CSE 446
K-means

Natasha Jaques

CSE 446

- Supervised learning
  - Linear models
    - Linear regression
    - Ridge regression
    - LASSO regression
    - Logistic regression
  - Non-parametric & non-linear
    - Nearest neighbors
    - Trees & Random forests
    - Boosting
    - Kernel methods
  - Neural networks
    - Backpropagation
    - CNN
    - RNN/LSTM
    - Attention/Transformer

X → Y: Previously we tried to predict output Y given input X

- **Unsupervised learning**
  - **Clustering**
    - **k-means**
    - **Gaussian Mixture Models (GMM)**
  - PCA/SVD

Now we are here

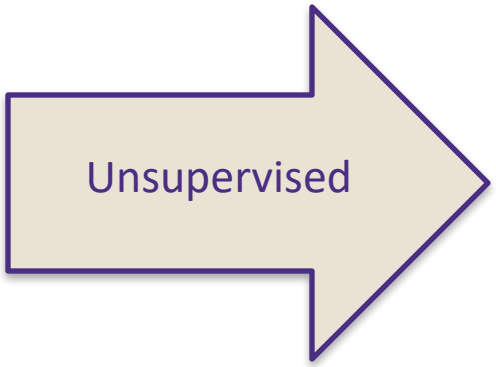
Just model P(X)

No labels Y (hence unsupervised)

Clustering

a fundamental unsupervised task

Clustering images



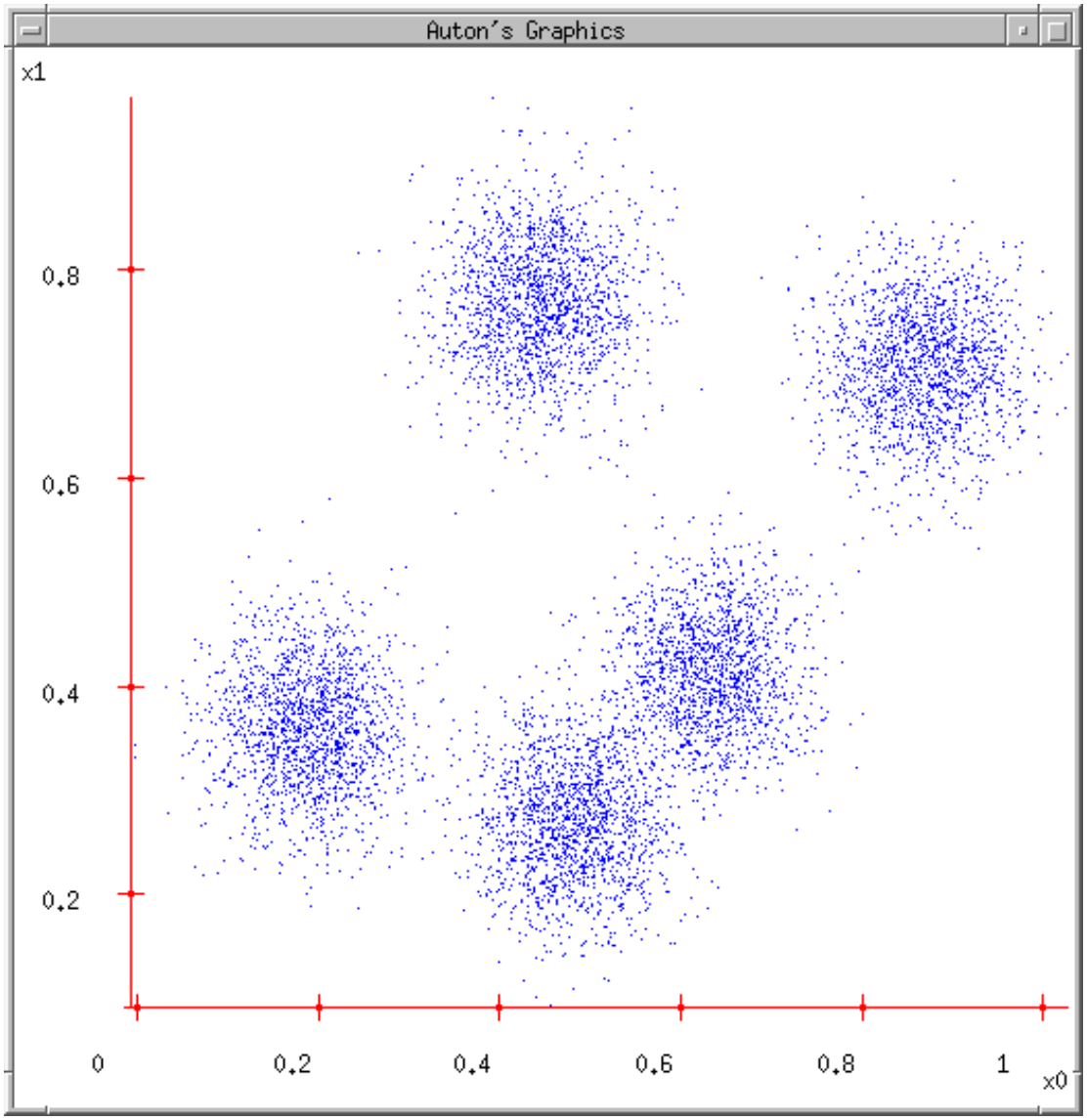
No labels Y

Can we just learn to group similar images?

[Goldberger et al.]

