different treatment regimens.

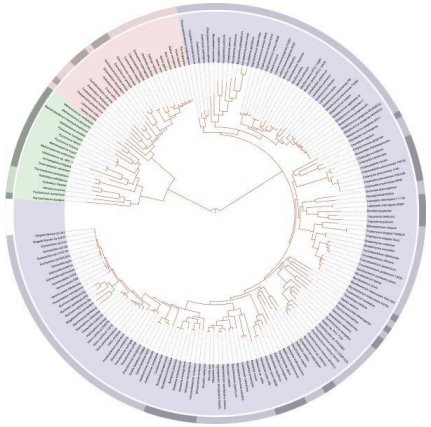
Applications of clustering

Main goal: Given a dataset of N records, we wish to partition the dataset into k << N groups such that records in the same group are similar to each other, and records in different groups are dissimilar.

Clustering is also very commonly used in **evolutionary biology** to create "phylogenetic trees" relating different species and showing when the species diverged from a common ancestor.



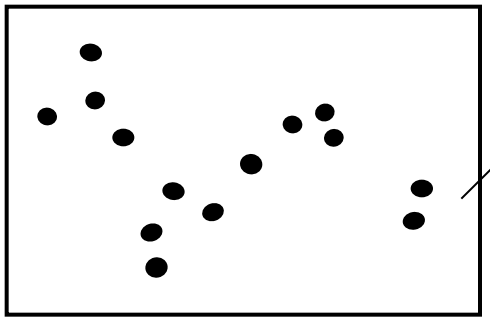
Here is a simple phylogenetic tree developed manually by evolutionary biologists.



Now much more detailed trees can be generated automatically by clustering DNA sequences, enhancing our understanding of evolution.

Here we are interested in learning the **hierarchy** of clusters!

Hierarchical clustering



14 records, 2 real-valued attributes, Euclidean distance

Given a set of records $x_1..x_N$ and a distance metric $d(x_i, x_j)$.

Records can have real and discrete-valued attributes.

We will create a **hierarchy** of clusters by merging similar records.

How to define "nearest" clusters?

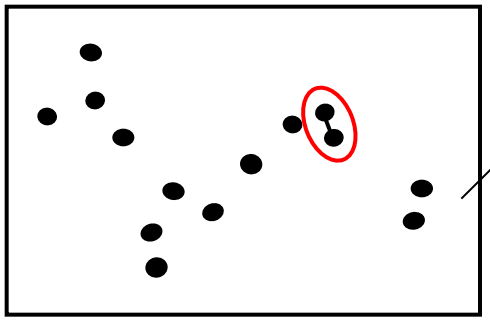
$$D(C,C') = \min_{x \in C,\, x' \in C'} d(x,x')$$

Single-link clustering

Bottom-up hierarchical clustering

- Start with N clusters, each containing one record.
- Choose the two "nearest" clusters, and merge them into a single cluster.
- Repeat until all points are merged into a single cluster.

Single-link clustering



14 records, 2 real-valued attributes, Euclidean distance

Given a set of records $x_1..x_N$ and a distance metric $d(x_i, x_j)$.

Records can have real and discrete-valued attributes.

We will create a **hierarchy** of clusters by merging similar records.

How to define "nearest" clusters?

$$D(C,C') = \min_{x \in C,\, x' \in C'} d(x,x')$$

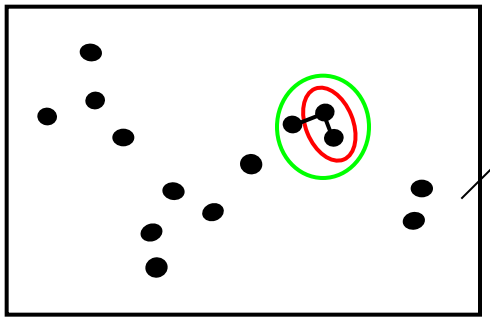
Single-link clustering

Bottom-up hierarchical clustering

- Start with N clusters, each containing one record.
- Choose the two "nearest" clusters, and merge them into a single cluster.
- Repeat until all points are merged into a single cluster.

(After 1 merge)

Single-link clustering



14 records, 2 real-valued attributes, Euclidean distance

Given a set of records $x_1..x_N$ and a distance metric $d(x_i, x_j)$.

Records can have real and discrete-valued attributes.

We will create a **hierarchy** of clusters by merging similar records.

How to define "nearest" clusters?

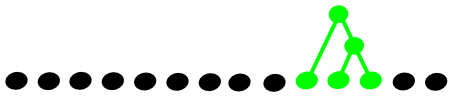
$$D(C,C') = \min_{x \in C,\, x' \in C'} d(x,x')$$

Single-link clustering

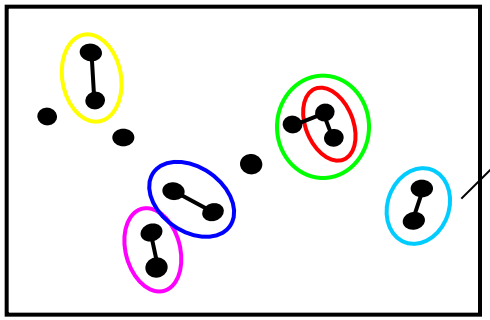
Bottom-up hierarchical clustering

- Start with N clusters, each containing one record.
- Choose the two "nearest" clusters, and merge them into a single cluster.
- Repeat until all points are merged into a single cluster.

(After 2 merges)



Single-link clustering



14 records, 2 real-valued attributes, Euclidean distance

Given a set of records $x_1..x_N$ and a distance metric $d(x_i, x_j)$.

Records can have real and discrete-valued attributes.

We will create a **hierarchy** of clusters by merging similar records.

How to define "nearest" clusters?

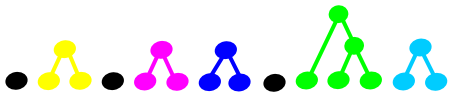
$$D(C,C') = \min_{x \in C,\, x' \in C'} d(x,x')$$

Single-link clustering

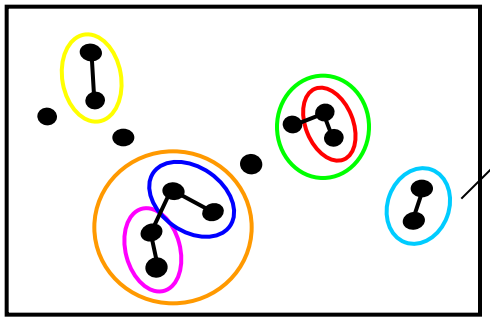
Bottom-up hierarchical clustering

- Start with N clusters, each containing one record.
- Choose the two "nearest" clusters, and merge them into a single cluster.
- Repeat until all points are merged into a single cluster.

(After 6 merges)



Single-link clustering



14 records, 2 real-valued attributes, Euclidean distance

Given a set of records $x_1..x_N$ and a distance metric $d(x_i, x_j)$.

Records can have real and discrete-valued attributes.

We will create a **hierarchy** of clusters by merging similar records.

How to define "nearest" clusters?

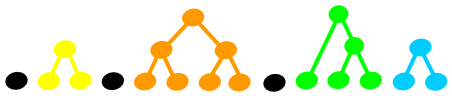
$$D(C,C') = \min_{x \in C,\, x' \in C'} d(x,x')$$

Single-link clustering

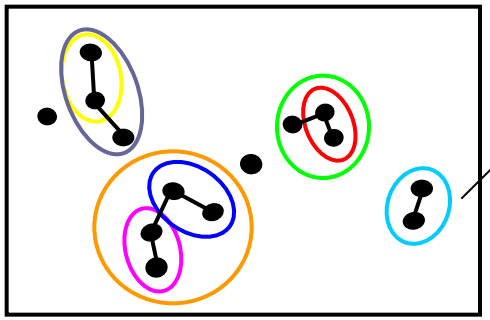
Bottom-up hierarchical clustering

- Start with N clusters, each containing one record.
- Choose the two "nearest" clusters, and merge them into a single cluster.
- Repeat until all points are merged into a single cluster.

(After 7 merges)



Single-link clustering



14 records, 2 real-valued attributes, Euclidean distance

Given a set of records $x_1..x_N$ and a distance metric $d(x_i, x_j)$.

Records can have real and discrete-valued attributes.

We will create a **hierarchy** of clusters by merging similar records.

How to define "nearest" clusters?

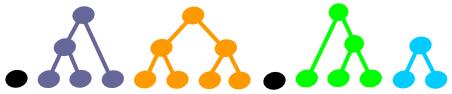
$$D(C,C') = \min_{x \in C,\, x' \in C'} d(x,x')$$

Single-link clustering

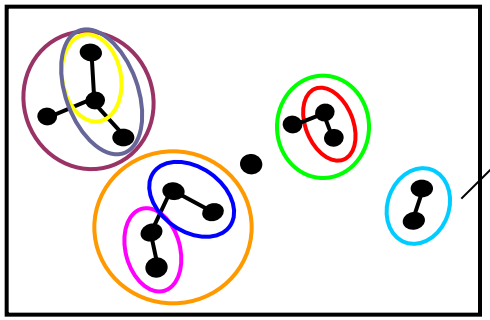
Bottom-up hierarchical clustering

- Start with N clusters, each containing one record.
- Choose the two "nearest" clusters, and merge them into a single cluster.
- Repeat until all points are merged into a single cluster.

(After 8 merges)



Single-link clustering



14 records, 2 real-valued attributes, Euclidean distance

Given a set of records $x_1..x_N$ and a distance metric $d(x_i, x_j)$.

Records can have real and discrete-valued attributes.

We will create a **hierarchy** of clusters by merging similar records.

How to define "nearest" clusters?

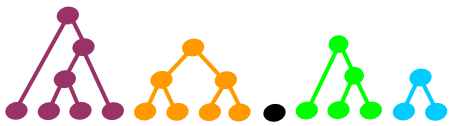
$$D(C,C') = \min_{x \in C,\, x' \in C'} d(x,x')$$

Single-link clustering

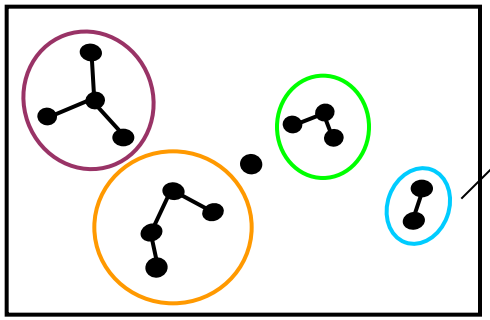
Bottom-up hierarchical clustering

- Start with N clusters, each containing one record.
- Choose the two "nearest" clusters, and merge them into a single cluster.
- Repeat until all points are merged into a single cluster.

(After 9 merges)



Single-link clustering



14 records, 2 real-valued attributes, Euclidean distance

Given a set of records $x_1..x_N$ and a distance metric $d(x_i, x_j)$.

Records can have real and discrete-valued attributes.

We will create a **hierarchy** of clusters by merging similar records.

How to define "nearest" clusters?

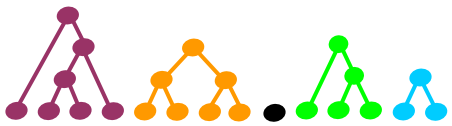
$$D(C,C') = \min_{x \in C,\, x' \in C'} d(x,x')$$

Single-link clustering

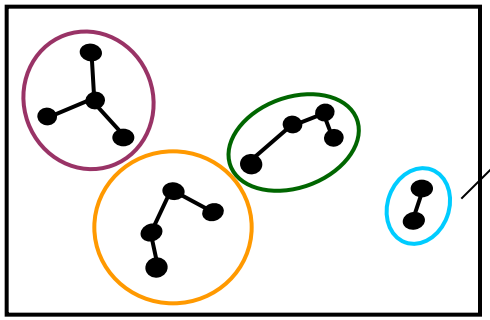
Bottom-up hierarchical clustering

- Start with N clusters, each containing one record.
- Choose the two "nearest" clusters, and merge them into a single cluster.
- Repeat until all points are merged into a single cluster.

(After 9 merges)



Single-link clustering



14 records, 2 real-valued attributes, Euclidean distance

Given a set of records $x_1..x_N$ and a distance metric $d(x_i, x_j)$.

Records can have real and discrete-valued attributes.

We will create a **hierarchy** of clusters by merging similar records.

How to define "nearest" clusters?

