COMP9321:
Data services engineering

Week 8: Clustering

Term3, 2019

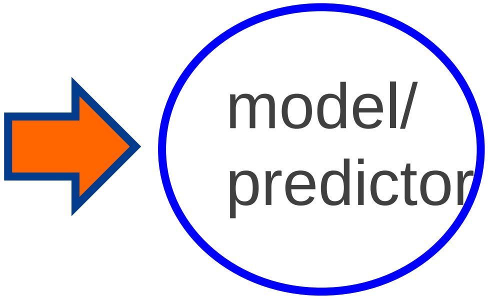
By Mortada Al-Banna, CSE UNSW

Supervised learning

label

label$_1$

label$_3$



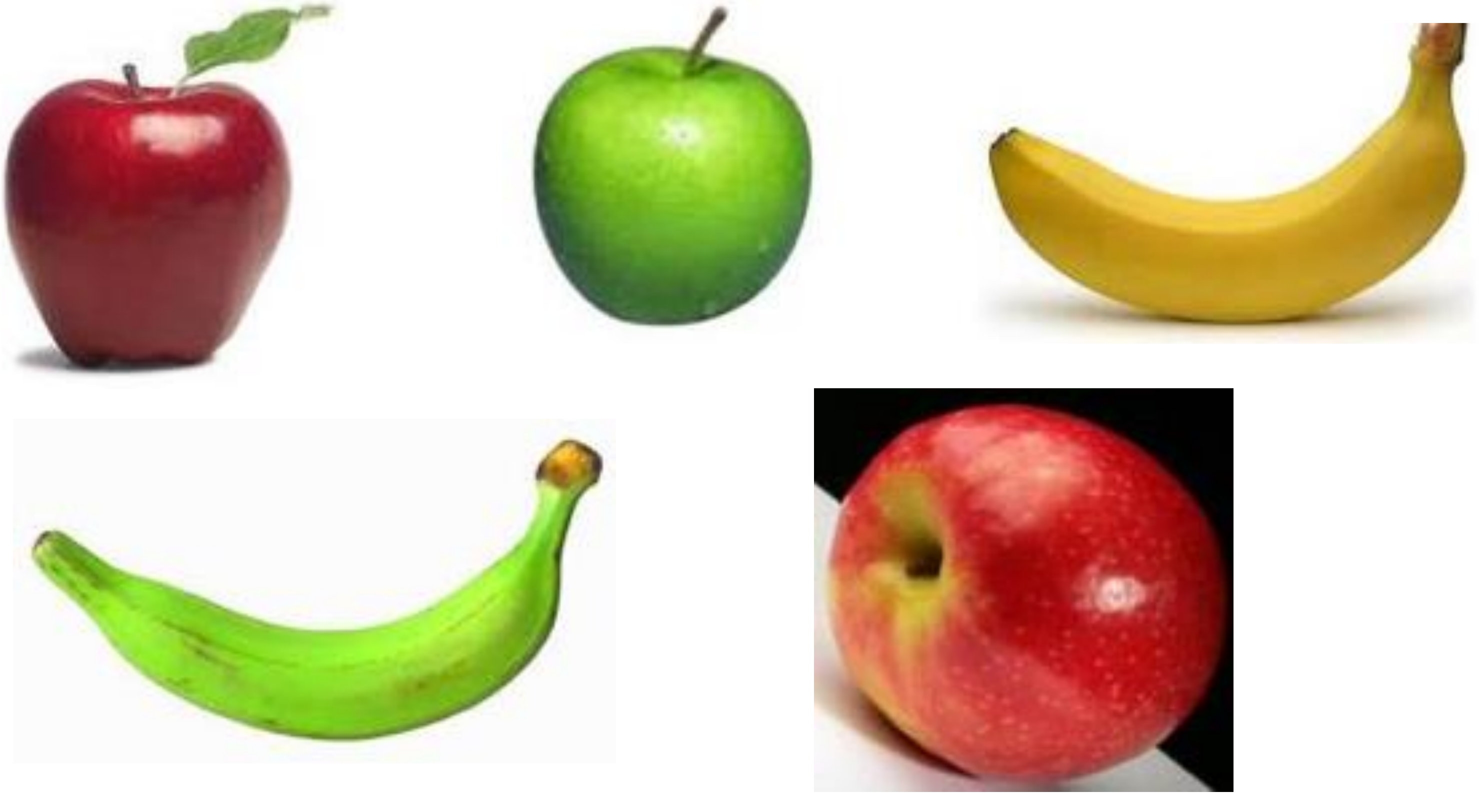
model/ predictor

label$_4$

label$_5$

Supervised learning: given labeled examples

Unsupervised learning



Unupervised learning: given data, i.e. examples, but no labels

Unsupervised Learning

Definition of Unsupervised Learning:

> Learning useful structure *without* labeled classes, optimization criterion, feedback signal, or any other information beyond the raw data

Unsupervised Learning

- Unsupervised learning involves operating on datasets without labelled responses or target values.
- The goal is to capture a structure of interest of useful information (e.g., relationships)
- Unsupervised learning good be used in:
  - ❑Visualizing the structure of a complex dataset
  - ❑Compressing and summarising the data (e.g, image compression)
  - ❑Extracting features for supervised learning
  - ❑Discover groups or outliers

Clustering

Unsupervised Learning

Clustering

– Unsupervised learning

– Requires data, but no labels

– Detect patterns

Motivations of Clustering

- exploratory data analysis

– understanding general characteristics of data

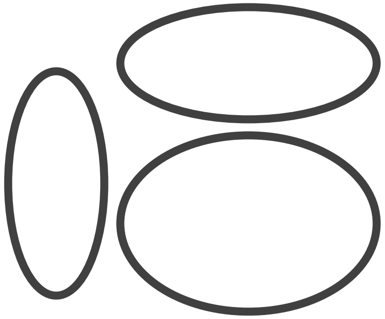
– visualizing data

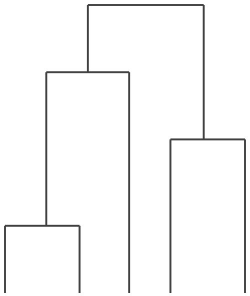
- generalization – infer something about an instance (e.g. a gene) based on how it relates to other instances

Paradigms

Flat algorithms

- Usually start with a random (partial) partitioning
- Refine it iteratively
  - *K* means clustering
  - Model based clustering
- Spectral clustering





Hierarchical algorithms

- Bottom-up, agglomerative
- Top-down, divisive

Paradigms

Hard clustering: Each example belongs to exactly one cluster

Soft clustering: An example can belong to more than one cluster (probabilistic)

- Makes more sense for applications like creating browsable hierarchies
- You may want to put a pair of sneakers in two clusters: (i) sports apparel and (ii) shoes

