CS 412 Introduction to Machine Learning

K-means

Instructor: Wei Tang

Department of Computer Science
University of Illinois at Chicago
Chicago IL 60607

https://tangw.people.uic.edu
tangw@uic.edu

Slides credit: Xinhua Zhang

k-Means Clustering

- Find k reference vectors (prototypes/codebook vectors/codewords) which best represent data

- Sample $\mathcal{X} = \{\mathbf{x}^t\}_{t=1}^N$. Reference vectors: $\mathbf{m}_j$ ($j = 1,\ldots,k$)
- Use nearest (most similar) reference:

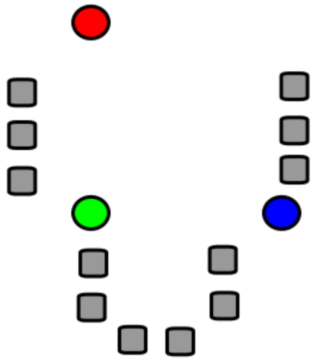
code book

$$\left\|\mathbf{x}^t - \mathbf{m}_i\right\| = \min_j \left\|\mathbf{x}^t - \mathbf{m}_j\right\|$$

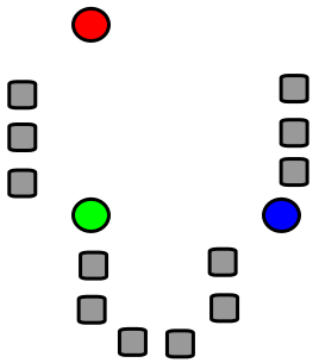
- Reconstruction error $E\left(\{\mathbf{m}_i\}_{i=1}^k \middle| \mathcal{X}\right) = \sum_t \sum_i b_i^t \left\|\mathbf{x}^t - \mathbf{m}_i\right\|^2$

no analytic minimizer
NP-hard to optimize $\{m_i\}$

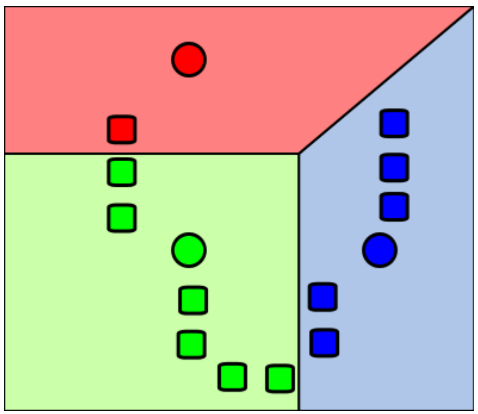
$$b_i^t = \begin{cases} 1 & \text{if } \left\|\mathbf{x}^t - \mathbf{m}_i\right\| = \min_j \left\|\mathbf{x}^t - \mathbf{m}_j\right\| \\ 0 & \text{otherwise} \end{cases}$$



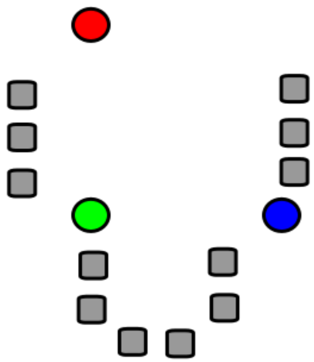
1. Select initial
centroids at random



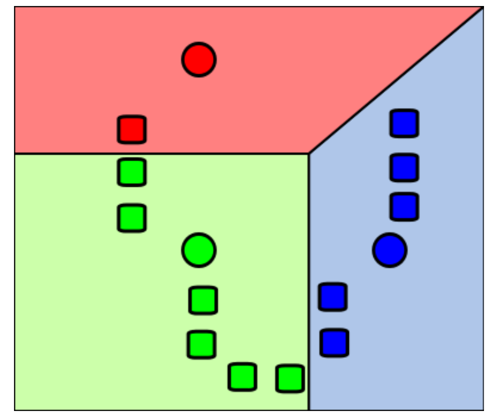
1. Select initial centroids at random



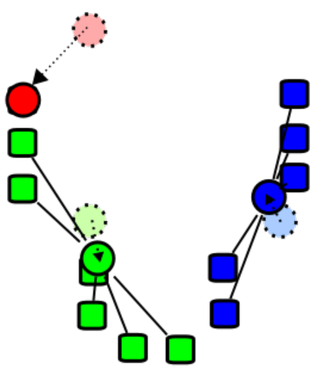
2. Assign each object to the cluster with the nearest centroid.



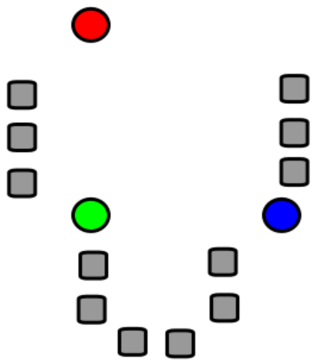
1. Select initial
centroids at random



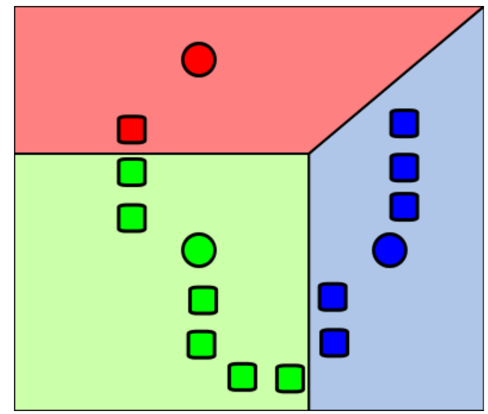
2. Assign each object to
the cluster with the
nearest centroid.