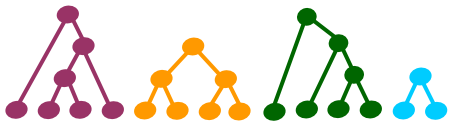
$$D(C,C') = \min_{x \in C,\, x' \in C'} d(x,x')$$

Single-link clustering

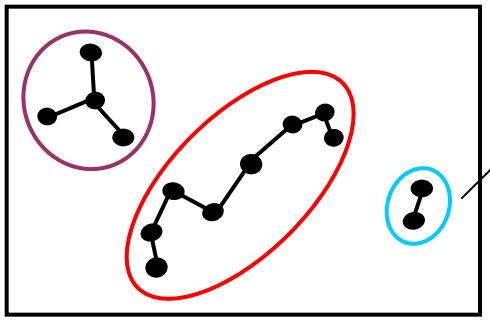
Bottom-up hierarchical clustering

- Start with N clusters, each containing one record.
- Choose the two "nearest" clusters, and merge them into a single cluster.
- Repeat until all points are merged into a single cluster.

(After 10 merges)



Single-link clustering



14 records, 2 real-valued attributes, Euclidean distance

Given a set of records $x_1..x_N$ and a distance metric $d(x_i, x_j)$.

Records can have real and discrete-valued attributes.

We will create a **hierarchy** of clusters by merging similar records.

How to define "nearest" clusters?

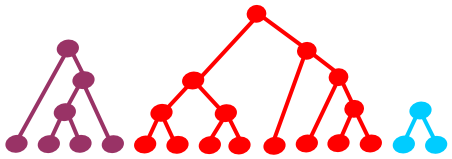
$$D(C,C') = \min_{x \in C,\, x' \in C'} d(x,x')$$

Single-link clustering

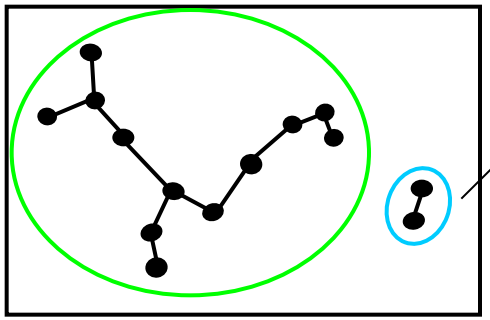
Bottom-up hierarchical clustering

- Start with N clusters, each containing one record.
- Choose the two "nearest" clusters, and merge them into a single cluster.
- Repeat until all points are merged into a single cluster.

(After 11 merges)



Single-link clustering



14 records, 2 real-valued attributes, Euclidean distance

Given a set of records $x_1..x_N$ and a distance metric $d(x_i, x_j)$.

Records can have real and discrete-valued attributes.

We will create a **hierarchy** of clusters by merging similar records.

How to define "nearest" clusters?

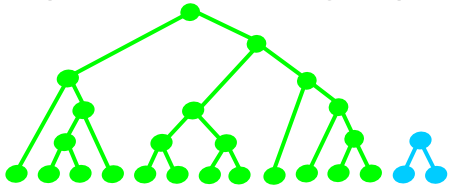
$$D(C,C') = \min_{x \in C,\, x' \in C'} d(x,x')$$

Single-link clustering

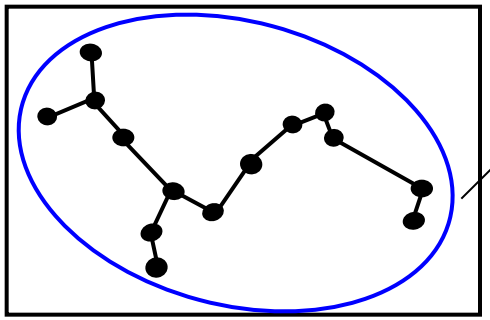
Bottom-up hierarchical clustering

- Start with N clusters, each containing one record.
- Choose the two "nearest" clusters, and merge them into a single cluster.
- Repeat until all points are merged into a single cluster.

(After 12 merges)



Single-link clustering



14 records, 2 real-valued attributes, Euclidean distance

Given a set of records $x_1..x_N$ and a distance metric $d(x_i, x_j)$.

Records can have real and discrete-valued attributes.

We will create a **hierarchy** of clusters by merging similar records.

How to define "nearest" clusters?

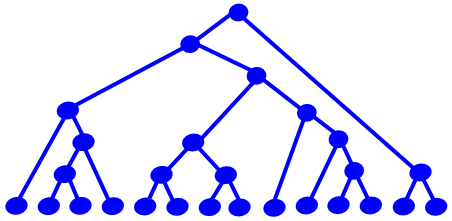
$$D(C,C') = \min_{x \in C,\, x' \in C'} d(x,x')$$

Single-link clustering

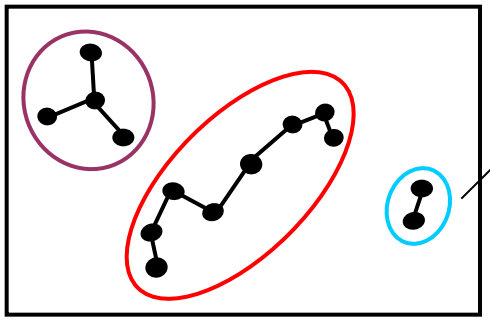
Bottom-up hierarchical clustering

- Start with N clusters, each containing one record.
- Choose the two "nearest" clusters, and merge them into a single cluster.
- Repeat until all points are merged into a single cluster.

Done!



Single-link clustering



14 records, 2 real-valued attributes, Euclidean distance

Given a set of records $x_1..x_N$ and a distance metric $d(x_i, x_j)$.

Records can have real and discrete-valued attributes.

We will create a **hierarchy** of clusters by merging similar records.

How to define "nearest" clusters?

$$D(C,C') = \min_{x \in C,\, x' \in C'} d(x,x')$$

Single-link clustering

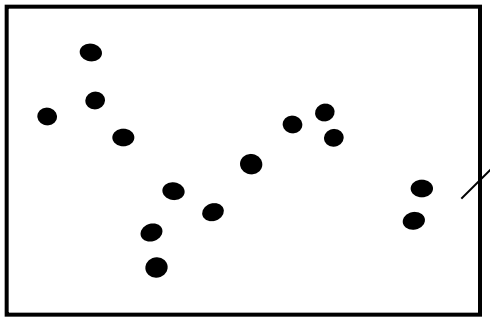
Bottom-up hierarchical clustering

- Start with N clusters, each containing one record.
- Choose the two "nearest" clusters, and merge them into a single cluster.
- Repeat until all points are merged into a single cluster.

Note that single-link clustering tends to produce highly elongated groups (or "chains") as shown above.

If we want to produce more spherical groups, we can instead consider the **maximum** distance between clusters.

Complete-link clustering



14 records, 2 real-valued attributes, Euclidean distance

Given a set of records $x_1..x_N$ and a distance metric $d(x_i, x_j)$.

Records can have real and discrete-valued attributes.

We will create a **hierarchy** of clusters by merging similar records.

How to define "nearest" clusters?

$$D(C,C') = \max_{x \in C,\, x' \in C'} d(x,x')$$

Complete-link clustering

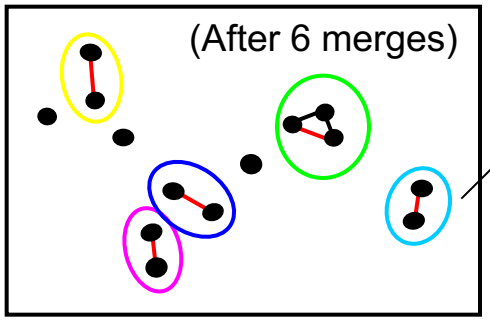
Bottom-up hierarchical clustering

- Start with N clusters, each containing one record.
- Choose the two "nearest" clusters, and merge them into a single cluster.
- Repeat until all points are merged into a single cluster.

Note that single-link clustering tends to produce highly elongated groups (or "chains") as shown above.

If we want to produce more spherical groups, we can instead consider the **maximum** distance between clusters.

Complete-link clustering



14 records, 2 real-valued attributes, Euclidean distance

Given a set of records $x_1..x_N$ and a distance metric $d(x_i, x_j)$.

Records can have real and discrete-valued attributes.

We will create a **hierarchy** of clusters by merging similar records.

How to define "nearest" clusters?

$$D(C,C') = \max_{x \in C,\, x' \in C'} d(x,x')$$

Complete-link clustering

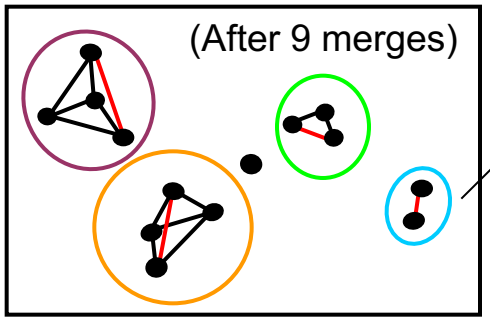
Bottom-up hierarchical clustering

- Start with N clusters, each containing one record.
- Choose the two "nearest" clusters, and merge them into a single cluster.
- Repeat until all points are merged into a single cluster.

Note that single-link clustering tends to produce highly elongated groups (or "chains") as shown above.

If we want to produce more spherical groups, we can instead consider the **maximum** distance between clusters.

Complete-link clustering



14 records, 2 real-valued attributes, Euclidean distance

Given a set of records $x_1..x_N$ and a distance metric $d(x_i, x_j)$.

Records can have real and discrete-valued attributes.

We will create a **hierarchy** of clusters by merging similar records.

How to define "nearest" clusters?

$$D(C,C') = \max_{x \in C,\, x' \in C'} d(x,x')$$

Complete-link clustering

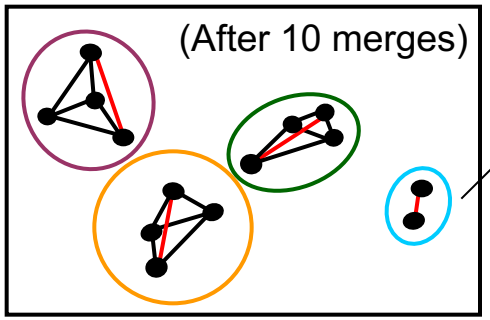
Bottom-up hierarchical clustering

- Start with N clusters, each containing one record.
- Choose the two "nearest" clusters, and merge them into a single cluster.
- Repeat until all points are merged into a single cluster.

Note that single-link clustering tends to produce highly elongated groups (or "chains") as shown above.

If we want to produce more spherical groups, we can instead consider the **maximum** distance between clusters.

Complete-link clustering



14 records, 2 real-valued attributes, Euclidean distance

Given a set of records $x_1..x_N$ and a distance metric $d(x_i, x_j)$.

Records can have real and discrete-valued attributes.

We will create a **hierarchy** of clusters by merging similar records.

How to define "nearest" clusters?

$$D(C,C') = \max_{x \in C,\, x' \in C'} d(x,x')$$

Complete-link clustering

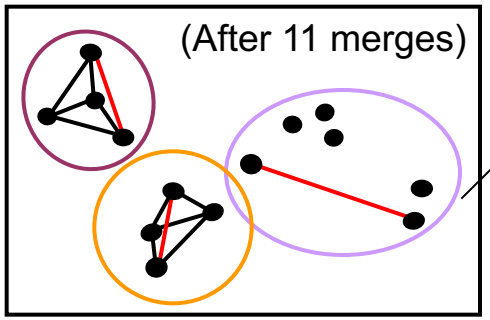
Bottom-up hierarchical clustering

- Start with N clusters, each containing one record.
- Choose the two "nearest" clusters, and merge them into a single cluster.
- Repeat until all points are merged into a single cluster.

Note that single-link clustering tends to produce highly elongated groups (or "chains") as shown above.

If we want to produce more spherical groups, we can instead consider the **maximum** distance between clusters.

Complete-link clustering



14 records, 2 real-valued attributes, Euclidean distance

Given a set of records $x_1..x_N$ and a distance metric $d(x_i, x_j)$.

Records can have real and discrete-valued attributes.

We will create a **hierarchy** of clusters by merging similar records.

How to define "nearest" clusters?

$$D(C,C') = \max_{x \in C,\, x' \in C'} d(x,x')$$

Complete-link clustering

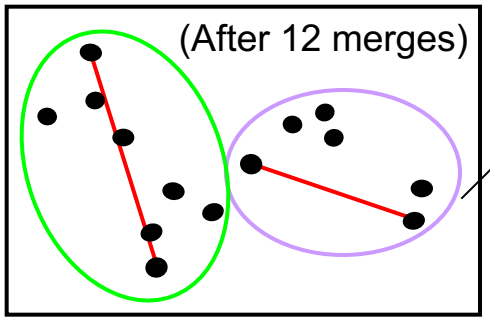
Bottom-up hierarchical clustering

- Start with N clusters, each containing one record.
- Choose the two "nearest" clusters, and merge them into a single cluster.
- Repeat until all points are merged into a single cluster.

