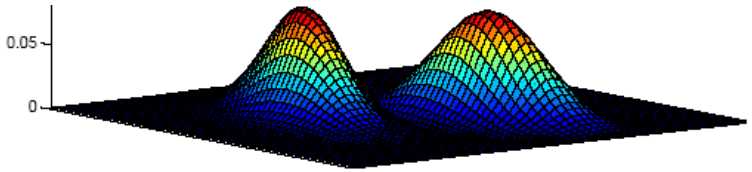
CHAPTER 7 (7.1-7.3, 7.8):

# CLUSTERİNG

# Semiparametric Density Estimation



- Parametric: Assume a single model for $p(\mathbf{x} \mid C_i)$ (Ch 4 & 5)
  - Reduces to the estimation of a small # of parameters
  - What if the model/bias isn't accurate?
- Semiparametric: $p(\mathbf{x} \mid C_i)$ is a mixture of densities
  - More flexible models
  - Multiple possible explanations/prototypes: Different handwriting styles (writing "7"), accents in speech




- Nonparametric: No model; data speaks for itself (Ch 8)

# Mixture Densities



$$p(\mathbf{x}) = \sum_{i=1}^{k} p(\mathbf{x} \mid G_i) P(G_i)$$

where $G_i$ the components/groups/clusters,

$P(G_i)$ mixture proportions (priors),

$p(\boldsymbol{x} \mid G_i)$ component densities

$k$: hyperparameter specified a priori

Gaussian mixture where $p(\boldsymbol{x} \mid G_i) \sim N(\boldsymbol{\mu}_i, \sum_i)$

parameters $\Phi = \{P(G_i), \boldsymbol{\mu}_i, \sum_i\}_{i=1}^{k}$

unlabeled sample $X=\{\boldsymbol{x}^t\}_t$ (unsupervised learning)

# Classes vs. Clusters
# Parametric classification v.s. clustering



- Supervised: $X = \{x^t, r^t\}_t$
- Classes $C_i$ $i=1,...,K$

$$p(\mathbf{x}) = \sum_{i=1}^{K} p(\mathbf{x} \mid C_i) P(C_i)$$

where $p(\mathbf{x} \mid C_i) \sim N(\mu_i, \sum_i)$

- $\Phi = \{P(C_i), \mu_i, \sum_i\}^K_{i=1}$

$$\hat{P}(C_i) = \frac{\sum_t r_i^t}{N} \quad \mathbf{m}_i = \frac{\sum_t r_i^t \mathbf{x}^t}{\sum_t r_i^t}$$

$$\mathbf{S}_i = \frac{\sum_t r_i^t (\mathbf{x}^t - \mathbf{m}_i)(\mathbf{x}^t - \mathbf{m}_i)^T}{\sum_t r_i^t}$$

- Unsupervised : $X = \{x^t\}_t$
- Clusters $G_i$ $i=1,...,k$

$$p(\mathbf{x}) = \sum_{i=1}^{k} p(\mathbf{x} \mid G_i) P(G_i)$$

where $p(\mathbf{x} \mid G_i) \sim N(\mu_i, \sum_i)$

- $\Phi = \{P(G_i), \mu_i, \sum_i\}^k_{i=1}$

How to figure out $\Phi$ when **no** label $r^t_i$ is available?

# *k*-Means Clustering



- Find *k* reference vectors (prototypes/codebook vectors/codewords) which best represent data
  - 24 bits/pixel image => 8 bit/pixel (uniformly? sea)
- 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

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

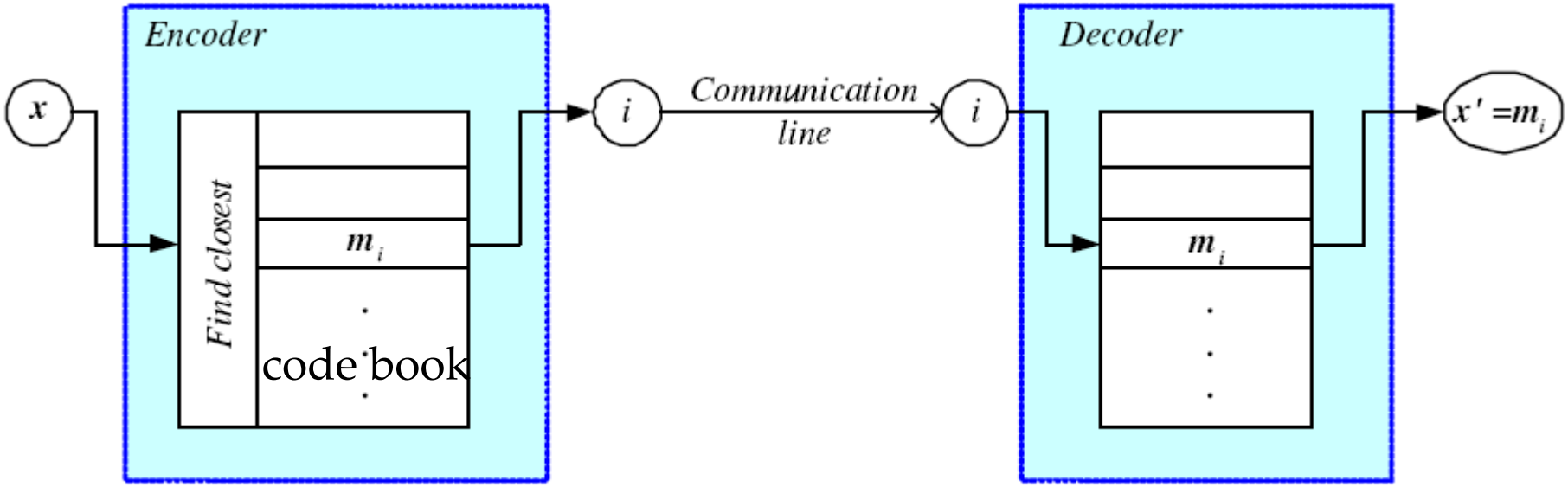
- Reconstruction error $E\left(\{\mathbf{m}_i\}_{i=1}^k \middle| \mathcal{X}\right) = \sum_t \sum_i b_i^t \left\|\mathbf{x}^t - \mathbf{m}_i\right\|^2$

