Fundamentals of AI

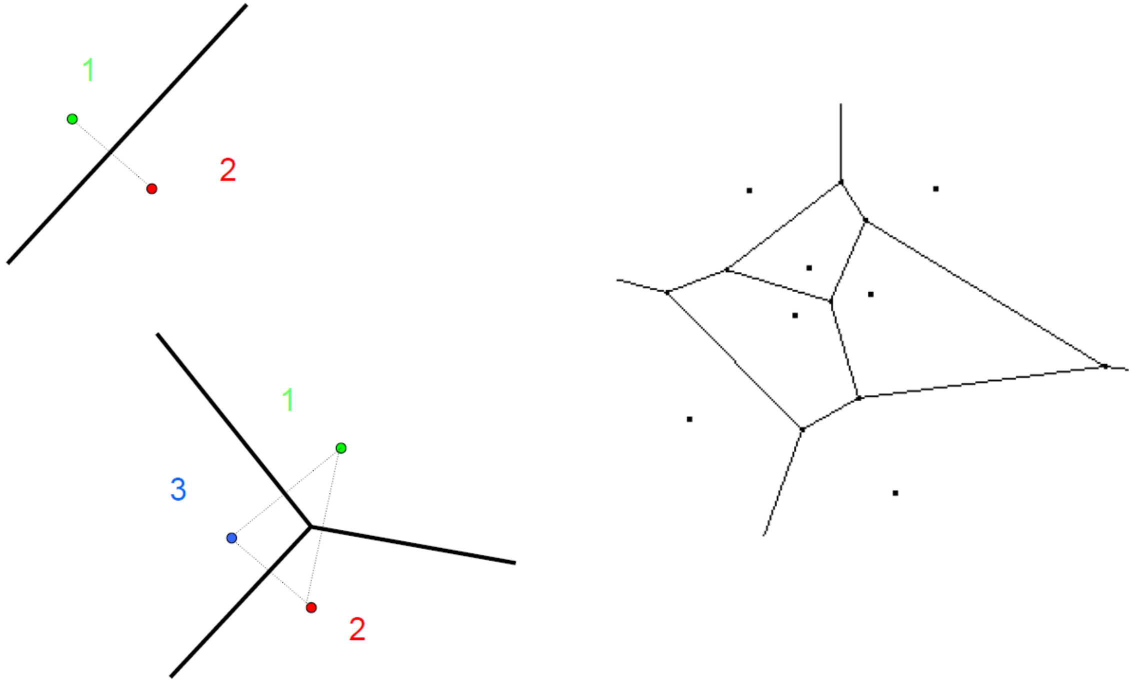
Clustering

K-means, the oldest clustering algorithm

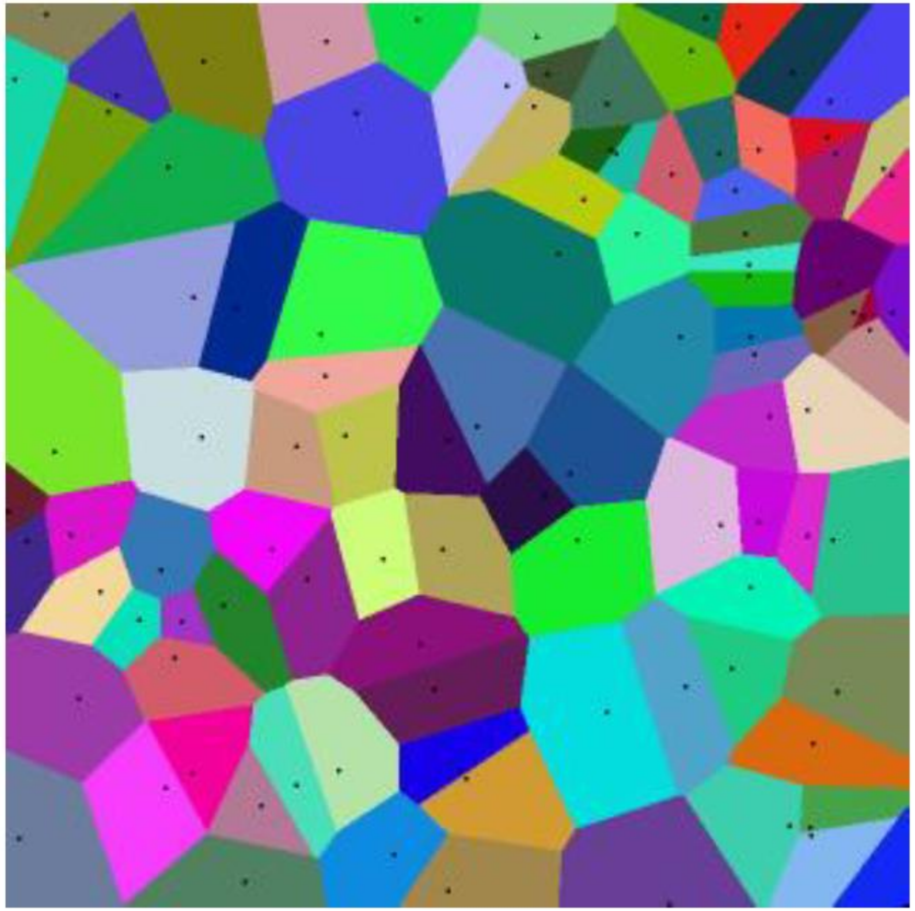
MacQueen, 1967; Steinhaus, 1956; Loyd, 1957

Let us introduce a simple notion: Voronoi cell

Examples of Voronoi decomposition. 1

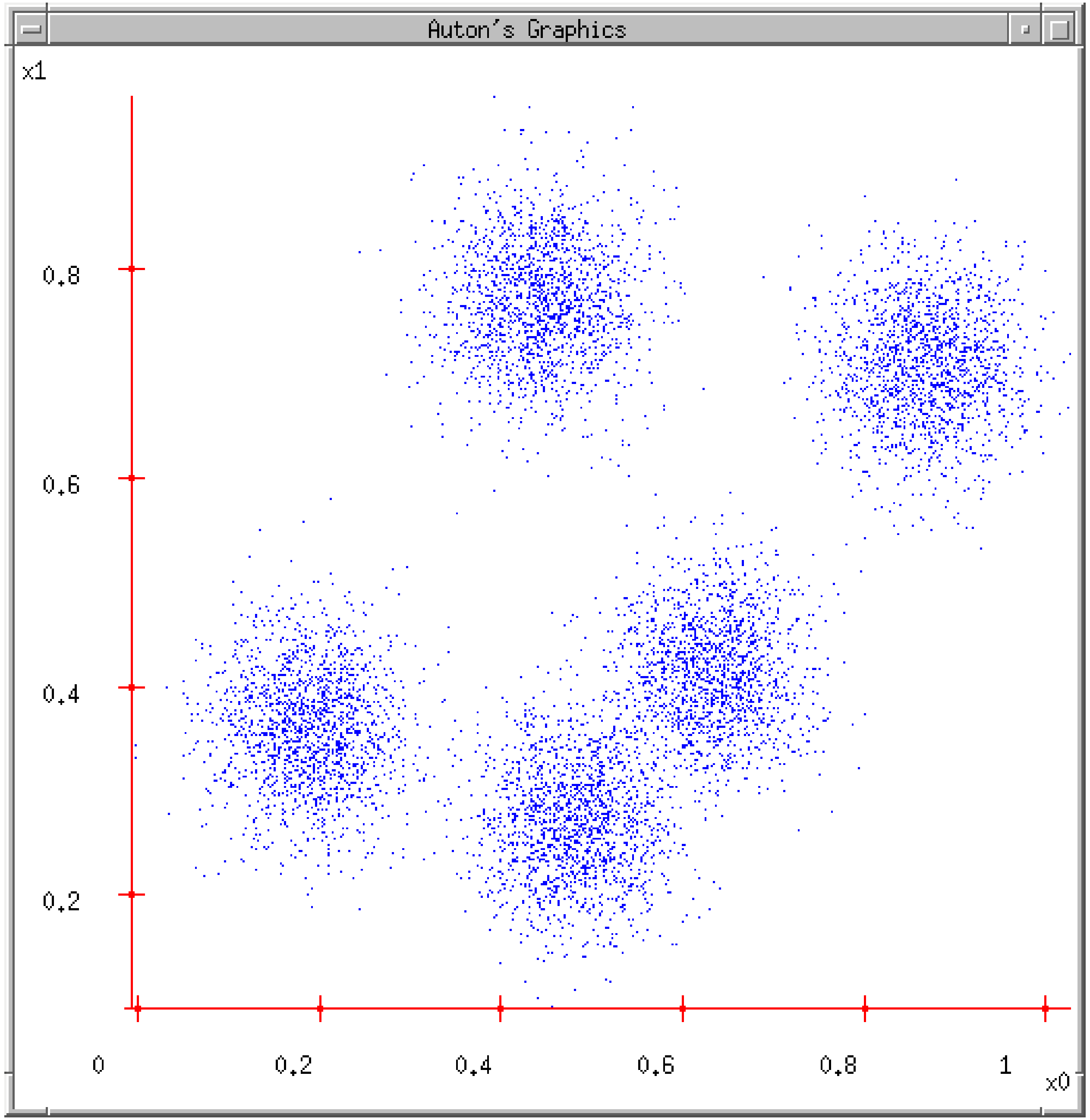


Examples of Voronoi decomposition. 2



K-means

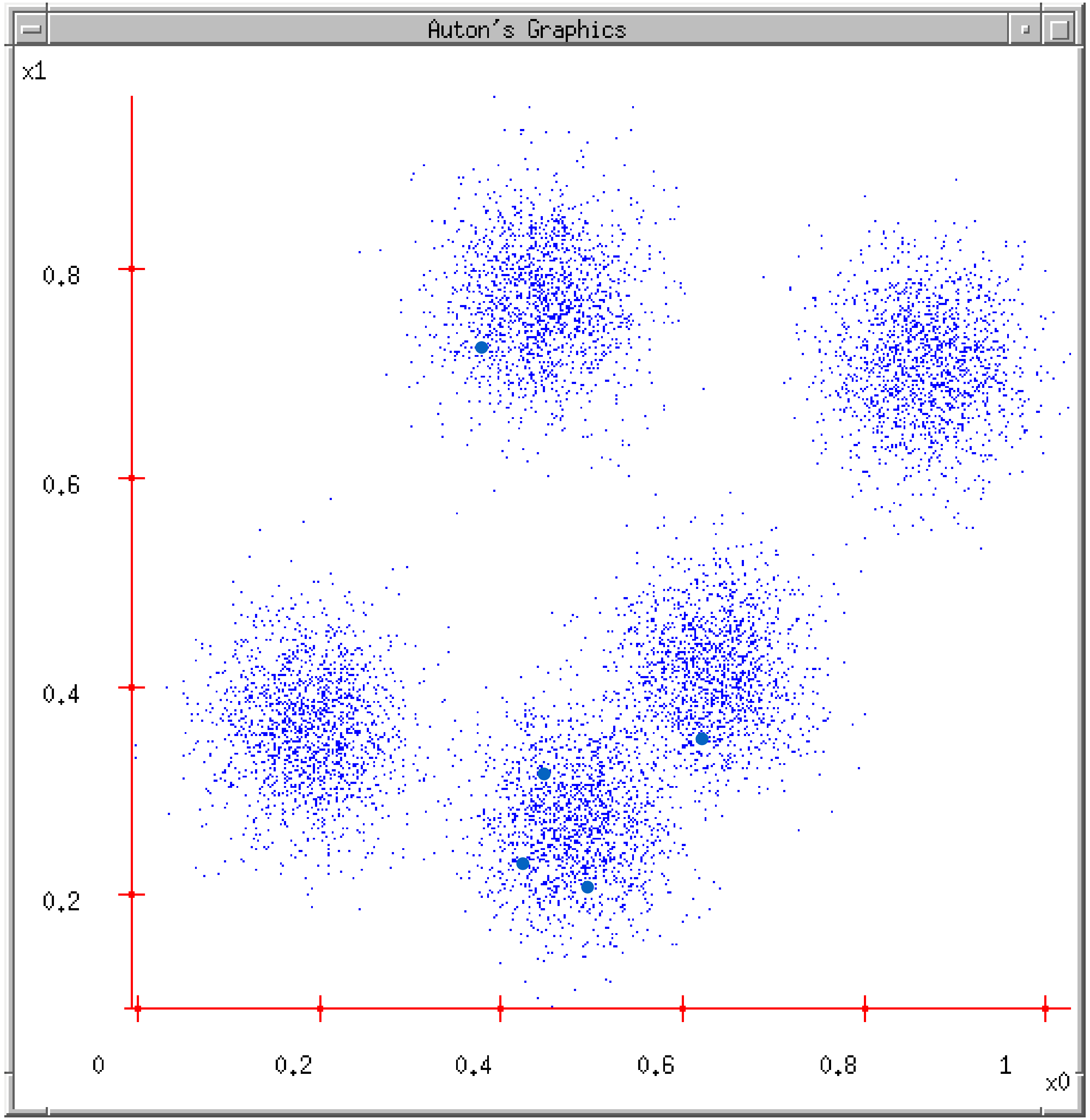
1. Ask user how many clusters they'd like. *(e.g. k=5)*