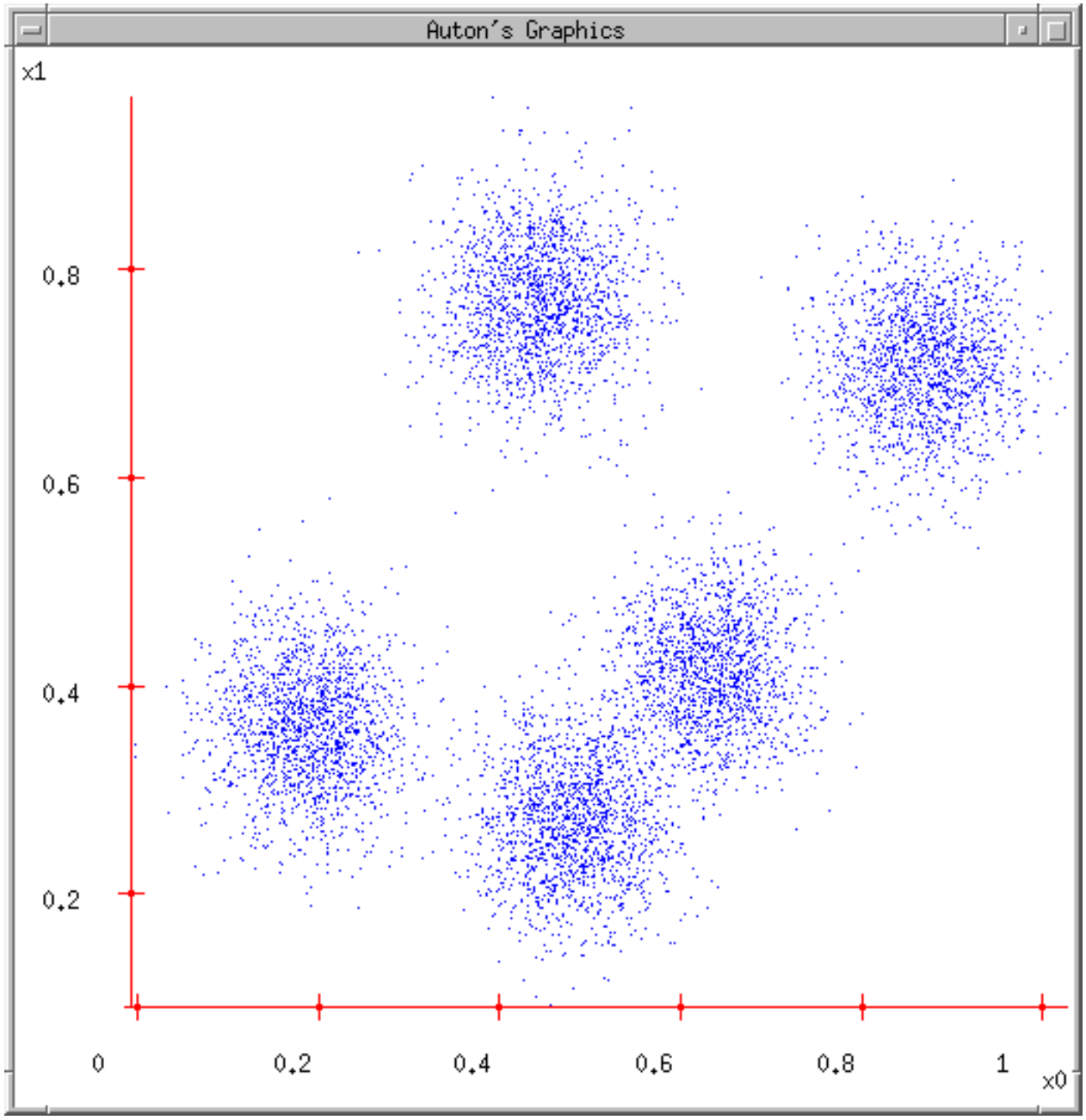
Example of 2-dimensional data points

- k-**means** algorithm assumes this kind of **structured** data, which will be clear once we learn what the algorithm does



k-means

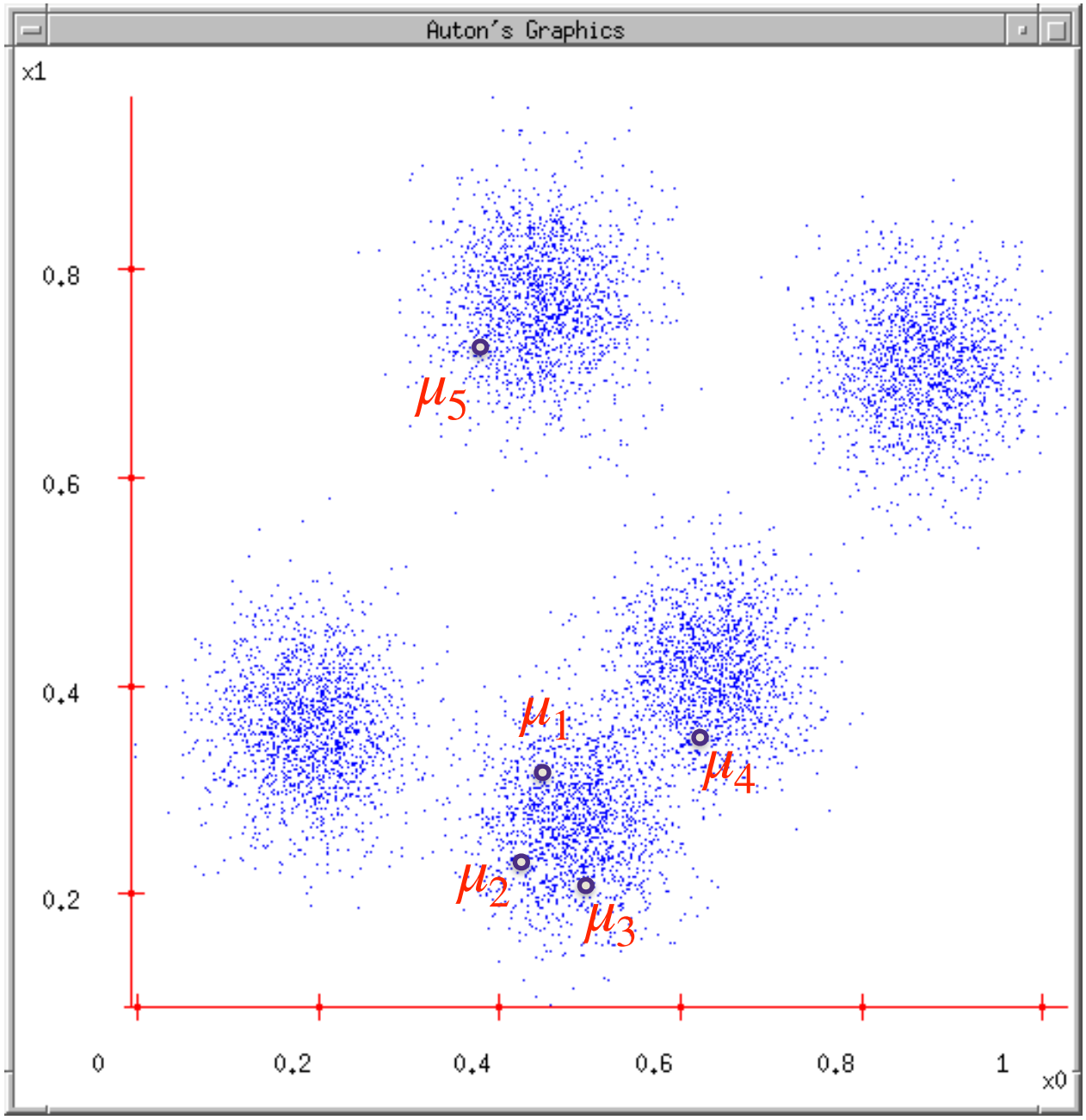
1. Ask user how many clusters they'd like. (e.g. k=5)



k-means

1. Ask user how many clusters they'd like. (e.g. k=5)
2. Randomly guess k cluster Center locations $\{\mu_1, \ldots, \mu_5\}$