Clustering: Image Segmentation

Break up the image into meaningful or perceptually similar regions

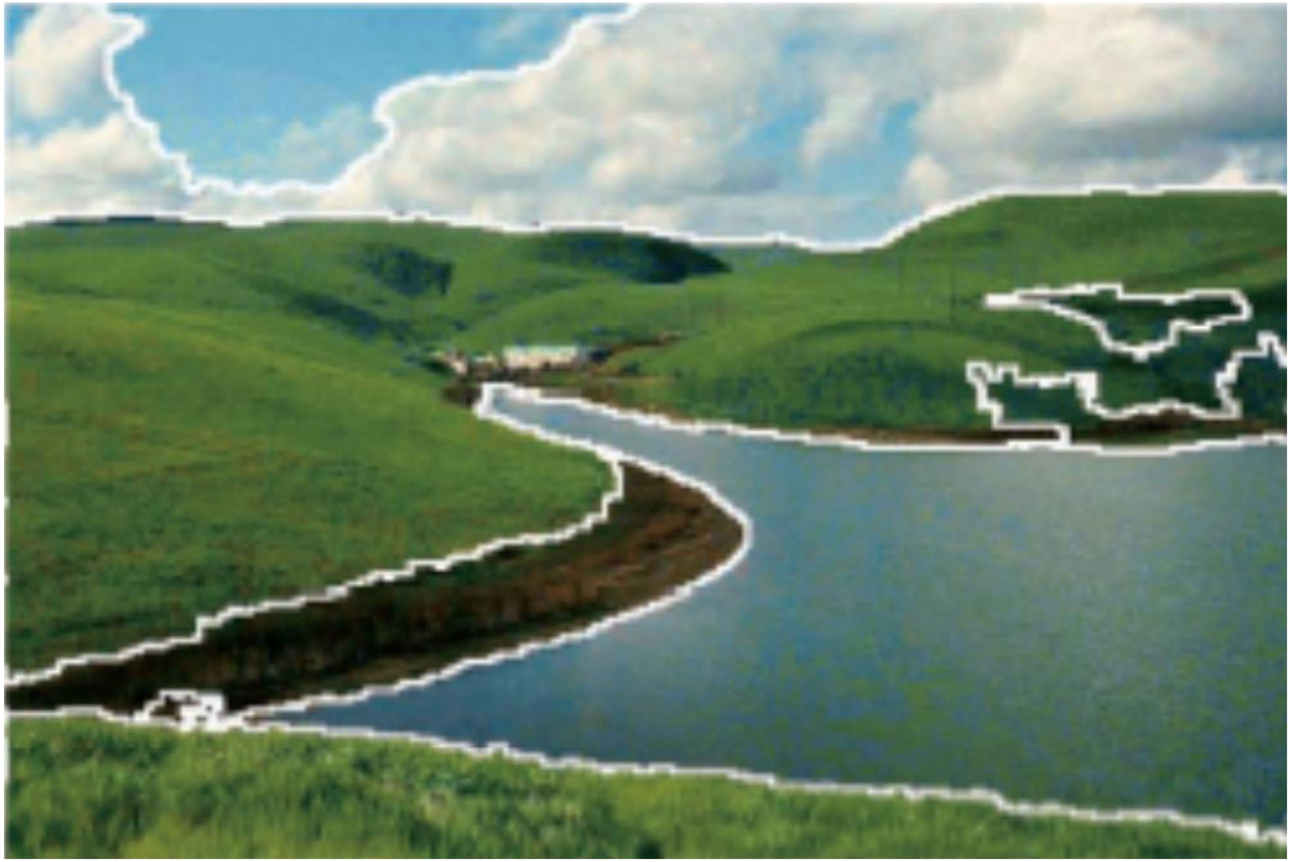


Figure from: James Hayes

Clustering: Edge Detection

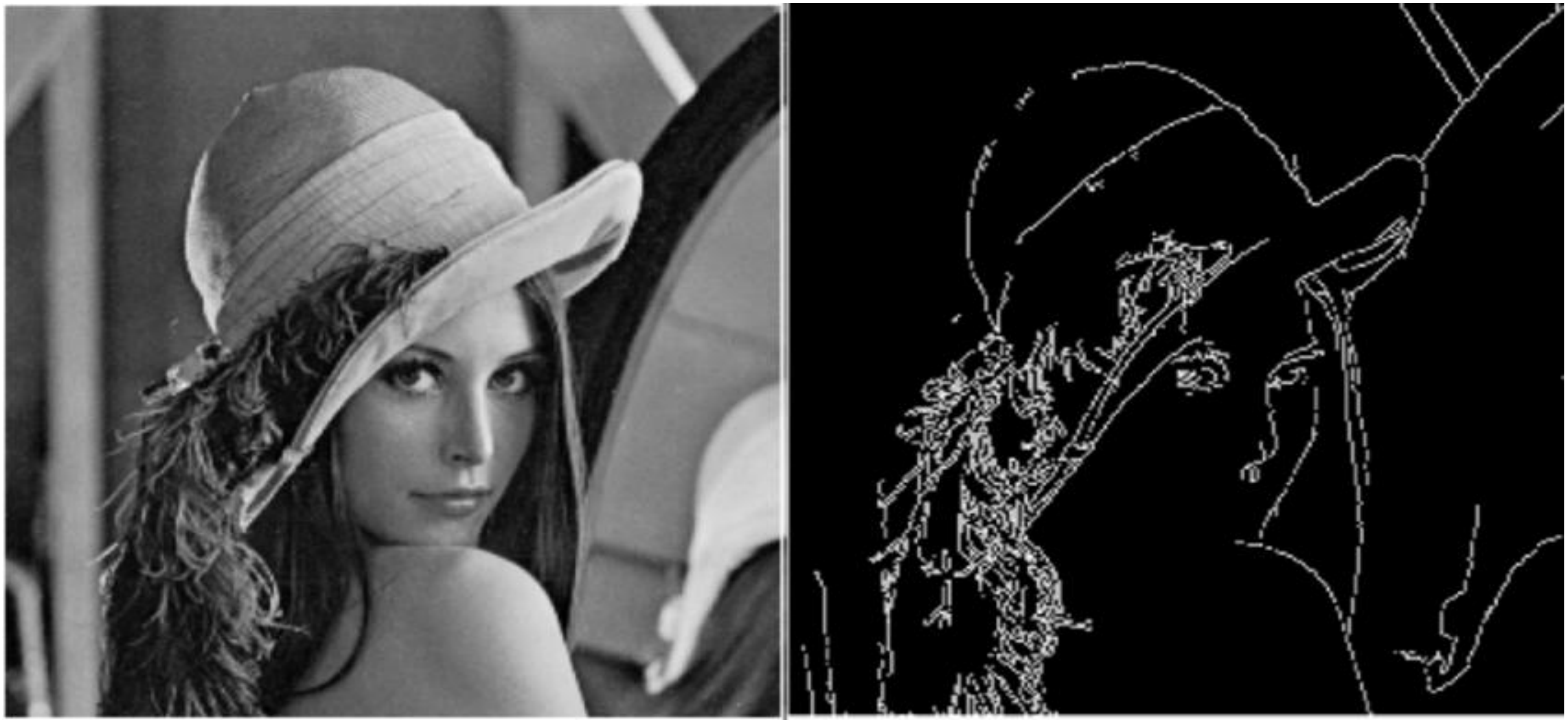
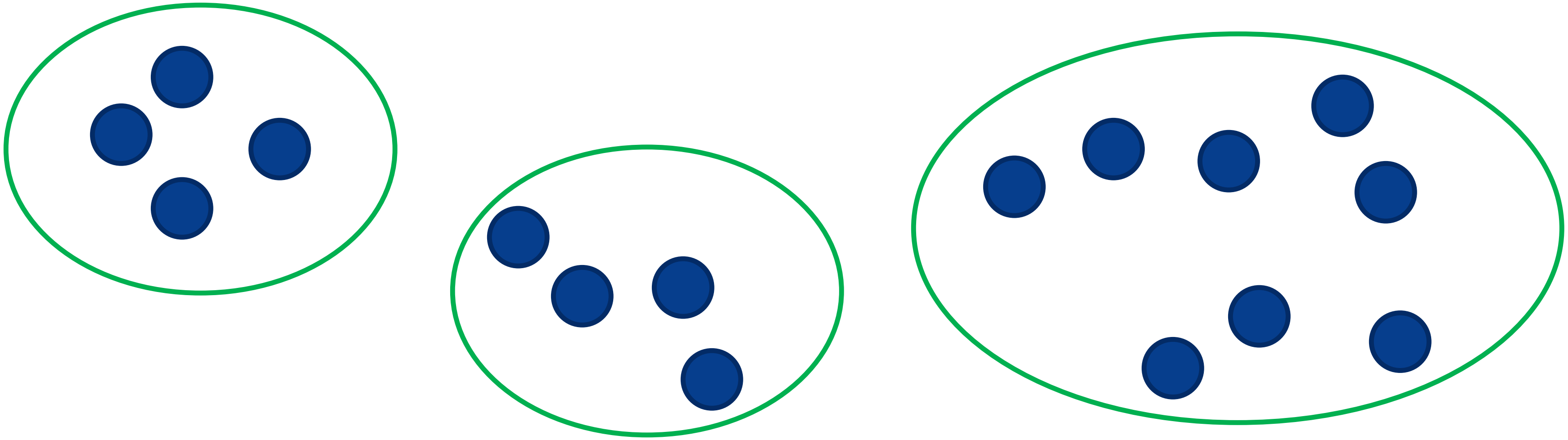
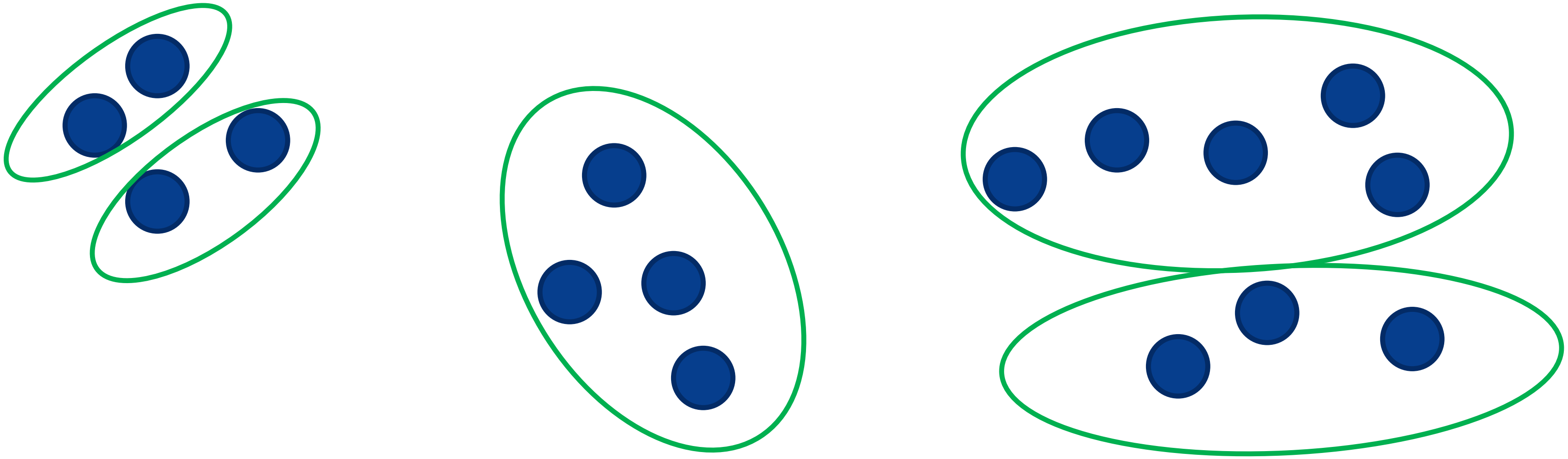


Figure from: James Hayes

Basic Idea of Clustering



Basic Idea of Clustering



Basic Idea of Clustering

Group together similar data points (instances)

- How to measure the similarity?

✓ What could similar mean?

- How many clusters do we need?