K-means

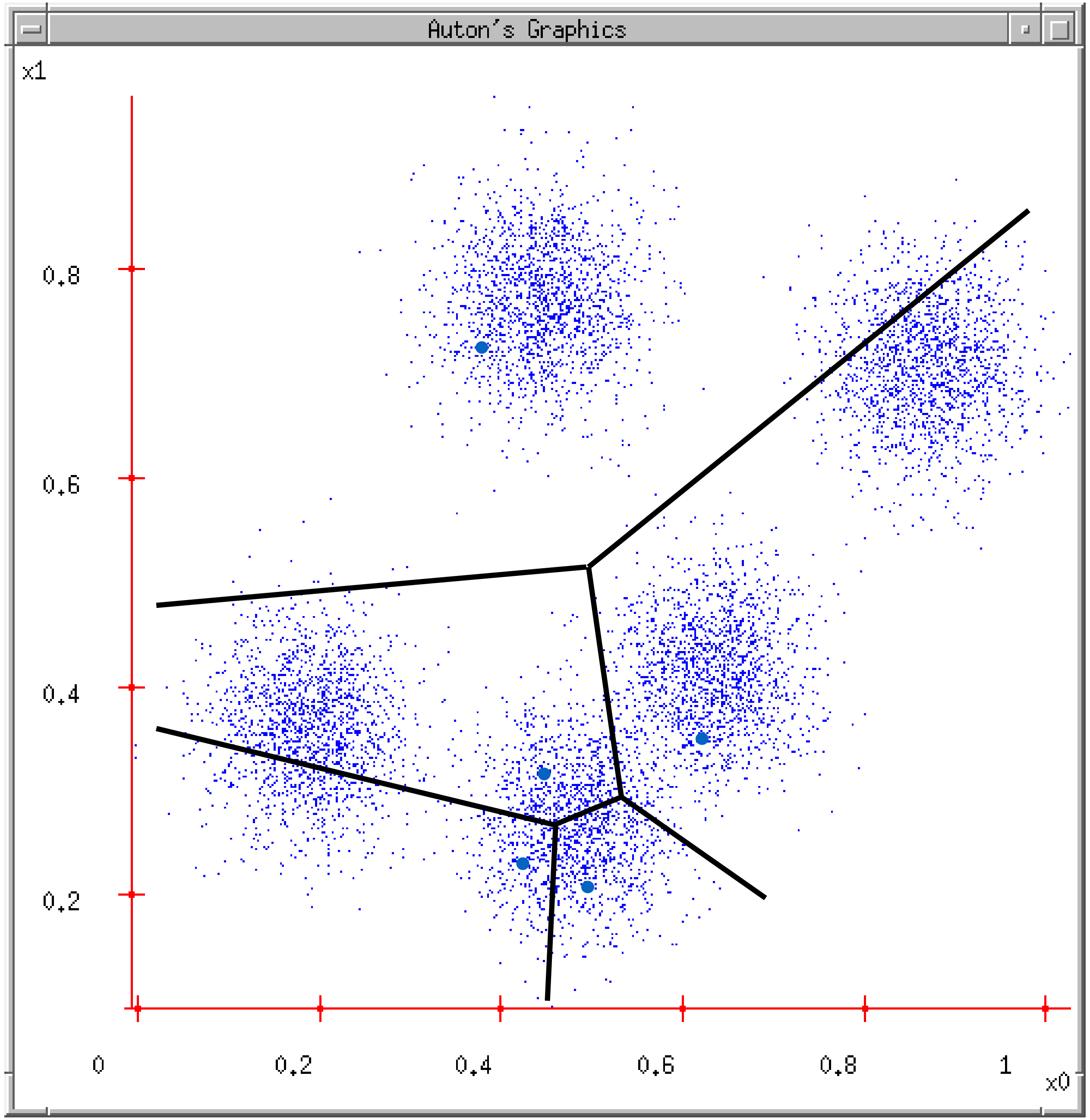
1. Ask user how many clusters they'd like. *(e.g. k=5)*

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)

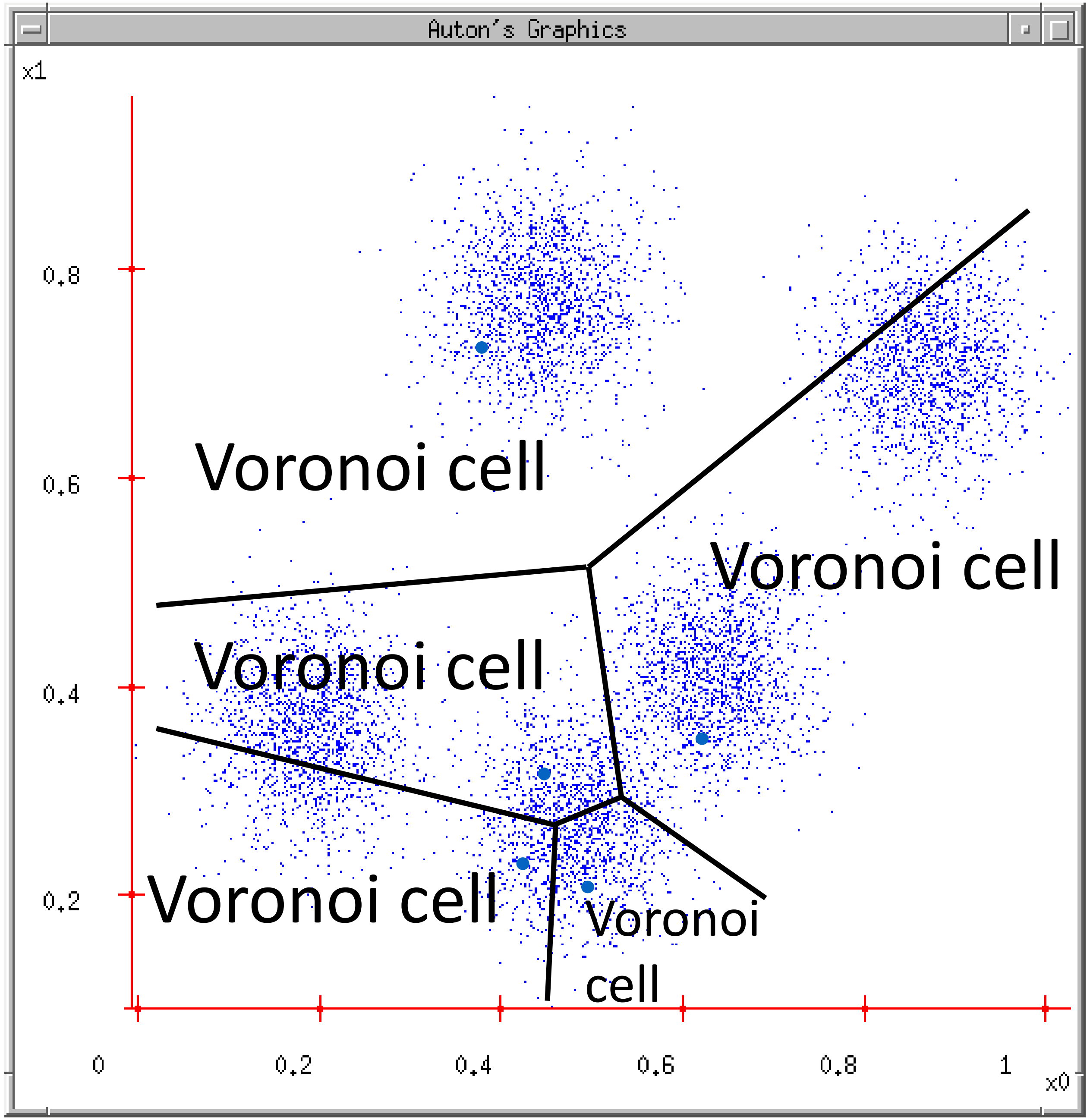


K-means

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

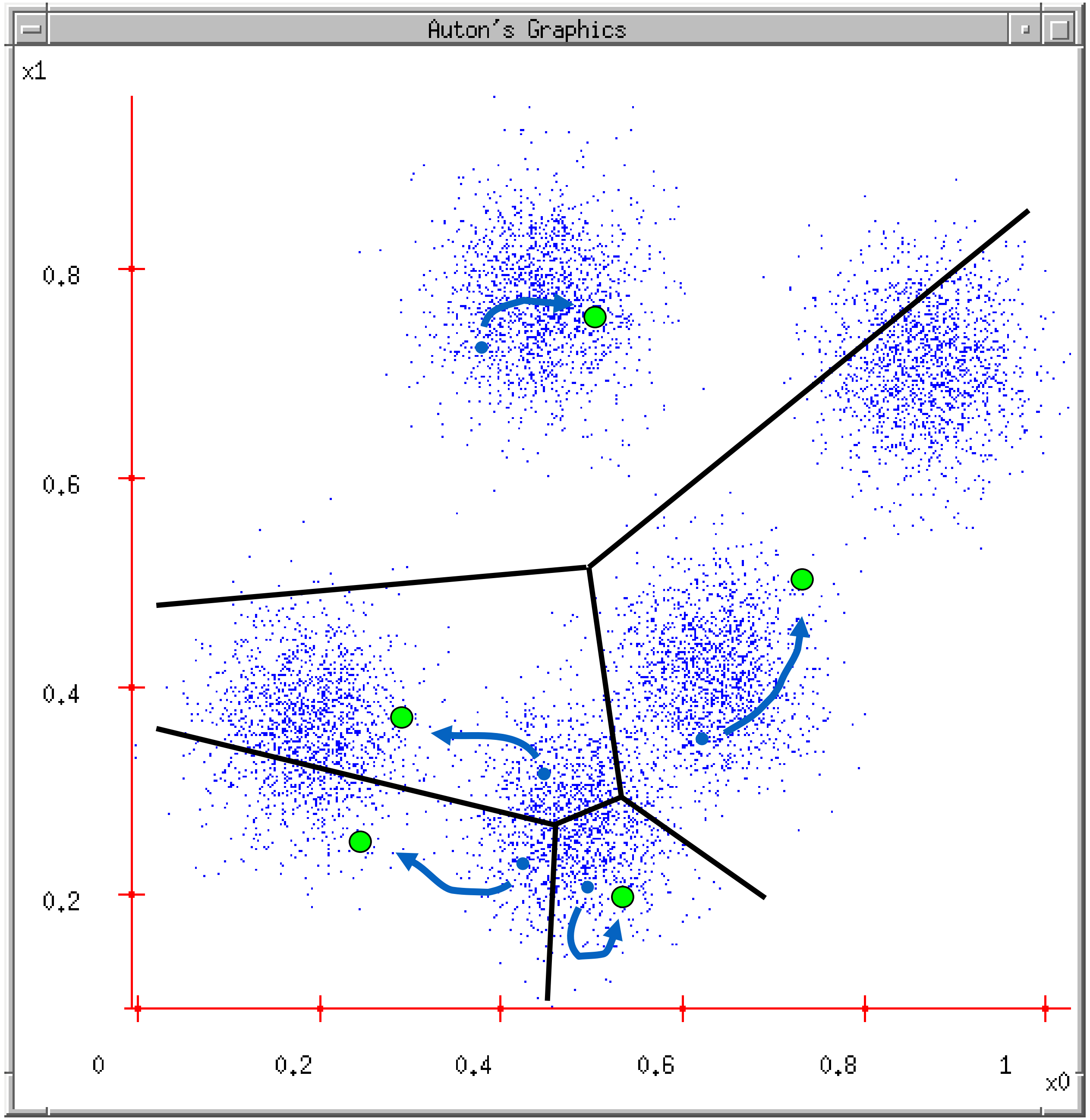
2. Randomly guess k cluster Center locations

3. Each datapoint finds out which Center it's closest to. (Thus each Center "owns" a set of datapoints)



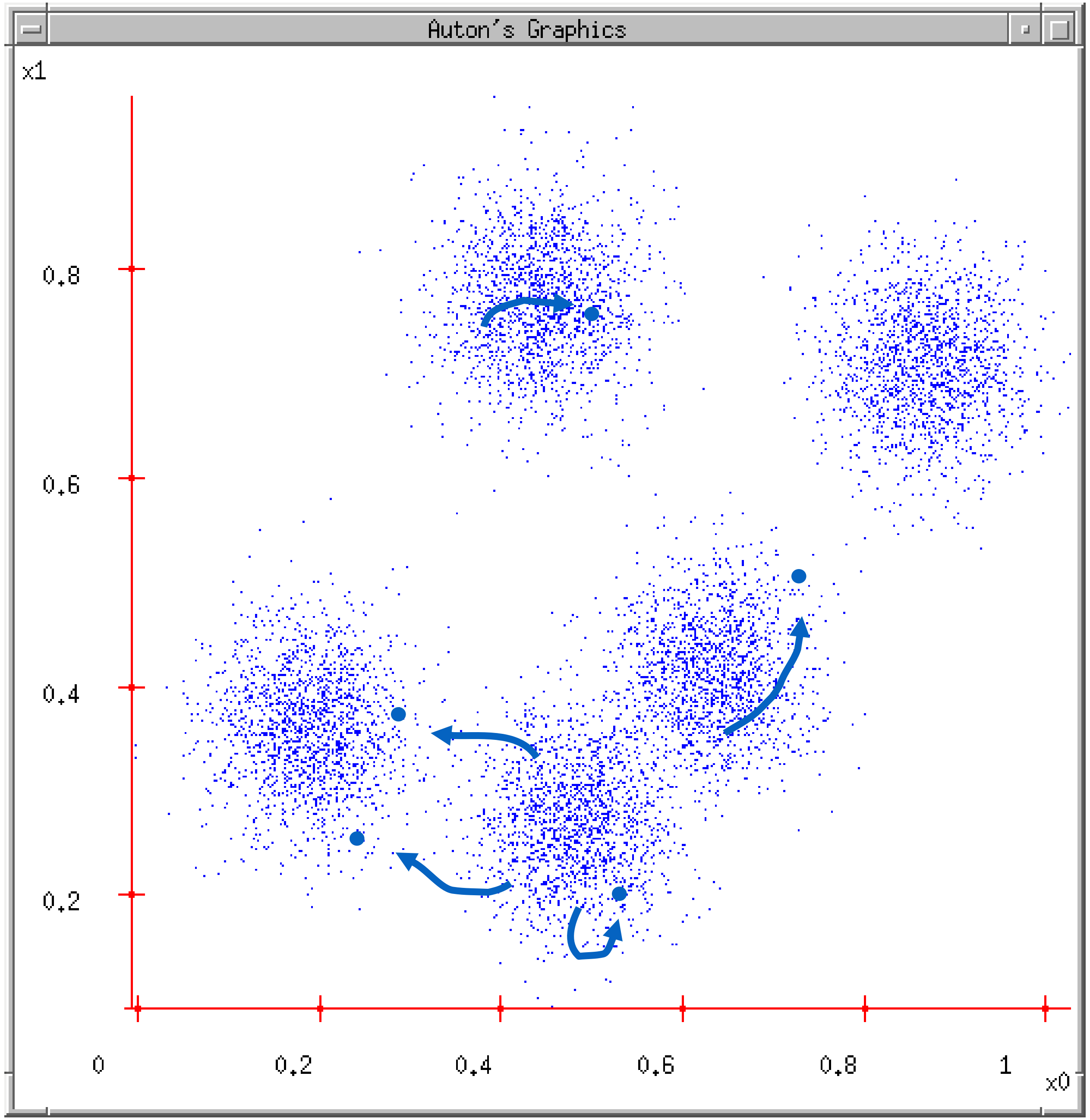
K-means

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. …and jumps there
6. …Repeat until terminated!