Initial guesses do matter

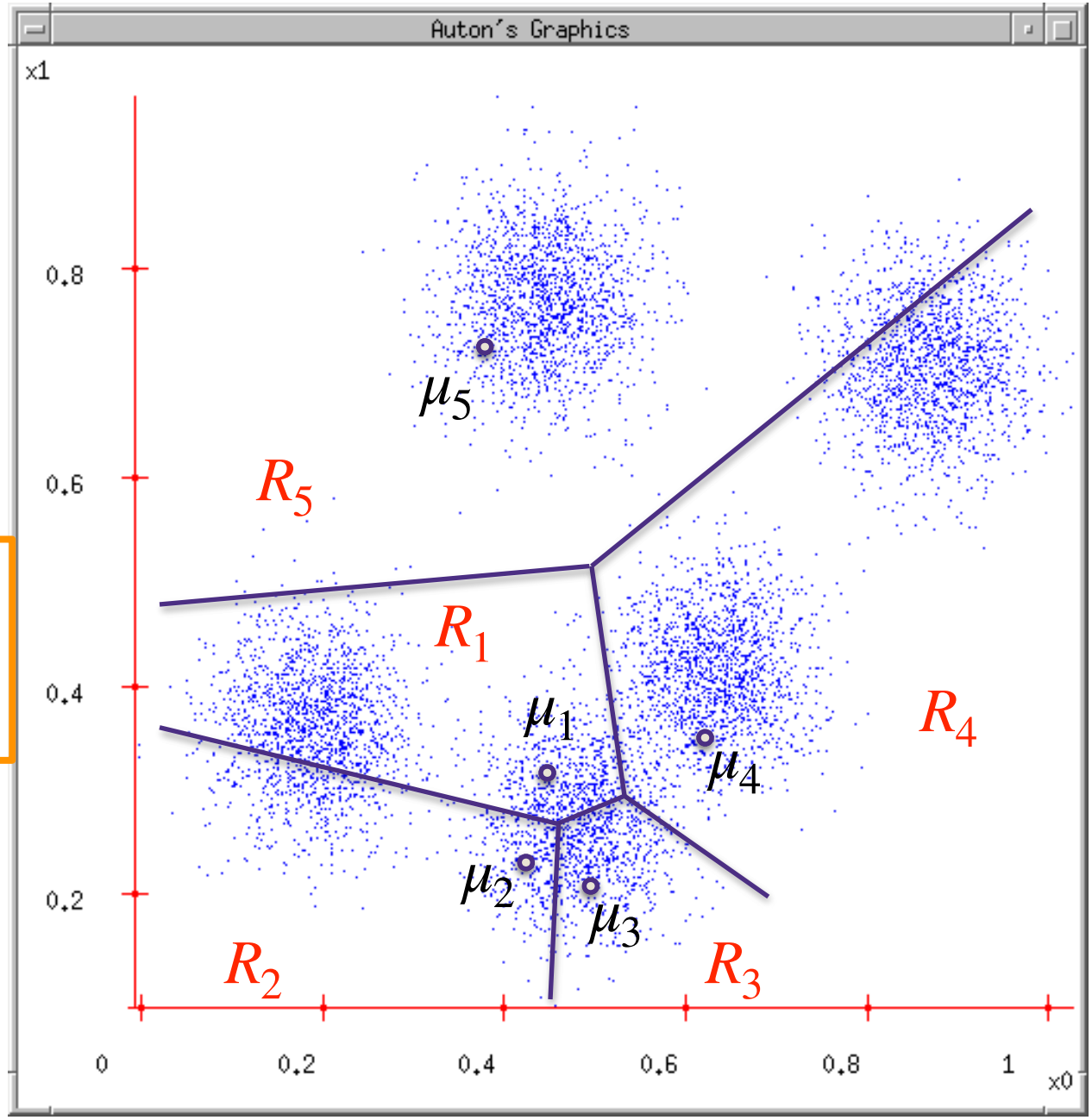


k-means

1. Ask user how many clusters they'd like. (e.g. k=5)
2. Randomly guess k cluster Center locations $\{\mu_1, \ldots, \mu_5\}$
3. Each datapoint finds out which Center it's closest to. (Thus each Center "owns" a set of datapoints)

Given {μ1…μ5}, determine Regions R1…R5.

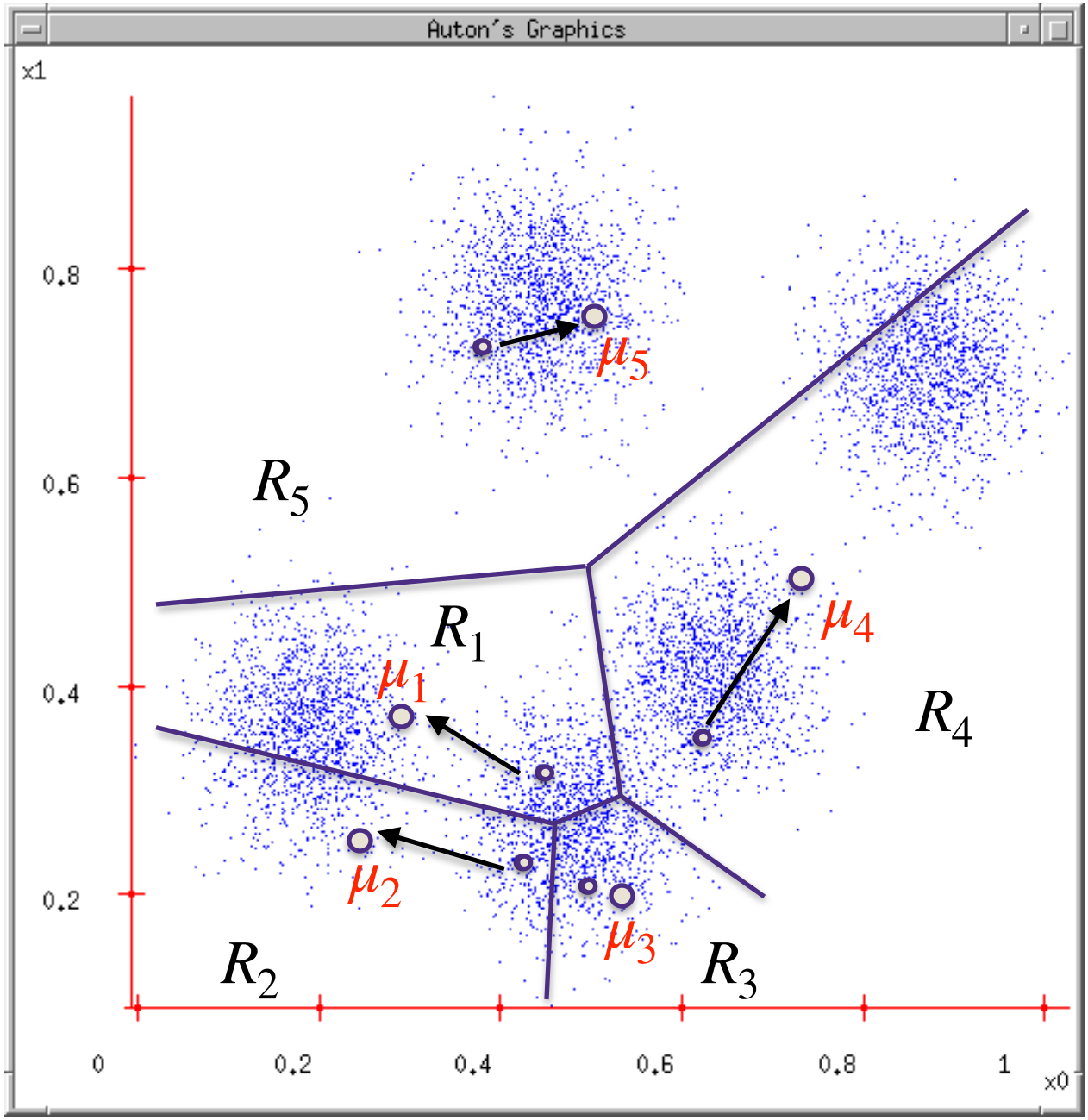
Determine membership of each point to a cluster C