K-means

Most well-known and popular clustering algorithm:

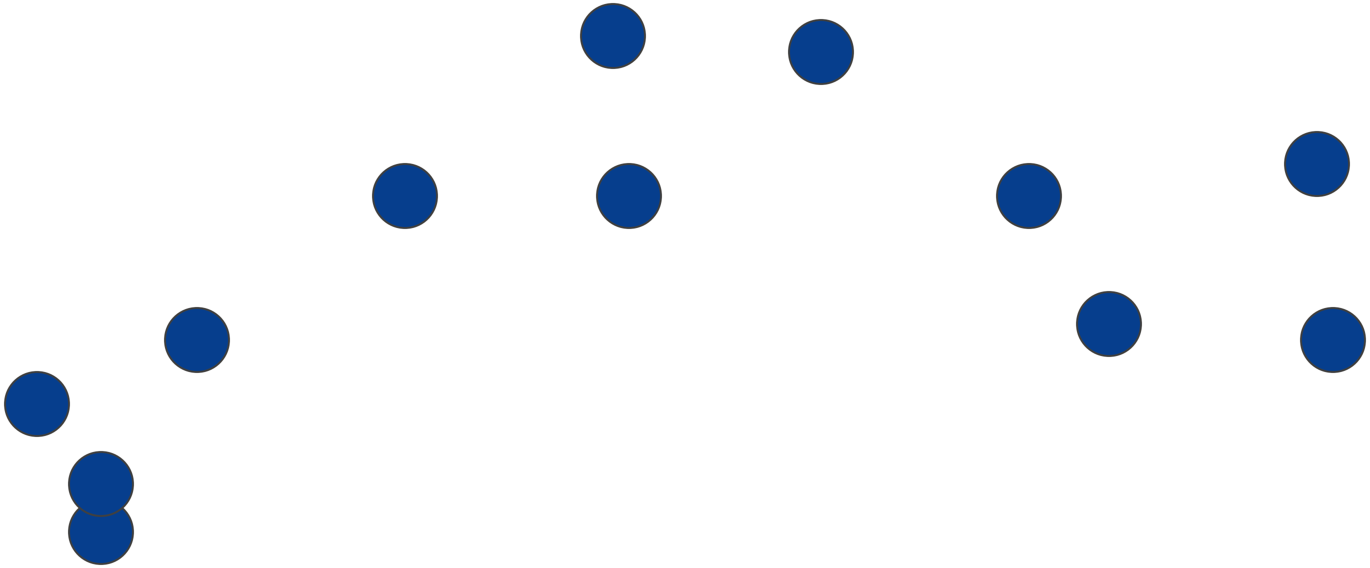
Step 1. Start with some initial cluster centers (k random points)

Step 2. Iterate:

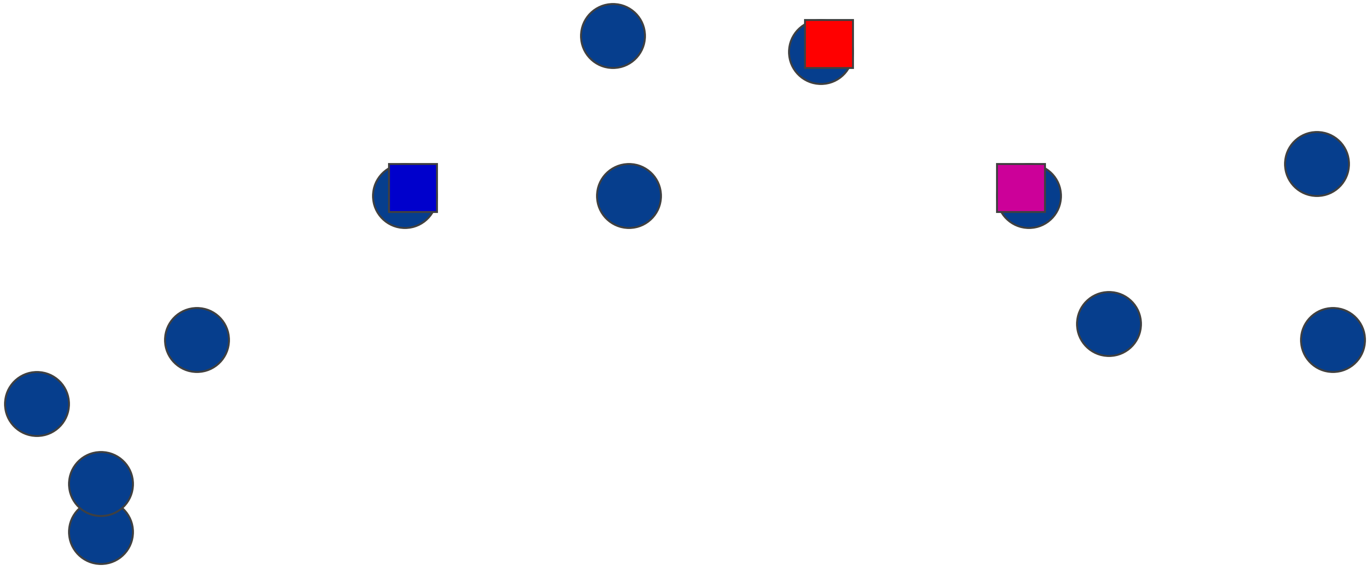
- Assign/cluster each example to closest center
- Recalculate and change centers as the mean of the points in the cluster.