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

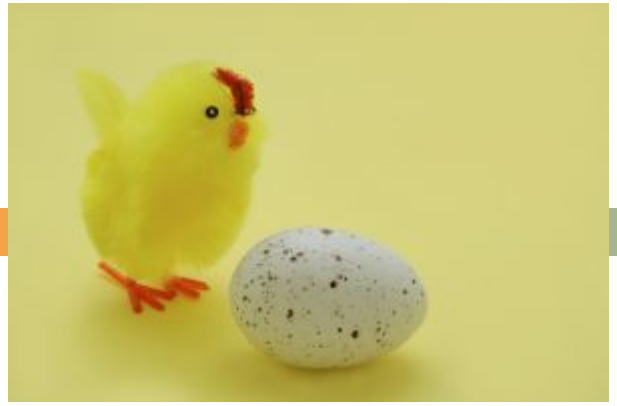
$$b_i^t = \begin{cases} 1 & \text{if } \left\|\mathbf{x}^t - \mathbf{m}_i\right\| = \min_j \left\|\mathbf{x}^t - \mathbf{m}_j\right\| \\ 0 & \text{otherwise} \end{cases}$$

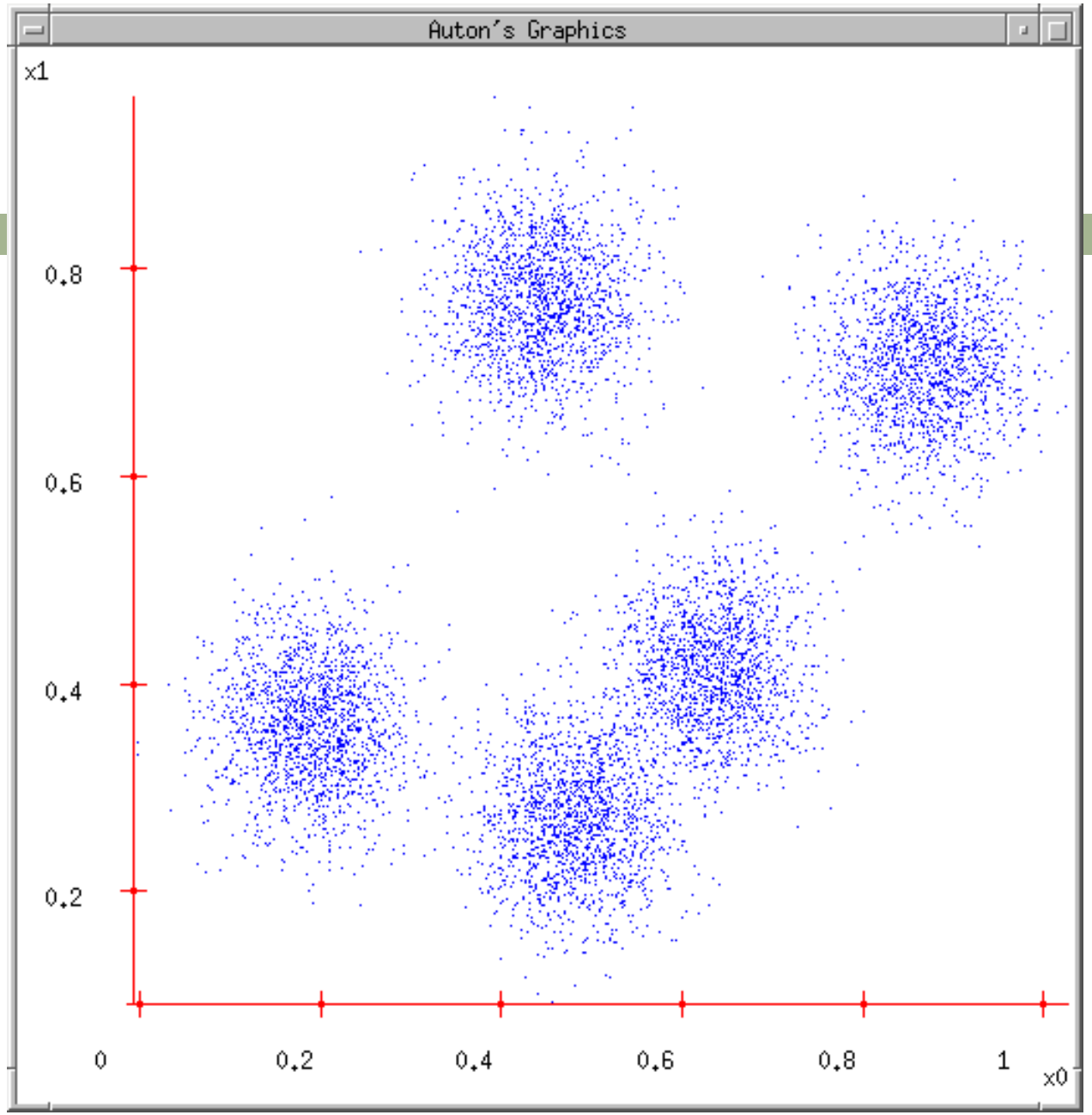
# Encoding/Decoding






Quantization allows compression (24 bit => 8 bit).
Also need to transfer color map.