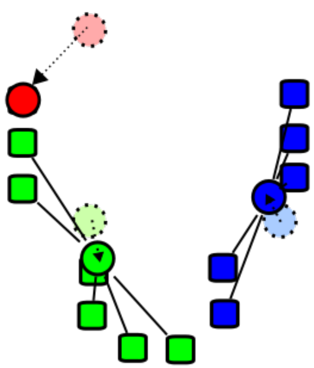
3. Compute each centroid as
the mean of the objects
assigned to it (go to 2)



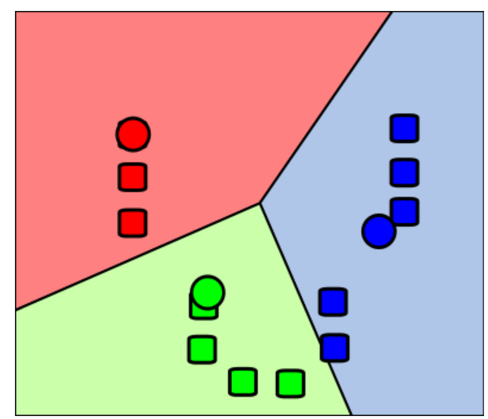
1. Select initial centroids at random



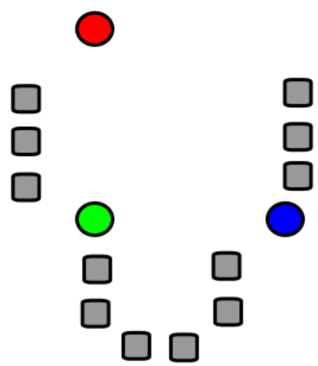
2. Assign each object to the cluster with the nearest centroid.



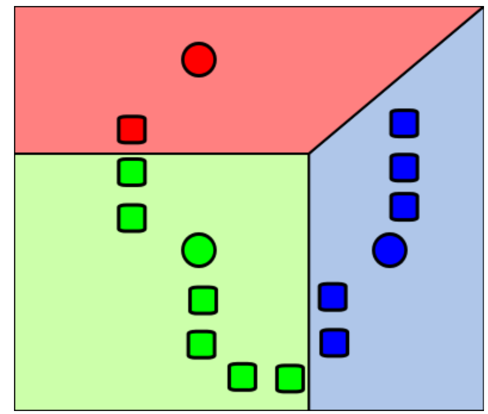
3. Compute each centroid as the mean of the objects assigned to it (go to 2)



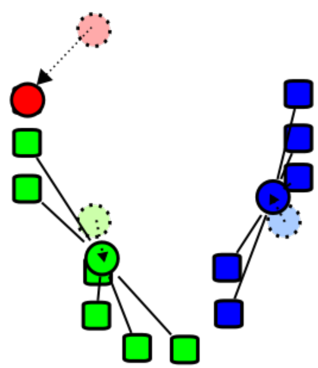
2. Assign each object to the cluster with the nearest centroid.



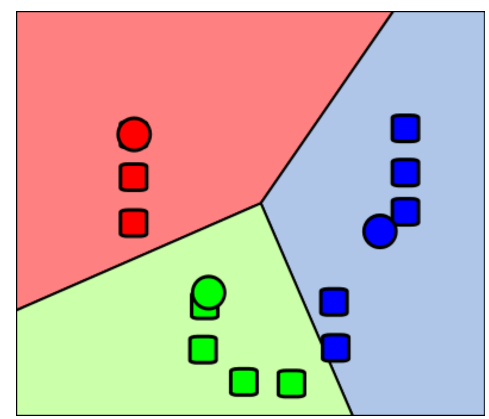
1. Select initial centroids at random



2. Assign each object to the cluster with the nearest centroid.



3. Compute each centroid as the mean of the objects assigned to it (go to 2)



2. Assign each object to the cluster with the nearest centroid.

Repeat previous 2 steps until no change

K-means Clustering

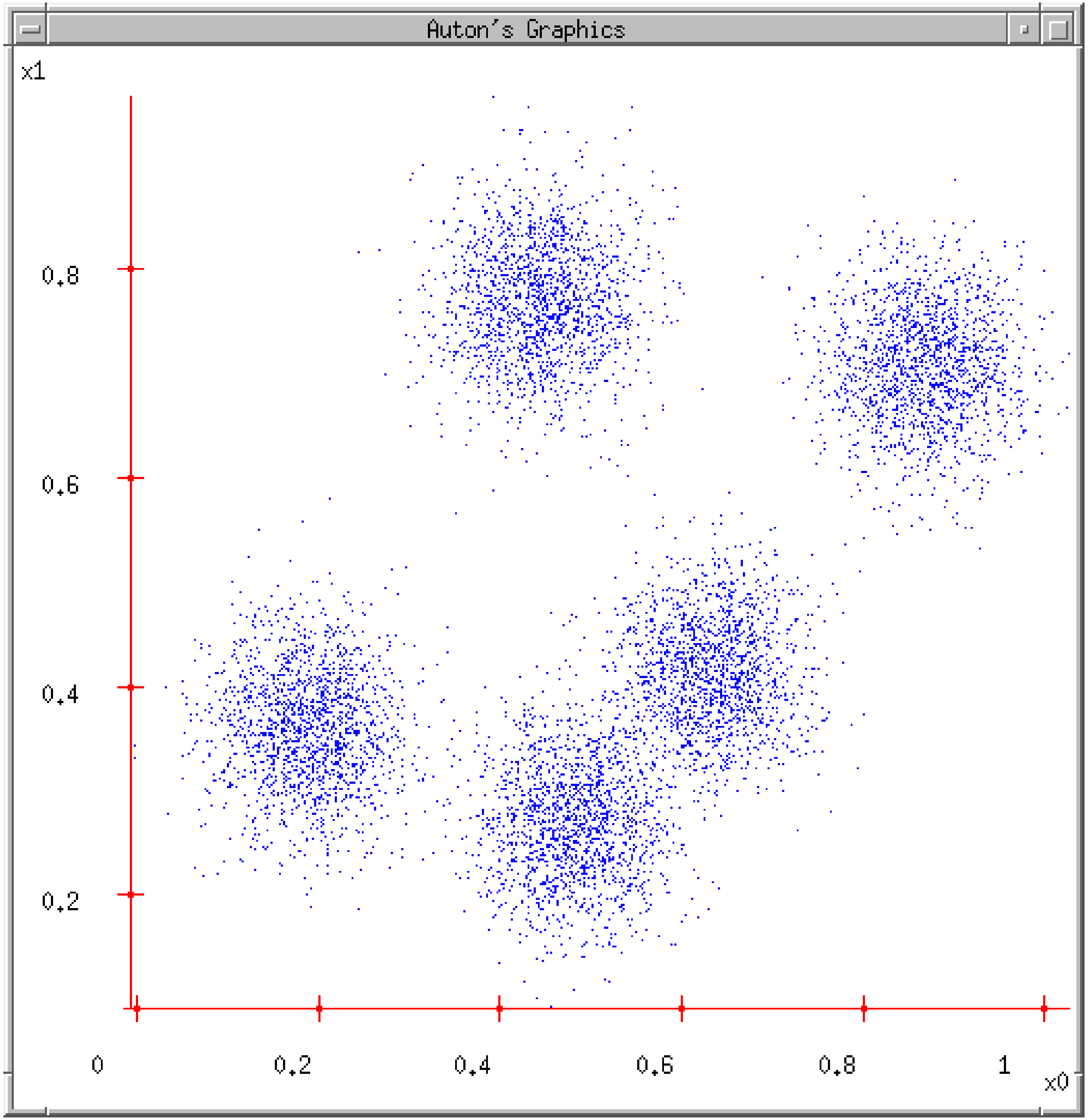
Given k:

1. Select initial centroids at random.

2. Assign each object to the cluster with the nearest centroid.

3. Compute each centroid as the mean of the objects assigned to it.

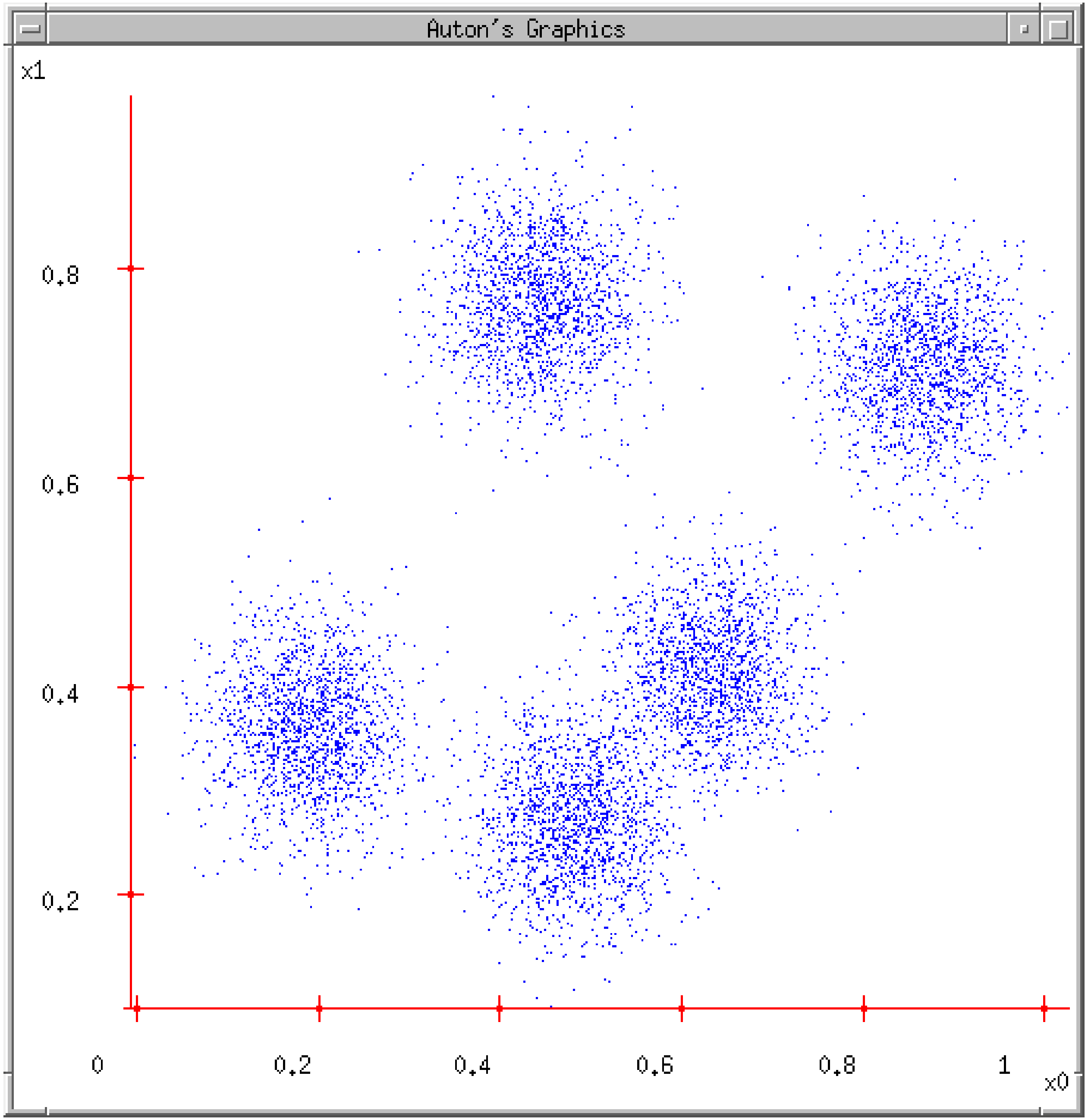
4. Repeat previous 2 steps until no change.