Step 3. Stop when no points’ assignments change

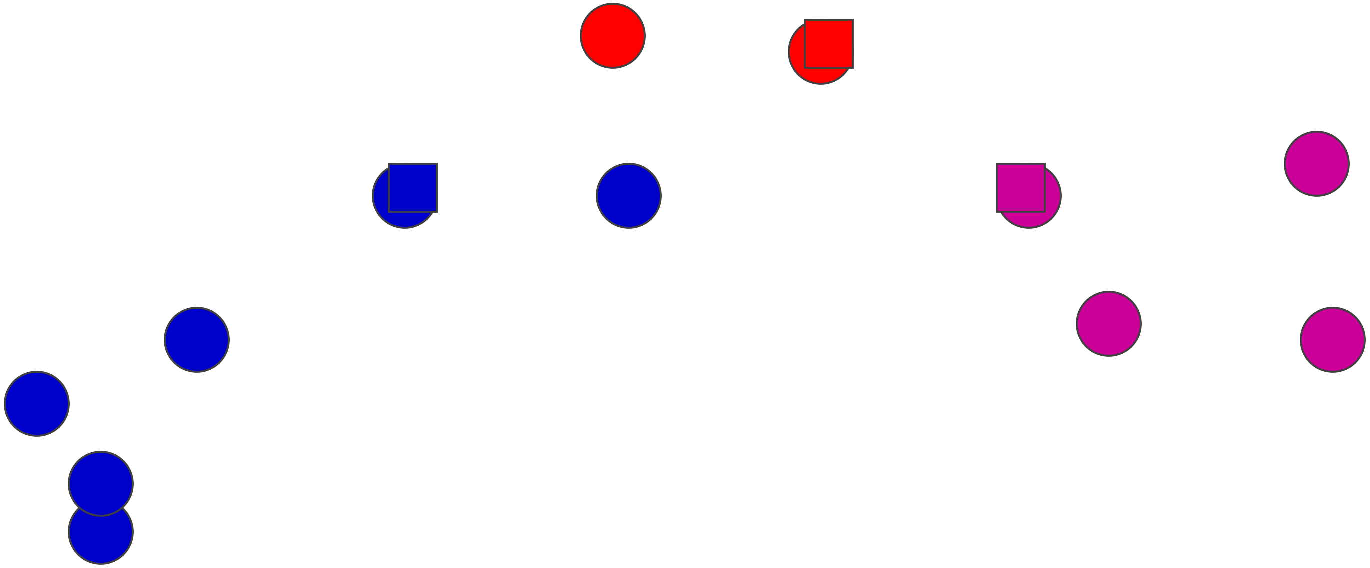
K-means: an example



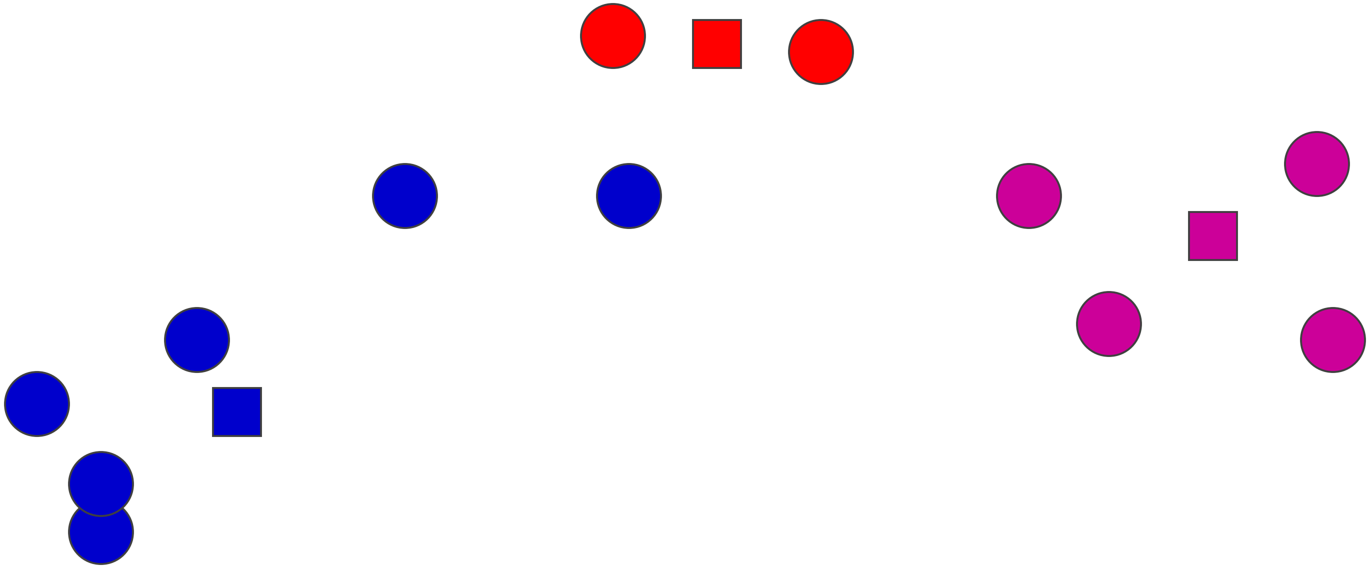
K-means: Initialize centers randomly



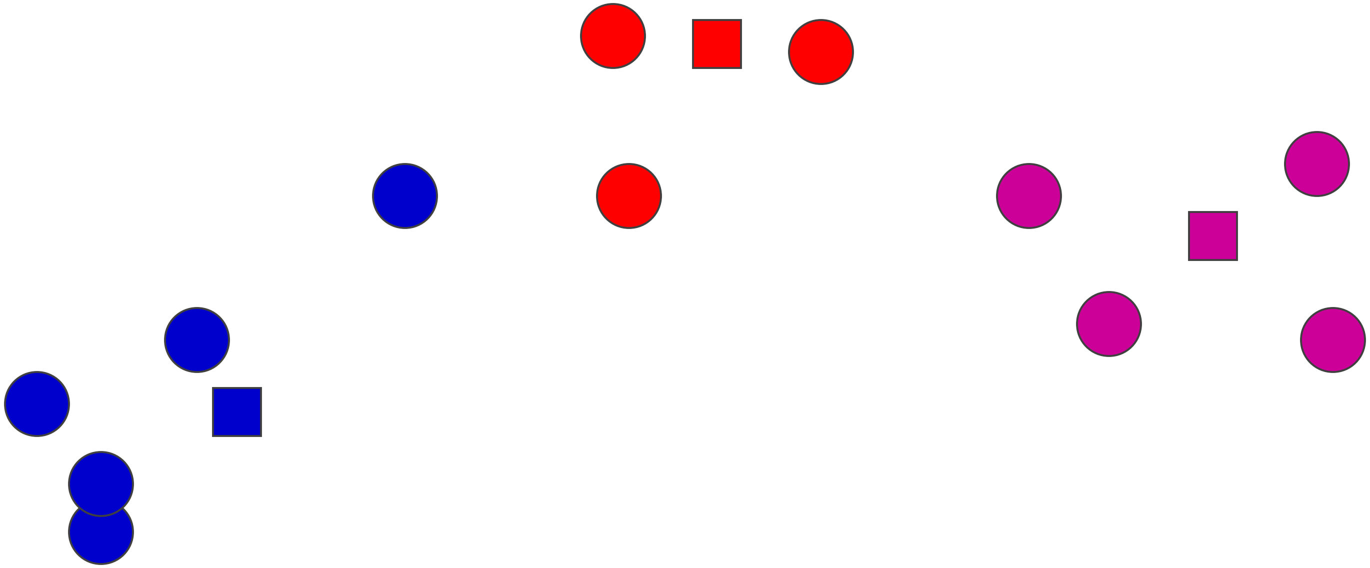
K-means: assign points to nearest center



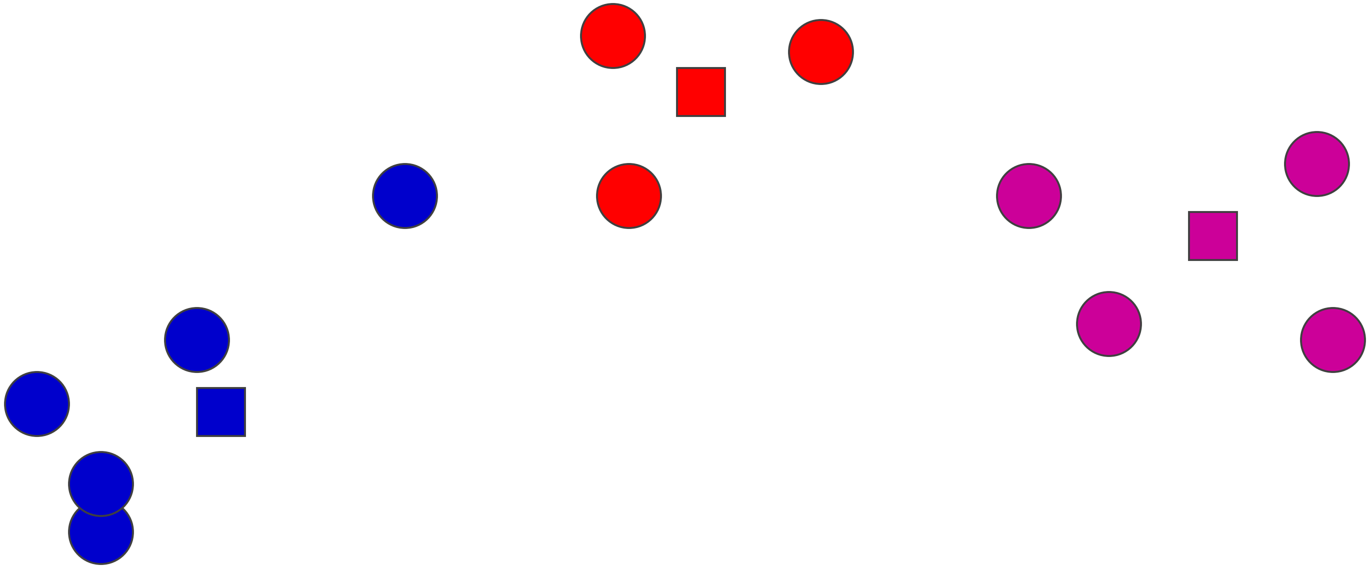
K-means: readjust centers



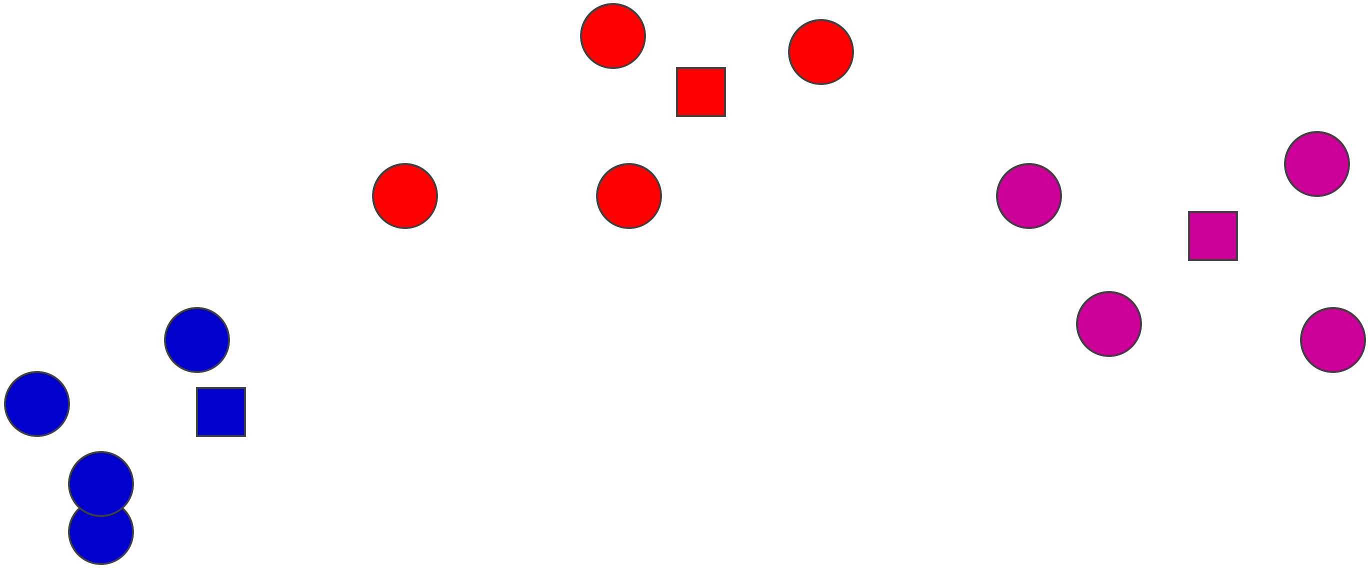
K-means: assign points to nearest center