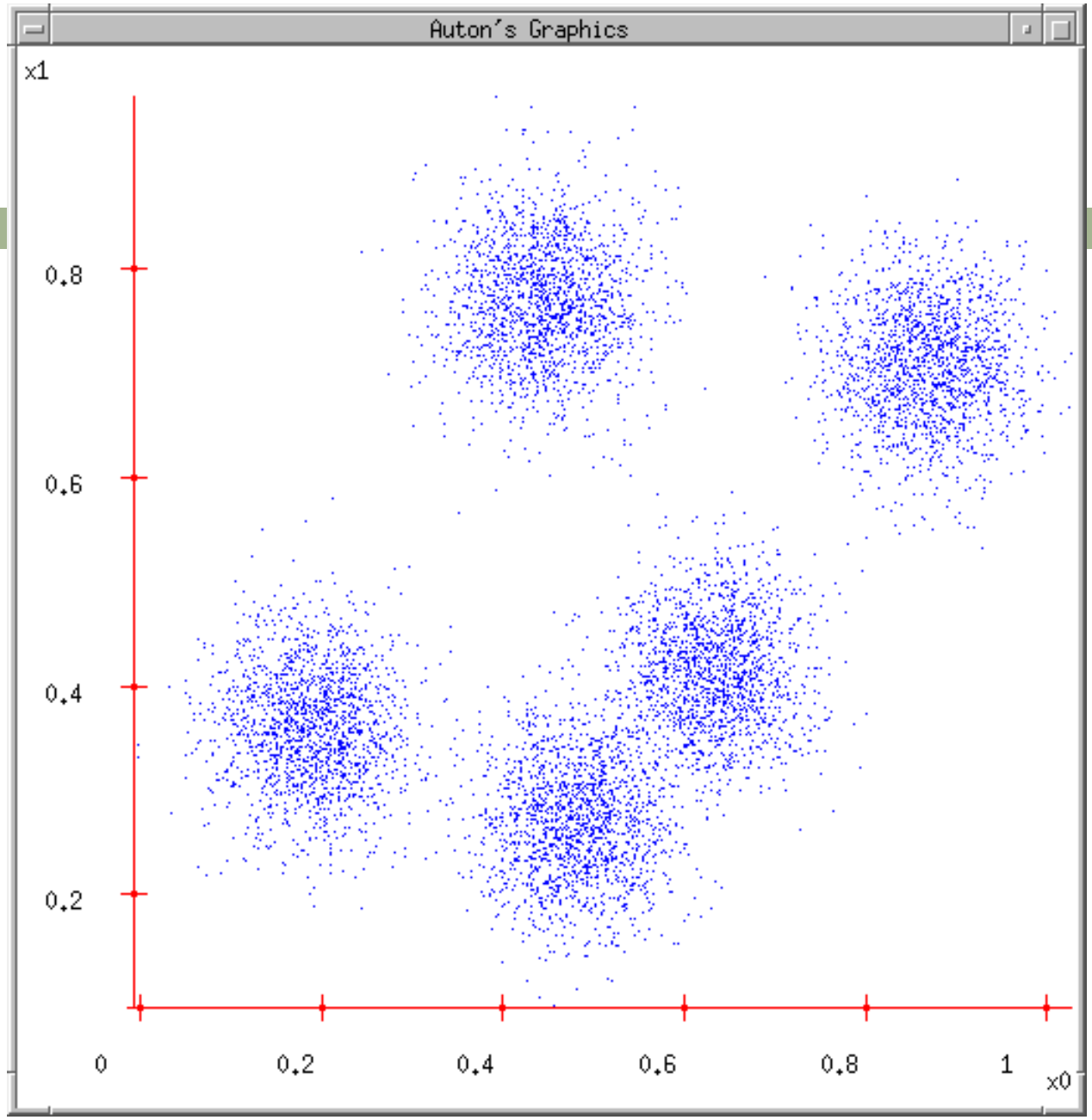



Auton's Graphics
x1
0.8
0.6
0.4
0.2
0
0.2
0.4
0.6
0.8
1
x0

# K-means



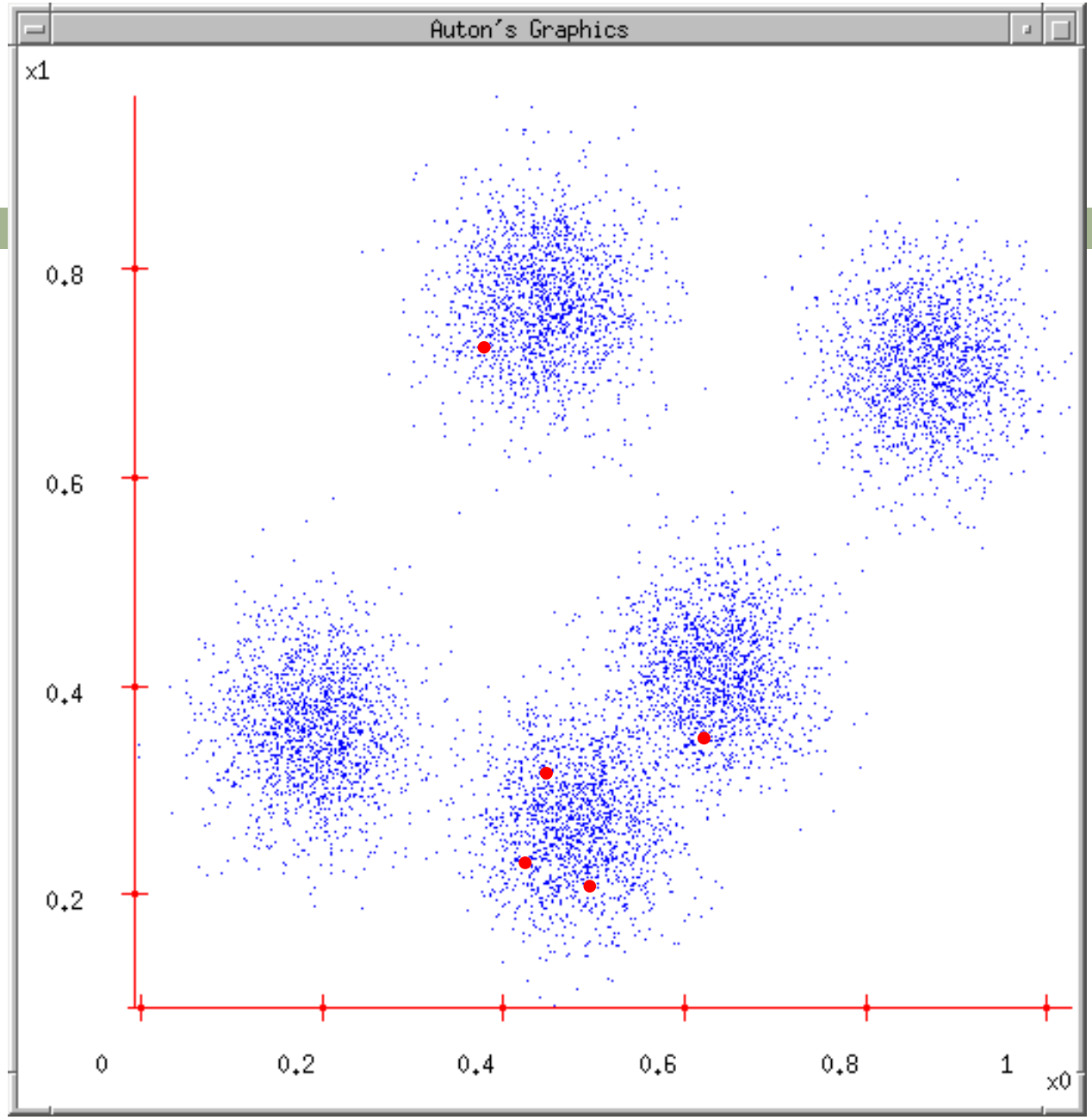
1. Ask user how many clusters they'd like. *(e.g. k=5)*