Auton's Graphics
x1
0.8
0.6
0.4
0.2
0
0.2
0.4
0.6
0.8
1
x0

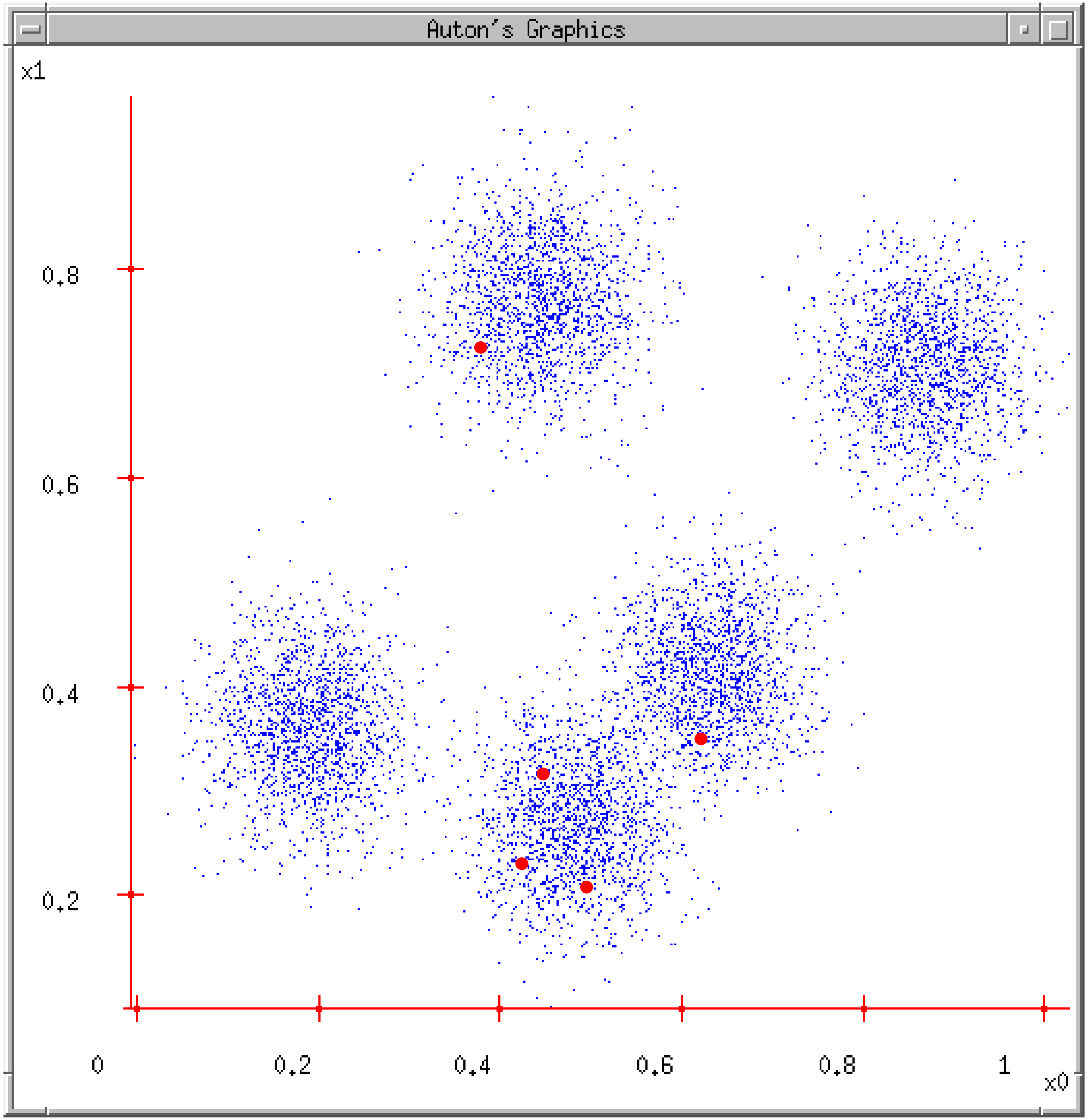
K-means

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



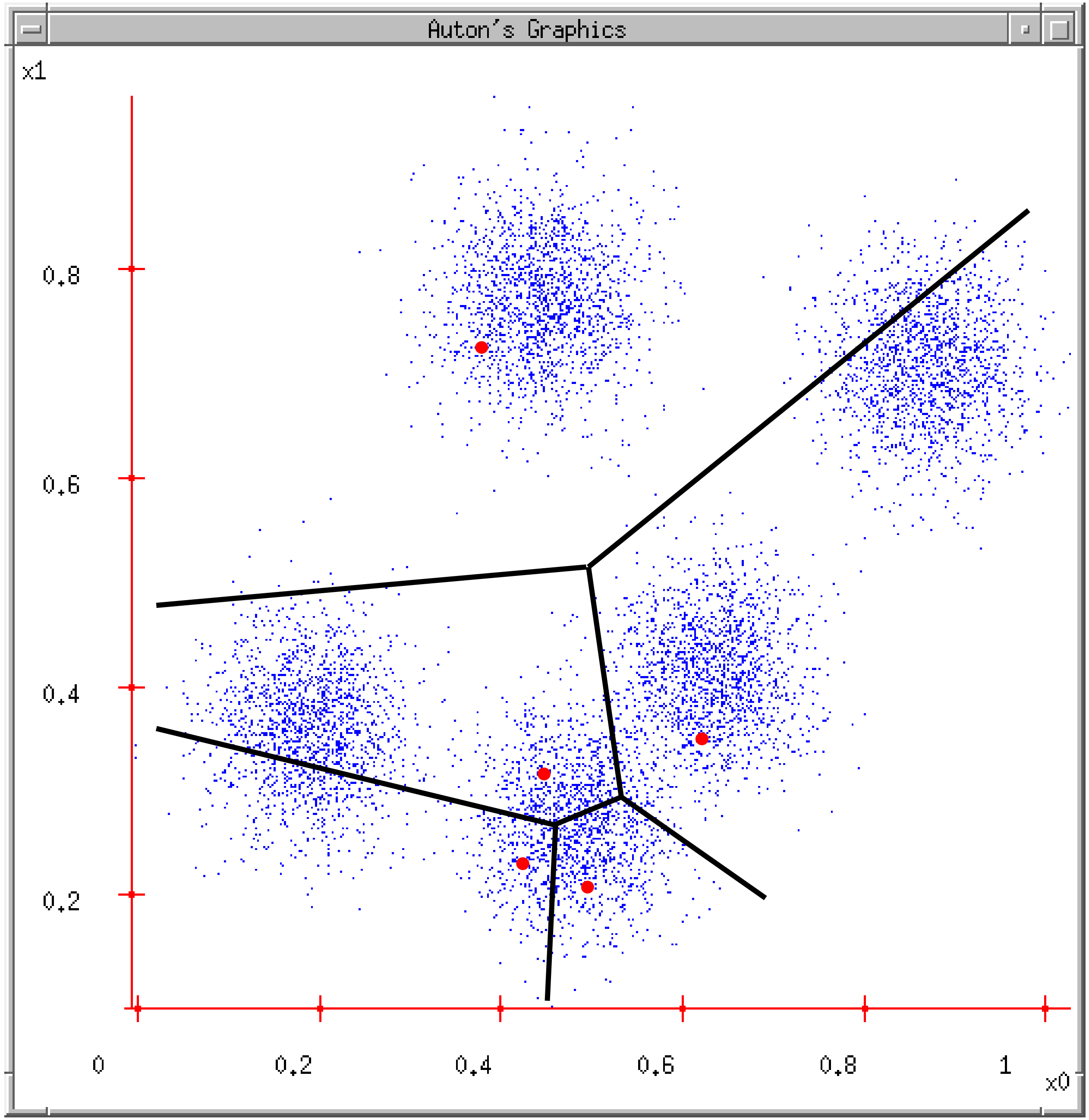