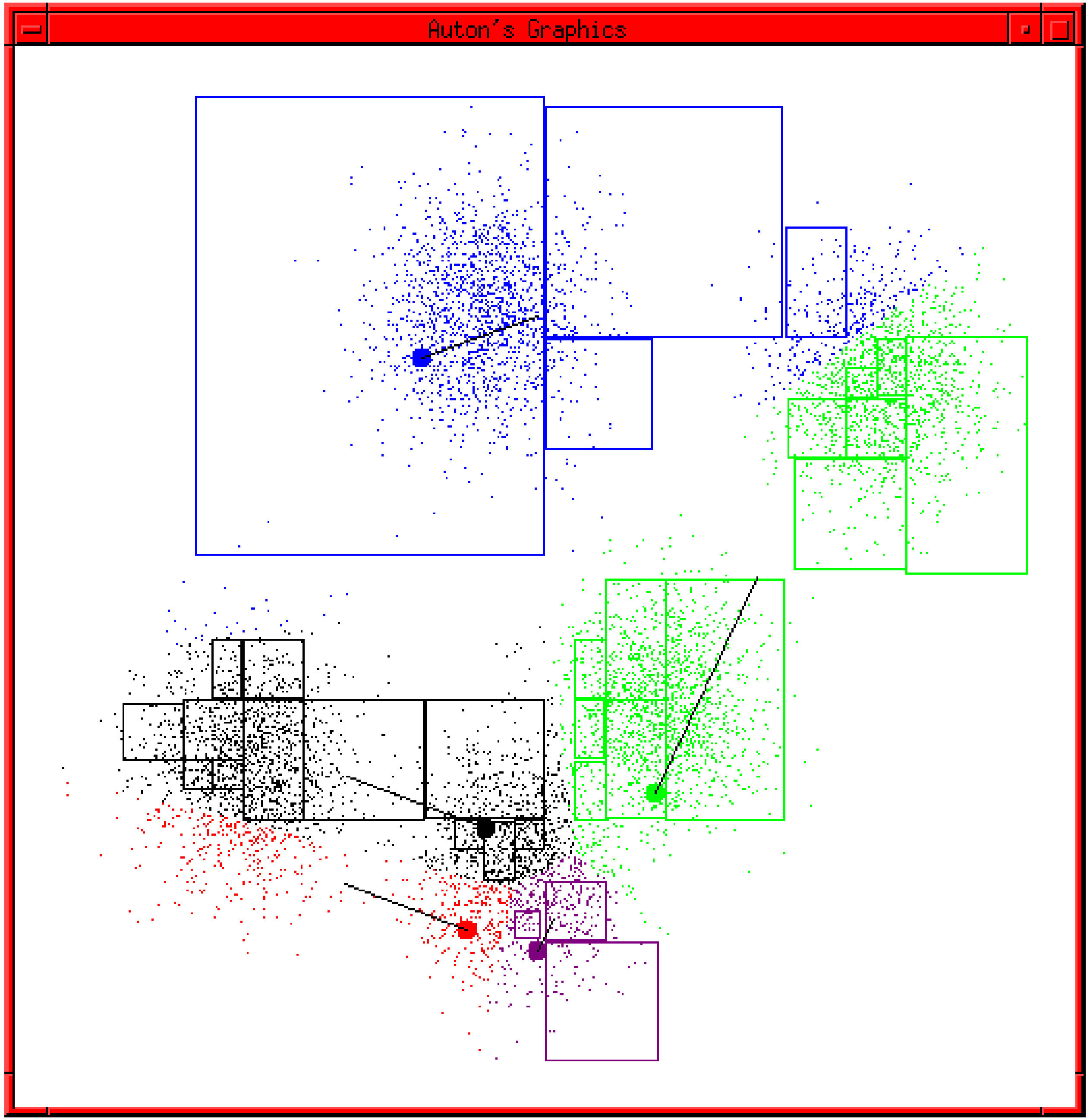


K-means Start

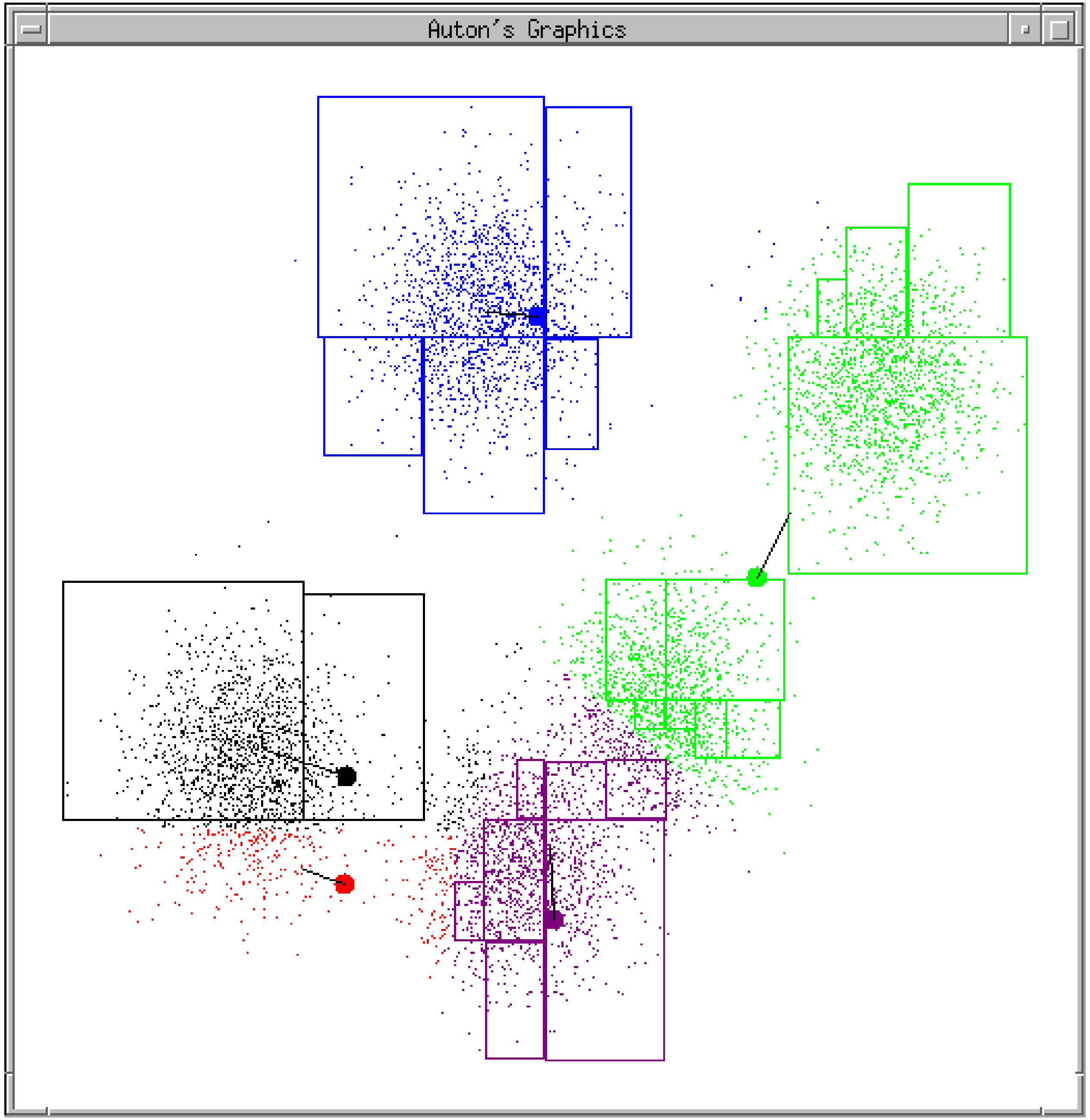
Advance apologies: in Black and White this example will deteriorate

Example generated by Dan Pelleg's super-duper fast K-means system:

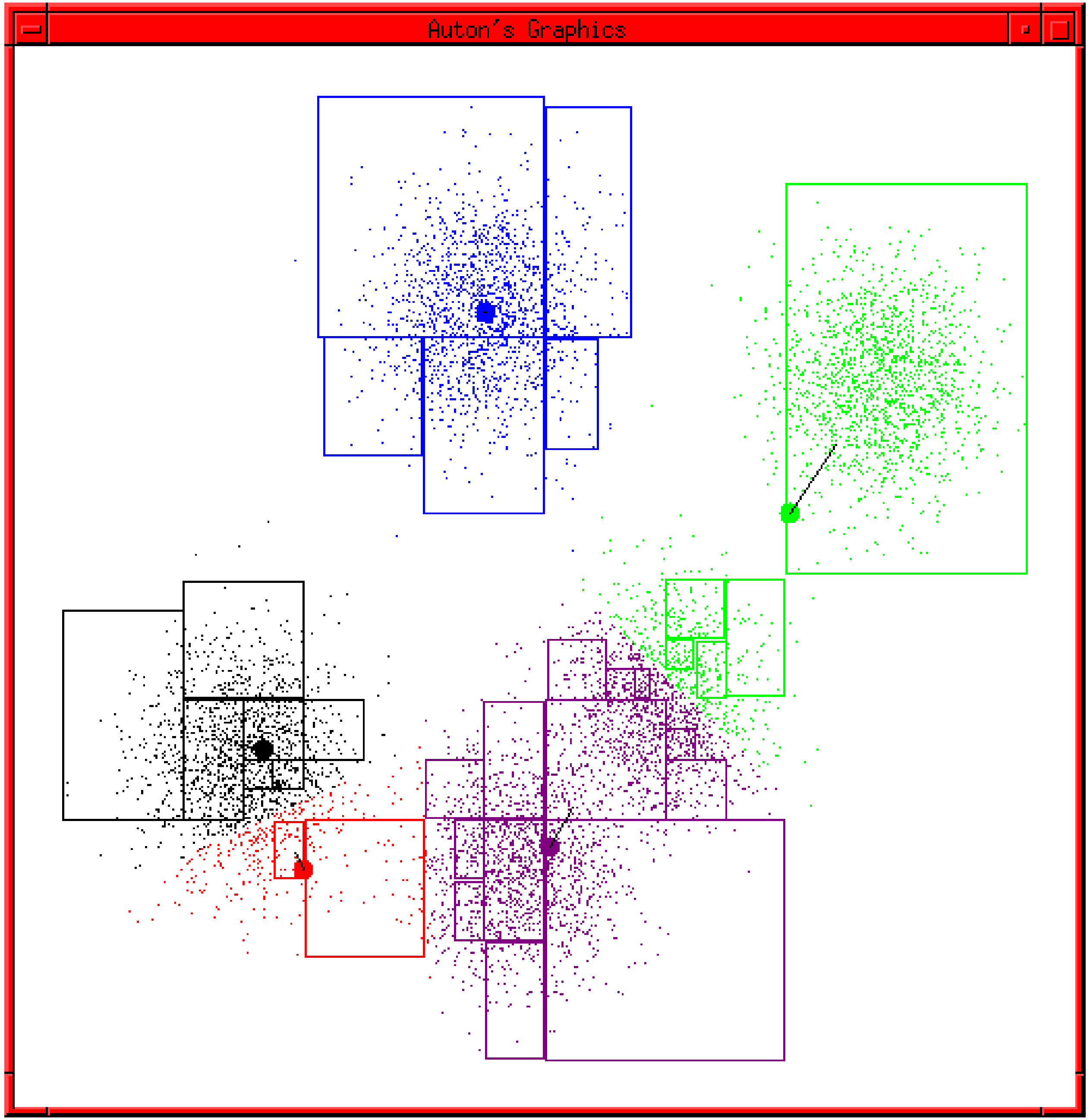
Dan Pelleg and Andrew Moore. Accelerating Exact k-means Algorithms with Geometric Reasoning. Proc. Conference on Knowledge Discovery in Databases 1999, (KDD99) (available on www.autonlab.org/pap.html*)*