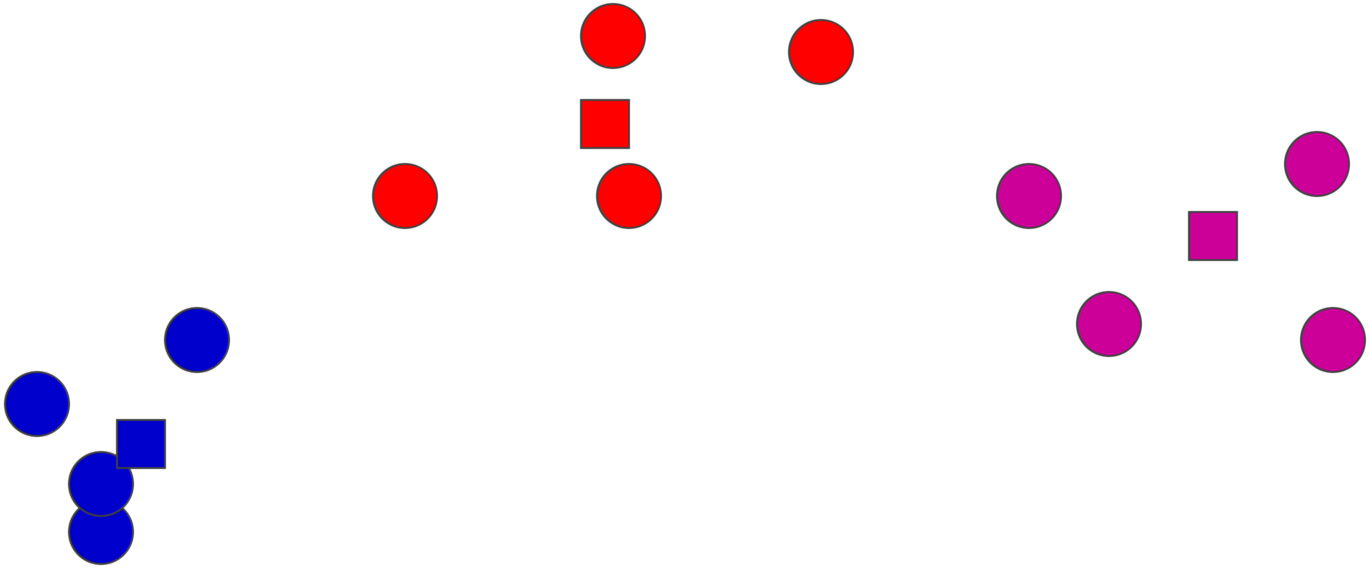
K-means: readjust centers



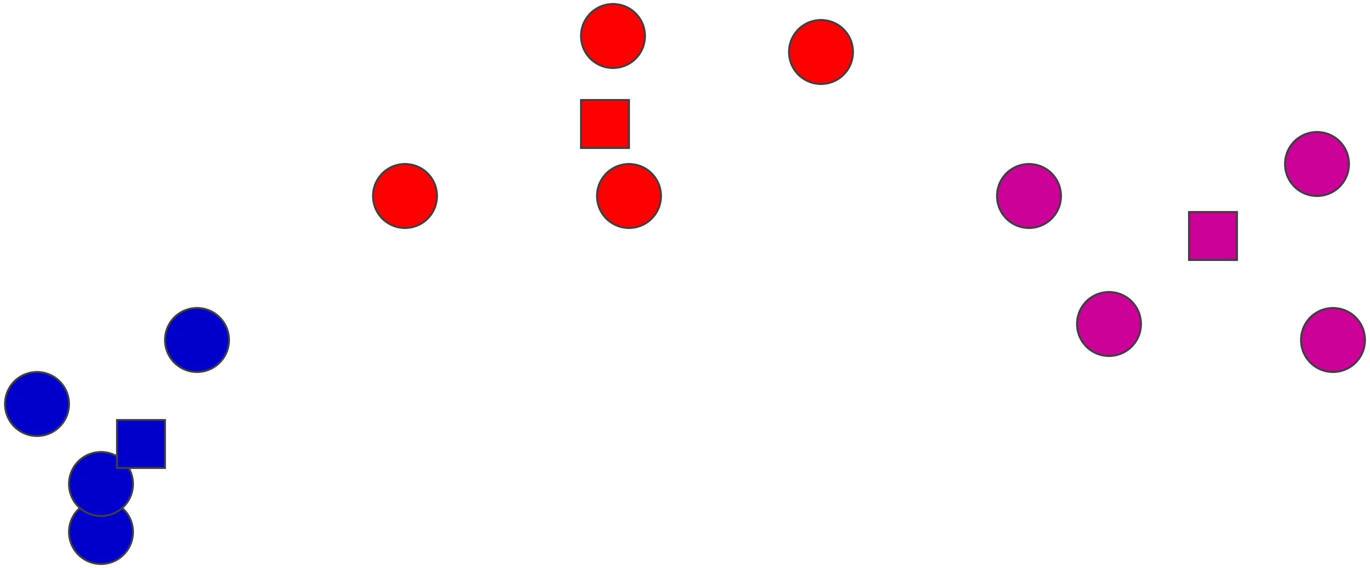
K-means: assign points to nearest center



K-means: readjust centers



K-means: assign points to nearest center

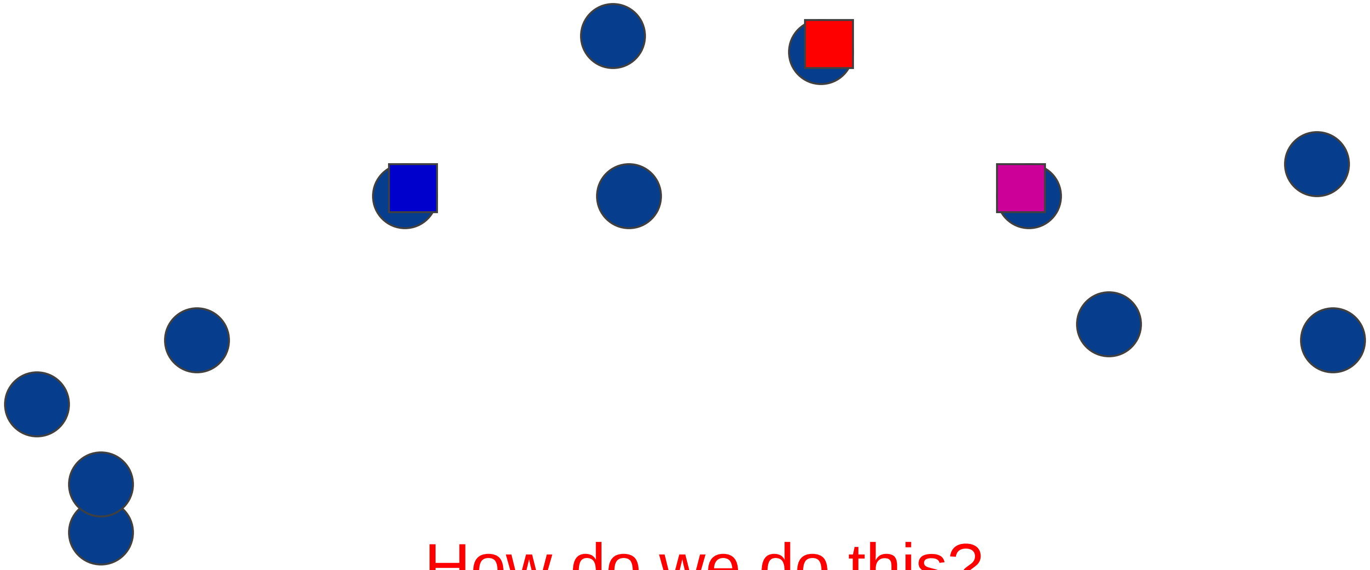


No changes: Done

K-means

Iterate:

- **Assign/cluster each example to closest center**
- Recalculate centers as the mean of the points in a cluster



How do we do this?

K-means

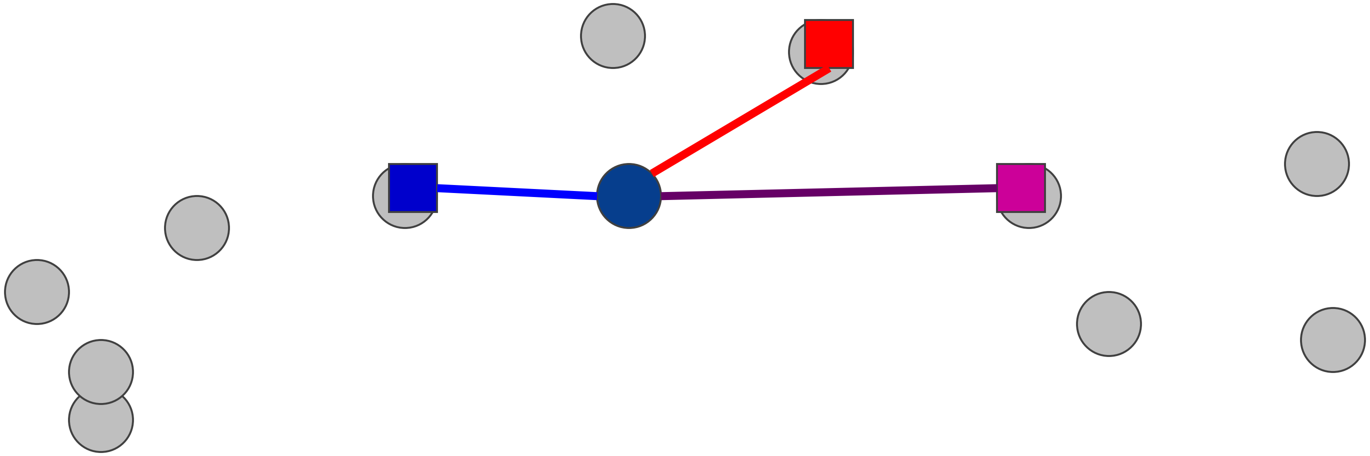
Iterate:

- **Assign/cluster each example to closest center**

  iterate over each point:

  - get distance to each cluster center
  - assign to closest center (hard cluster)

- Recalculate centers as the mean of the points in a cluster



K-means

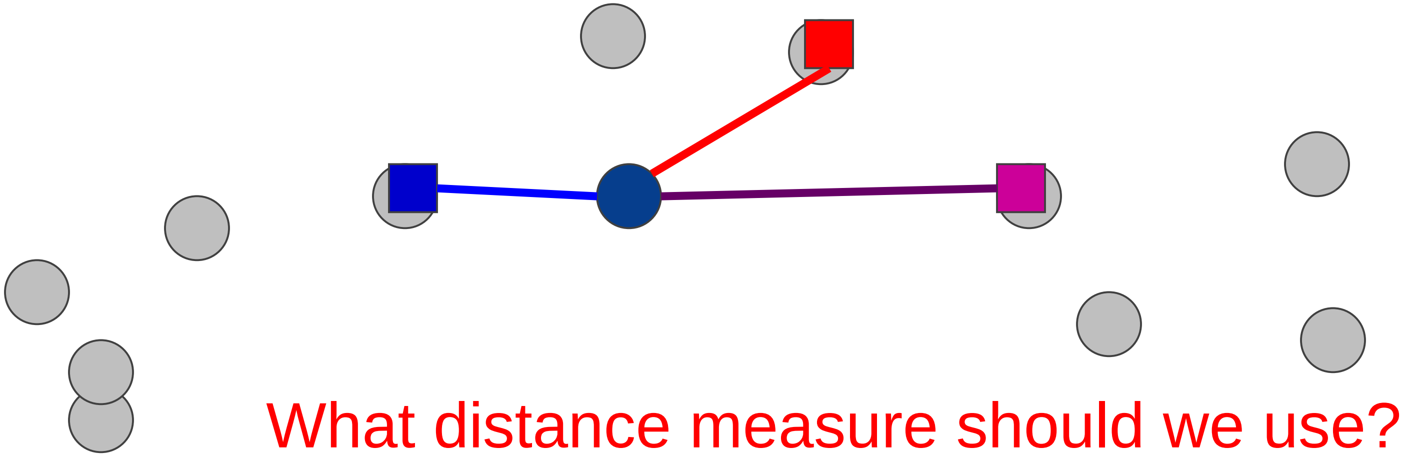
Iterate:

- **Assign/cluster each example to closest center**

  iterate over each point:

  - get **distance** to each cluster center
  - assign to closest center (hard cluster)

- Recalculate centers as the mean of the points in a cluster



What distance measure should we use?