K-means

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



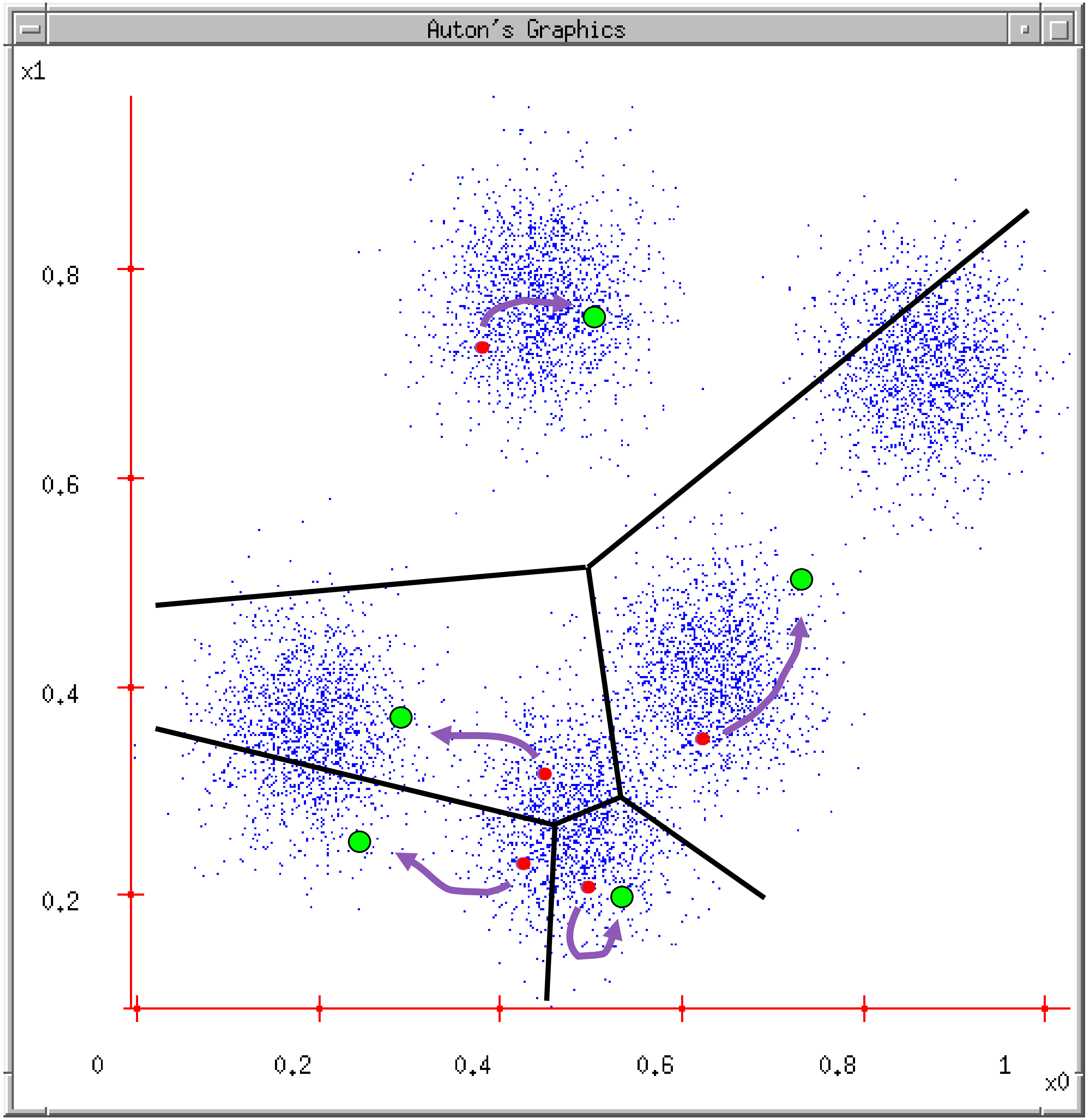
K-means

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)