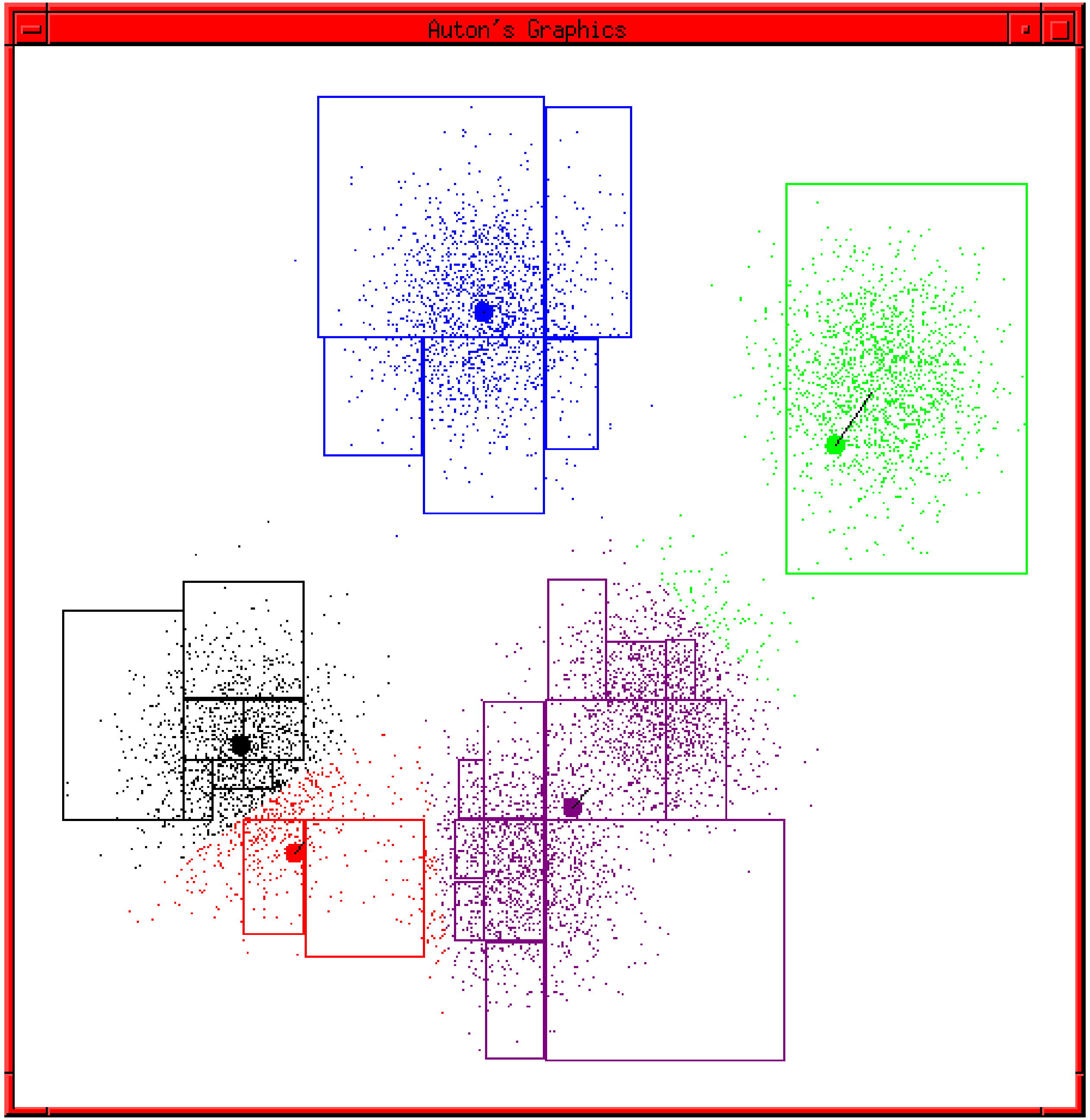
K-means continues …



K-means continues …

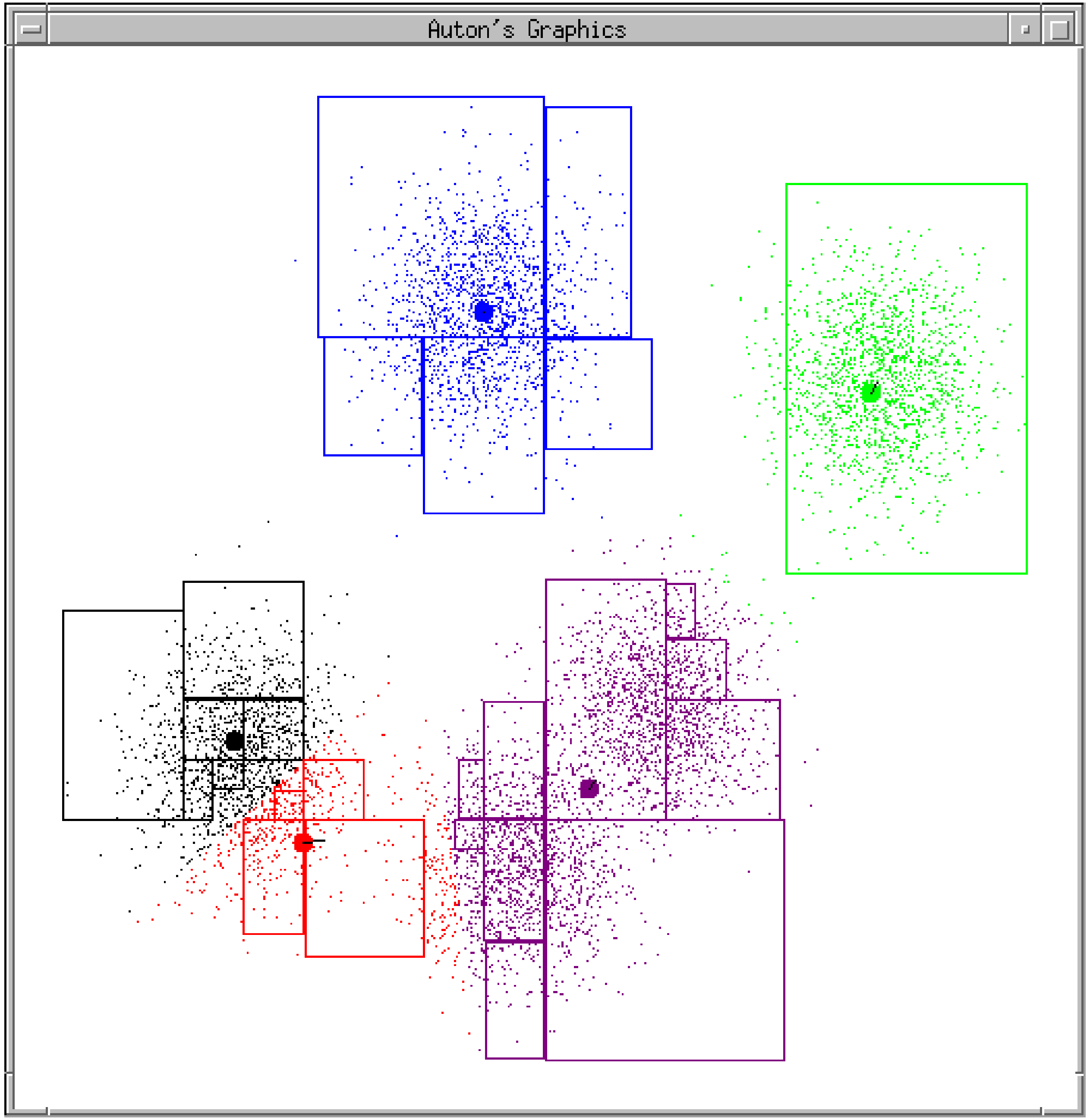


K-means continues …

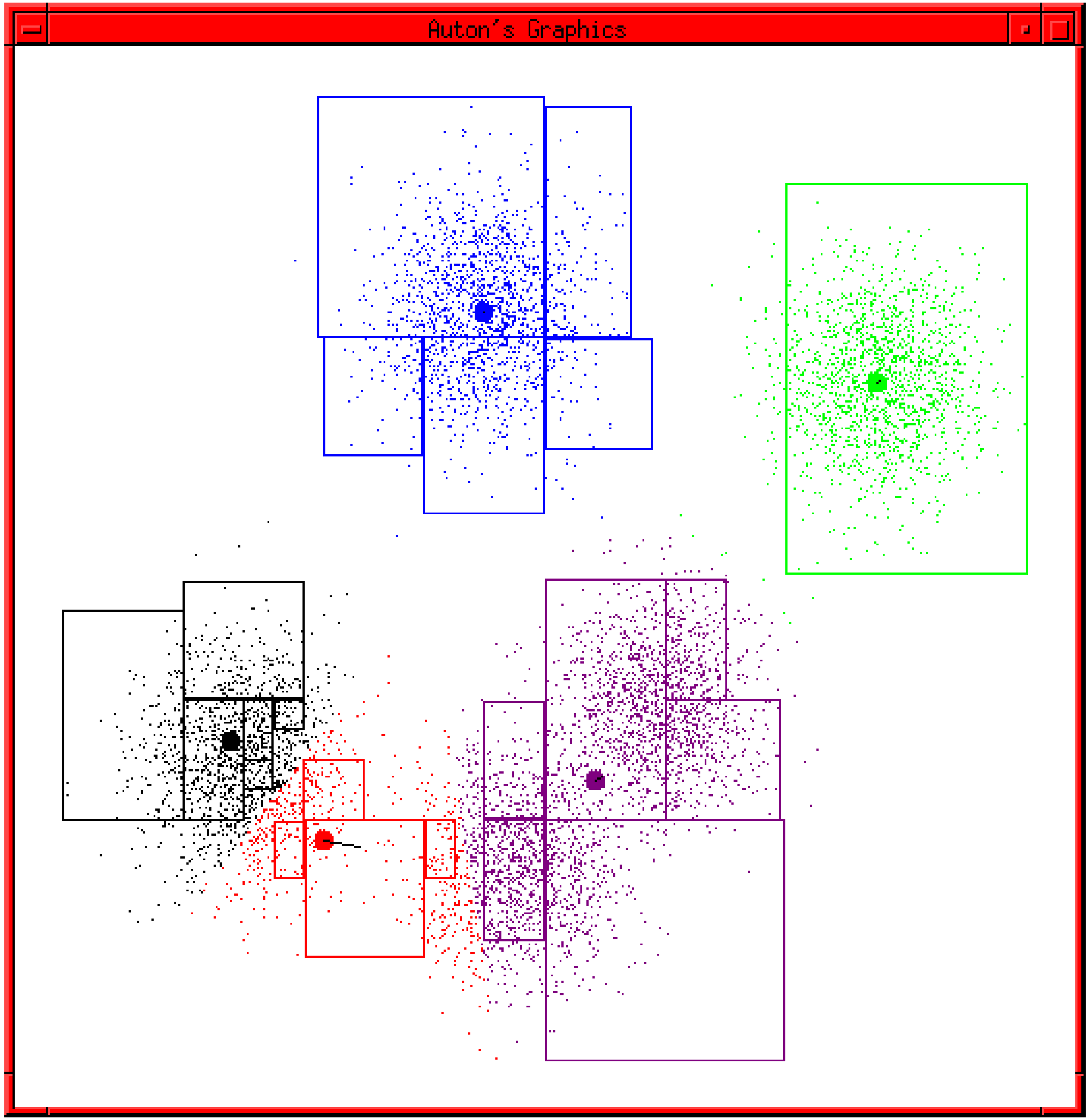


Copyright © 2001, 2004, Andrew W. Moore

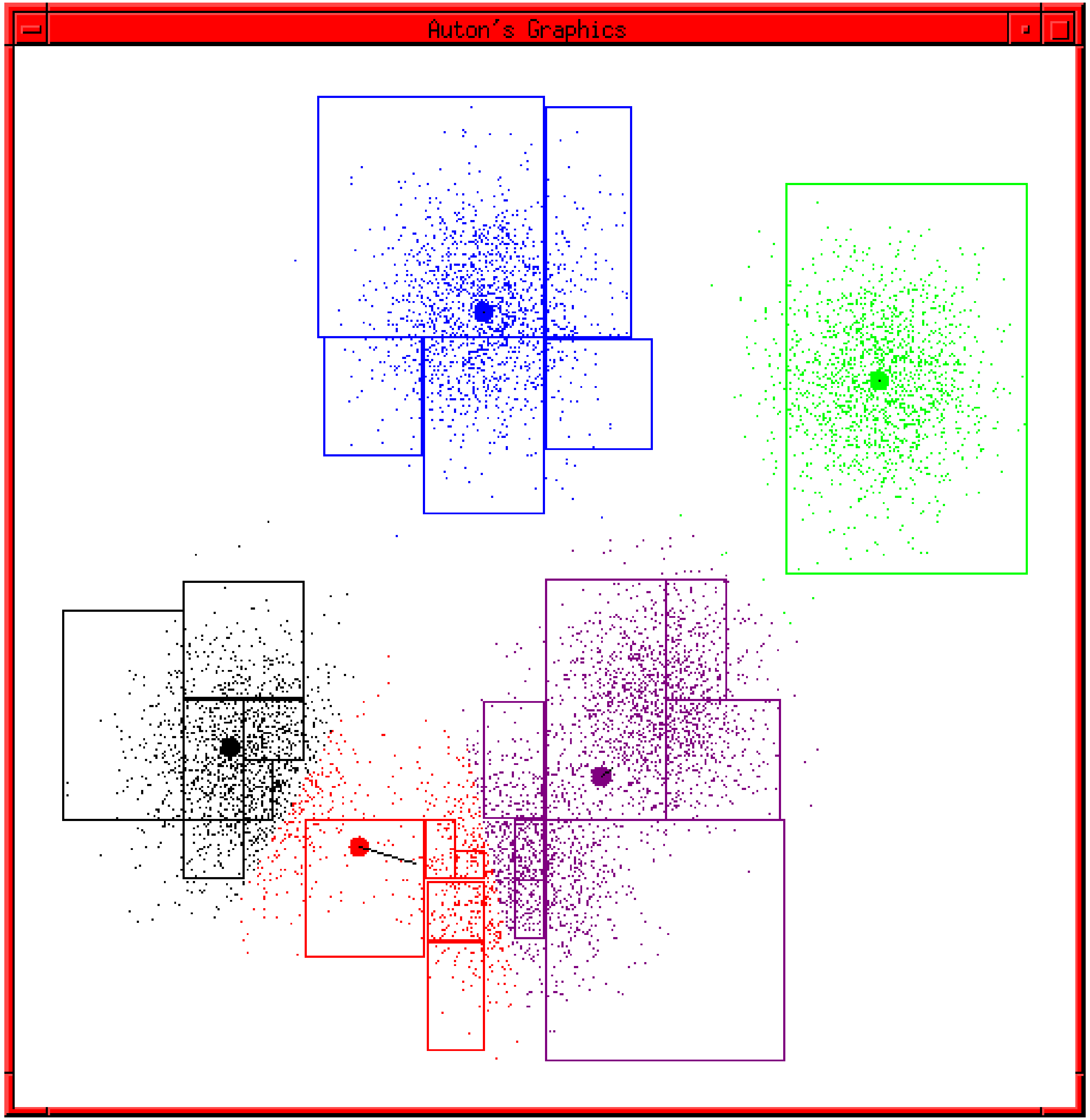
K-means continues …



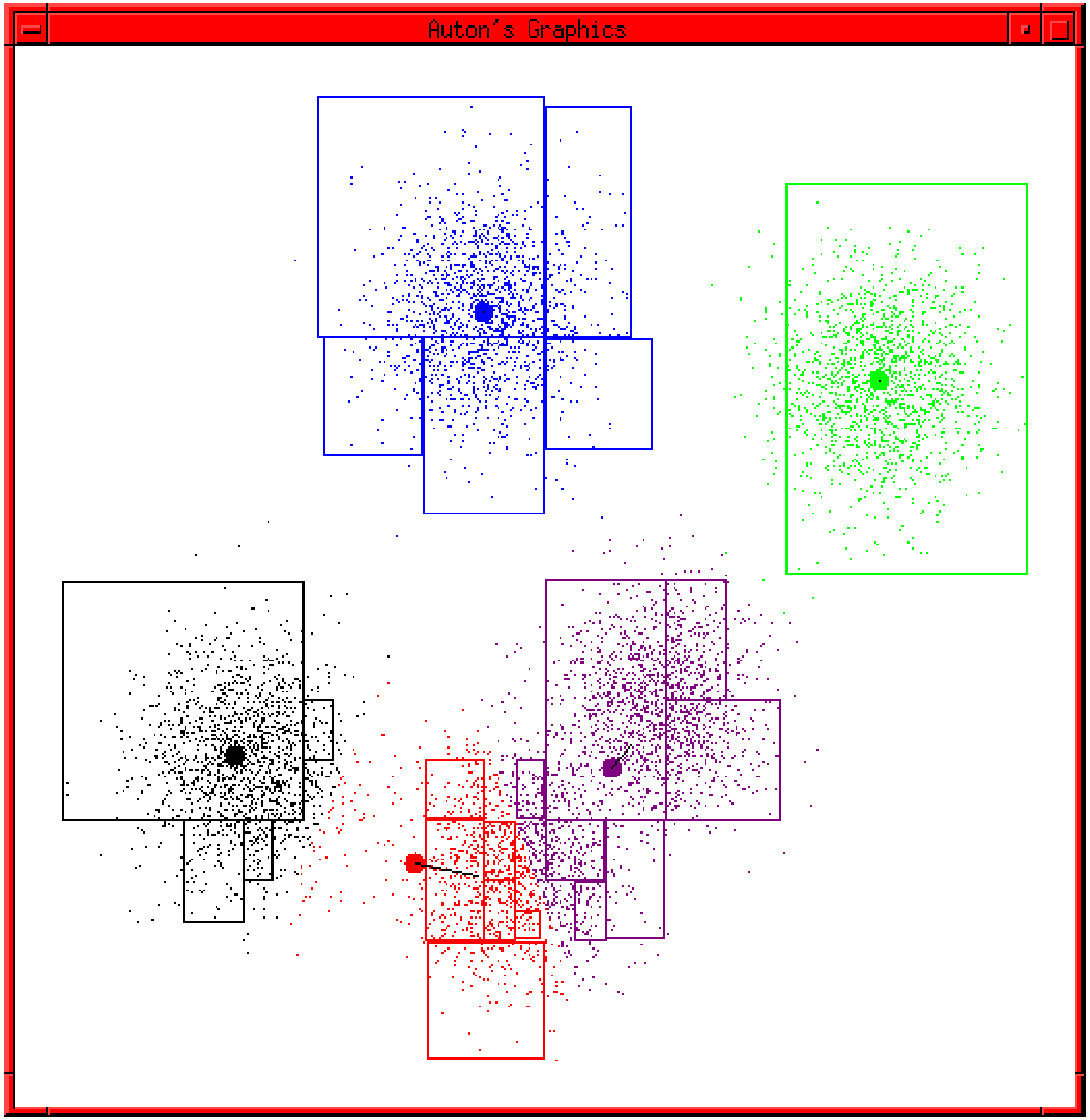
K-means continues …



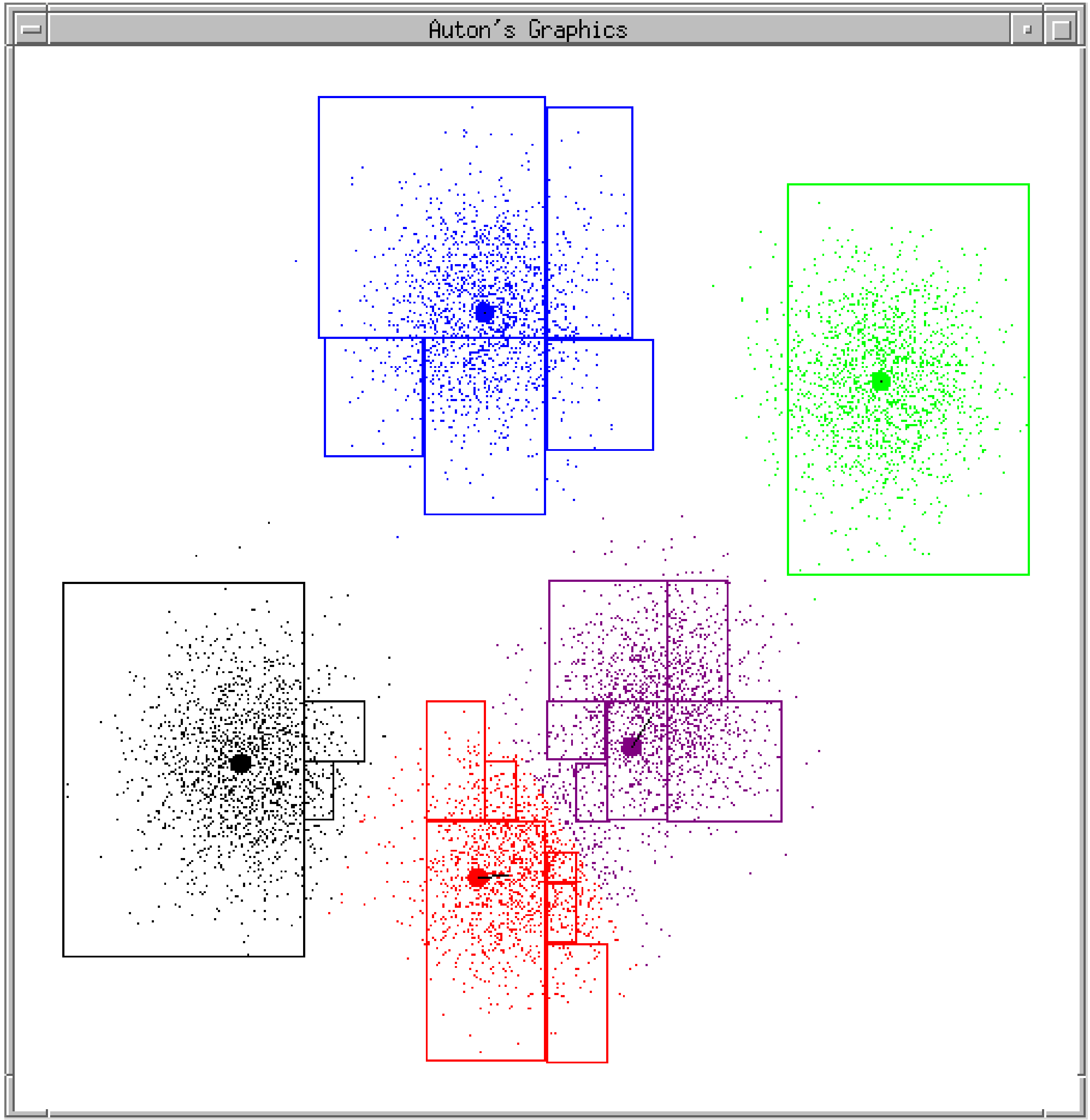