Note that single-link clustering tends to produce highly elongated groups (or "chains") as shown above.

If we want to produce more spherical groups, we can instead consider the **maximum** distance between clusters.

Complete-link clustering



14 records, 2 real-valued attributes, Euclidean distance

Given a set of records $x_1..x_N$ and a distance metric $d(x_i, x_j)$.

Records can have real and discrete-valued attributes.

We will create a **hierarchy** of clusters by merging similar records.

How to define "nearest" clusters?

$$D(C,C') = \max_{x \in C,\, x' \in C'} d(x,x')$$

Complete-link clustering

Bottom-up hierarchical clustering

- Start with N clusters, each containing one record.
- Choose the two "nearest" clusters, and merge them into a single cluster.
- Repeat until all points are merged into a single cluster.

Note that single-link clustering tends to produce highly elongated groups (or "chains") as shown above.

If we want to produce more spherical groups, we can instead consider the **maximum** distance between clusters.

More Clustering Metric: https://en.wikipedia.org/wiki/Hierarchical_clustering

Single-Link vs. Complete-Link Clustering

Both single-link and complete-link clustering are agglomerative hierarchical clustering methods, but they define the "distance" between two clusters differently, leading to distinct clustering results and susceptibilities to certain data structures.

Complete-Link Clustering:
Distance Measure: The distance between two clusters is defined as the distance between the two farthest points, one from each cluster.
Tendency: Complete-link clustering tends to produce more compact, spherical clusters because it considers the worst-case (farthest) link.
Avoiding Elongated Clusters: It avoids the chaining effect and elongated clusters as it demands that every point in a potential new cluster is reasonably close to every other point, ensuring a more globular shape.

Single-Link Clustering

Distance Measure: The distance between two clusters is defined as the distance between the two closest points, one from each cluster.

Tendency: Single-link clustering has a tendency to produce elongated or chain-like clusters because it considers the shortest link, which can result in joining distant clusters if they have a single close pair of points.

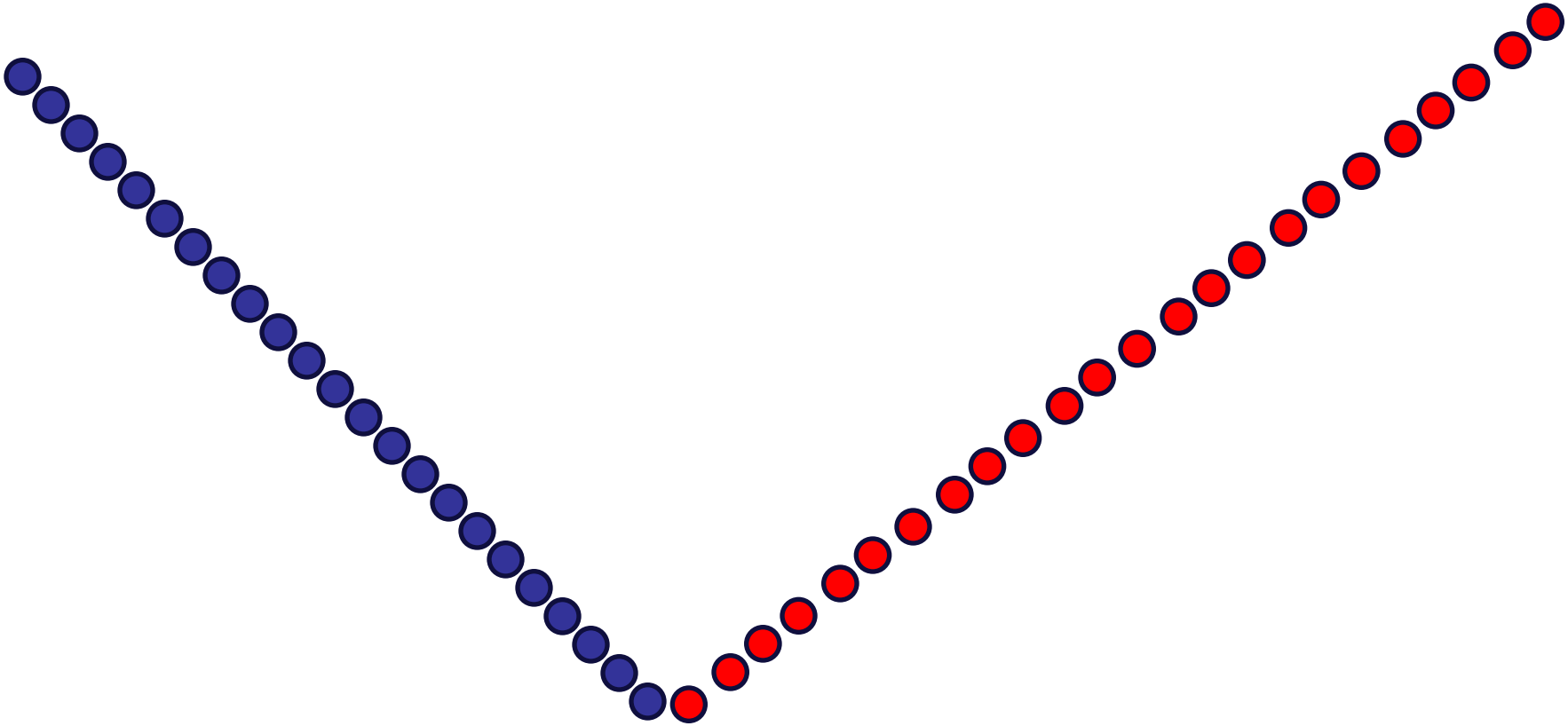
Chaining Effect: It can create a "chaining effect," where clusters can be strung along a line or curve, even if visually there are separate clusters. This is because it can link distant clusters via a series of individual points.

Complete-Link Clustering

Distance Measure: The distance between two clusters is defined as the distance between the two farthest points, one from each cluster.

Tendency: Complete-link clustering tends to produce more compact, spherical clusters because it considers the worst-case (farthest) link.

Avoiding Elongated Clusters: It avoids the chaining effect and elongated clusters as it demands that every point in a potential new cluster is reasonably close to every other point, ensuring a more globular shape.



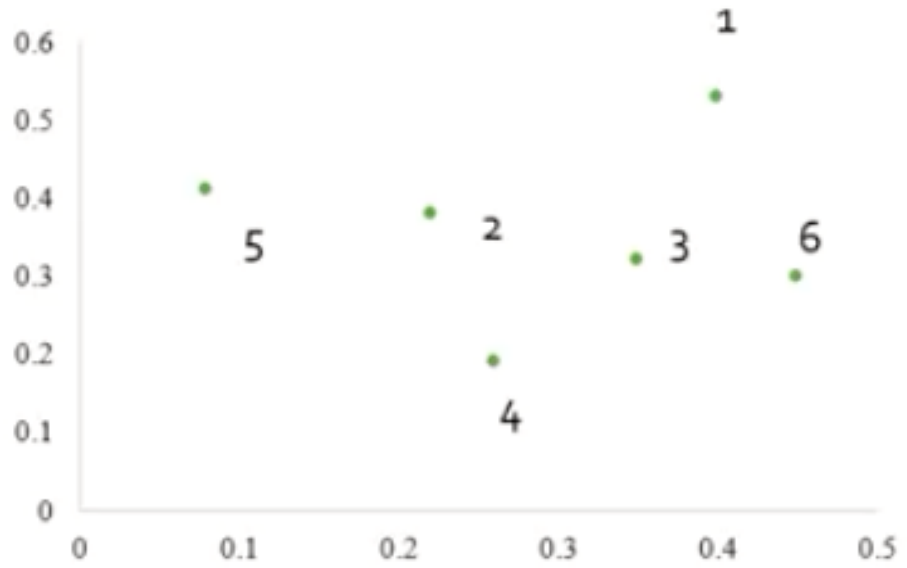
Examples-Dendrogram

Find the clusters using single link technique. Use Euclidean distance, and draw the dendrogram.

	X	Y
P1	0.40	0.53
P2	0.22	0.38
P3	0.35	0.32
P4	0.26	0.19
P5	0.08	0.41
P6	0.45	0.30

Reference: https://www.anuradhabhatia.com

	X	Y
P1	0.40	0.53
P2	0.22	0.38
P3	0.35	0.32
P4	0.26	0.19
P5	0.08	0.41
P6	0.45	0.30



Clustering Procedures: Single-Link

Bottom-up hierarchical clustering

- Start with N clusters, each containing one record.
- Choose the two “nearest” clusters, and merge them into a single cluster.
- Repeat until all points are merged into a single cluster.

The distance matrix is

	P1	P2	P3	P4	P5	P6
P1	0					
P2	0.23	0				
P3	0.22	0.15	0			
P4	0.37	0.20	0.15	0		
P5	0.34	0.14	0.28	0.29	0	
P6	0.23	0.25	0.11	0.22	0.39	0

Merge two closest clusters:

	P1	P2	P3	P4	P5	P6
P1	0					
P2	0.24	0				
P3	0.22	0.15	0			
P4	0.37	0.20	0.15	0		
P5	0.34	0.14	0.28	0.29	0	
P6	0.23	0.25	0.11	0.22	0.39	0

Updated Distance Matrix:

- To update the distance matrix MIN[dist(P3,P6),P1)]
- MIN(dist(P3,P1), (P6,P1))

  = min[(0.22,0.23)]

  = 0.22
- To update the distance matrix MIN[dist(P3,P6),P2)]
- MIN(dist(P3,P2), (P6,P2))

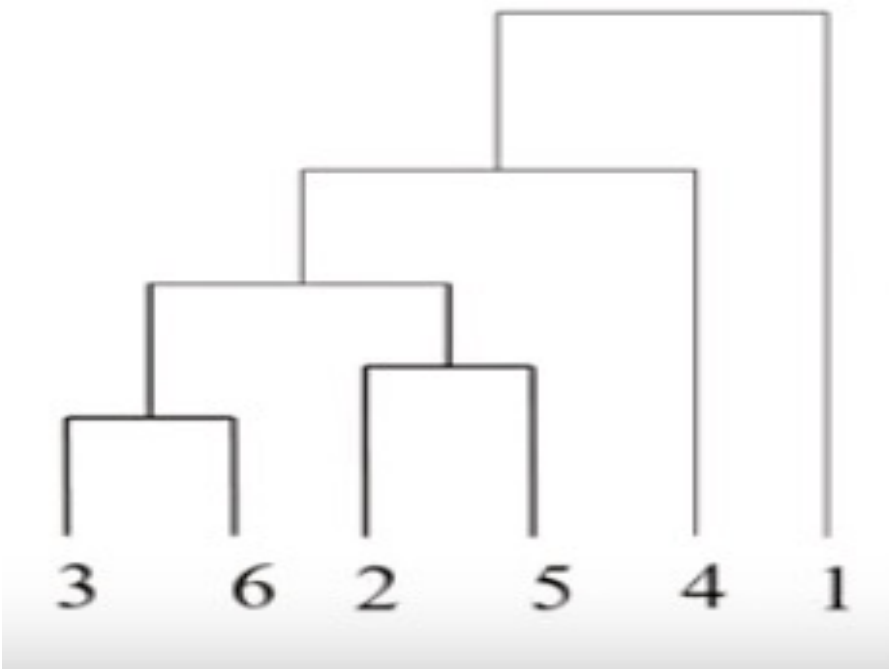
  = min[(0.15,0.25)]

  = 0.15

The updated distance matrix for cluster P3, P6

	P1	P2	P3,P6	P4	P5
P1	0				
P2	0.23	0			
P3,P6	0.22	0.15	0		
P4	0.37	0.20	0.15	0	
P5	0.34	0.14	0.28	0.29	0

Final Merge to One Cluster



Dendrogram

K-means clustering

Hierarchical clustering is a simple, useful clustering method, but it gives you no guarantees on the quality of the resulting groups.

An alternative is to define some objective function that describes the quality of the groups, and attempt to optimize that function.

Assume that all attributes are real-valued, so we can compute the centroid of any cluster (i.e. the mean value of each attribute)

Possible objective:
minimize distortion

$$\sum_{C_k} \sum_{x_i \in C_k} (x_i - \mu_k)^2$$

This is the sum of squared errors for each data point x_i, assuming that each x_i is mapped to the closest cluster center μ_k.

Possible moves for state-space search
Change position of cluster center μ_k
Change mapping of point x_i to center μ_k
Change number of clusters K

How to perform this search efficiently?