# K-means



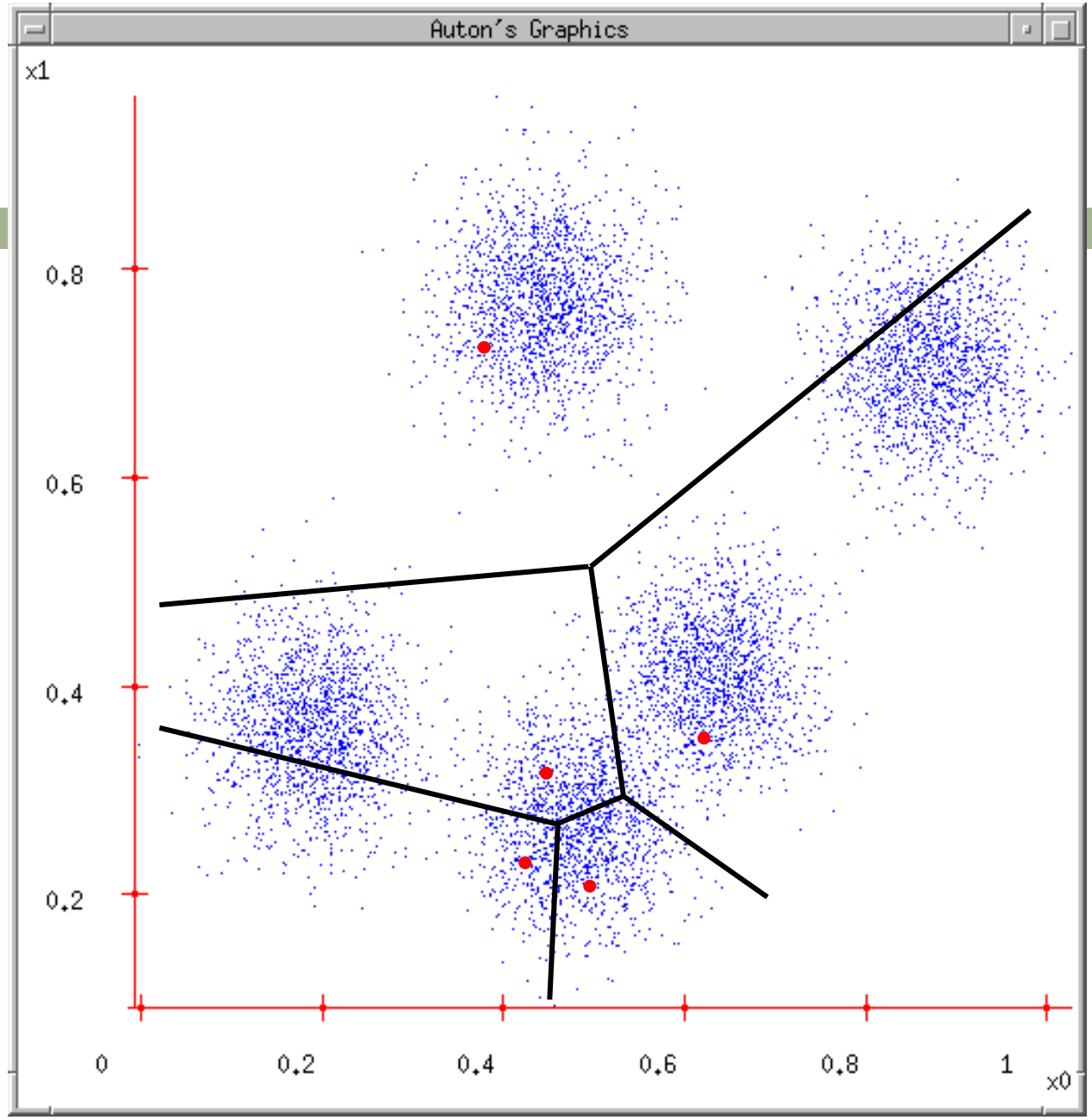
1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations



# K-means



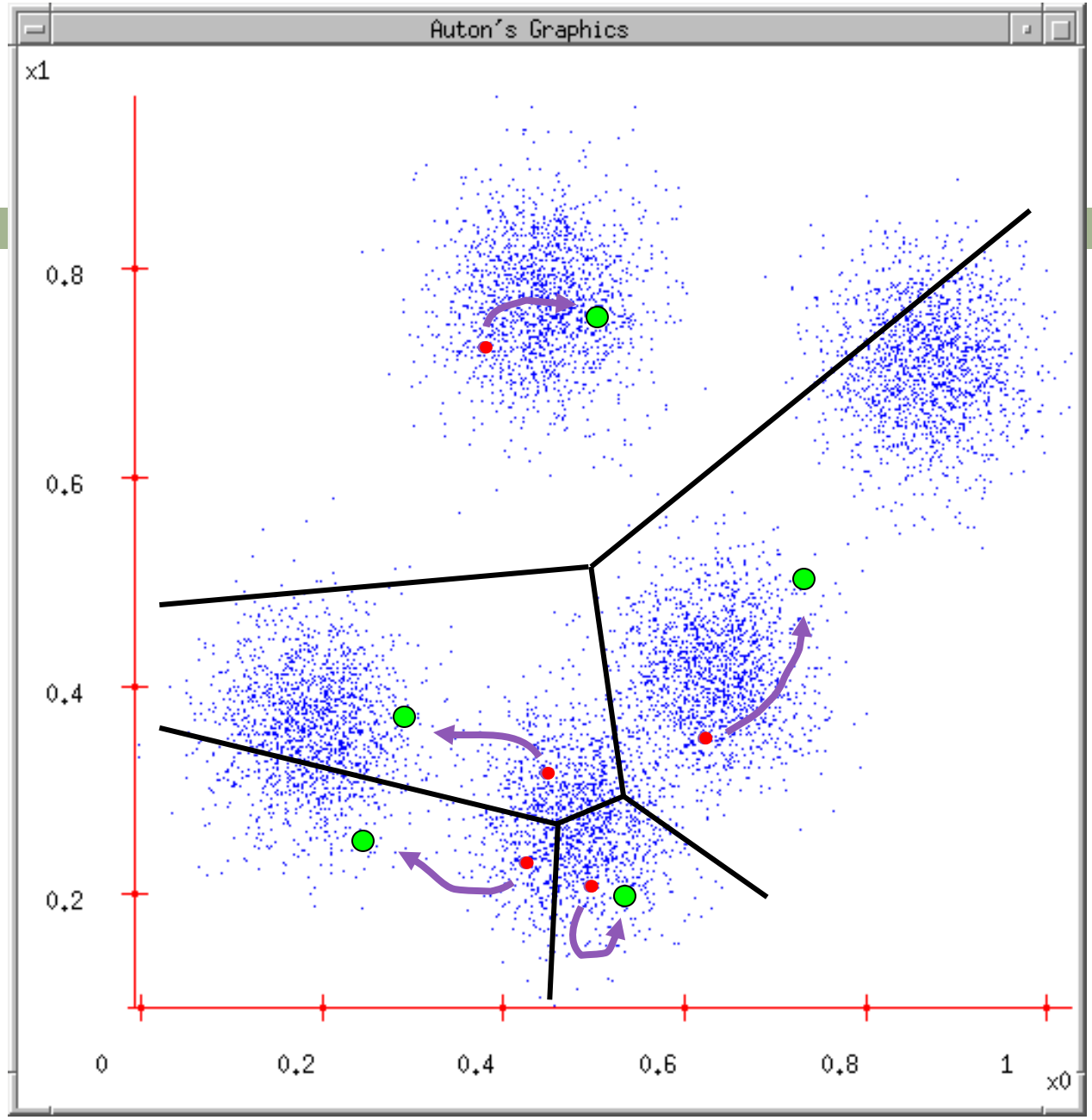
1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to. (Thus each Center "owns" a set of datapoints)