k-means

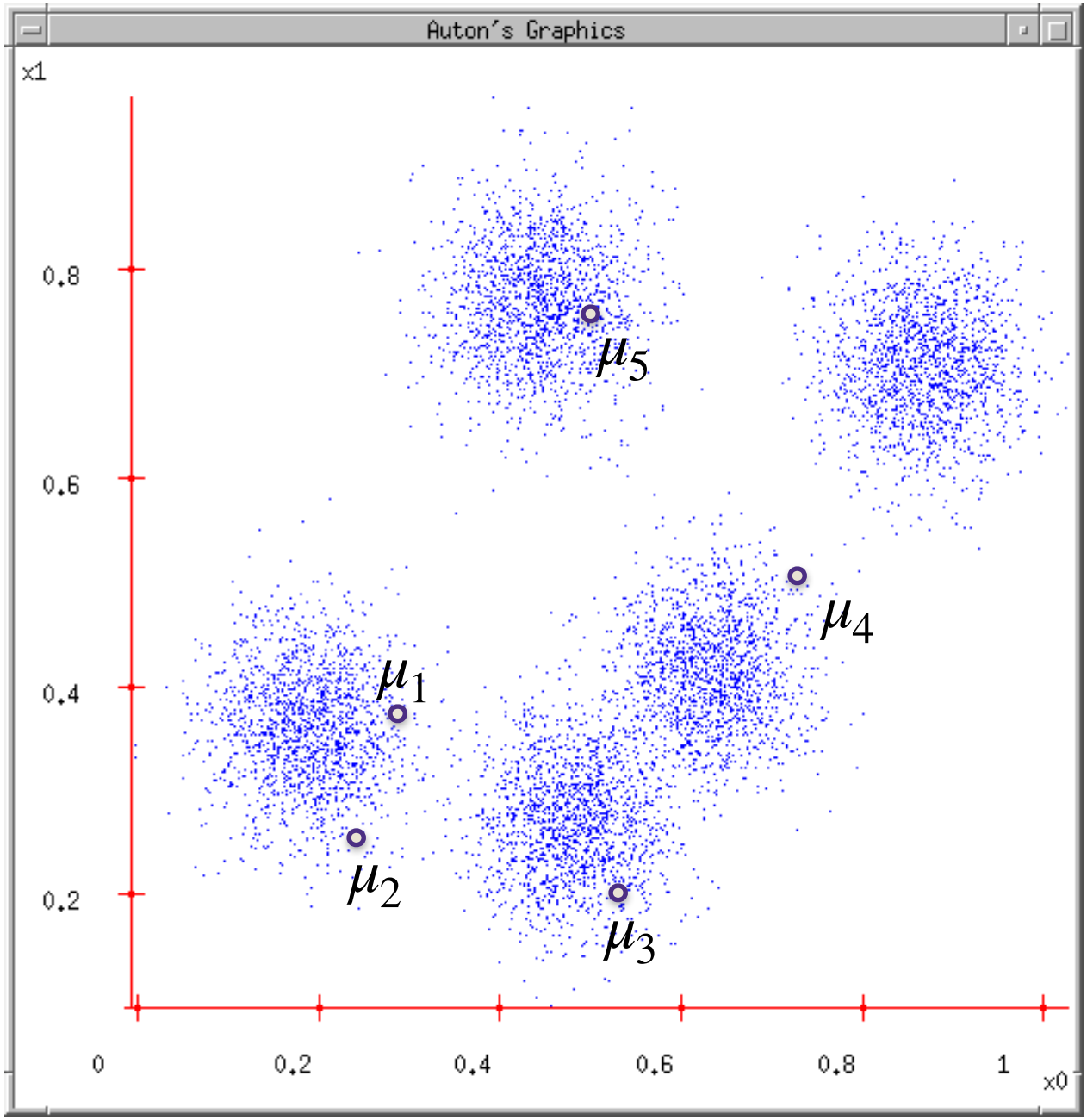
1. Ask user how many clusters they'd like. (e.g. k=5)
2. Randomly guess k cluster Center locations $\{\mu_1, \ldots, \mu_5\}$
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds a new centroid of the points it owns

$\{C_1, \ldots, C_n\}$
$\{R_1, \ldots, R_5\}$ $\rightarrow \{\mu_1, \ldots, \mu_5\}$



k-means

1. Ask user how many clusters they'd like. (e.g. k=5)
2. Randomly guess k cluster Center locations $\{\mu_1, \ldots, \mu_5\}$
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds a new centroid of the points it owns…
5. …and jumps there
6. …Repeat until terminated!