Distance measures

Euclidean:

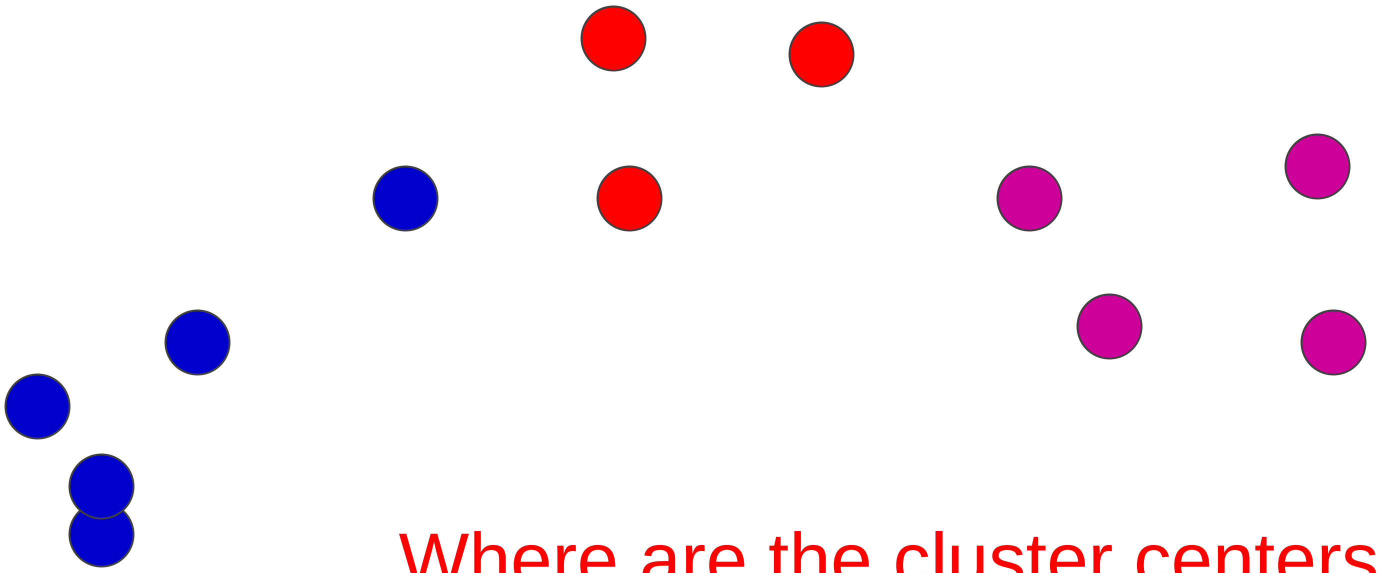
$$d(x,y) = \sqrt{\text{å}_{i=1}^{n}(x_i - y_i)^2}$$

good for spatial data

K-means

Iterate:

- Assign/cluster each example to closest center
- Recalculate centers as the mean of the points in a cluster

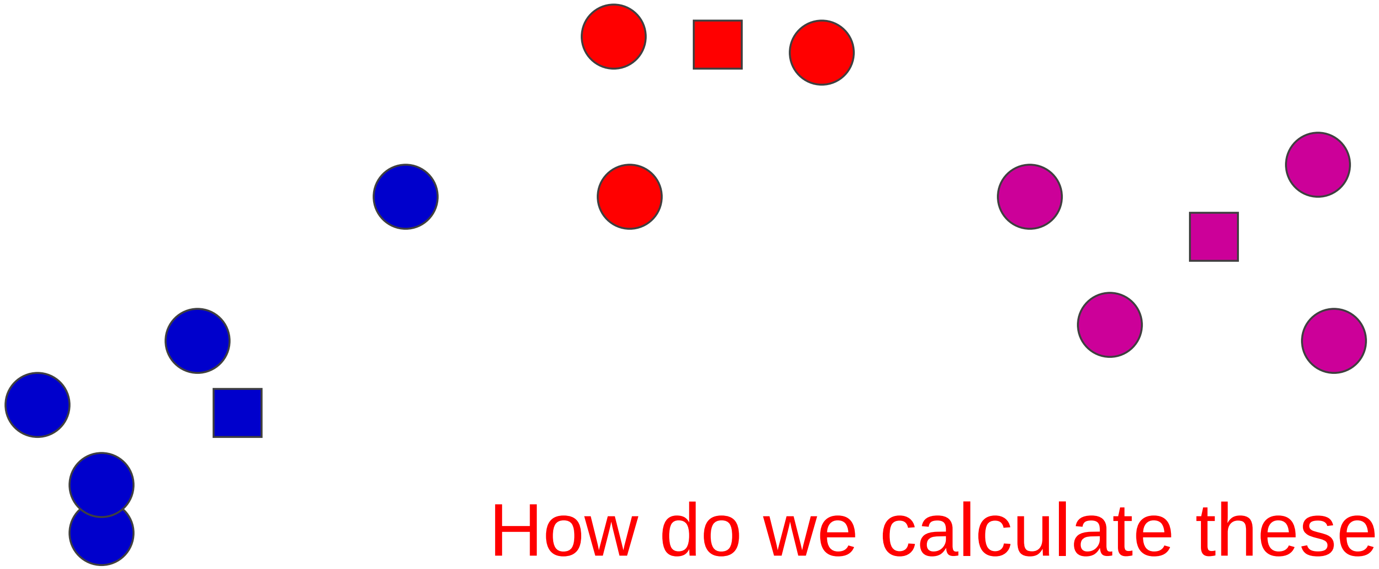


Where are the cluster centers?

K-means

Iterate:

- Assign/cluster each example to closest center
- Recalculate centers as the mean of the points in a cluster



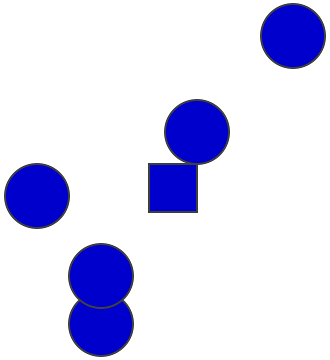
How do we calculate these?

K-means

Iterate:

- Assign/cluster each example to closest center
- Recalculate centers as the mean of the points in a cluster

e.g., for a set of instances that have been assigned to a cluster , we c_j -compute the mean of the cluster as follow



$$\mu(c_j) = \frac{\sum\limits_{\vec{x}_i \in c_j} \vec{x}_i}{|c_j|}$$

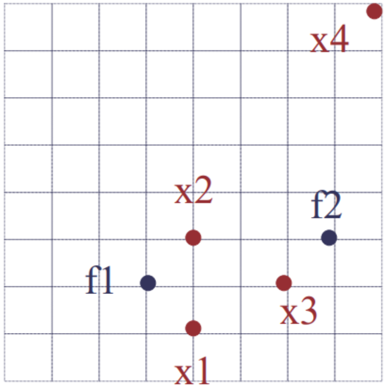
K-means

```
given : a set $X = \{\vec{x}_1 ... \vec{x}_n\}$ of instances
select $k$ initial cluster centers $\vec{f}_1 ... \vec{f}_k$
while stopping criterion not true do
    for all clusters $c_j$ do
        // determine which instances are assigned to this cluster
        $c_j = \left\{ \vec{x}_i \mid \forall f_l \ \text{dist}\left(\vec{x}_i, \vec{f}_j\right) < \text{dist}\left(\vec{x}_i, \vec{f}_l\right) \right\}$
    for all means $\vec{f}_j$ do
        // update the cluster center
        $\vec{f}_j = \mu(c_j)$
```

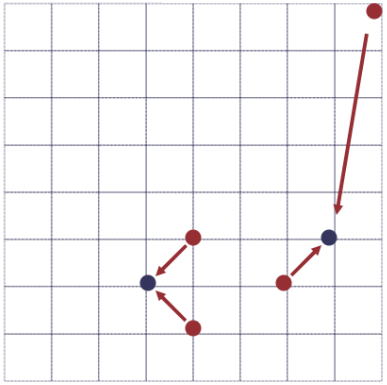
Run an example together ~~

Initialization: 4 points, 2 clusters and distance function

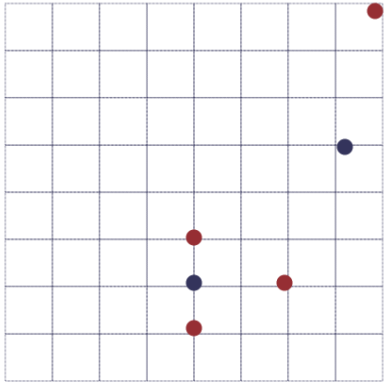
$$\mathrm{dist}(x_i, x_j) = \sum_e \left| x_{i,e} - x_{j,e} \right|$$