# K-means



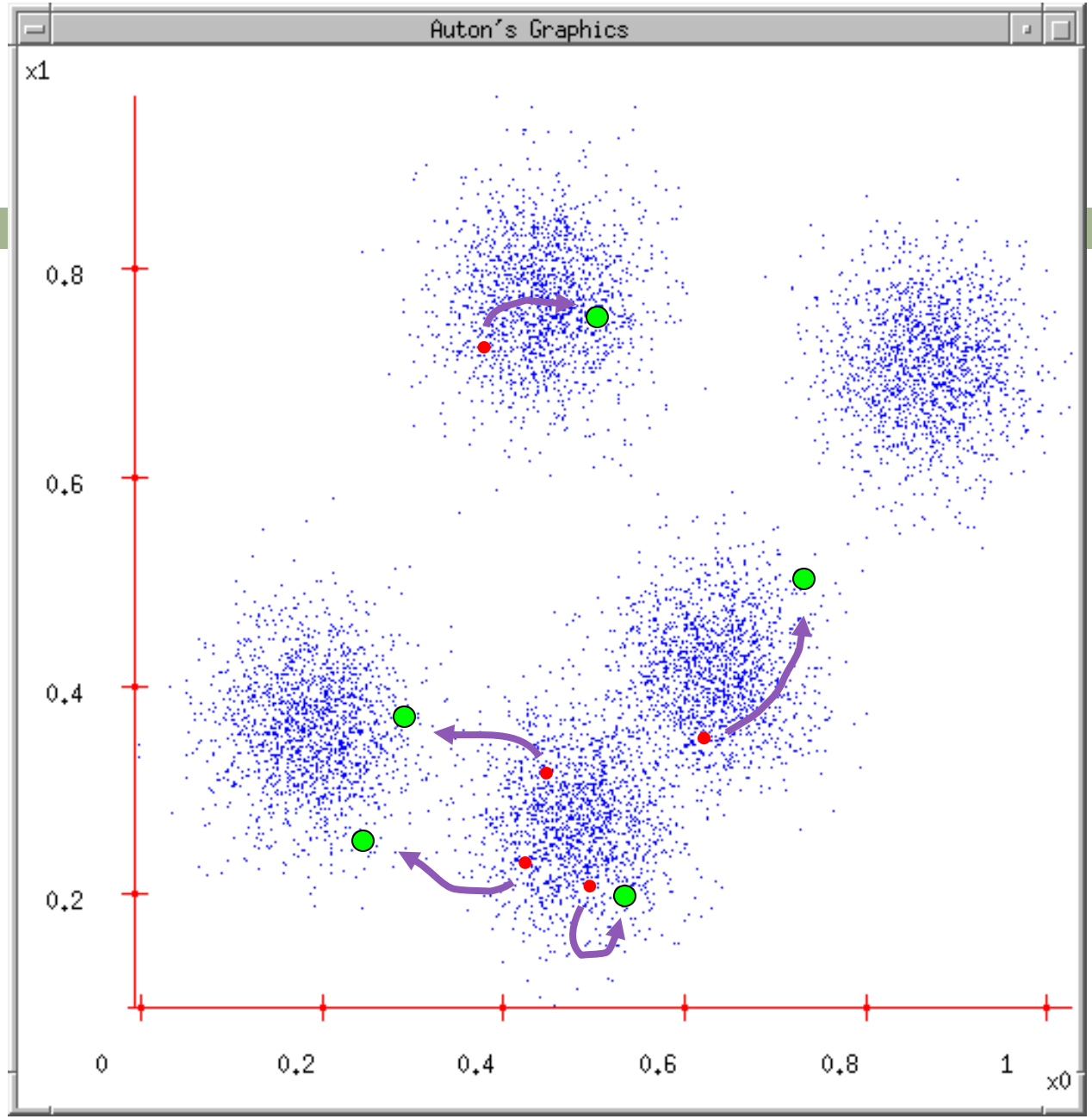
1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns