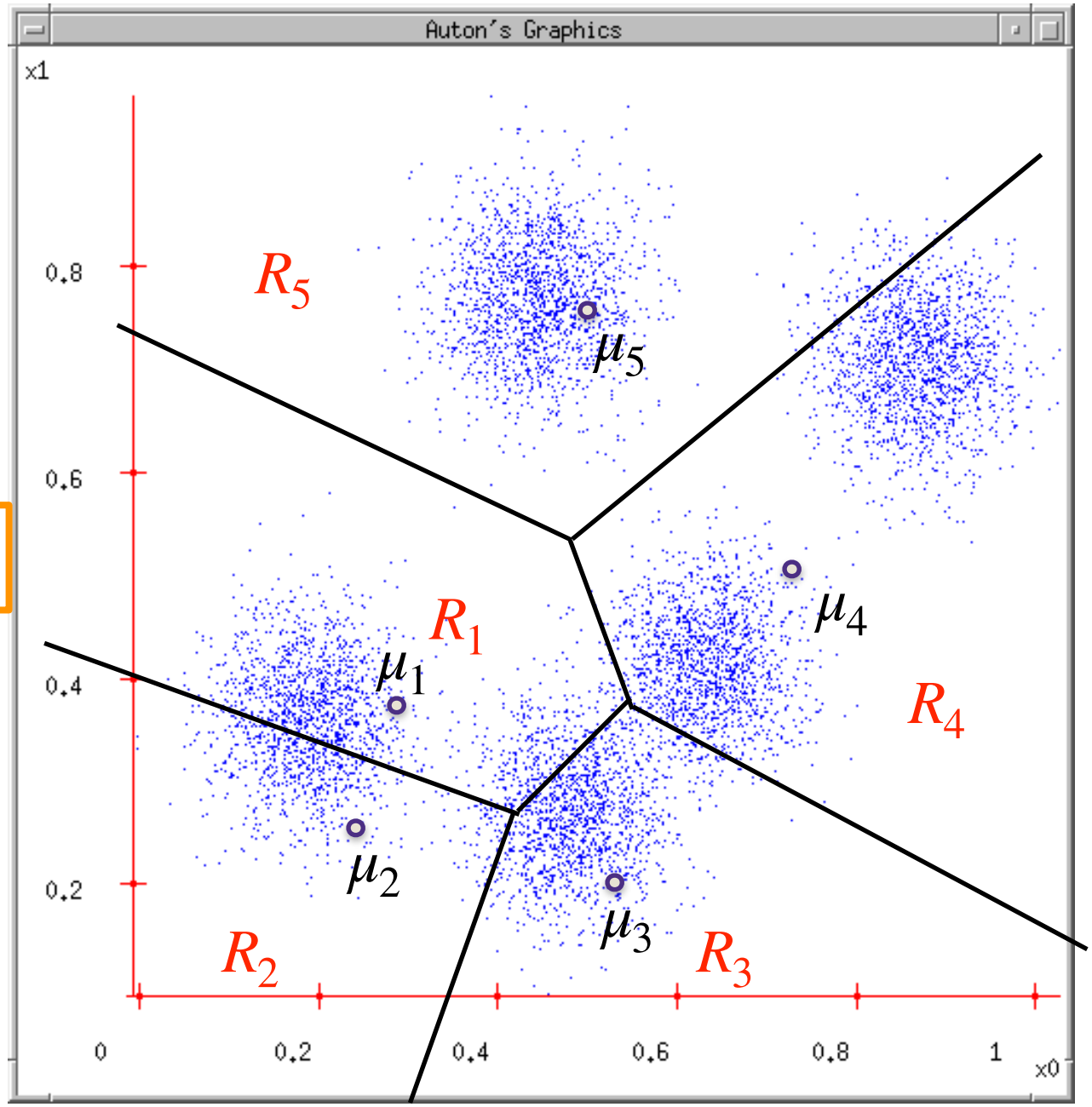


k-means

1. Ask user how many clusters they'd like. (e.g. k=5)
2. Randomly guess k cluster Center locations $\{\mu_1, \ldots, \mu_5\}$
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!

Terminated = points' membership stops changing

Was this the clustering you expected?



k-means

https://www.naftaliharris.com/blog/visualizing-k-means-clustering/

K-means algorithm

1. Choose k, how many clusters to find
2. Randomly initialize k centers
   - $\mu^{(0)} = [\mu_1^{(0)}, \ldots, \mu_k^{(0)}] \in \mathbb{R}^{d \times k}$
   - Usually randomly chosen from the data points, to make sure they are in the right domain

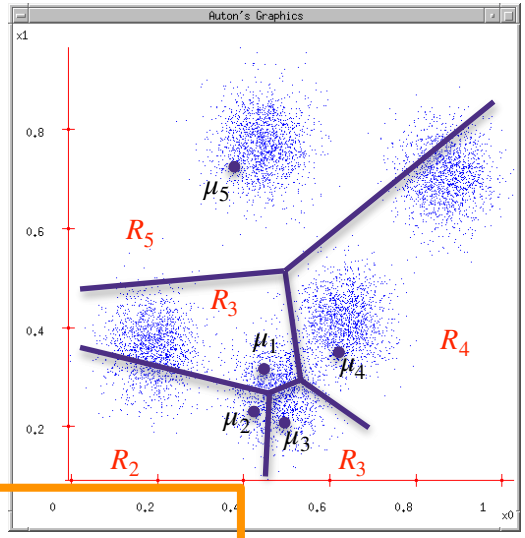
- For t=0,1,2,… repeat

K-means

1. Choose k, how many clusters to find
2. Randomly initialize k centers
   - $\mu^{(0)} = [\mu_1^{(0)}, \ldots, \mu_k^{(0)}] \in \mathbb{R}^{d \times k}$
   - Usually randomly chosen from the data points, to make sure they are in the right domain
- For t=0,1,2,… repeat
3. Assign each point j∈{1,…,n} to its nearest center:

$$\underbrace{C_j^{(t)}}_{\text{Membership of the j-th data point}} \leftarrow \arg\min_{i \in \{1,2,\ldots,k\}} \|x_j - \mu_i^{(t)}\|^2$$

Assignment:



K-means

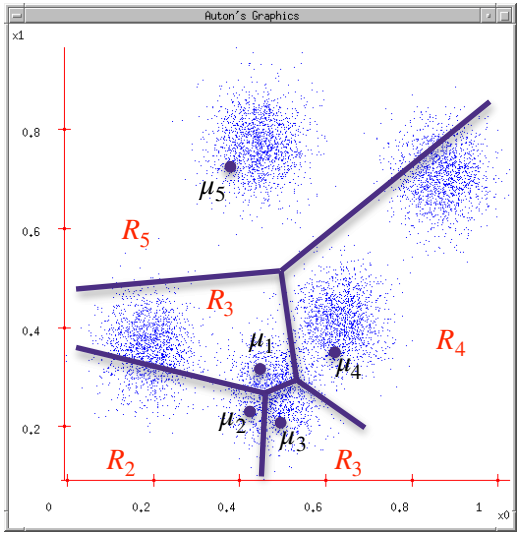
1. Choose k, how many clusters to find
2. Randomly initialize k centers
   - $\mu^{(0)} = [\mu_1^{(0)}, \ldots, \mu_k^{(0)}] \in \mathbb{R}^{d \times k}$
   - Usually randomly chosen from the data points, to make sure they are in the right domain
- For t=0,1,2,… repeat
3. Assign each point j∈{1,…,n} to its nearest center:

$$C_j^{(t)} \leftarrow \arg\min_{i \in \{1,2,\ldots,k\}} \|x_j - \mu_i^{(t)}\|^2$$