K-means clustering

Hierarchical clustering is a simple, useful clustering method, but it gives you no guarantees on the quality of the resulting groups.

An alternative is to define some objective function that describes the quality of the groups, and attempt to optimize that function.

Assume that all attributes are real-valued, so we can compute the centroid of any cluster (i.e. the mean value of each attribute)

Possible objective:
minimize distortion

$$\sum_{C_k} \sum_{x_i \in C_k} (x_i - \mu_k)^2$$

This is the sum of squared errors for each data point x_i, assuming that each x_i is mapped to the closest cluster center μ_k.

Possible moves for state-space search

Change position of cluster center μ_k

Change mapping of point x_i to center μ_k

Change number of clusters K

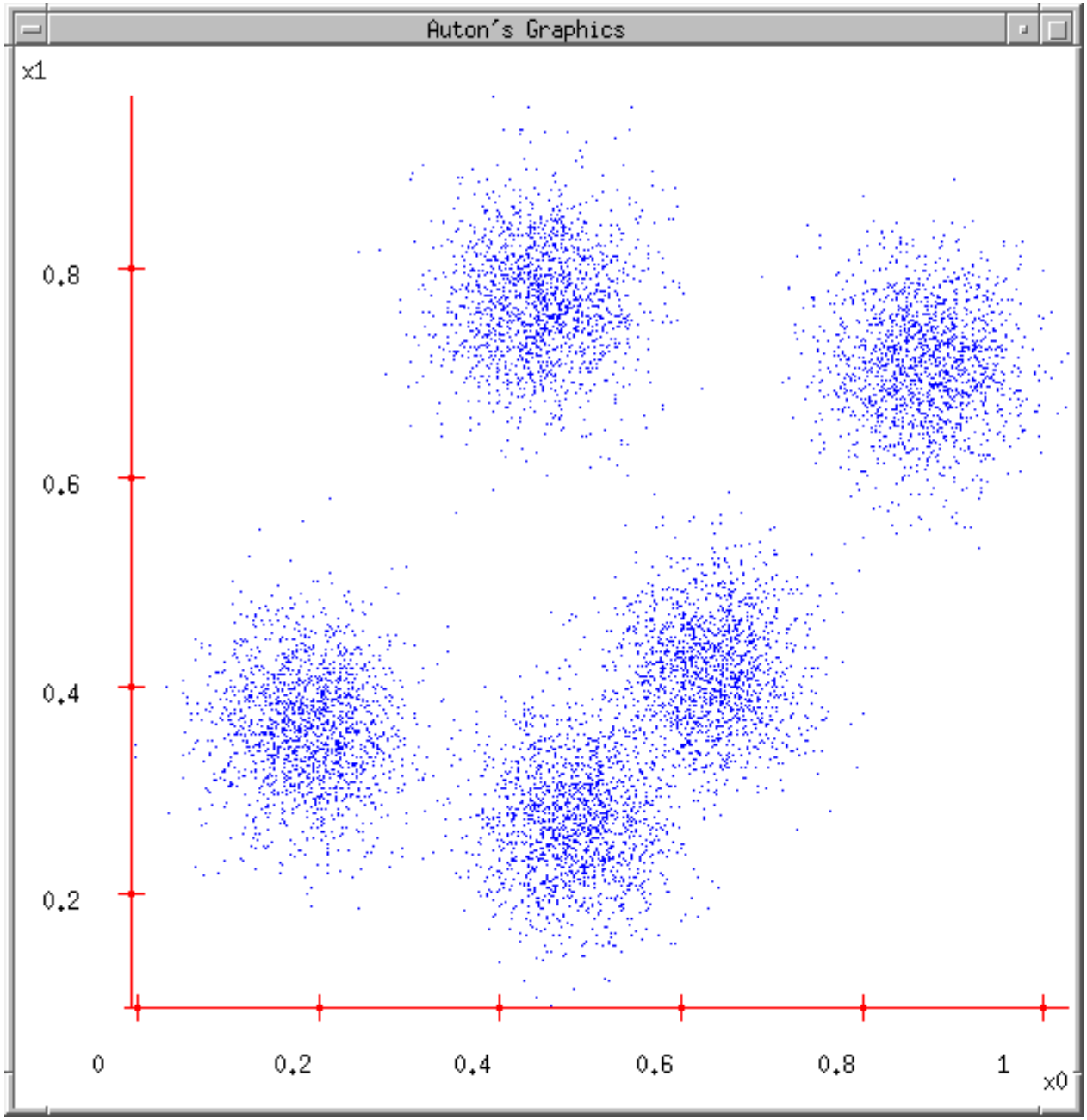
1. Moving μ_k to centroid of all points in C_k reduces distortion

2. Mapping point x_i to nearest center μ_k reduces distortion.

Keep number of centers fixed, and alternate between these two moves.

K-means

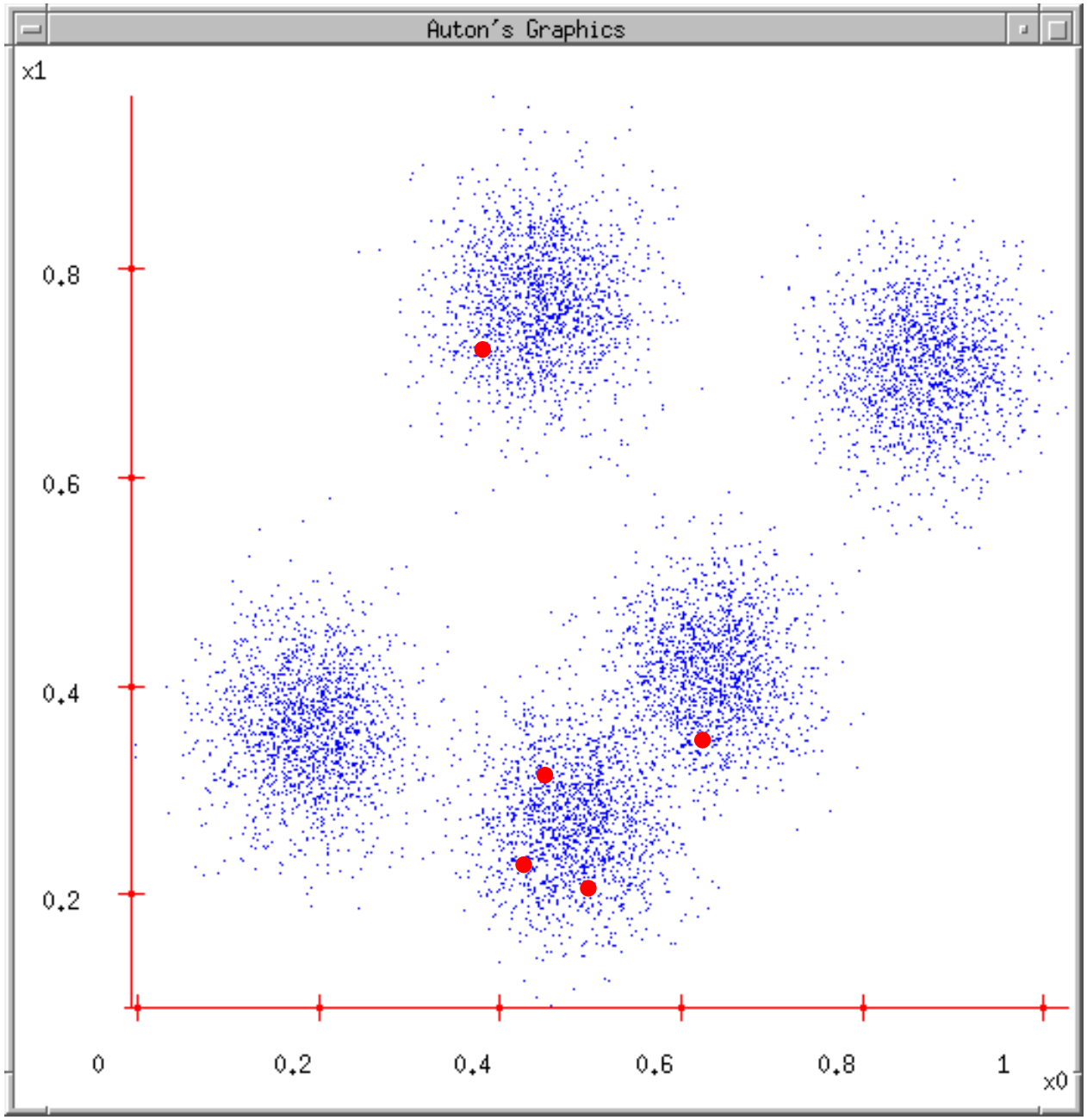
1. Ask user how many clusters they'd like. (e.g. k = 5)



Thanks to Andrew Moore for providing this example.

K-means

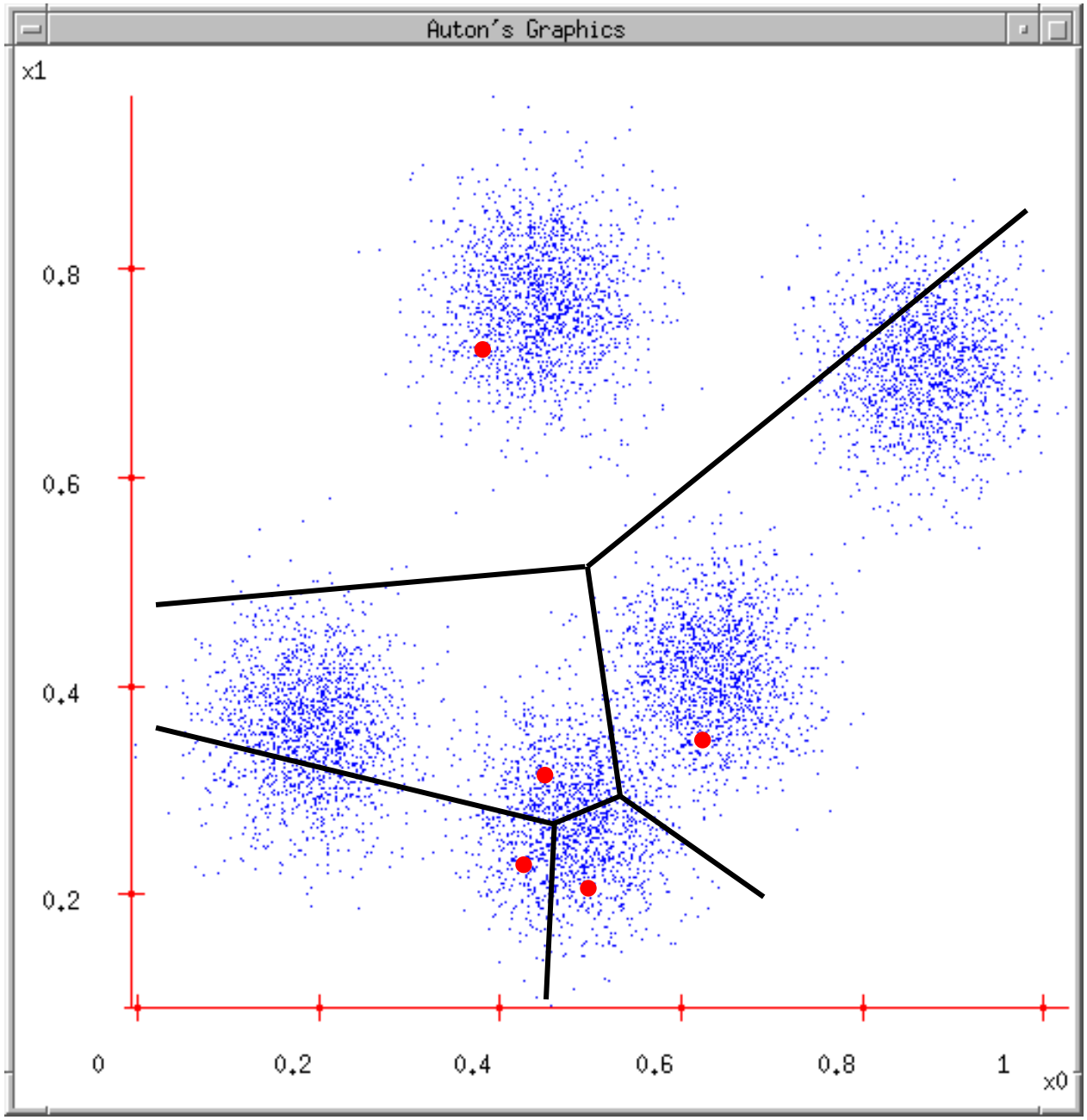
1. Ask user how many clusters they'd like. (e.g. k = 5)
2. Randomly guess k cluster center locations



Thanks to Andrew Moore for providing this example.