# K-means



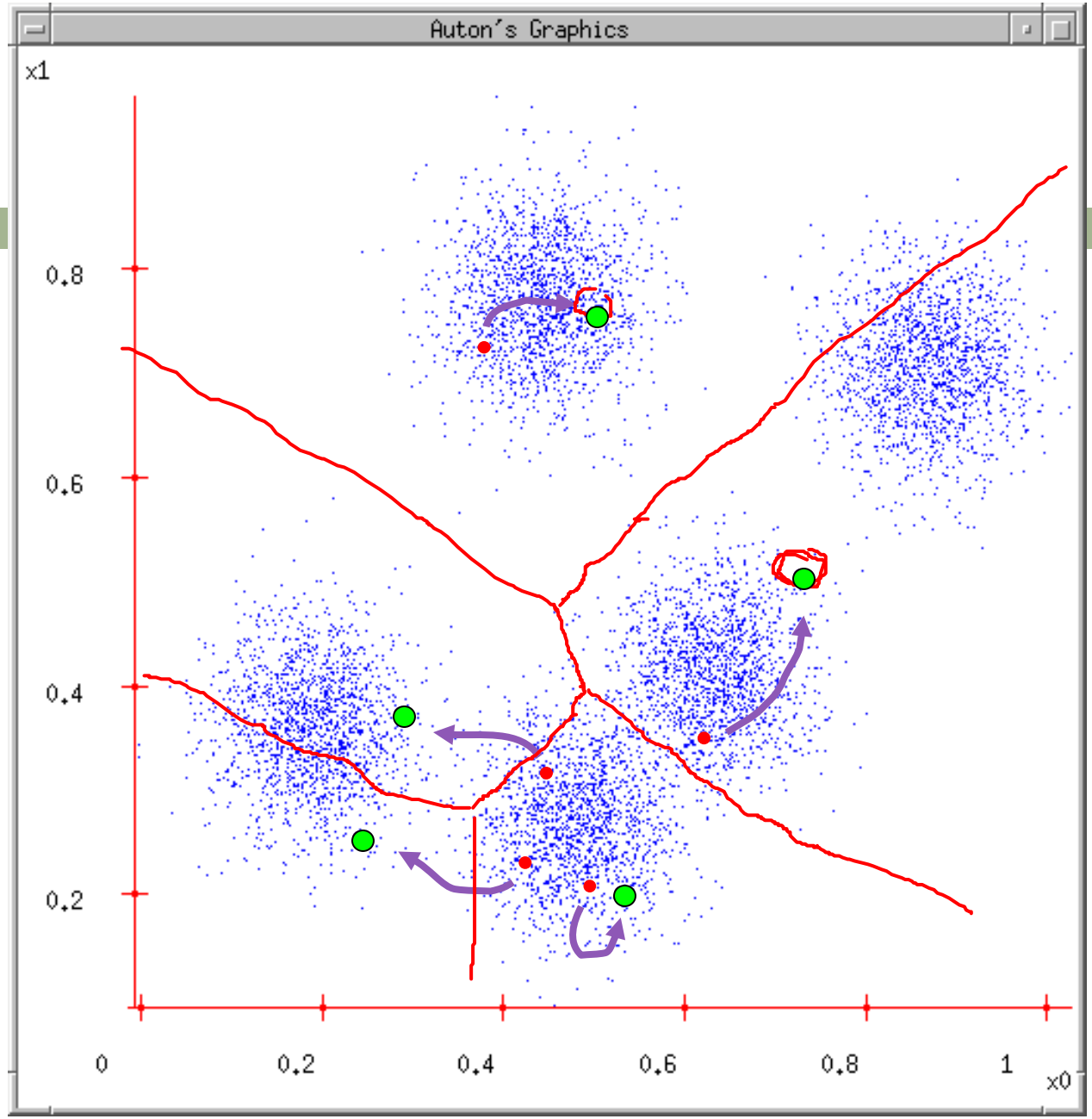
1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns…
5. …and jumps there
6. …Repeat until terminated!



# 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 $\boldsymbol{x}^t \in \mathcal{X}$

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

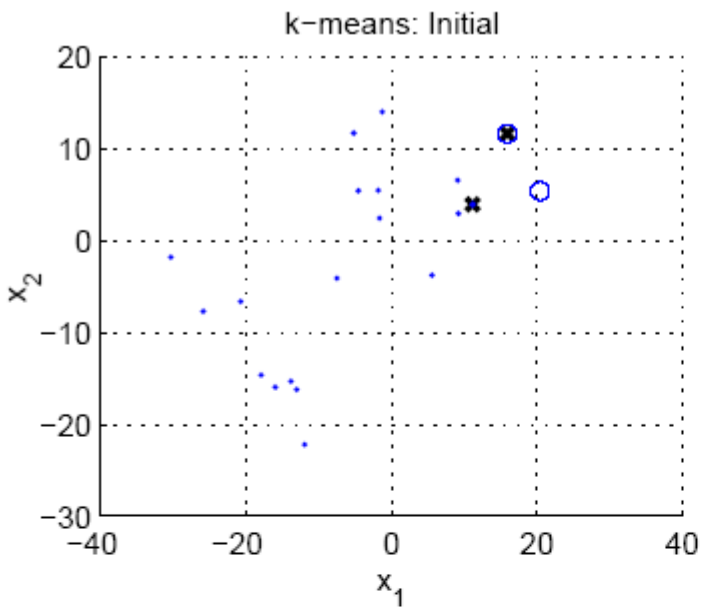
For all $\boldsymbol{m}_i, i = 1, \ldots, k$

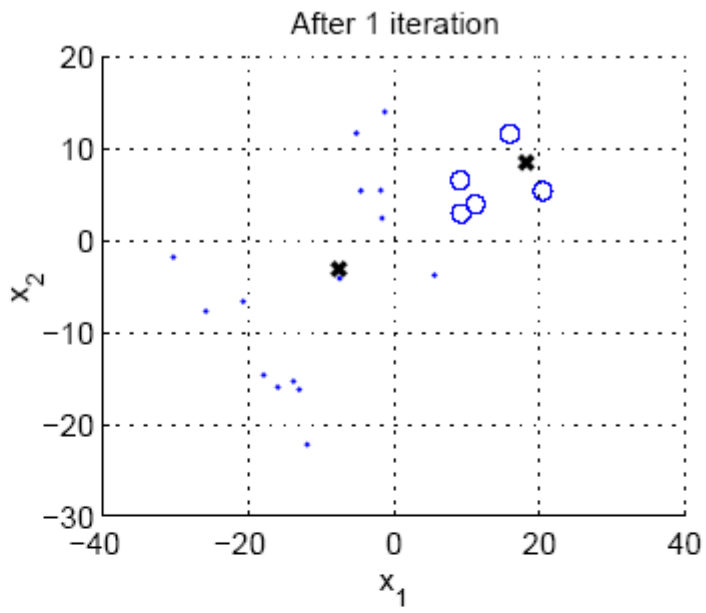
$$\boldsymbol{m}_i \leftarrow \sum_t b_i^t \boldsymbol{x}^t / \sum_t b_i^t$$