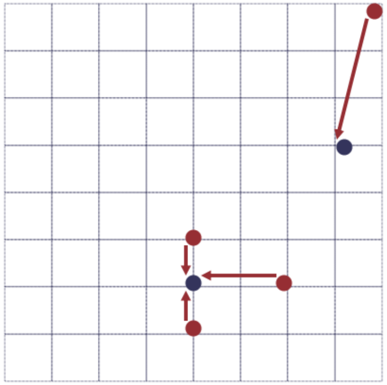
$$dist(x_1, f_1) = 2, \quad dist(x_1, f_2) = 5$$
$$dist(x_2, f_1) = 2, \quad dist(x_2, f_2) = 3$$
$$dist(x_3, f_1) = 3, \quad dist(x_3, f_2) = 2$$
$$dist(x_4, f_1) = 11, \quad dist(x_4, f_2) = 6$$



$$f_1 = \left\langle \frac{4+4}{2}, \frac{1+3}{2} \right\rangle = \langle 4,2 \rangle$$
$$f_2 = \left\langle \frac{6+8}{2}, \frac{2+8}{2} \right\rangle = \langle 7,5 \rangle$$



$$dist(x_1, f_1) = 1, \quad dist(x_1, f_2) = 7$$
$$dist(x_2, f_1) = 1, \quad dist(x_2, f_2) = 5$$
$$dist(x_3, f_1) = 2, \quad dist(x_3, f_2) = 4$$
$$dist(x_4, f_1) = 10, \quad dist(x_4, f_2) = 4$$



$$f_1 = \left\langle \frac{4+4+6}{3}, \frac{1+3+2}{3} \right\rangle = \langle 4.67,2 \rangle$$
$$f_2 = \left\langle \frac{8}{1}, \frac{8}{1} \right\rangle = \langle 8,8 \rangle$$

Properties of K-means

Guaranteed to converge in a finite number of iterations

Running time per iteration

1. Assign data points to closest cluster center O(KN) time
2. Change the cluster center to the average of its assigned points O(N)

K-means variations/parameters

Start with some initial cluster centers

Iterate:

- Assign/cluster each example to closest center
- Recalculate centers as the mean of the points in a cluster

What are some other variations/parameters we haven’t specified?

K-means variations/parameters

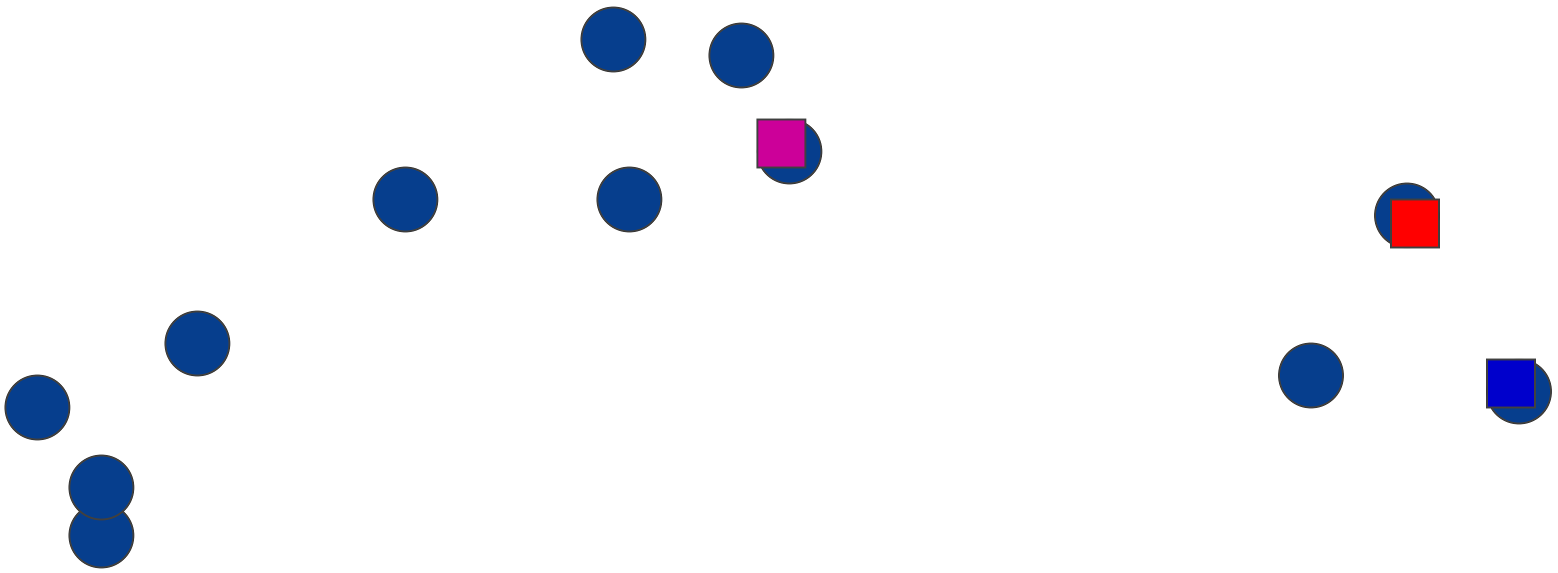
Initial (seed) cluster centers

Convergence

- A fixed number of iterations
- partitions unchanged
- Cluster centers don’t change

K!

K-means: Initialize centers randomly



What would happen here?

Seed selection ideas?

Seed choice

Results can vary drastically based on random seed selection

Some seeds can result in poor convergence rate, or convergence to sub-optimal clustering

Common heuristics

- Random centers in the space
- Randomly pick examples
- Points least similar to any existing center (furthest centers heuristic)
- **Try out multiple starting points**
- Initialize with the results of another clustering method

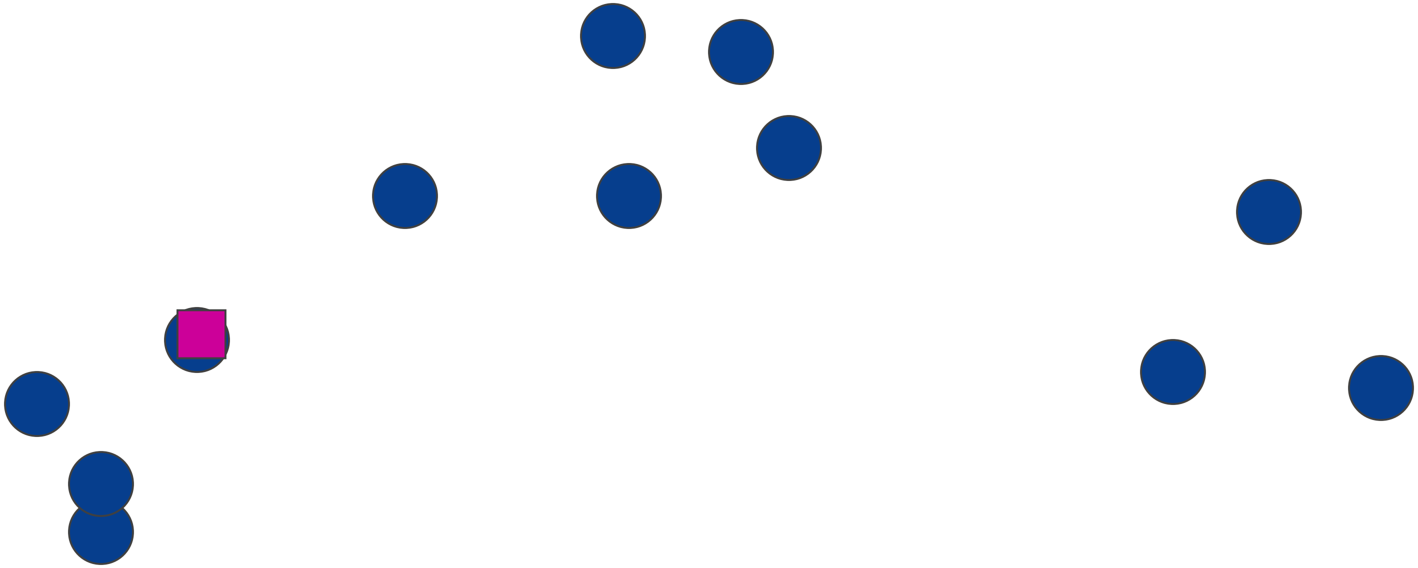
Furthest centers heuristic

```
$\mu_1$ = pick random point

for i = 2 to K:
    $\mu_i$ = point that is furthest from any previous centers
```