K-means

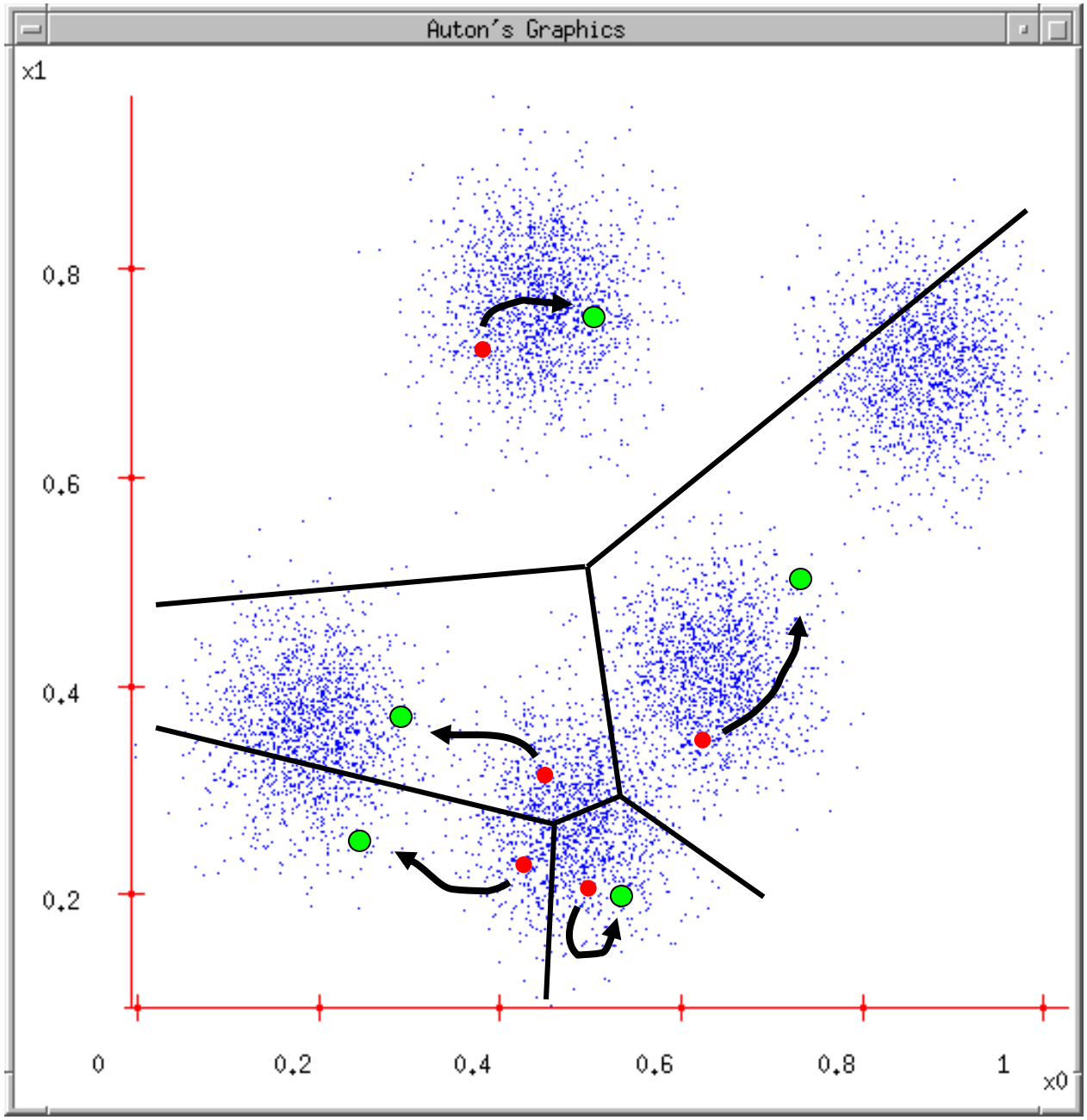
1. Ask user how many clusters they'd like. (e.g. k = 5)
2. Randomly guess k cluster center locations
3. Each datapoint finds out which center it's closest to.



Thanks to Andrew Moore for providing this example.

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, and moves there.

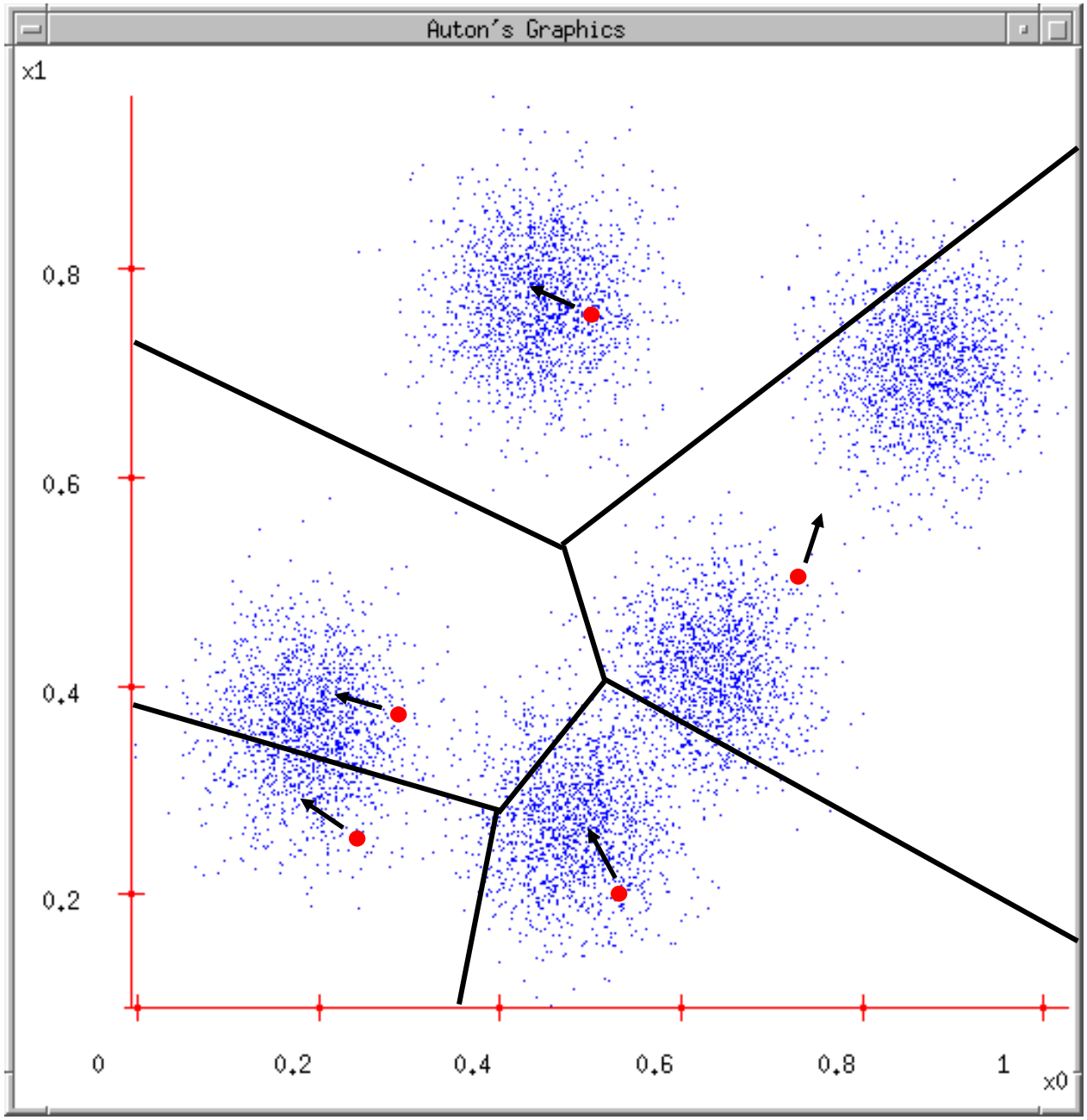


Thanks to Andrew Moore for providing this example.

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, and moves there.

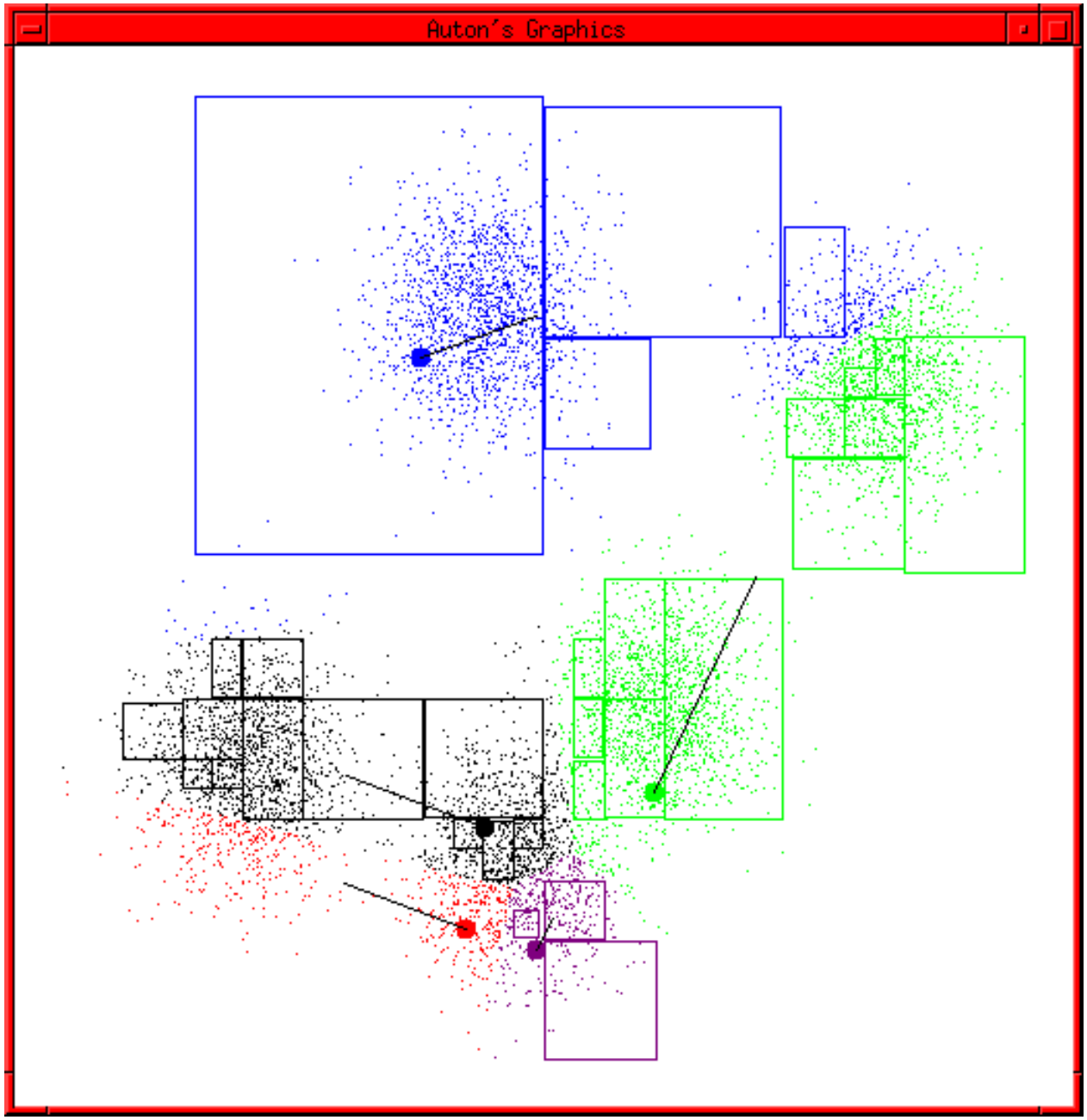
Repeat steps 3-4 until convergence!



Thanks to Andrew Moore for providing this example.