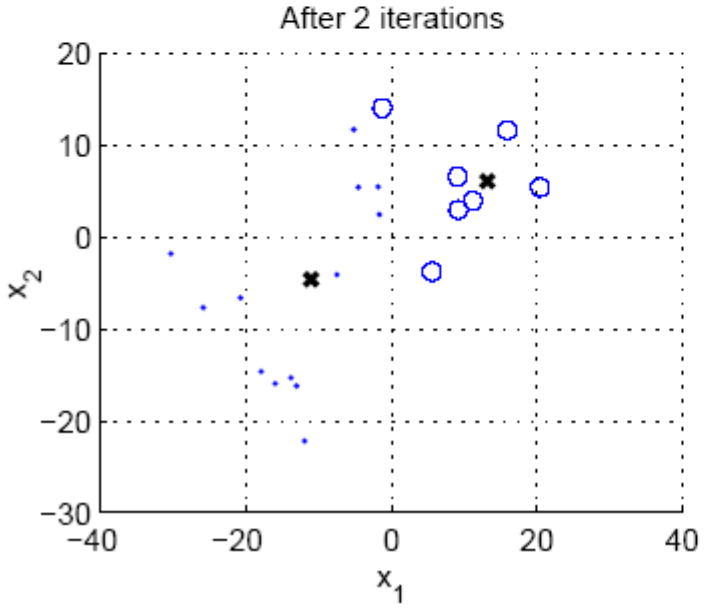
Until $\boldsymbol{m}_i$ converge

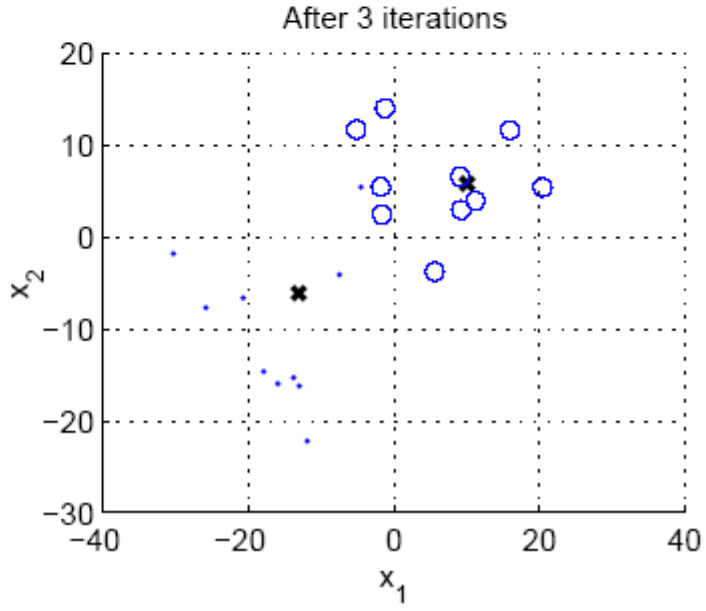
$$E\left(\{\mathbf{m}_i\}_{i=1}^k \middle| \mathcal{X}\right) = \sum_t \sum_i b_i^t \left\|\mathbf{x}^t - \mathbf{m}_i\right\|^2$$

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


k−means: Initial
20
10
0
−10
−20
−30
−40
−20
0
20
40
x2
x1


After 1 iteration
20
10
0
−10
−20
−30
−40
−20
0
20
40
x2
x1


After 2 iterations
20
10
0
−10
−20
−30
−40
−20
0
20
40
x2
x1


After 3 iterations
20
10
0
−10
−20
−30
−40
−20
0
20
40
x2
x1

# Local procedure



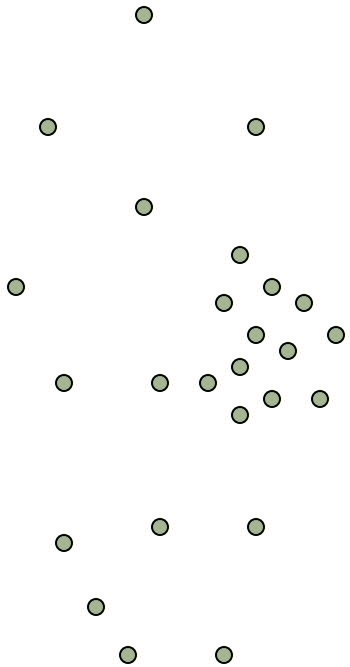
- May converge to suboptimal

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

- 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

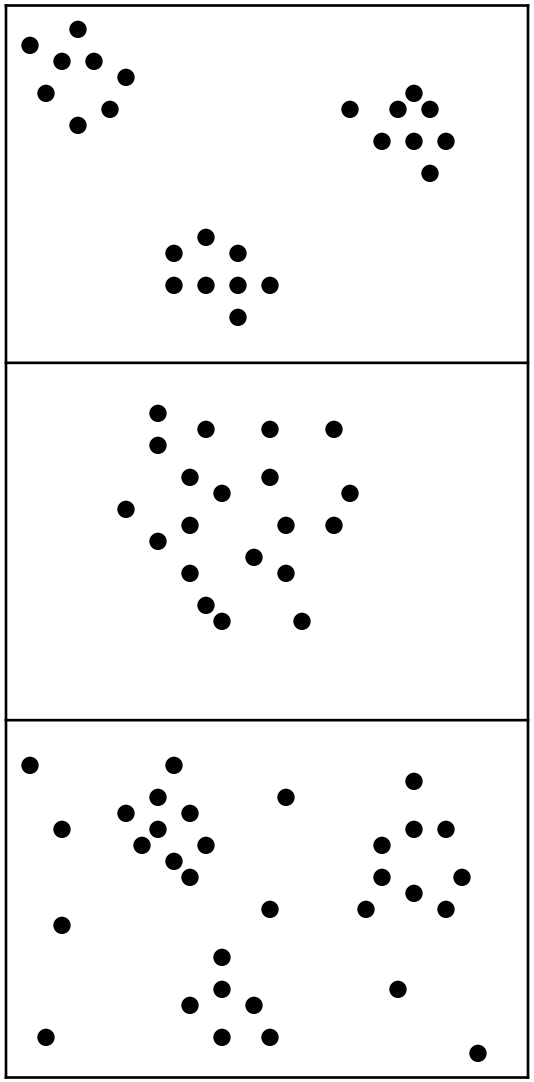




- Clusters may overlap
- Some clusters may be "wider" than others

# Unsupervised Learning





Sometimes easy

Sometimes impossible

and sometimes in between

# Choosing *k*



- Defined by the application, e.g., color quantization
- Plot data (Projection to low dimension) and check for clusters
- Incremental (leader-cluster) algorithm: Add one at a time until “elbow” (reconstruction error/log likelihood/intergroup distances)
- Manually check for meaning

# After Clustering



- Clustering methods find similarities between instances and use it to group instances
- Allows knowledge extraction through
  - number of clusters,
  - prior probabilities,
  - cluster parameters, i.e., center, range of features (demographic/transaction).