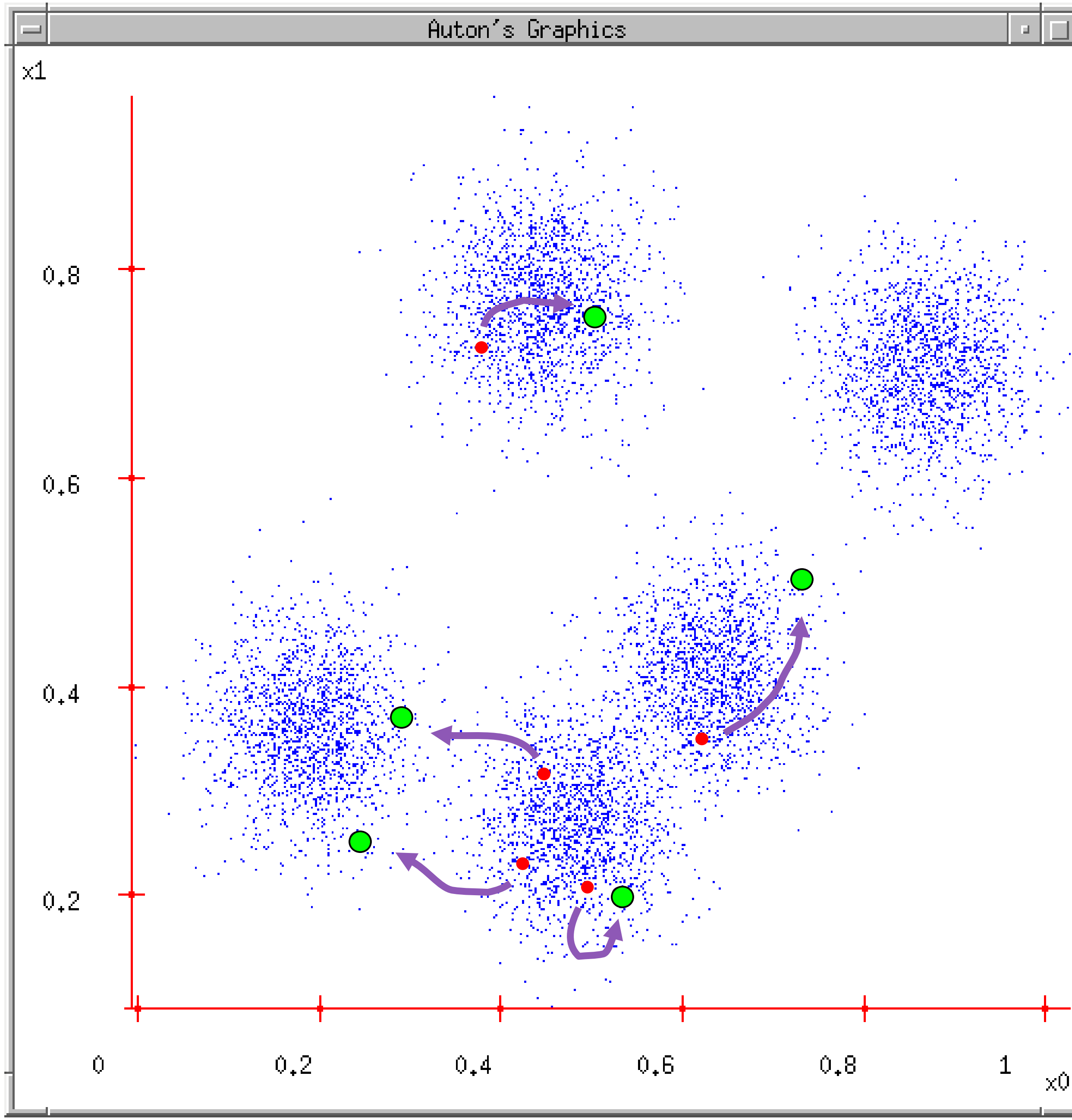
K-means

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



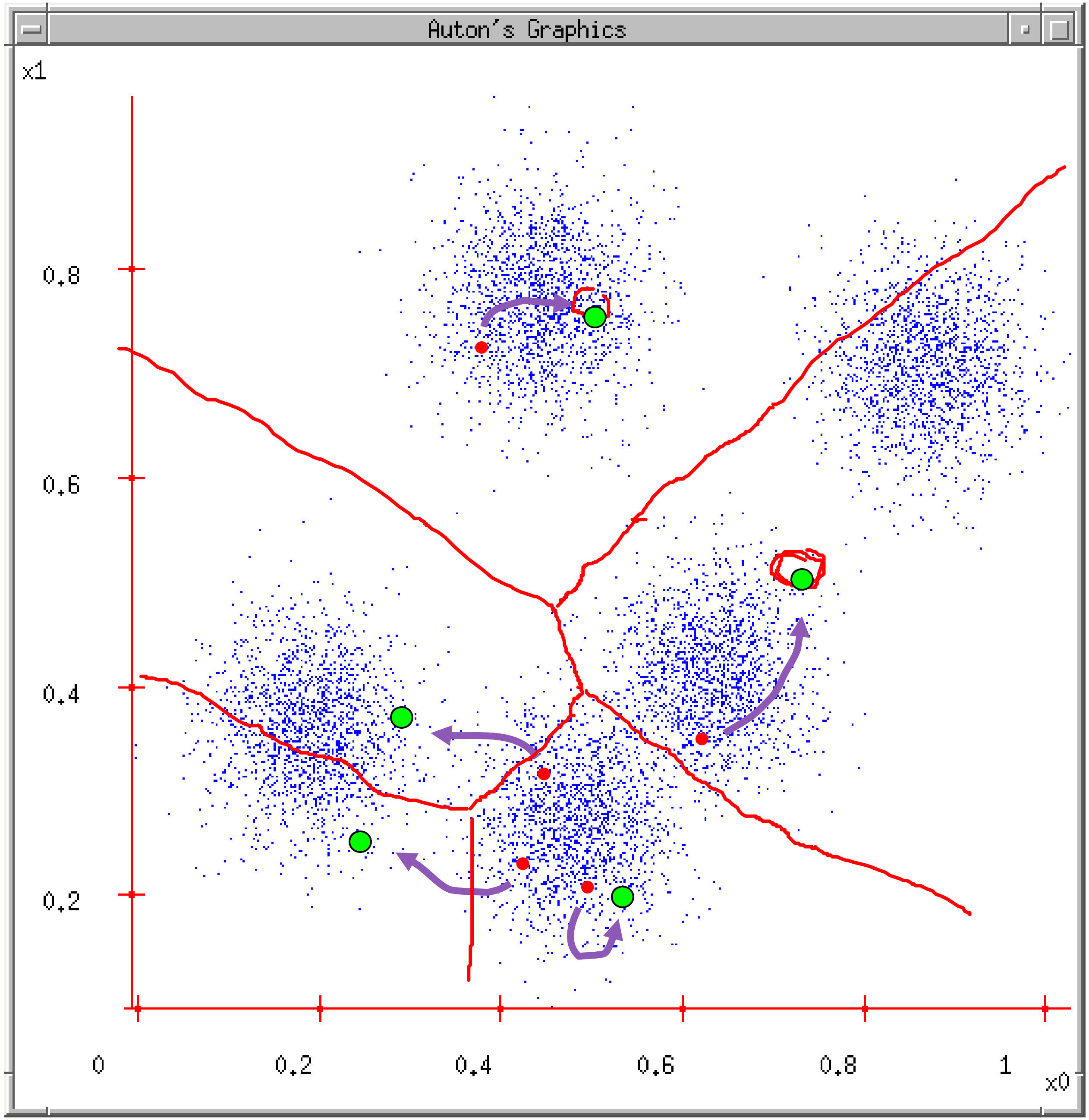
K-means

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



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 Clustering

Initialize $\boldsymbol{m}_i, i = 1, \ldots, k$, for example, to k random $\boldsymbol{x}^t$