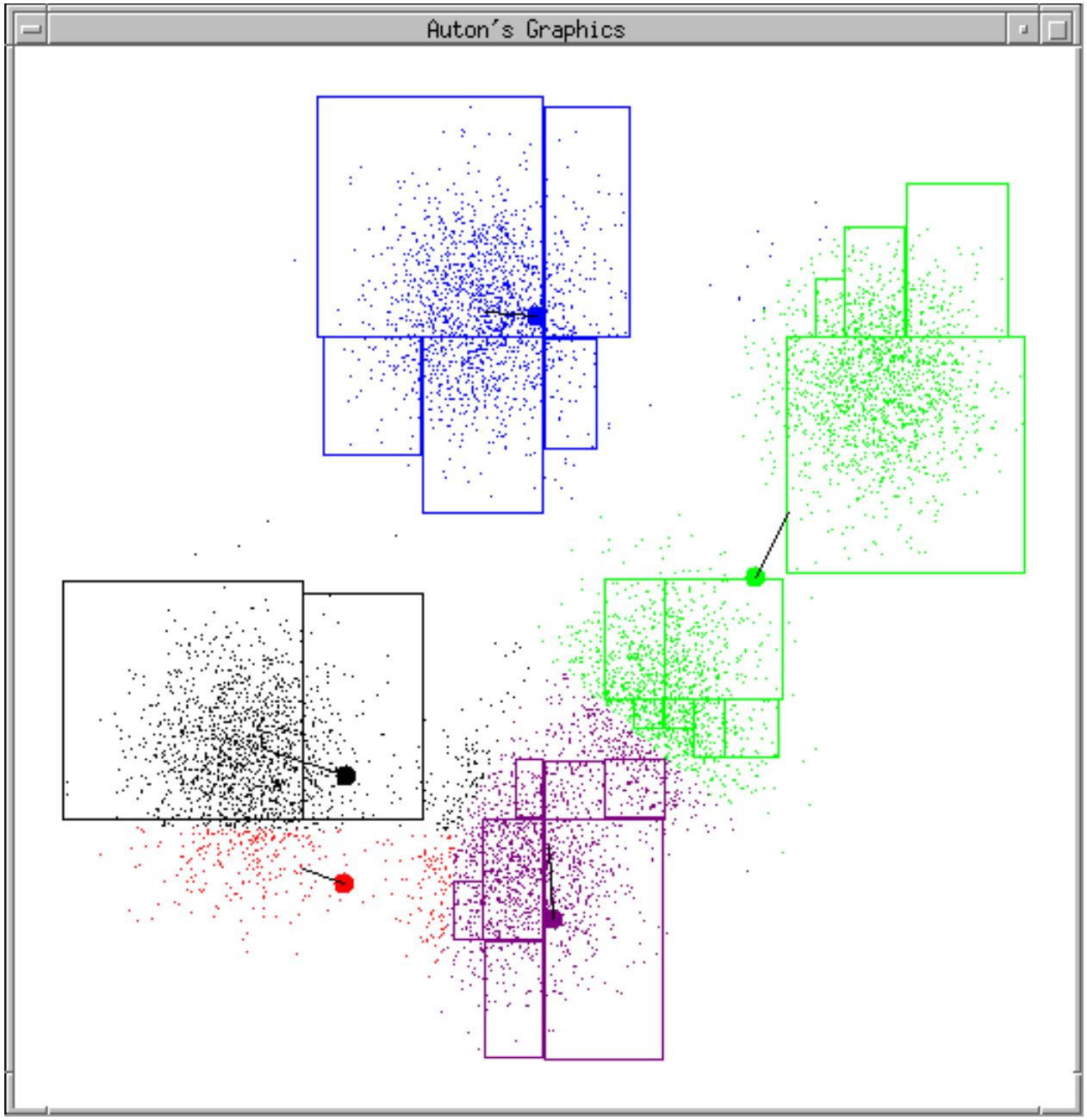
K-means

Here's an example run of k-means, generated by Dan Pelleg's software.



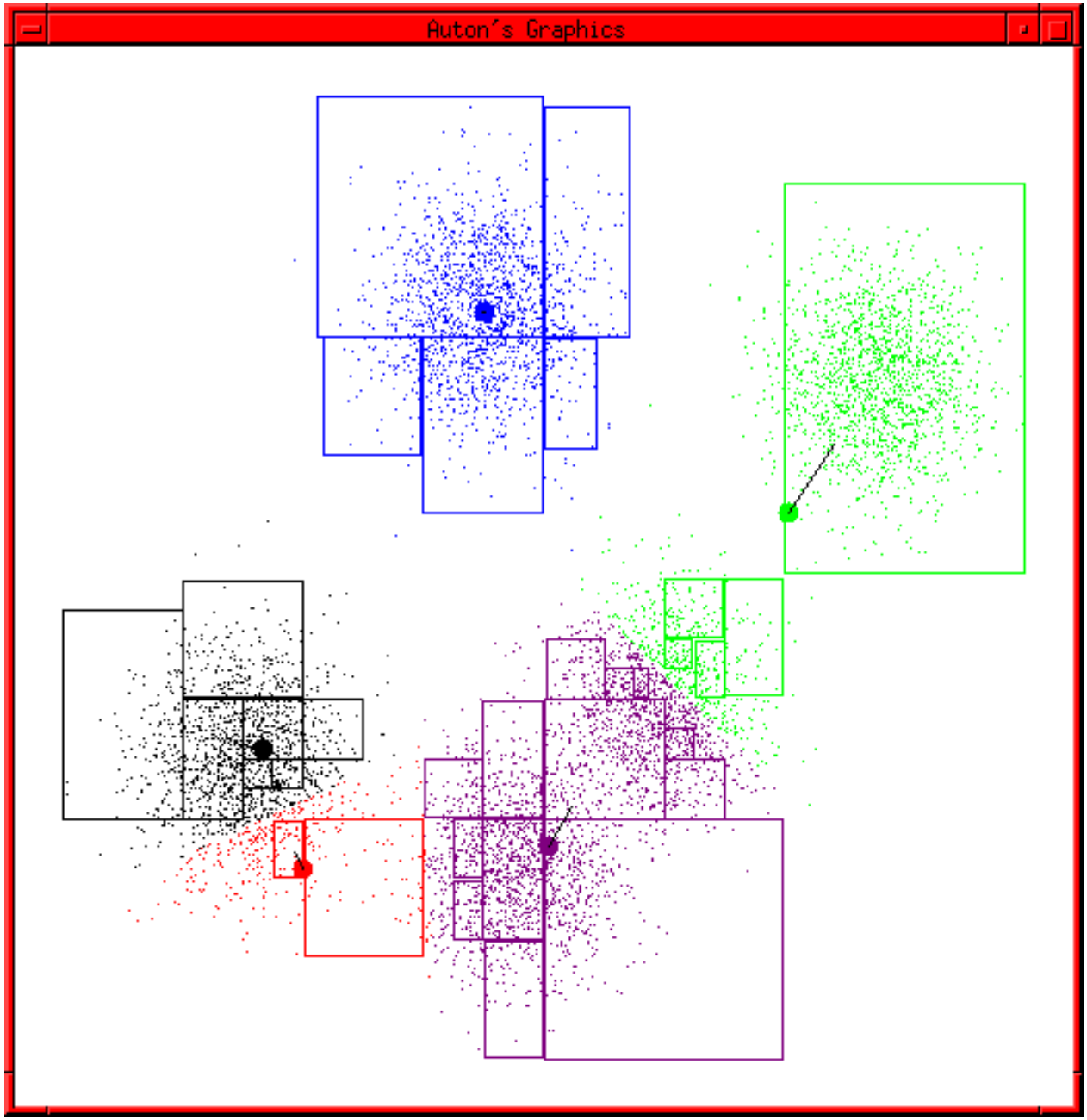
K-means

Here's an example run of k-means, generated by Dan Pelleg's software.



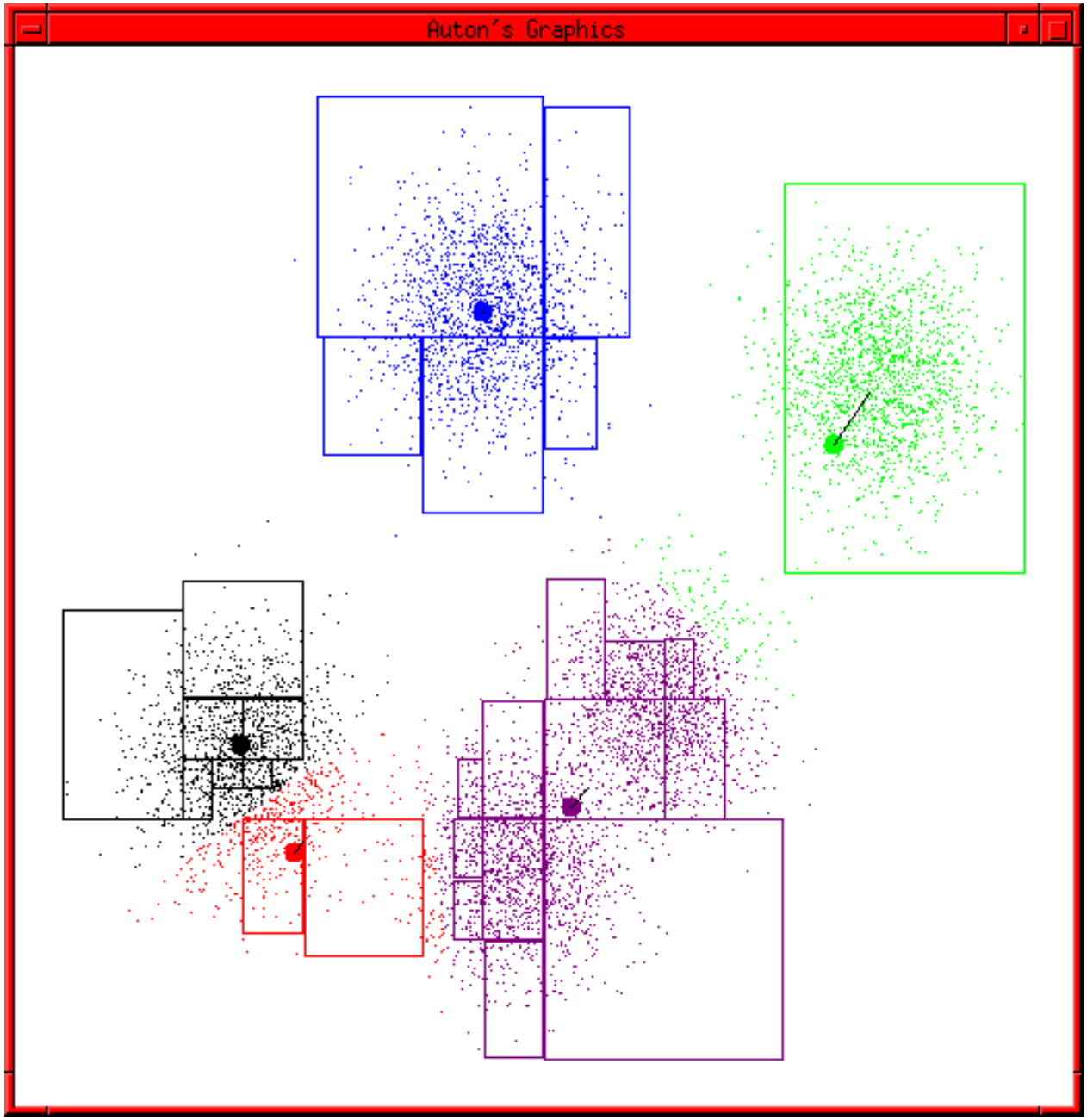
K-means

Here's an example run of k-means, generated by Dan Pelleg's software.