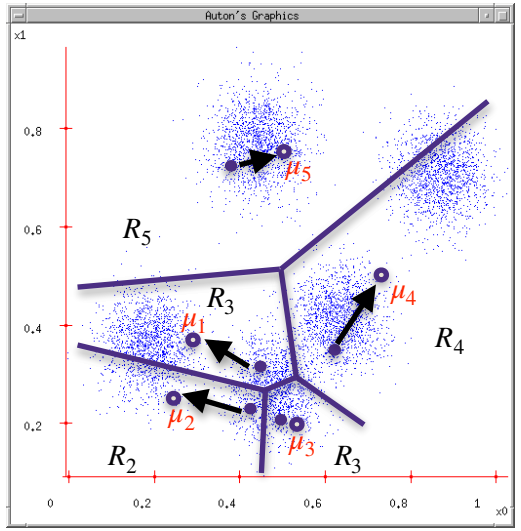
4. Recenter: μ_i becomes centroid of its point:

$$\mu_i^{(t+1)} \leftarrow \arg\min_{\mu} \sum_{j:C_j^{(t)}=i} \|\mu - x_j\|^2$$

Assignment:



Recenter:



K-means

1. Choose k, how many clusters to find
2. Randomly initialize k centers
   - $\mu^{(0)} = [\mu_1^{(0)}, \ldots, \mu_k^{(0)}] \in \mathbb{R}^{d \times k}$
   - Usually randomly chosen from the data points, to make sure they are in the right domain
- For t=0,1,2,… repeat
3. Assign each point j∈{1,…,n} to its nearest center:

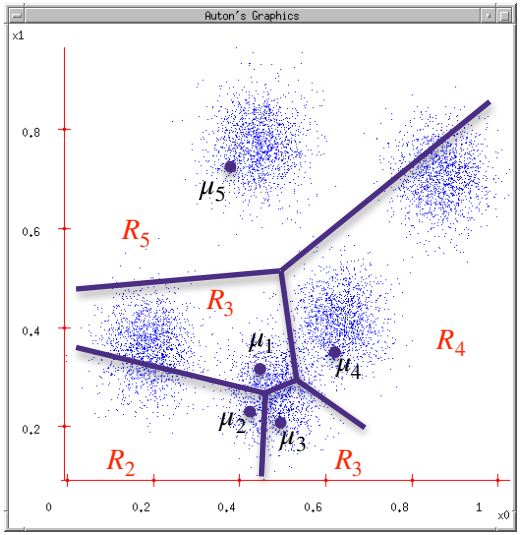
$$C_j^{(t)} \leftarrow \arg\min_{i \in \{1,2,\ldots,k\}} \|x_j - \mu_i^{(t)}\|^2$$

4. Recenter: μ_i becomes centroid of its point:

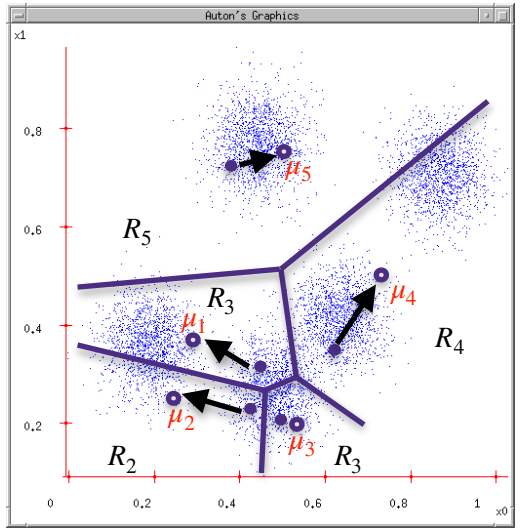
$$\mu_i^{(t+1)} \leftarrow \arg\min_{\mu} \sum_{j: C_j^{(t)} = i} \|\mu - x_j\|^2$$

- Equivalent to
  $\mu_i^{(t+1)} \leftarrow$ average of all the points assigned to $\mu_i^{(t)}$

Assignment:

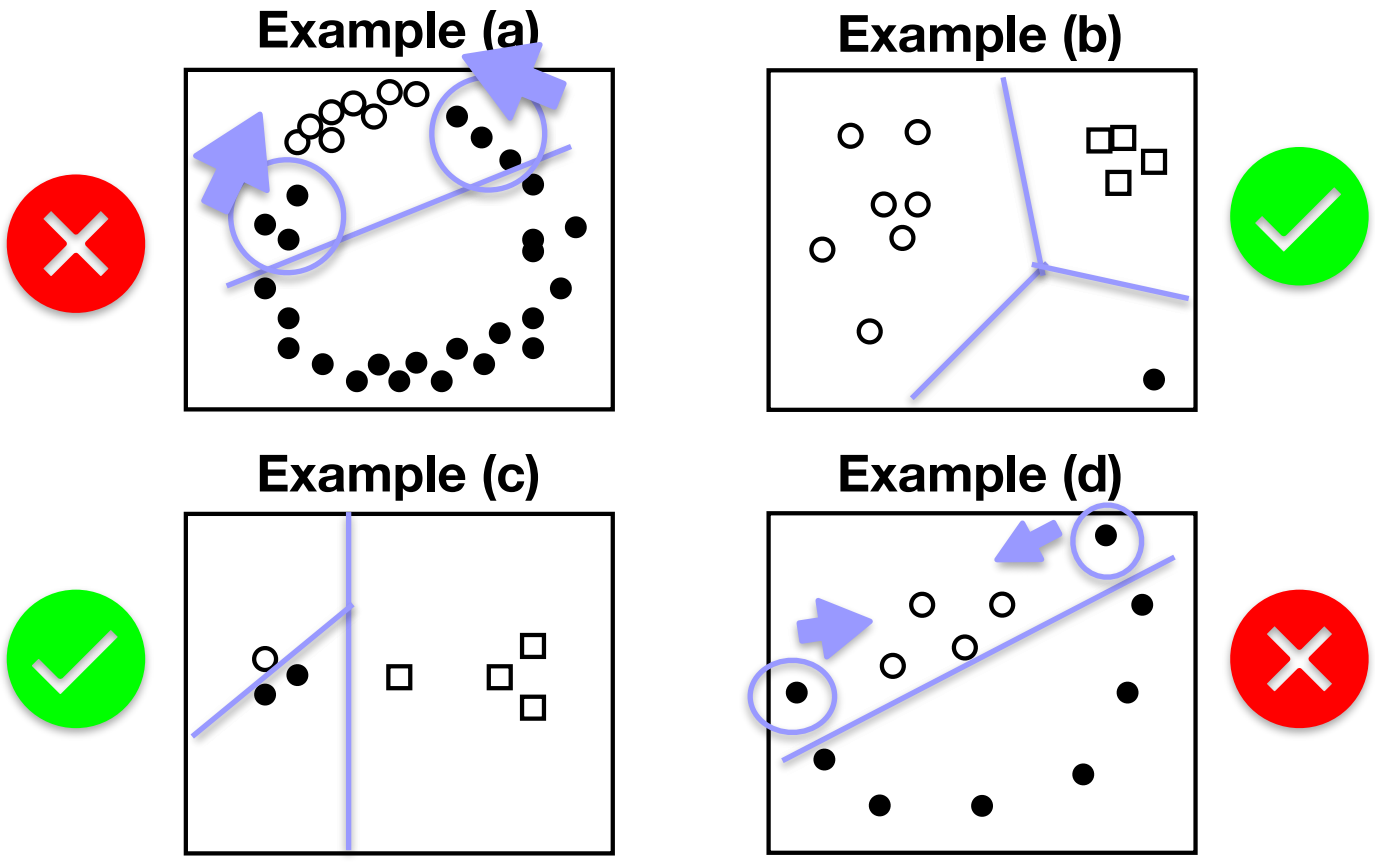


Recenter:



Which one is a snapshot of a converged k-means

When k-means is converged, there should be a set of centers and assignments that do not change when applying 1 step of k-means



Does k-means converge?