K-means continues …



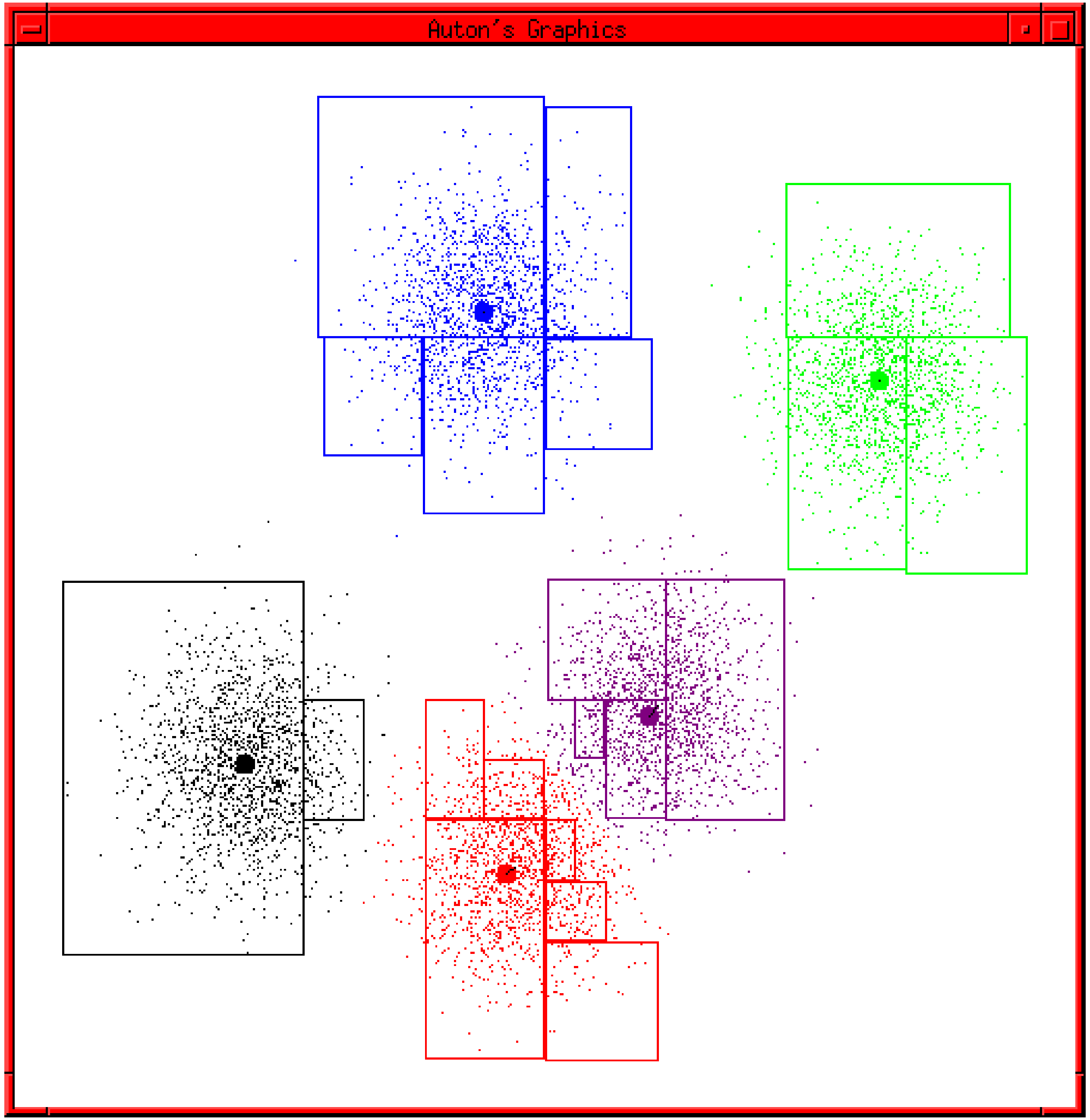
K-means continues …



K-means continues …



K-means terminates

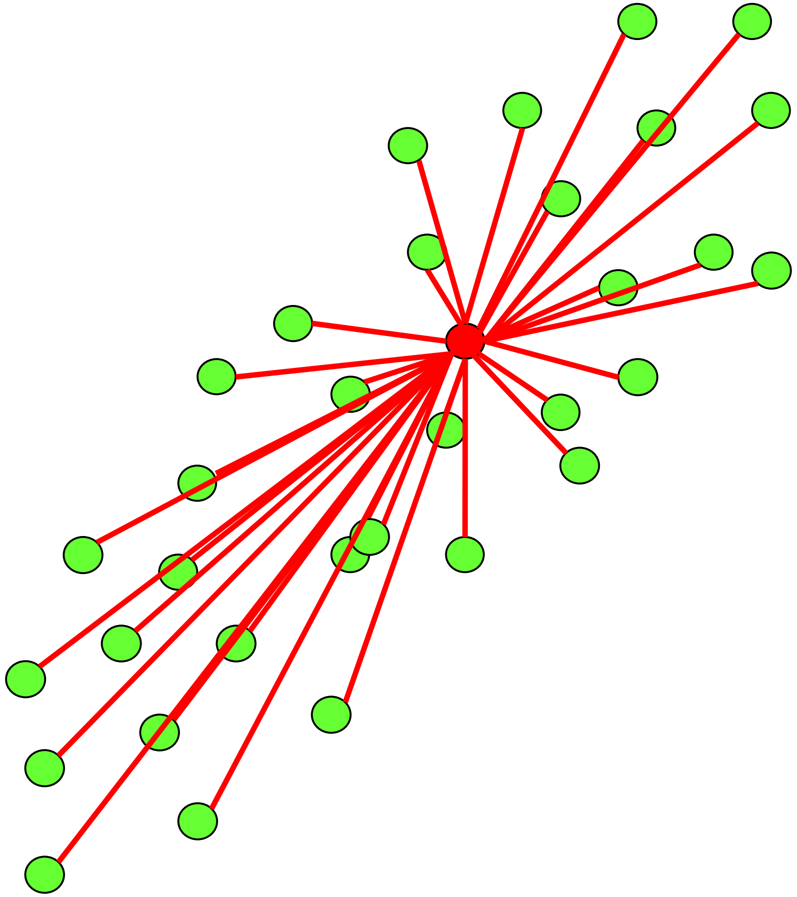


K-means Questions

- What is it trying to optimize?
- Are we sure it will terminate?
- Are we sure it will find an optimal clustering?
- How should we start it?
- *How could we automatically choose the number of centers?*