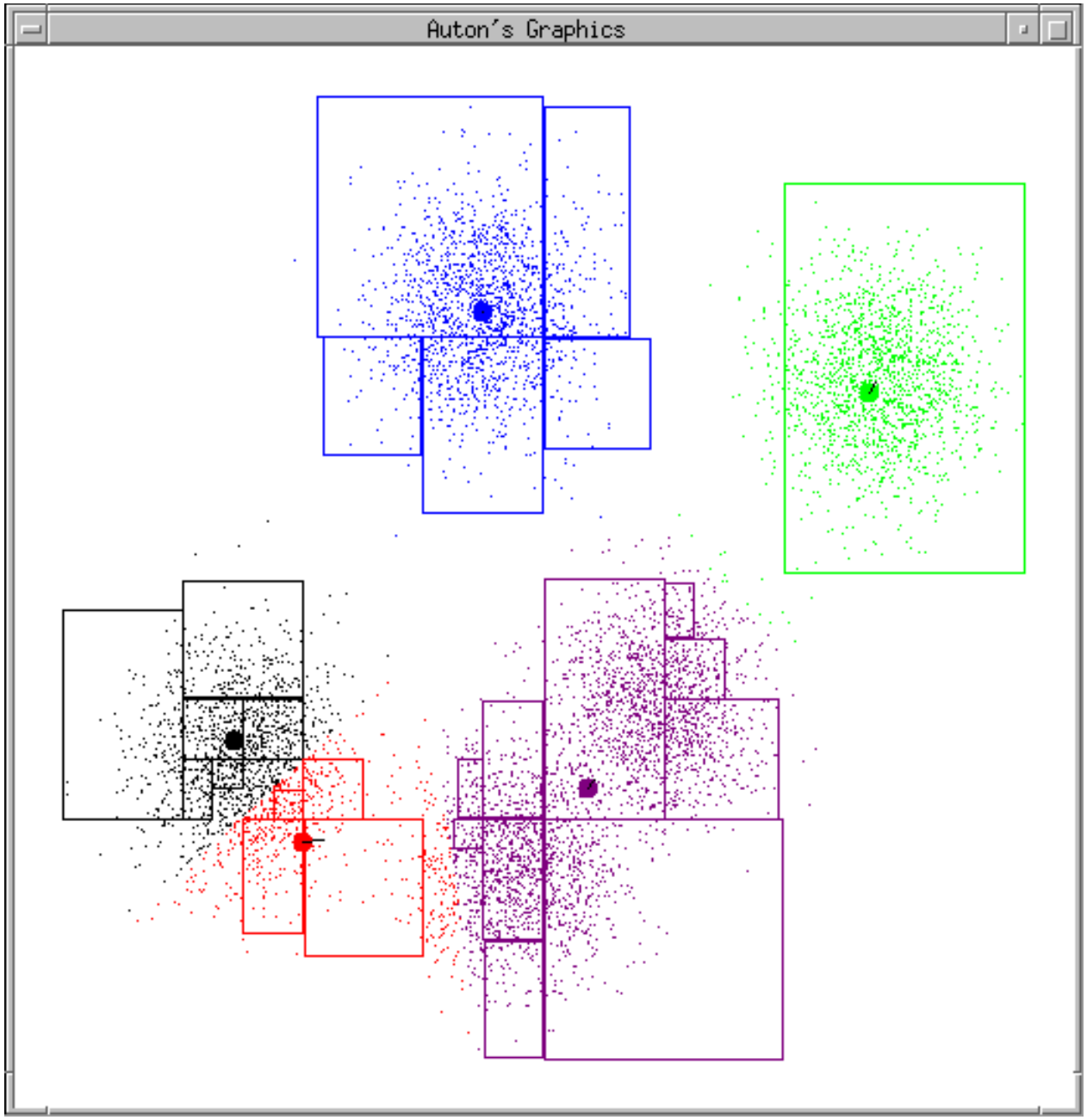
K-means

Here's an example run of k-means, generated by Dan Pelleg's software.



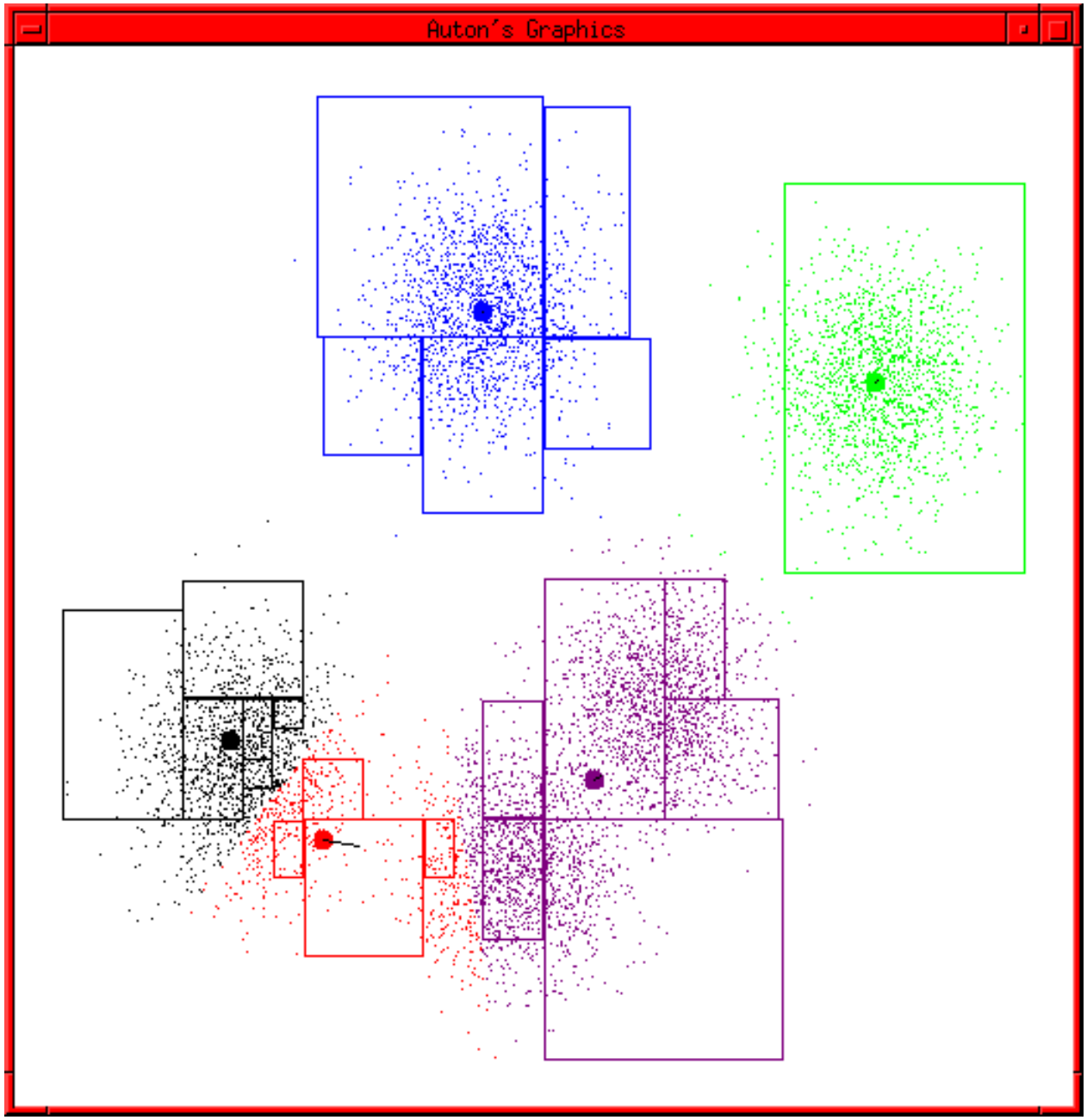
K-means

Here's an example run of k-means, generated by Dan Pelleg's software.



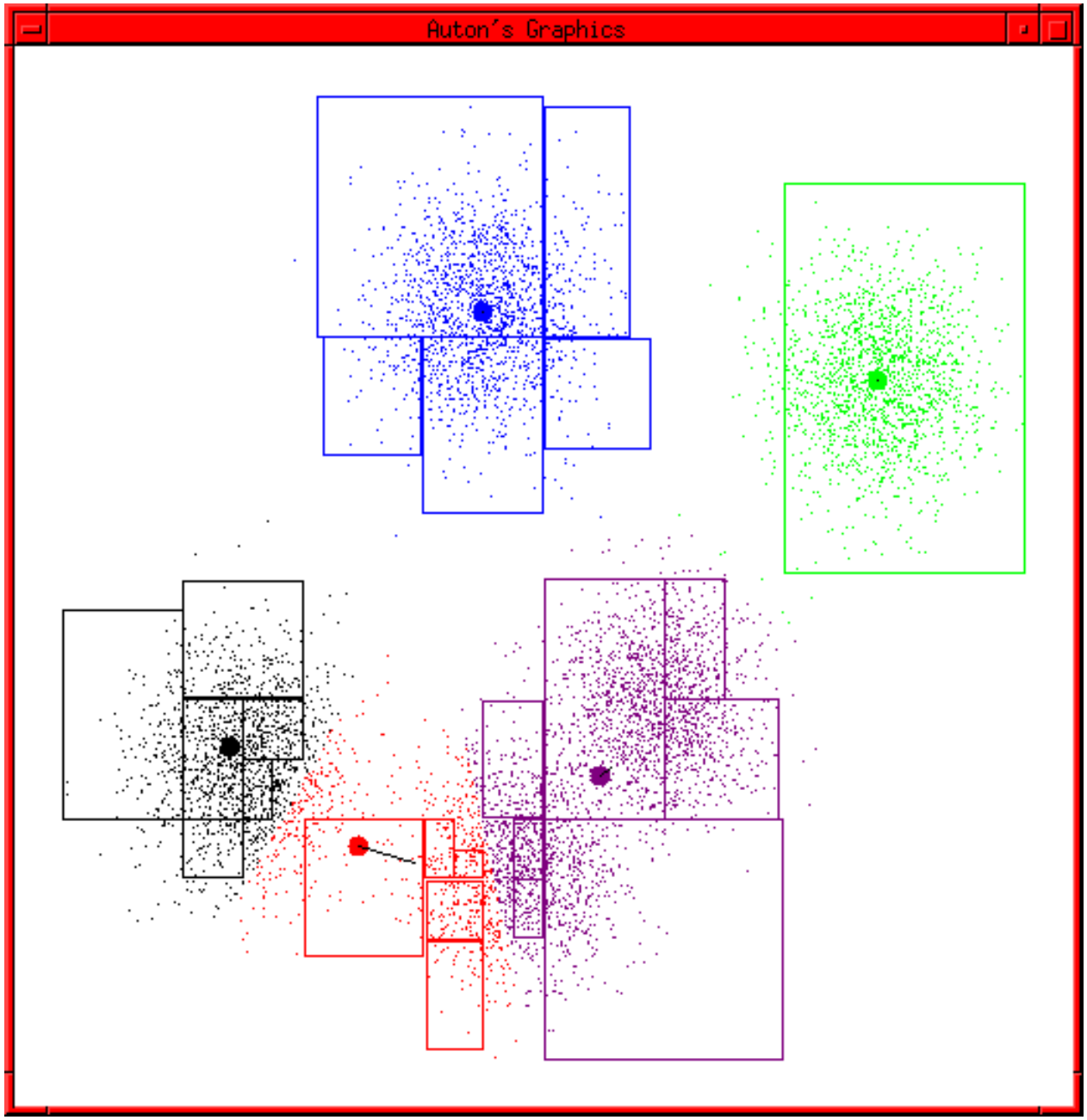
K-means

Here's an example run of k-means, generated by Dan Pelleg's software.



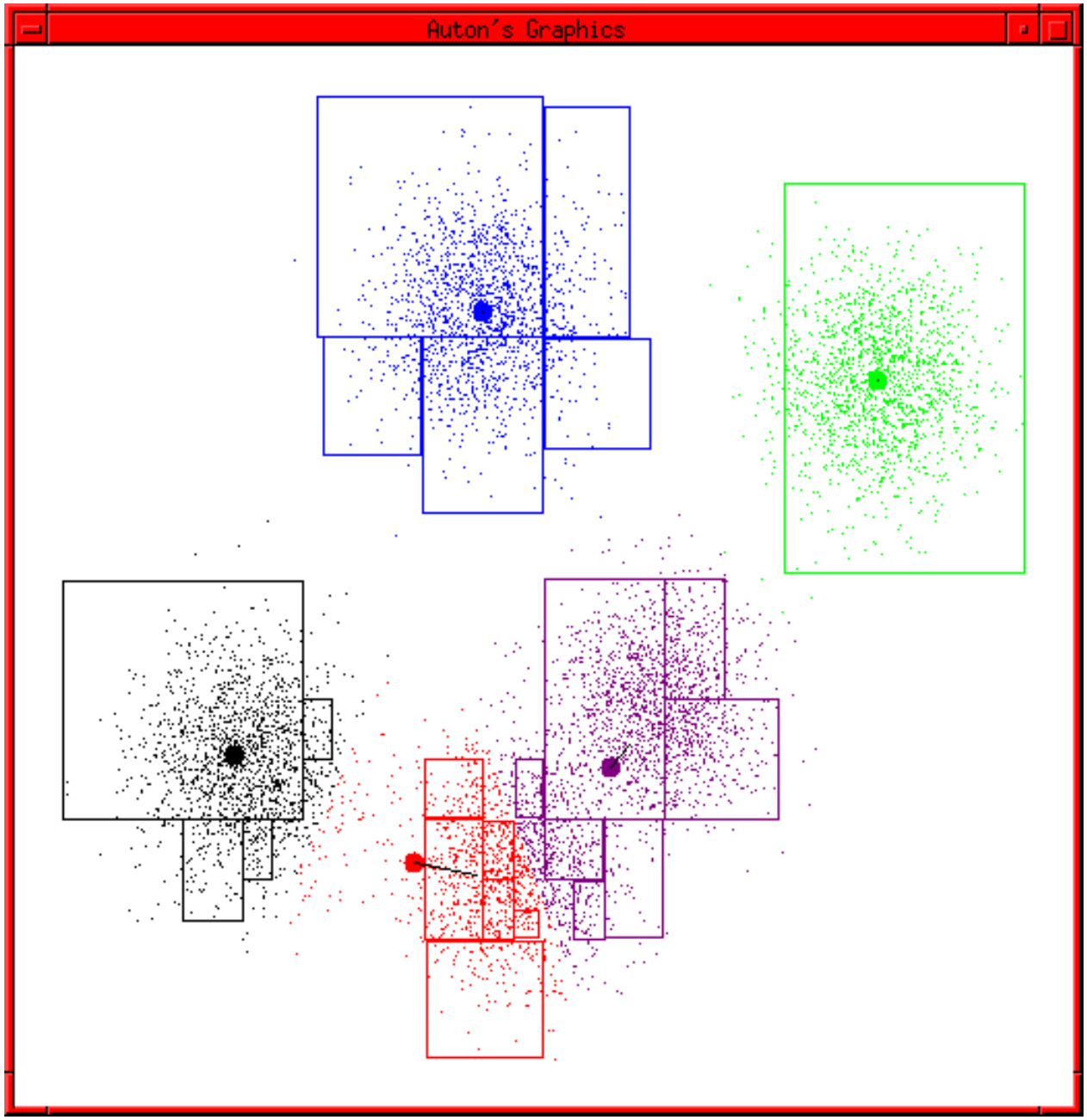
K-means

Here's an example run of k-means, generated by Dan Pelleg's software.