- k-means is trying to minimize the following objective

$$\min_{\{\mu_i\}_{i=1}^{k}} \min_{\{C_\ell\}_{\ell=1}^{n}} \sum_{j=1}^{n} \|x_j - \mu_{C_j}\|^2$$

via alternating minimization
(equivalent to coordinate descent)

- Fix μ, optimize C # Convex
- Fix C, optimize μ # Convex

"Expectation Maximization"
(also used for GMM)

- Does this converge? Does this terminate in finite time?

Does k-means converge?

- there is only a finite set of values that $\{C_j\}_{j=1}^n \in \{1,\ldots,k\}^n$ can take (k^n is large but finite)
- so there is only finite, k^n at most, values for cluster-centers, $\{\mu_i\}_{i=1}^k$, also
- each time we update them, we will never increase the objective function:

$$\sum_{j=1}^{n} \|x_j - \mu_{C_j}\|_2^2$$

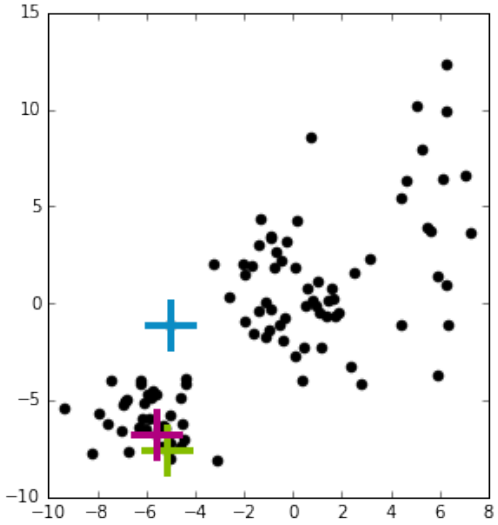
- the objective is lower bounded by zero
- after at most k^n steps, the algorithm must converge
  (as the assignments $\{C_j\}_{j=1}^n$ cannot return to previous assignments in the course of k-means iterations)

downsides of k-means

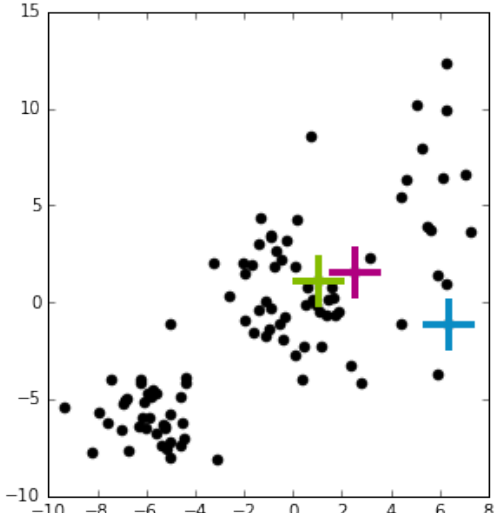
1. it requires the number of clusters k to be specified by us
2. the final solution depends on the initialization
   (does not find global minimum of the objective)

Initial position of centers

Trial 1



Trial 2

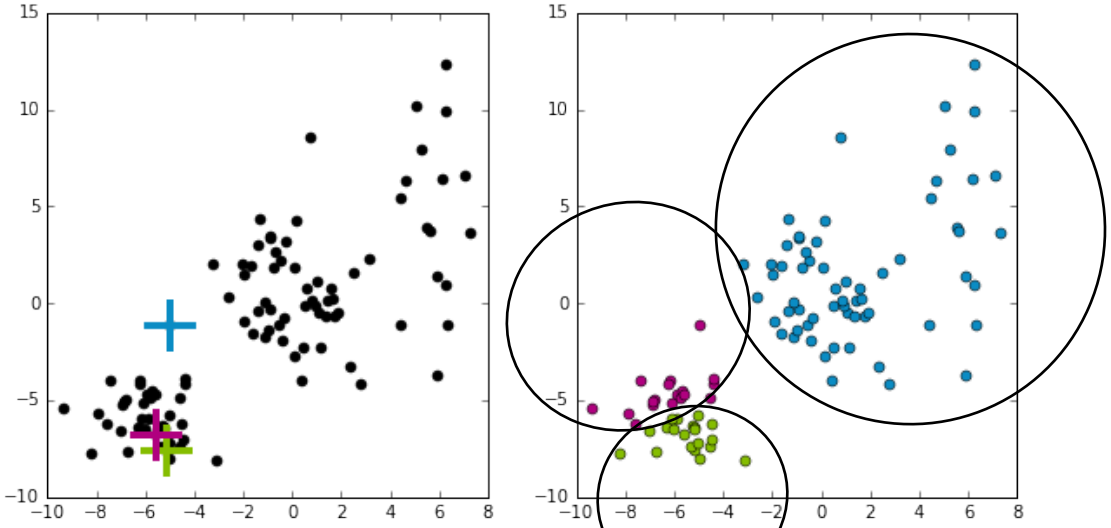


downsides of k-means

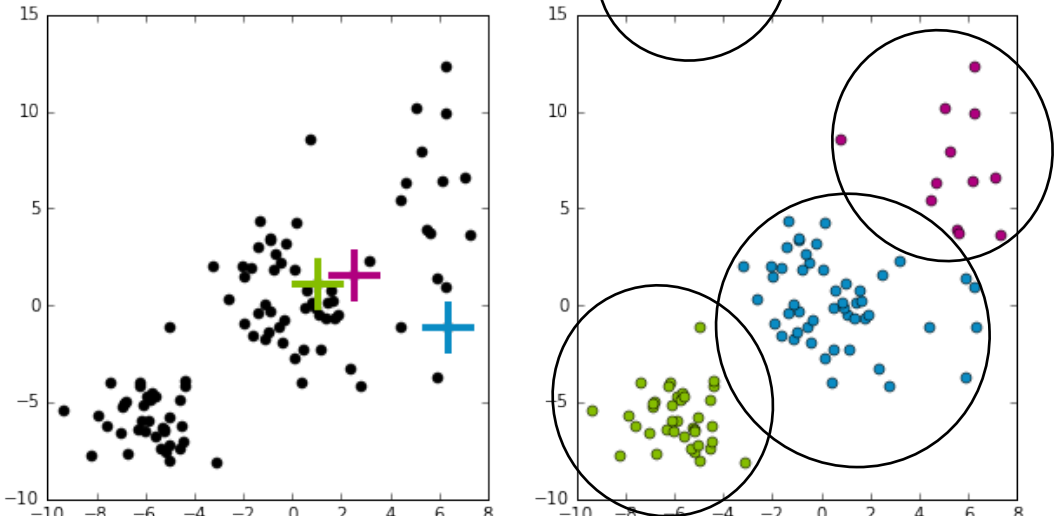
1. it requires the number of clusters k to be specified by us
2. the final solution depends on the initialization
   (does not find global minimum of the objective)