What is it trying to optimize?

$$\sum_{i=1}^{m} \| \text{▬▬} \|^2 \to \min$$

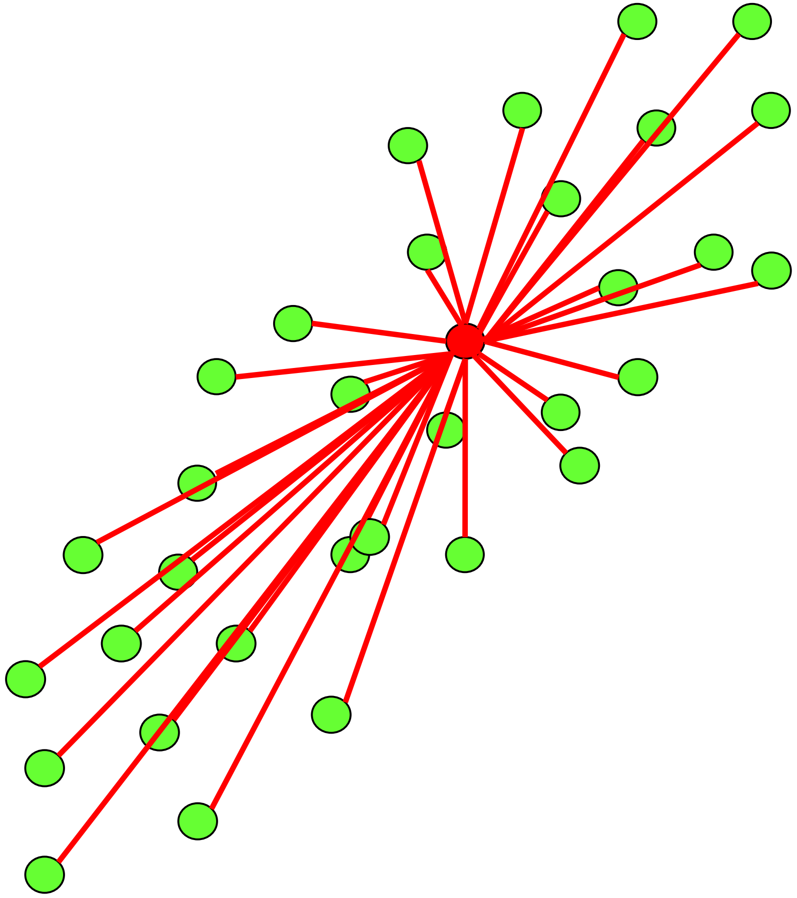


1-Mean

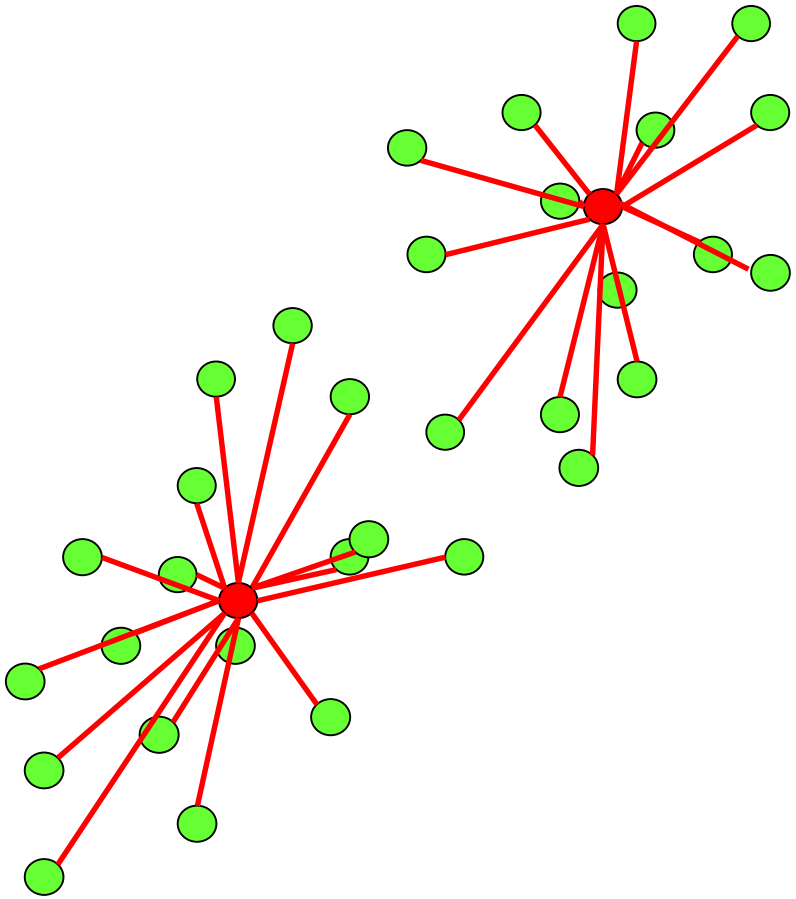
What is it trying to optimize?

$$\sum_{i=1}^{m} \left\| \rule{2cm}{2pt} \right\|^2 \to \min$$

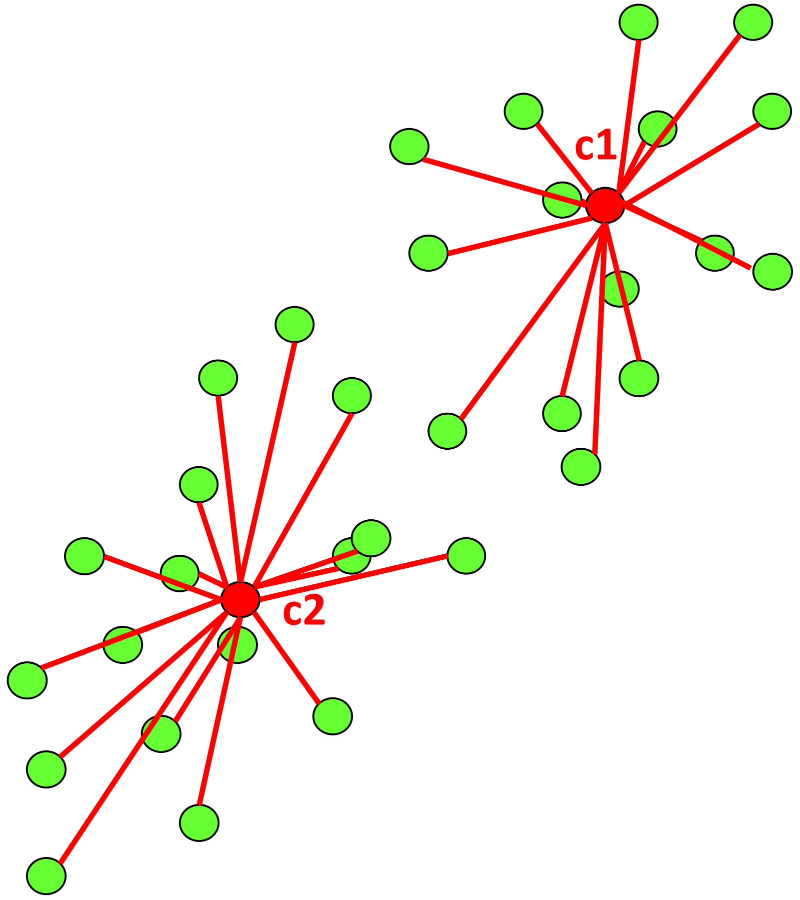
1-Mean



2-Mean



Self-consistency, *principal points*



K-means as data encoder*

Given..

- an encoder function: ENCODE : $\Re^m \rightarrow [1..k]$
- a decoder function: DECODE : $[1..k] \rightarrow \Re^m$

Define…

$$\text{Distortion} = \sum_{i=1}^{R} \left(\mathbf{x}_i - \text{DECODE}[\text{ENCODE}(\mathbf{x}_i)]\right)^2$$

*This formulation is good for building a neural network!

Distortion

$$\sum_{i=1}^{m} \left\| \qquad \right\|^2 \to \min$$

Given..

- an encoder function: ENCODE : $\Re^m \to [1..k]$
- a decoder function: DECODE : $[1..k] \to \Re^m$

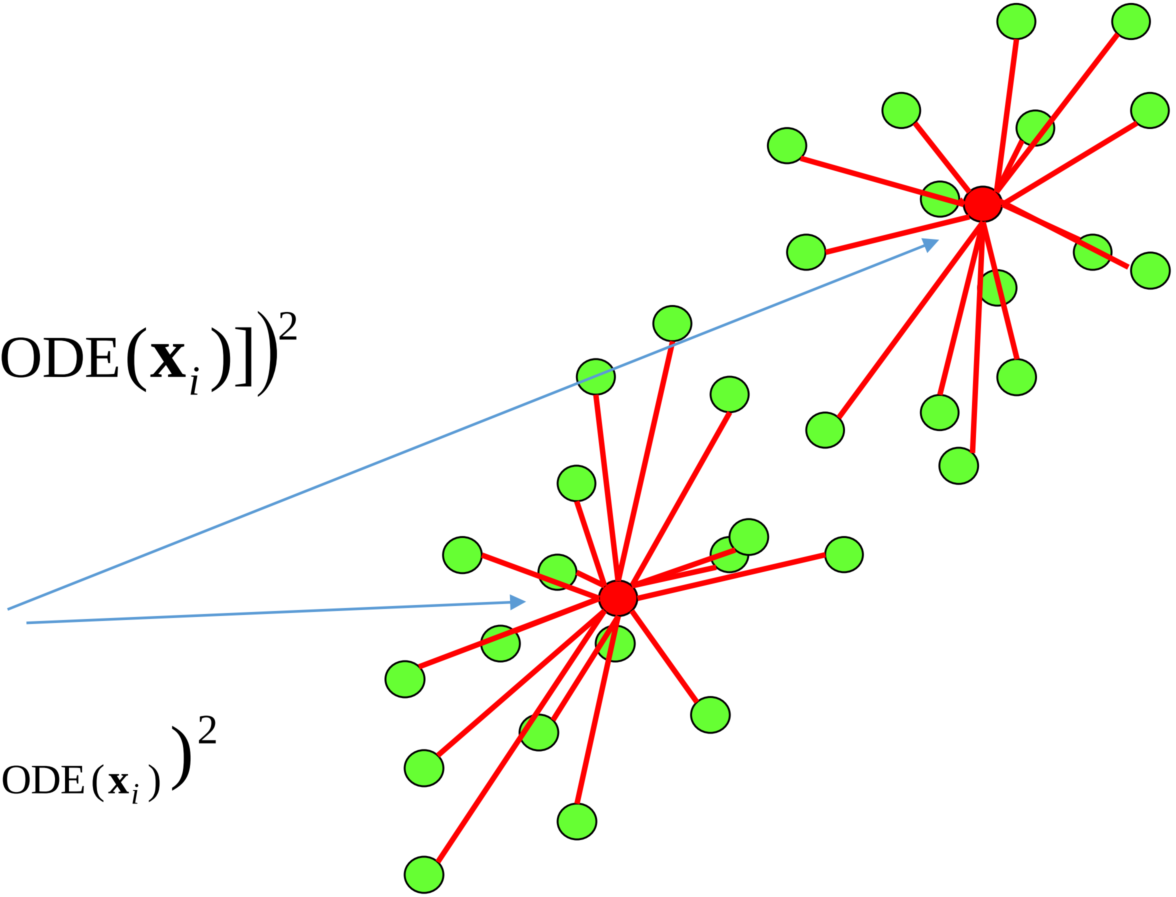
Define…

$$\text{Distortion} = \sum_{i=1}^{R} \left(\mathbf{x}_i - \text{DECODE}[\text{ENCODE}(\mathbf{x}_i)]\right)^2$$

We may as well write

$$\text{DECODE}[j] = \mathbf{c}_j$$

so $$\text{Distortion} = \sum_{i=1}^{R} (\mathbf{x}_i - \mathbf{c}_{\text{ENCODE}(\mathbf{x}_i)})^2$$



The Minimal Distortion

$$\text{Distortion} = \sum_{i=1}^{R} (\mathbf{x}_i - \mathbf{c}_{\text{ENCODE}(\mathbf{x}_i)})^2$$

What properties must centers $\boldsymbol{c}_1$, $\boldsymbol{c}_2$, … , $\boldsymbol{c}_k$ have when distortion is minimized?