  Example: CRM, customer segmentation

# Clustering as Preprocessing



- Estimated group labels $h_i$ (soft) or $b_i$ (hard) may be seen as the dimensions of a new $k$ dimensional space, where we can then learn our discriminant or regressor.
- Local representation (only one $b_i$ is 1, all others are 0; only few $h_i$ are nonzero) vs

  Distributed representation (many $h_i$ are nonzero)

$$h_j \propto \exp\left(-\|\mathbf{x}-\mathbf{m}_j\|^2\right)$$

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

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

# Hierarchical Clustering



- Cluster based on similarities/distances
  - Sometimes easier to define (e.g. sequences)
  - No need of probabilistic perspective (mixture model)
- Distance measure between instances $\mathbf{x}^r$ and $\mathbf{x}^s$

  Minkowski ($L_p$) (Euclidean when $p = 2$)

$$d_m\left(\mathbf{x}^r, \mathbf{x}^s\right) = \left[\sum_{j=1}^{d} \left|x_j^r - x_j^s\right|^p\right]^{1/p}$$

  City-block distance ($p=1$)

$$d_{cb}\left(\mathbf{x}^r, \mathbf{x}^s\right) = \sum_{j=1}^{d} \left|x_j^r - x_j^s\right|$$

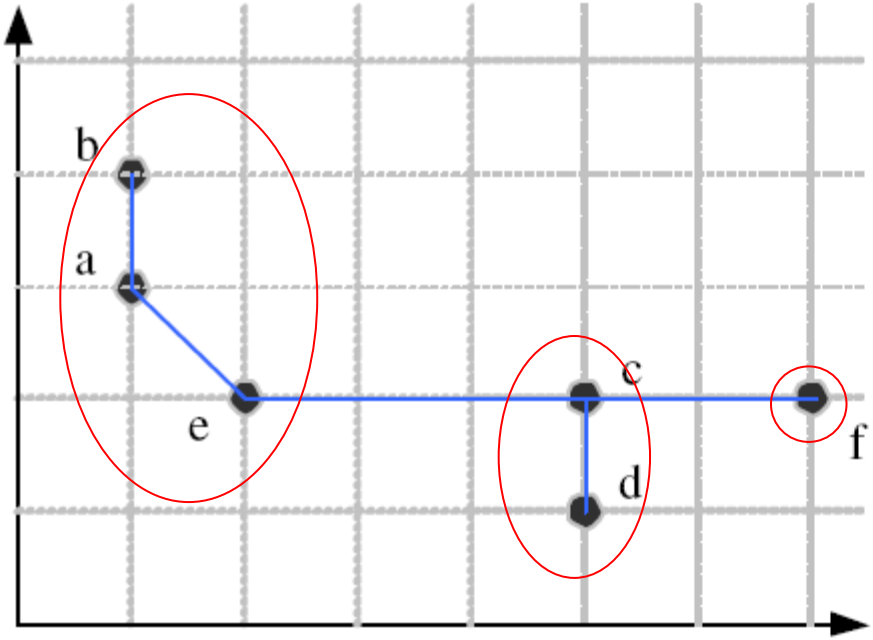
# Agglomerative Clustering

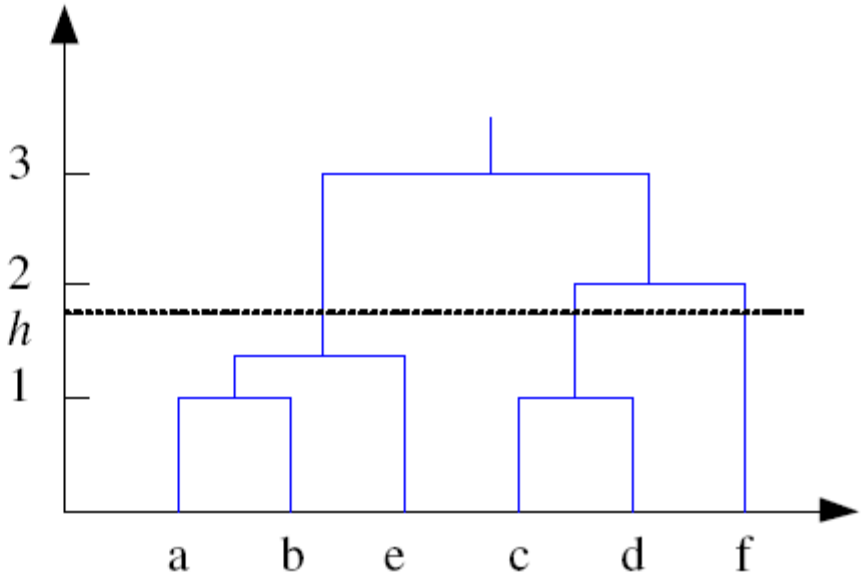


- Start with $N$ groups each with one instance and merge two closest groups at each iteration
- Distance between two groups $G_i$ and $G_j$:
  - Single-link:
$$d(G_i, G_j) = \min_{\mathbf{x}^r \in G_i, \mathbf{x}^s \in G_j} d(\mathbf{x}^r, \mathbf{x}^s)$$
  - Complete-link:
$$d(G_i, G_j) = \max_{\mathbf{x}^r \in G_i, \mathbf{x}^s \in G_j} d(\mathbf{x}^r, \mathbf{x}^s)$$
  - Average-link, centroid
$$d(G_i, G_j) = \text{ave}_{\mathbf{x}^r \in G_i, \mathbf{x}^s \in G_j} d(\mathbf{x}^r, \mathbf{x}^s)$$

# Example: Single-Link Clustering






*Dendrogram*