Initial position of centers final converged assignment

Trial 1



Trial 2



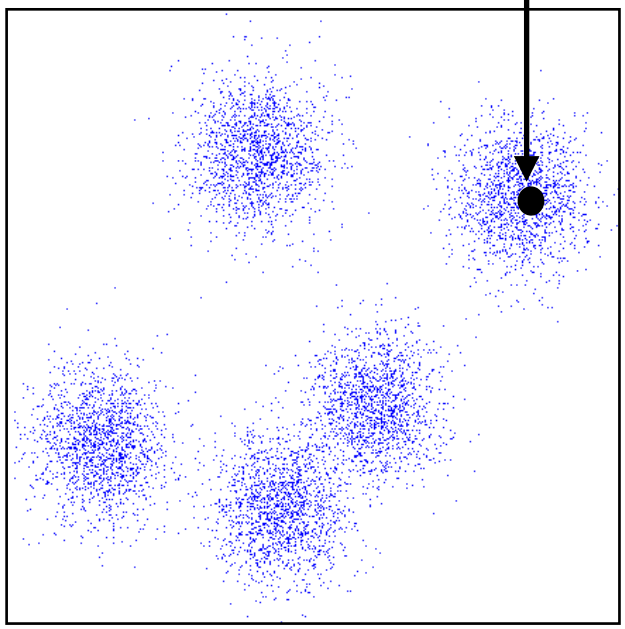
How could we improve initialization?

Choose initial points by incorporating distance and number of points

k-means++: a smart initialization

Smart initialization:

1. Choose **first** cluster center μ_1 uniformly at random from data points



k-means++: a smart initialization

Smart initialization:

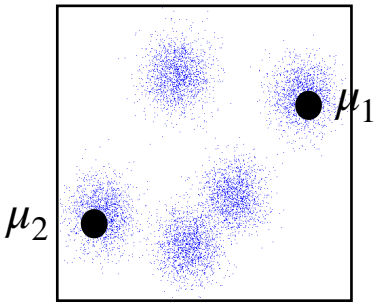
1. Choose **first** cluster center μ_1 uniformly at random from data points
2. For k=2,… K # for each initial cluster center

3. For each data point x_i, compute distance d_i to nearest cluster center
4. Choose new cluster center from amongst data points, with probability of x_i being chosen proportional to $(d_i)^2$

precisely,

$$d_i \leftarrow \min_{j \in \{1,\ldots k-1\}} \|\mu_j - x_i\|, \text{ for all } i \text{ that is not chosen already}$$

$$\text{Prob}(x_i \text{ chosen as the next center}) = \frac{(d_i)^2}{\sum_\ell (d_\ell)^2}$$



k-means++: a smart initialization

Smart initialization:

1. Choose **first** cluster center μ_1 uniformly at random from data points
2. For k=2,… K

   3. For each data point x_i, compute distance d_i to nearest cluster center

   4. Choose new cluster center from amongst data points, with probability

of x_i being chosen proportional to $(d_i)^2$

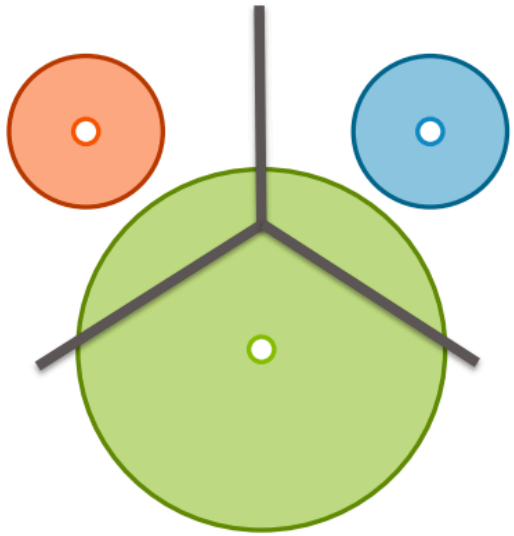
precisely,

$$d_i \leftarrow \min_{j \in \{1,\ldots k-1\}} \|\mu_j - x_i\|, \text{ for all } i \text{ that is not chosen already}$$

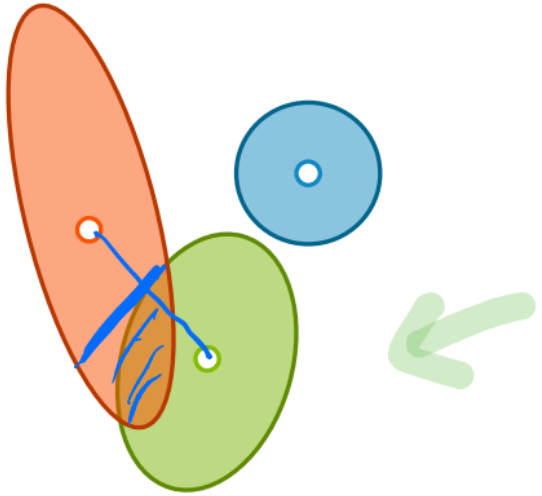
$$\text{Prob}(x_i \text{ chosen as the next center}) = \frac{(d_i)^2}{\sum_\ell (d_\ell)^2}$$

- apply standard K-means after this initialization

- K-means algorithm fails, when the data has:



disparate cluster sizes



different shaped/oriented clusters

- What can we do?

CSE 446

- Supervised learning
  - Linear models
    - Linear regression
    - Ridge regression
    - LASSO regression
    - Logistic regression
  - Non-parametric & non-linear
    - Nearest neighbors
    - Trees & Random forests
    - Boosting
    - Kernel methods
  - Neural networks
    - Backpropagation
    - CNN
    - RNN/LSTM
    - Attention/Transformer

- Unsupervised learning
  - Clustering
    - k-means
    - **Gaussian Mixture Models (GMM)**
  - PCA/SVD