The Minimal Distortion (1)

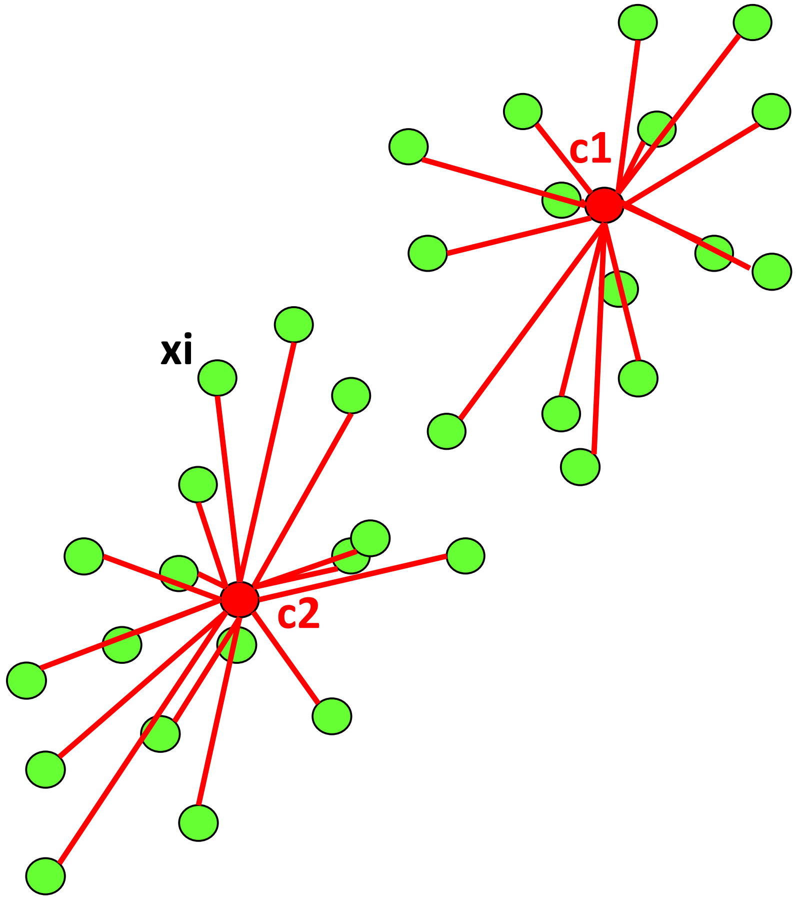
$$\text{Distortion} = \sum_{i=1}^{R} (\mathbf{x}_i - \mathbf{c}_{\text{ENCODE}(\mathbf{x}_i)})^2$$

$$\sum_{i=1}^{m} \left\| \rule{3em}{3pt} \right\|^2 \rightarrow \min$$

What properties must centers $\boldsymbol{c}_1 , \boldsymbol{c}_2 , \ldots , \boldsymbol{c}_k$ have when distortion is minimized?

(1) $\boldsymbol{x}_i$ must be encoded by its nearest center

....why?



$$\mathbf{c}_{\text{ENCODE}(\mathbf{x}_i)} = \underset{\mathbf{c}_j \in \{\mathbf{c}_1, \mathbf{c}_2, \ldots \mathbf{c}_k\}}{\arg\min} (\mathbf{x}_i - \mathbf{c}_j)^2$$

..at the minimal distortion

The Minimal Distortion (2)

$$\text{Distortion} = \sum_{i=1}^{R} (\mathbf{x}_i - \mathbf{c}_{\text{ENCODE}(\mathbf{x}_i)})^2$$

What properties must centers $\boldsymbol{c}_1$, $\boldsymbol{c}_2$, … , $\boldsymbol{c}_k$ have when distortion is minimized?

(2) The partial derivative of Distortion with respect to each center location must be zero.

(2) The partial derivative of Distortion with respect to each center location must be zero.

$$
\begin{aligned}
\text{Distortion} &= \sum_{i=1}^{R} (\mathbf{x}_i - \mathbf{c}_{\text{ENCODE}(\mathbf{x}_i)})^2 \\
&= \sum_{j=1}^{k} \sum_{i \in \text{OwnedBy}(\mathbf{c}_j)} (\mathbf{x}_i - \mathbf{c}_j)^2
\end{aligned}
$$

OwnedBy($\boldsymbol{c}_j$) = the set of records owned by Center $\boldsymbol{c}_j$.

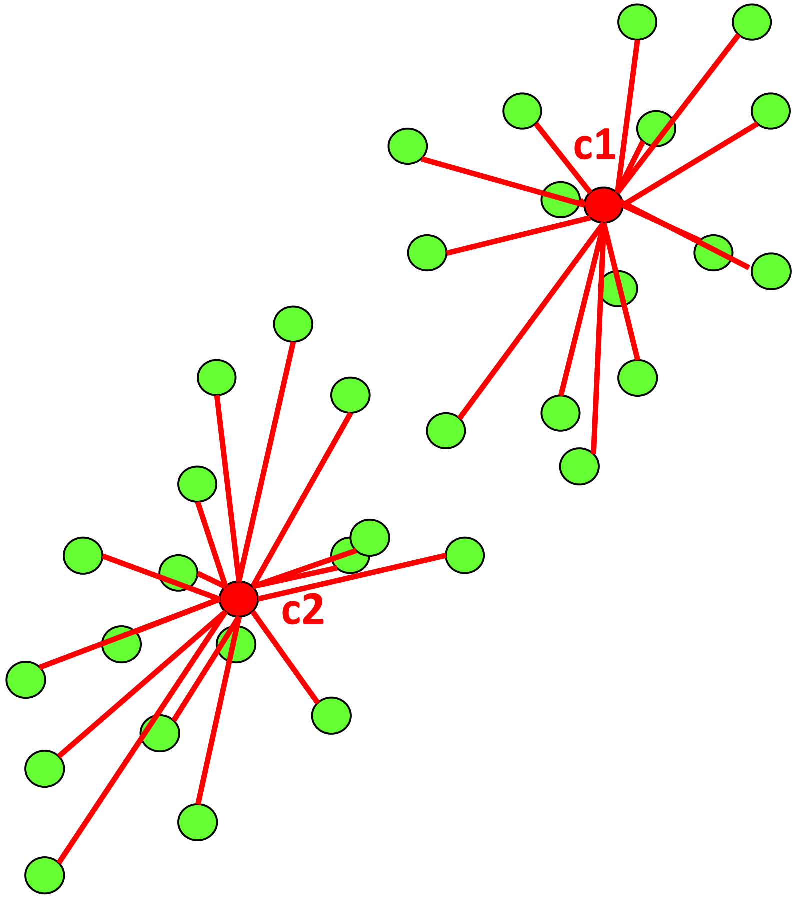
$$
\begin{aligned}
\frac{\partial \text{Distortion}}{\partial \mathbf{c}_j} &= \frac{\partial}{\partial \mathbf{c}_j} \sum_{i \in \text{OwnedBy}(\mathbf{c}_j)} (\mathbf{x}_i - \mathbf{c}_j)^2 \\
&= -2 \sum_{i \in \text{OwnedBy}(\mathbf{c}_j)} (\mathbf{x}_i - \mathbf{c}_j) \\
&= 0 \text{ (for a minimum)}
\end{aligned}
$$

(2) The partial derivative of Distortion with respect to each center location must be zero.

$$\sum_{i=1}^{m} \left\| \rule{2cm}{3pt} \right\|^2 \to \min$$

$$\begin{aligned}
\text{Distortion} &= \sum_{i=1}^{R} (\mathbf{x}_i - \mathbf{c}_{\text{ENCODE}(\mathbf{x}_i)})^2 \\
&= \sum_{j=1}^{k} \sum_{i \in \text{OwnedBy}(\mathbf{c}_j)} (\mathbf{x}_i - \mathbf{c}_j)^2
\end{aligned}$$

$$\begin{aligned}
\frac{\partial \text{Distortion}}{\partial \mathbf{c}_j} &= \frac{\partial}{\partial \mathbf{c}_j} \sum_{i \in \text{OwnedBy}(\mathbf{c}_j)} (\mathbf{x}_i - \mathbf{c}_j)^2 \\
&= -2 \sum_{i \in \text{OwnedBy}(\mathbf{c}_j)} (\mathbf{x}_i - \mathbf{c}_j) \\
&= 0 \text{ (for a minimum)}
\end{aligned}$$



Thus, at a minimum:

$$\mathbf{c}_j = \frac{1}{|\text{OwnedBy}(\mathbf{c}_j)|} \sum_{i \in \text{OwnedBy}(\mathbf{c}_j)} \mathbf{x}_i$$

At the minimum distortion

$$\text{Distortion} = \sum_{i=1}^{R} (\mathbf{x}_i - \mathbf{c}_{\text{ENCODE}(\mathbf{x}_i)})^2$$

What properties must centers $\boldsymbol{c}_1$, $\boldsymbol{c}_2$, … , $\boldsymbol{c}_k$ have when distortion is minimized?

(1) $\boldsymbol{x}_i$ must be encoded by its nearest center

(2) Each Center must be at the centroid of points it owns.

Improving a suboptimal configuration…

$$\text{Distortion} = \sum_{i=1}^{R} (\mathbf{x}_i - \mathbf{c}_{\text{ENCODE}(\mathbf{x}_i)})^2$$

What properties can be changed for centers ***c**$_1$* , ***c**$_2$* , … , ***c**$_k$* have when distortion is not minimized?

(1) Change encoding so that ***x**$_i$* is encoded by its nearest center

(2) Set each Center to the centroid of points it owns.

Alternate! …And that's K-means!

Easy to prove this procedure will terminate in a state at which neither (1) or (2) change the configuration. Why?

Improving a suboptimal configuratio

What p e when distortio

(1) Chan

(2) Set e

There's n

But it can be profitable to alternate.

…And that's K-means!

Easy to prove this procedure will terminate in a state at which neither (1) or (2) change the configuration. Why?

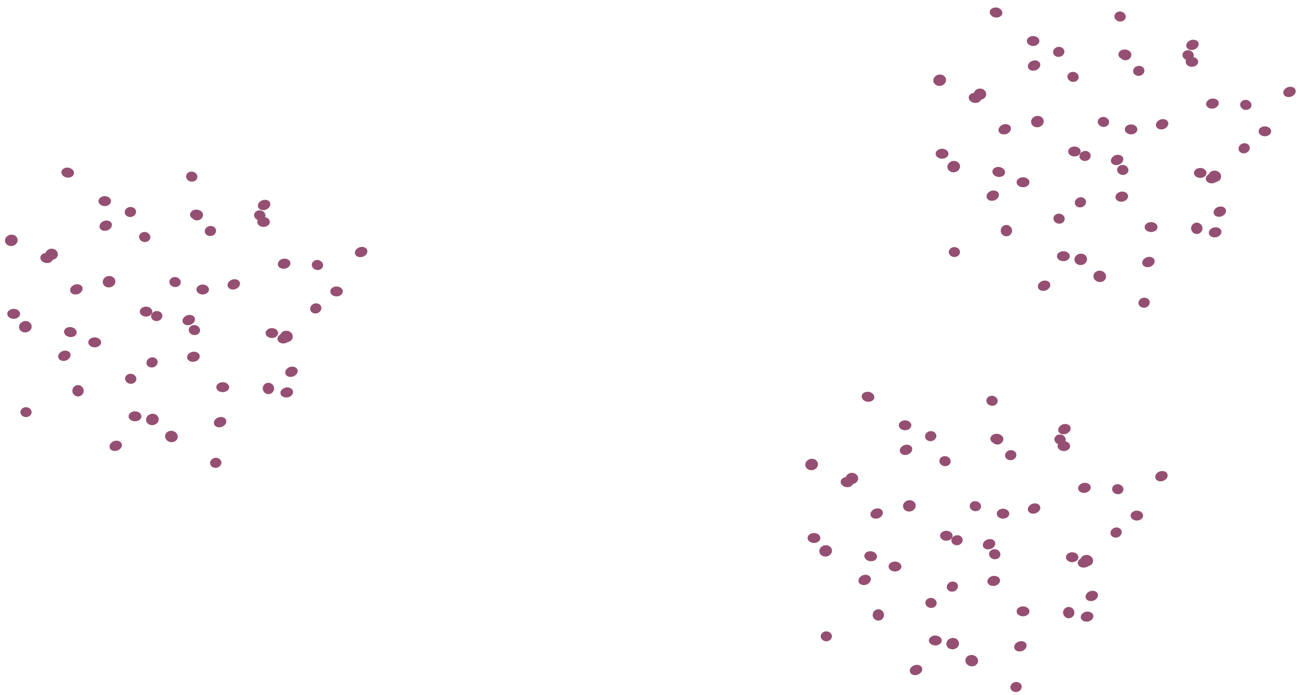
There are only a finite number of ways of partitioning R records into k groups.
So there are only a finite number of possible configurations in which all Centers are the centroids of the points they own.
If the configuration changes on an iteration, it must have improved the distortion.
So each time the configuration changes it must go to a configuration it's never been to before.
So if it tried to go on forever, it would eventually run out of configurations.

Will we find the optimal configuration?

- Not necessarily.
- Can you invent a configuration that has converged, but does not have the minimum distortion?