Repeat

For all $\boldsymbol{x}^t \in \mathcal{X}$

$$b_i^t \leftarrow \begin{cases} 1 & \text{if } \|\boldsymbol{x}^t - \boldsymbol{m}_i\| = \min_j \|\boldsymbol{x}^t - \boldsymbol{m}_j\| \\ 0 & \text{otherwise} \end{cases}$$

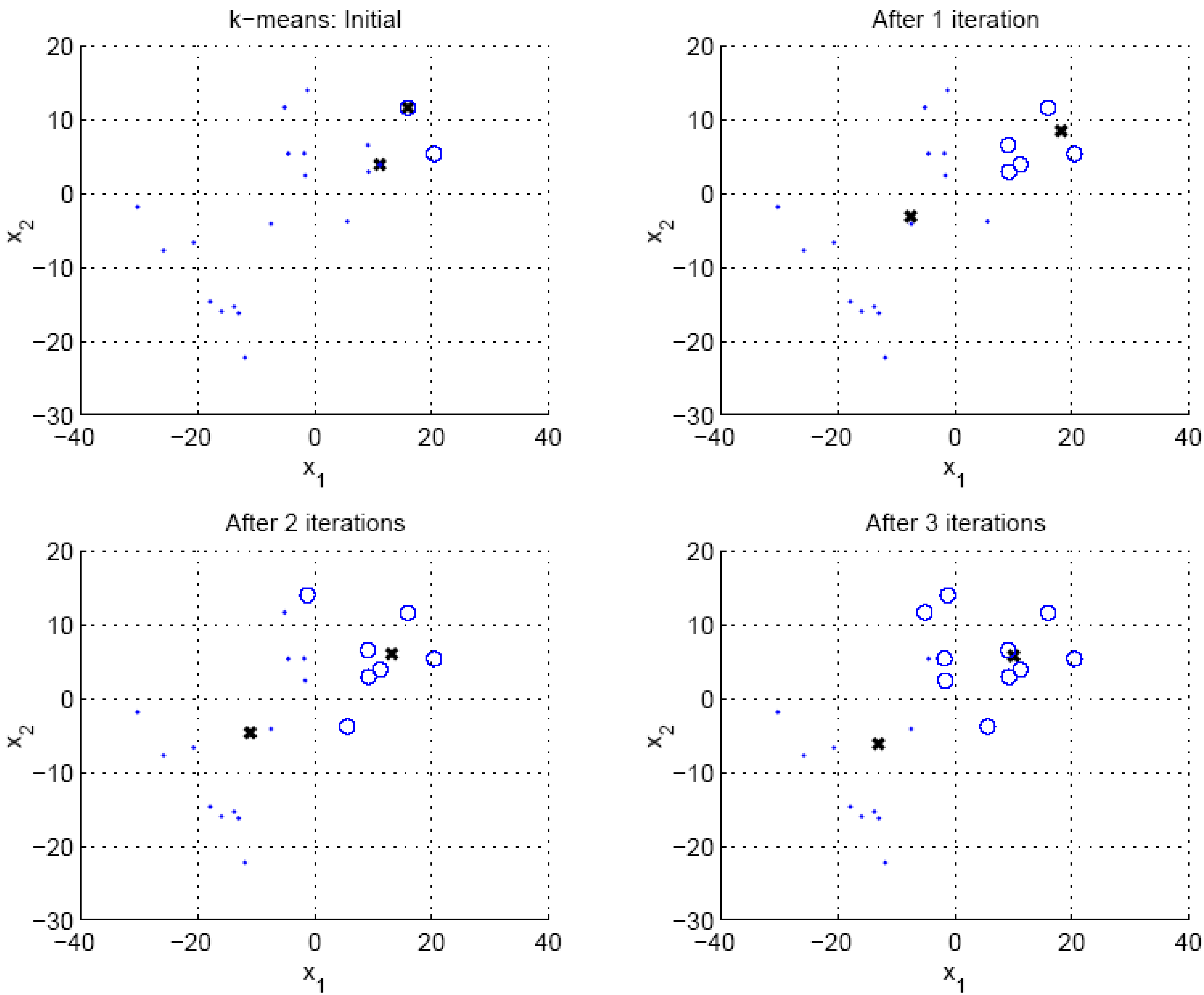
For all $\boldsymbol{m}_i, i = 1, \ldots, k$