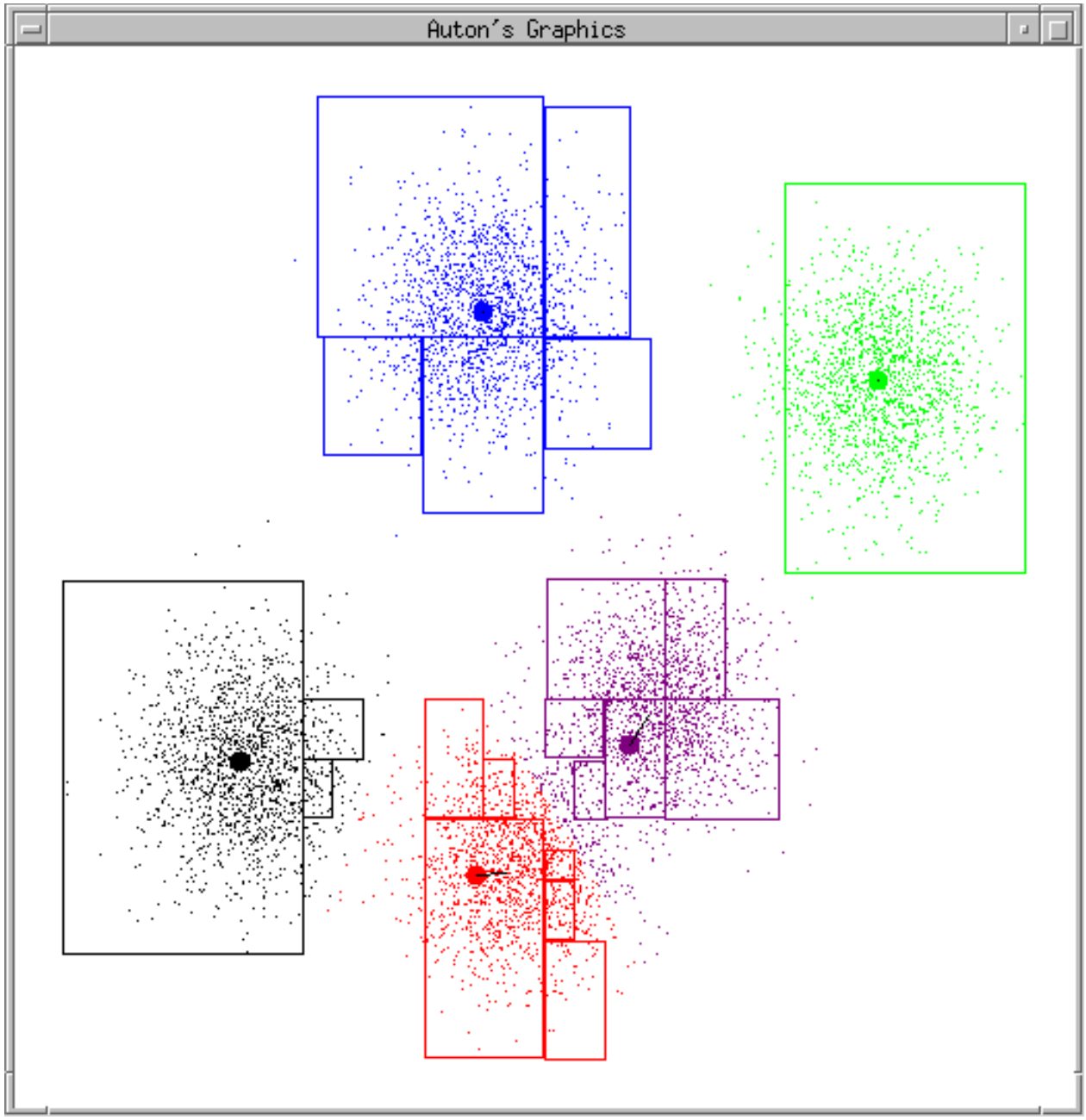
K-means

Here's an example run of k-means, generated by Dan Pelleg's software.



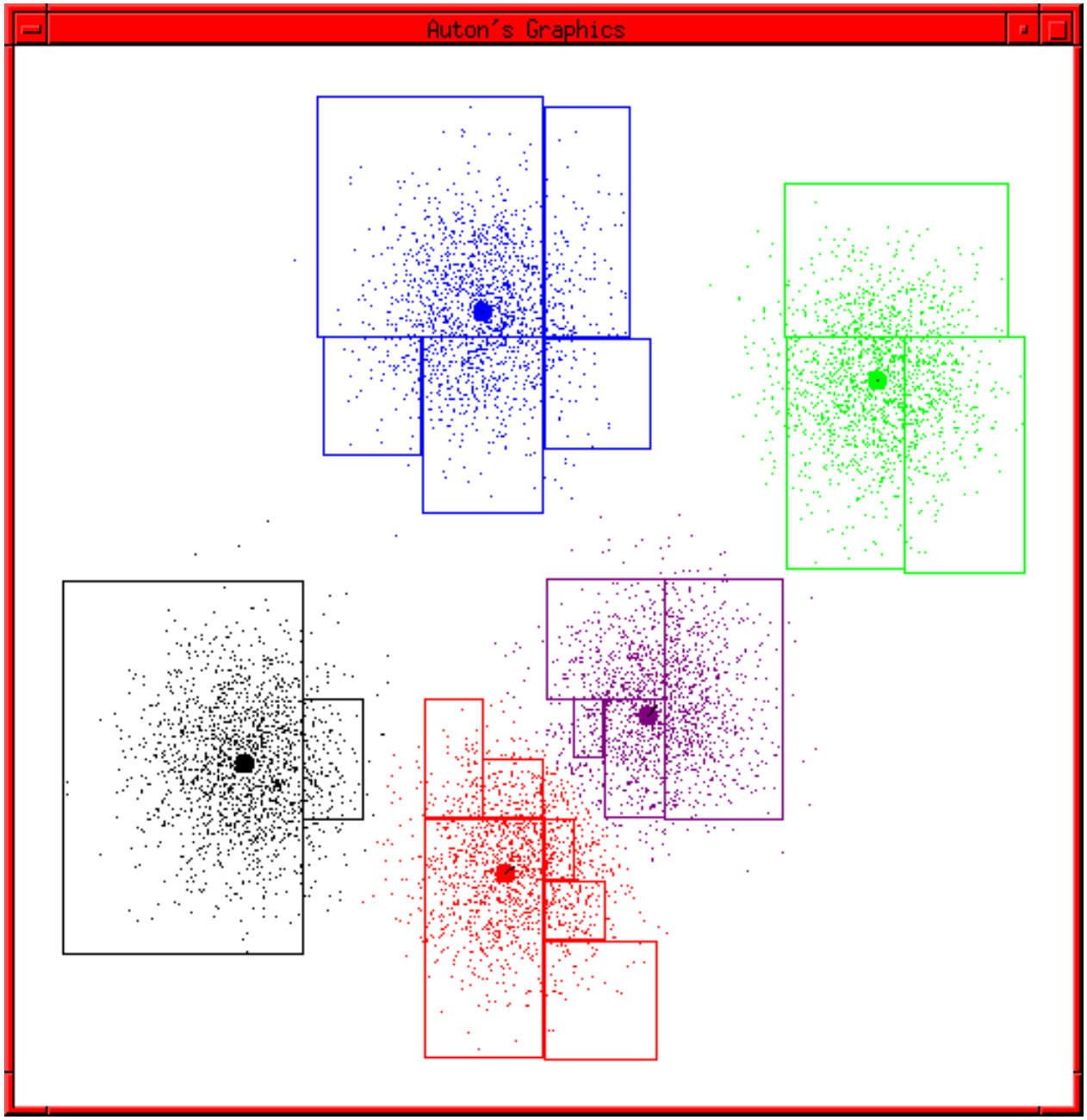
K-means

Here's an example run of k-means, generated by Dan Pelleg's software.



K-means

Here's the final result when the algorithm converges.



Brief on EM Algorithm

The Expectation-Maximization (EM) algorithm is a statistical technique that iteratively optimizes the likelihood function. It's particularly useful for estimating the parameters of a model when there are hidden or latent variables. The EM algorithm has two main steps, repeated until convergence:

Expectation Step (E-Step): Calculate the expected value of the hidden variables given the observed data and the current estimation of model parameters.

Maximization Step (M-Step): Update the parameters to maximize the expected likelihood found in the E-Step.

The EM algorithm iteratively performs these two steps until the log-likelihood of the observed data plateaus.

K-Means as a Special Case of EM

K-Means clustering can be viewed as a specific instance of the EM algorithm. In K-Means, the goal is to partition your data into

K clusters, each described by the mean of the data points belonging to the cluster. Here's how the K-Means algorithm can be interpreted in terms of the EM framework:

Initialization:
Choose K initial cluster centers.

E-Step:
Assign each data point to the nearest cluster center. Here, the "expectation" is assigning data points to clusters based on the current means (cluster centers). The hidden variable here is the cluster assignment of each data point, which is not observed in the data.

M-Step:
Update the cluster centers (means) by computing the mean of all points assigned to each cluster center. This step "maximizes" the likelihood of the assigned data under the assumption that the data in each cluster comes from a Gaussian distribution centered at the mean.

Repeat:
Continue iterating between the E and M steps until the cluster assignments no longer change or change very little.

K-means clustering

Question 1: Will k-means always converge to a solution?

Yes! Distortion decreases monotonically, and it will end in a state where neither type of move will further reduce the distortion.

Question 2: Will k-means always find the optimal solution?

No! Here is one example where k-means converges to a locally optimal solution that does not have the global minimum distortion.

K-means clustering

Question 1: Will k-means always converge to a solution?

Yes! Distortion decreases monotonically, and it will end in a state where neither type of move will further reduce the distortion.

Question 2: Will k-means always find the optimal solution?

No! Here is one example where k-means converges to a locally optimal solution that does not have the global minimum distortion.

Question 3: How can we avoid these poor local optima?

K-means is a form of hill-climbing, so we can use many of the same tricks!

1. Run multiple times with different start states, and choose the best result.
2. Allow some moves that increase distortion, as in simulated annealing.
3. Choose a start state that is less likely to result in a poor local optimum.

Center 1 = randomly chosen data point
Center 2 = data point that's farthest from center 1
Center 3 = data point that's farthest from the closest of centers 1 and 2, etc.

References

- Scikit-learn clustering documentation:
  http://scikit-learn.org/stable/modules/clustering.html
- C.C. Aggarwal and C.K. Reddy, eds. *Data Clustering: Algorithms and Applications*, 2014.
  http://www.crcnetbase.com/doi/book/10.1201/b15410
- A.K. Jain et al. Data clustering: a review. *ACM Computing Surveys* 31(3), 1999.
- The Auton Laboratory (www.autonlab.org) has very fast k-means (and X-means) software, created by D. Pelleg.