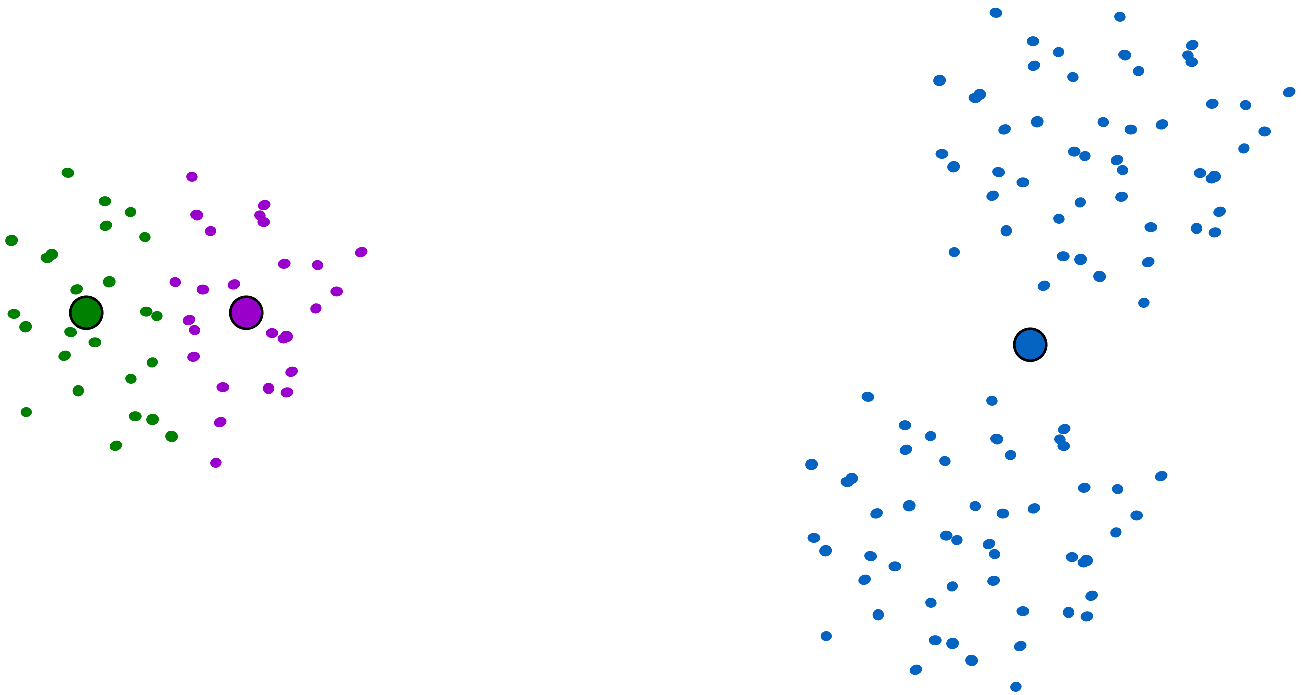
Will we find the optimal configuration?

- Not necessarily.
- Can you invent a configuration that has converged, but does not have the minimum distortion?



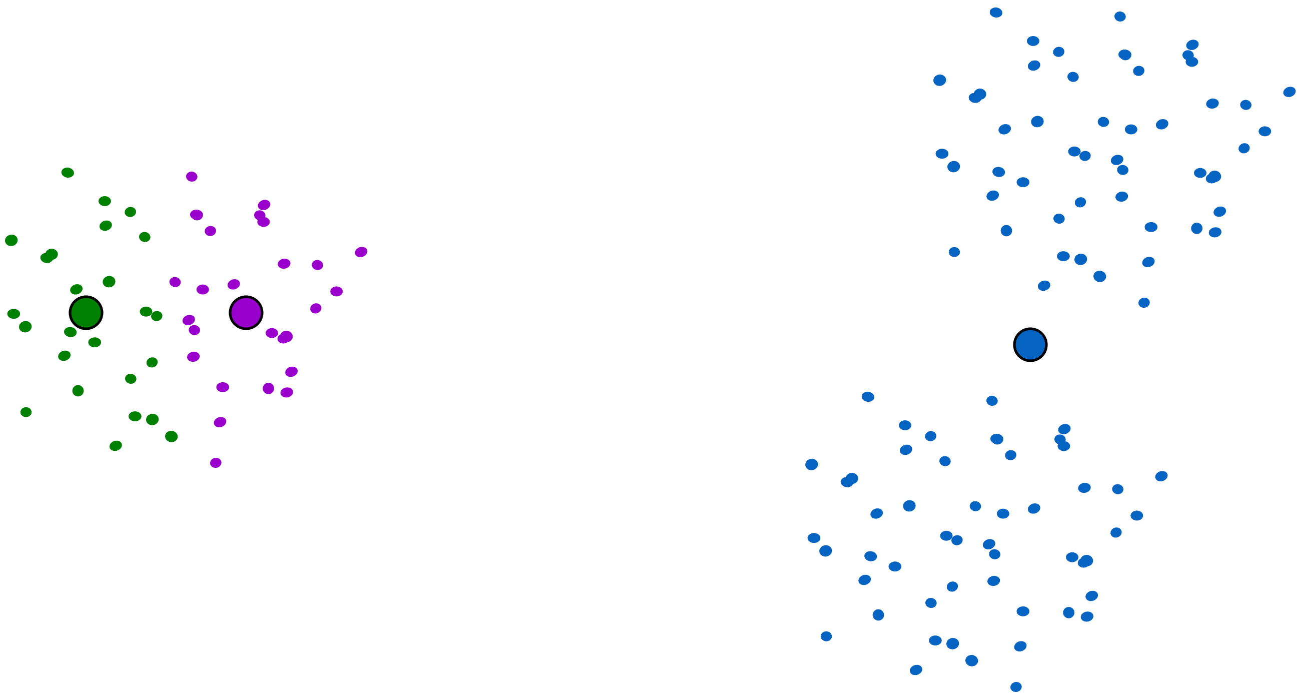
Will we find the optimal configuration?

- Not necessarily.
- Can you invent a configuration that has converged, but does not have the minimum distortion?



Trying to find good optima

- Idea 1: Be careful about where you start
- Idea 2: Do many runs of k-means, each from a different random start configuration
- Many other ideas floating around.



Trying to find good optima

- Idea 1: Be careful about where you start
- Idea 2: Do ny runs of k-means, each from a different random start tion
- Many

Neat trick:

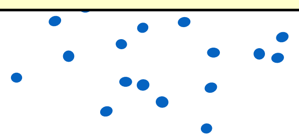
Place first center on top of randomly chosen datapoint.

Place second center on datapoint that's as far away as possible from first center

:

Place j'th center on datapoint that's as far away as possible from the closest of Centers 1 through j-1

:



Common uses of K-means

- Often used as an exploratory data analysis tool
- In one-dimension, a good way to quantize real-valued variables into k non-uniform buckets
- **Coarse-graining of big data!** (reducing the effect of outliers)

Pros and Cons of K-means

- Relatively efficient: O(tknm)
  - n: # objects, k: # clusters, t: # iterations, m: dimension of data; k, t << n.
- Often terminate at a local optimum
- Applicable only when mean is defined
  - What about categorical data?
- Need to specify the number of clusters
- Unable to handle noisy data and outliers
- unsuitable to discover non-convex clusters

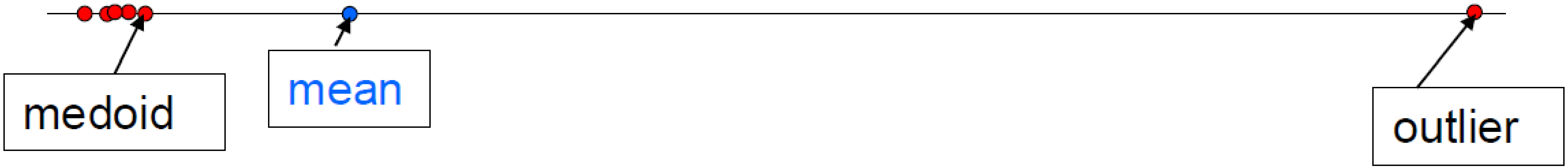
K-medoids or PAM (partitioning around medoids)

- Arbitrarily choose k objects as the initial medoids
- Until no change, do

–(Re)assign each object to the cluster to which the nearest medoid

–Randomly select a non-medoid object o’, compute the total cost, S, of swapping medoid o with o’

–If S < 0 then swap o with o’ to form the new set of k medoids



Pros and Cons of PAM

- PAM is more robust than k-means in the presence of noise and outliers

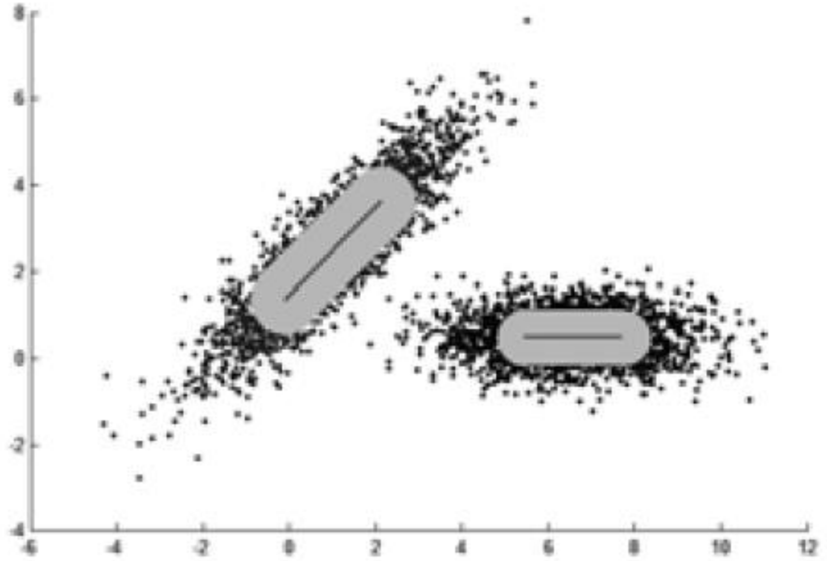
Medoids are less influenced by outliers

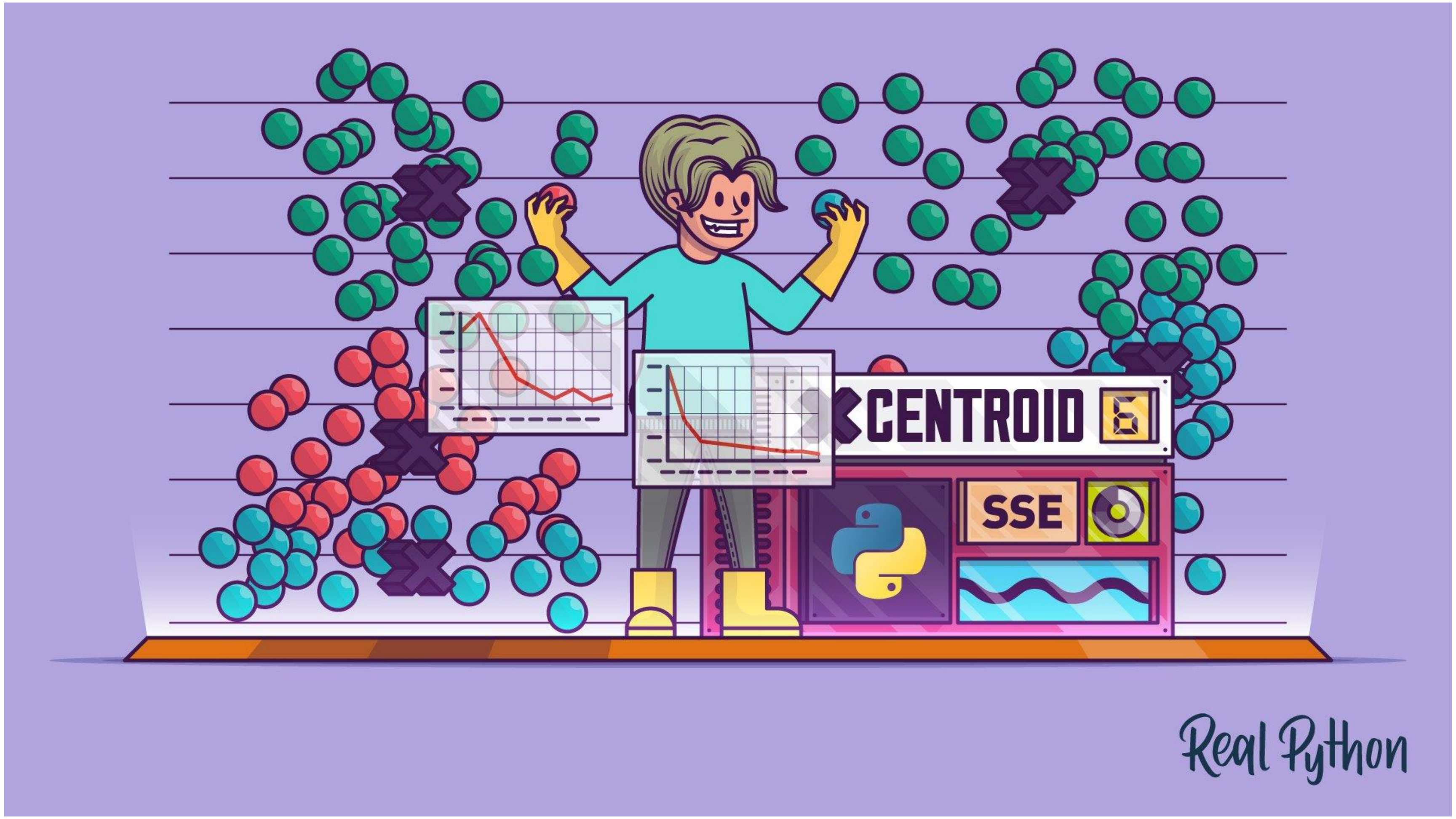
- PAM is efficient for small data sets but does not scale well for large data sets

$O(kn^2)$ for each iteration

Building on top of k-means

- **K-lines clustering :** centroid is not a point but a line segment
- **Soft or fuzzy k-means**: Each point can belong to more than one cluster
- **K-means with trimming:** points too distant (>R) from the centroid do not contribute at a given iteration to define the new centroid position





https://realpython.com/k-means-clustering-python/