$$\boldsymbol{m}_i \leftarrow \sum_t b_i^t \boldsymbol{x}^t / \sum_t b_i^t$$

Until $\boldsymbol{m}_i$ converge

$$E\left(\{\mathbf{m}_i\}_{i=1}^k \middle| \mathcal{X}\right) = \sum_t \sum_i b_i^t \left\|\mathbf{x}^t - \mathbf{m}_i\right\|^2$$

$$b_i^t = \begin{cases} 1 & \text{if } \left\|\mathbf{x}^t - \mathbf{m}_i\right\| = \min_j \left\|\mathbf{x}^t - \mathbf{m}_j\right\| \\ 0 & \text{otherwise} \end{cases}$$


k−means: Initial
After 1 iteration
After 2 iterations
After 3 iterations
x2
x1
20
10
0
−10
−20
−30
−40
−20
0
20
40

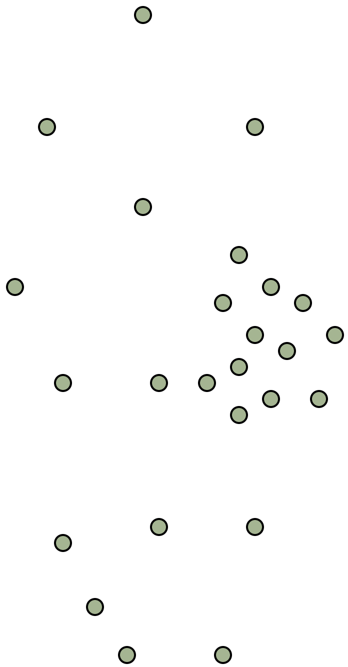
Local procedure

- May converge to suboptimal

$$min_{\{m_i\}} \quad E\left(\{\mathbf{m}_i\}_{i=1}^{k} \middle| \mathcal{X}\right) = \sum_t \sum_i b_i^t \left\| \mathbf{x}^t - \mathbf{m}_i \right\|^2$$

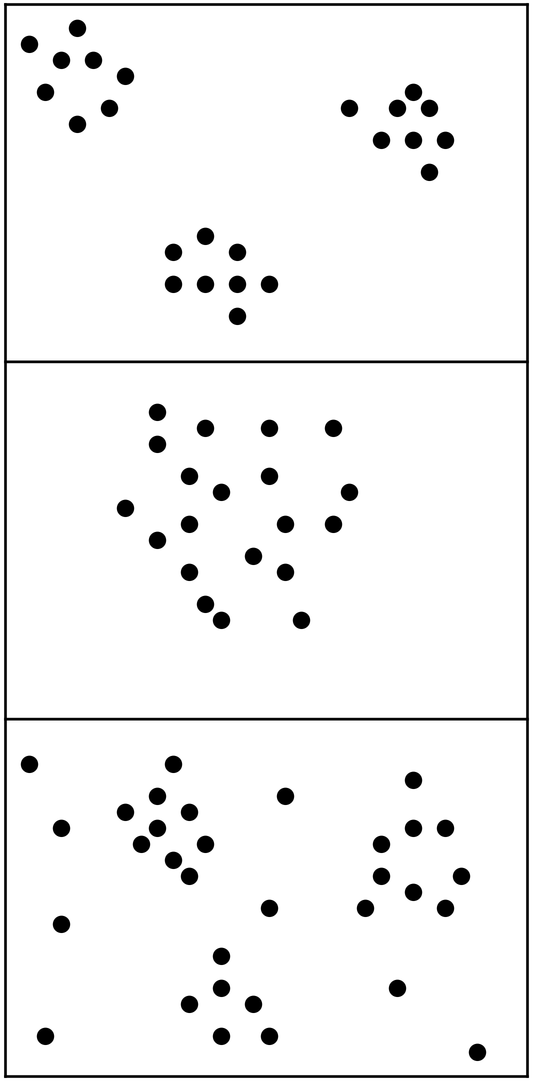
- Randomly reinitialize
  - take randomly selected k instances as the initial $\boldsymbol{m}_i$
  - 1) calculate the mean of all data; 2) add small random vectors to the mean to get the k initial $\boldsymbol{m}_i$.

Bad cases for k-means



- Clusters may overlap
- Some clusters may be "wider" than others

Unsupervised Learning



Sometimes easy

Sometimes impossible

and sometimes in between

K-means clustering using intensity or color

Image

Clusters on intensity

Clusters on color

Choosing *k*

- Defined by the application, e.g., image segmentation or color quantization
- Plot data (projection to low dimension) and check for clusters
- Add one at a time until small change in reconstruction error
- Cross-validation