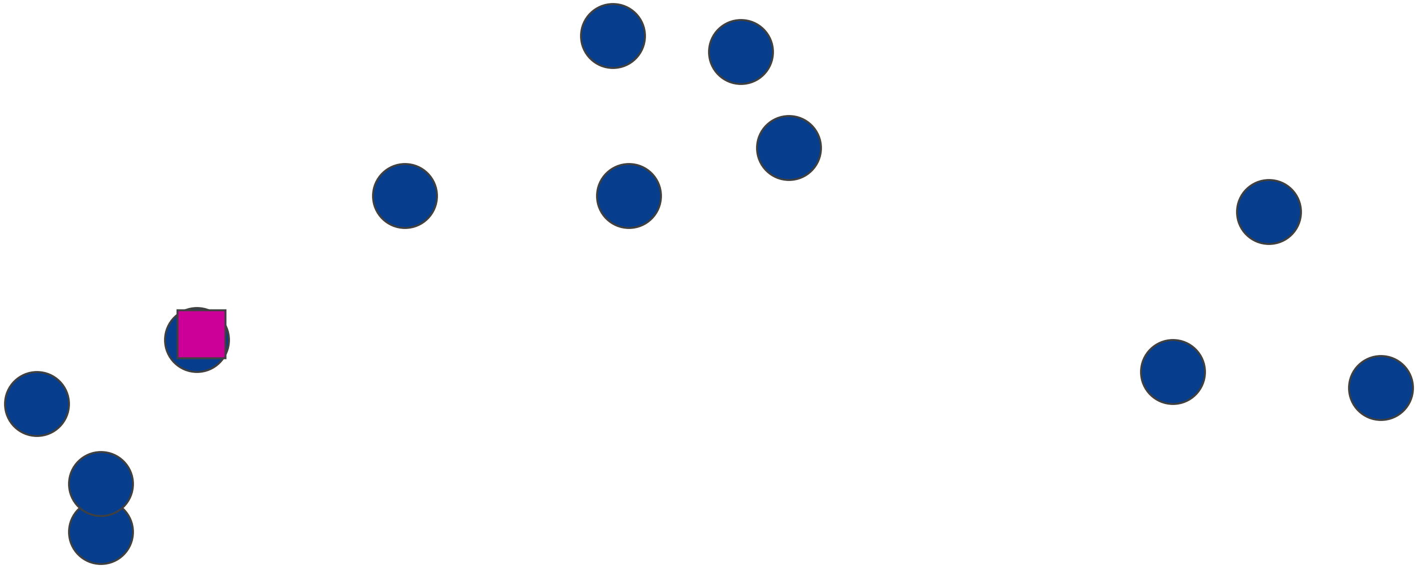
$$\mu_i = \underbrace{\arg\max_{x}}_{\text{point with the largest distance to any previous center}} \underbrace{\min_{\mu_j : 1 < j < i} d(x, \mu_j)}_{\text{smallest distance from x to any previous center}}$$

K-means: Initialize furthest from centers



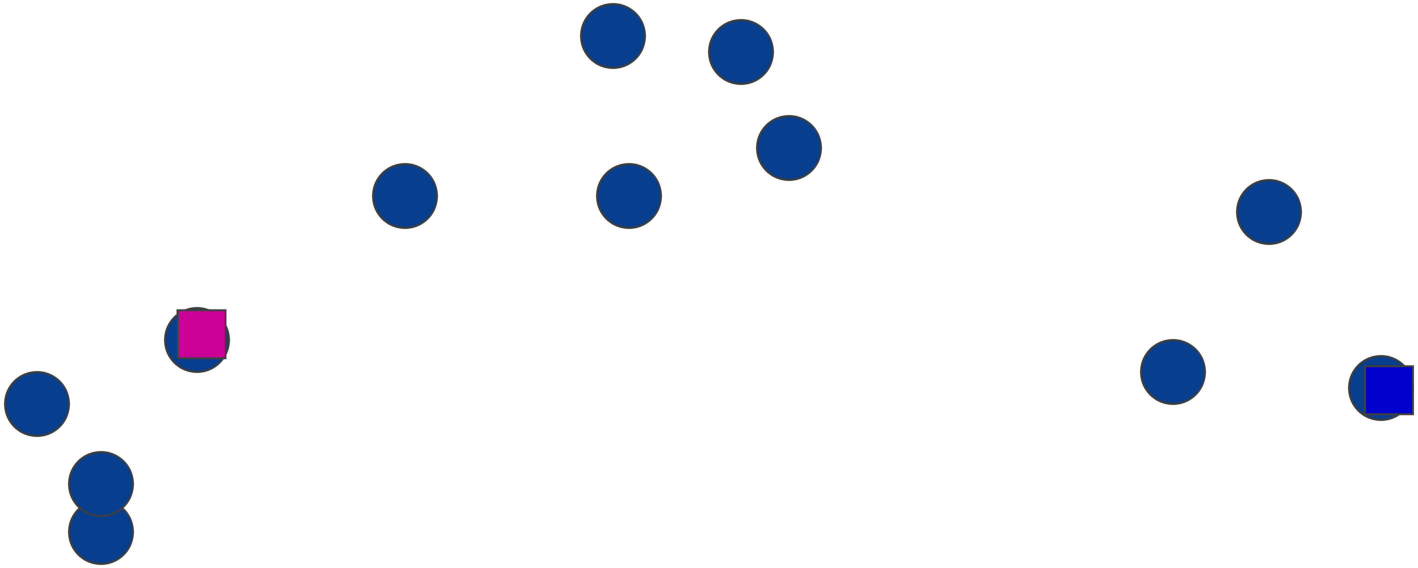
Pick a random point for the first center

K-means: Initialize furthest from centers



What point will be chosen next?

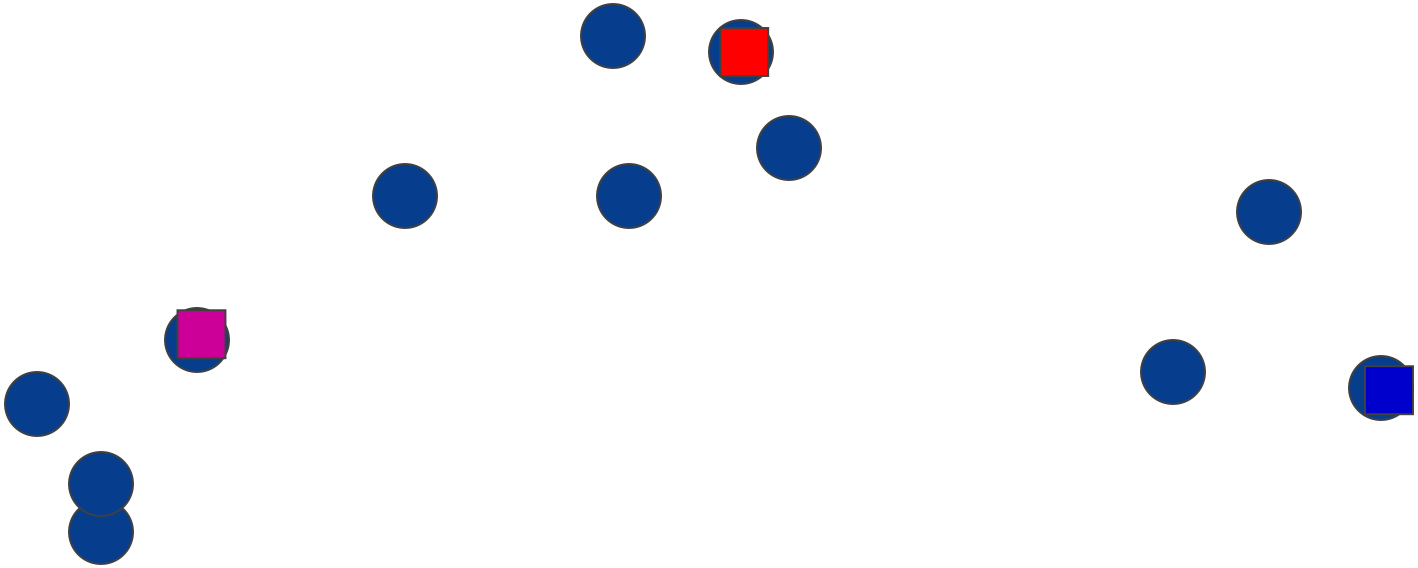
K-means: Initialize furthest from centers



Furthest point from center

What point will be chosen next?

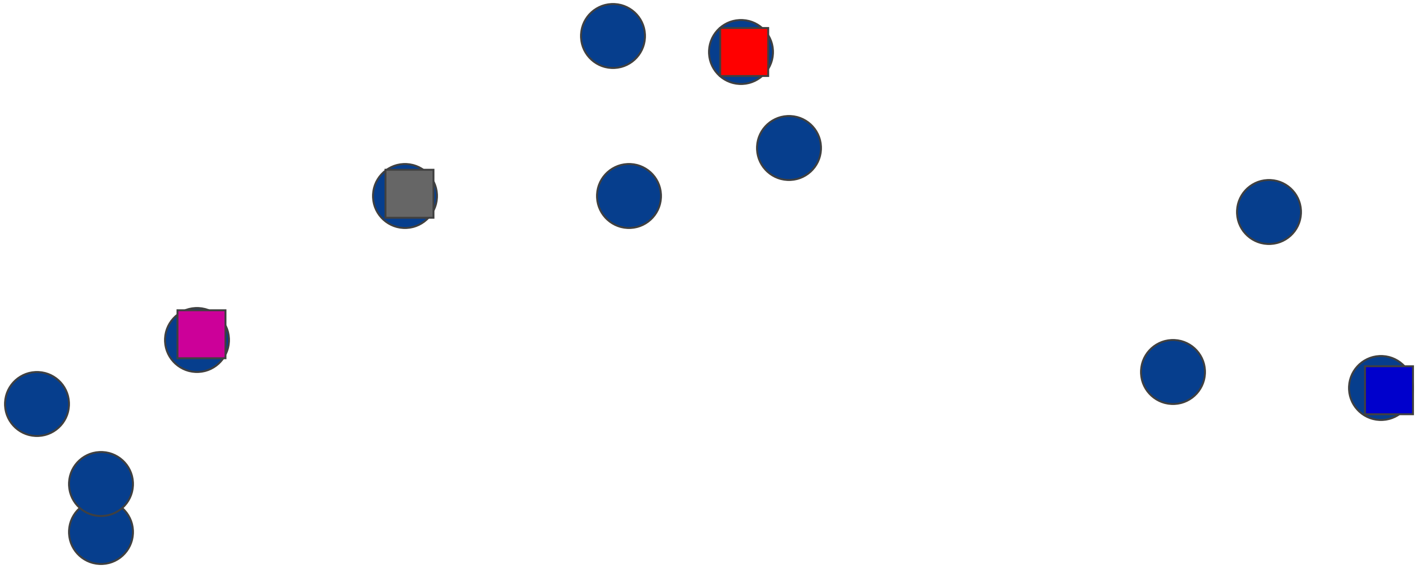
K-means: Initialize furthest from centers



Furthest point from center

What point will be chosen next?

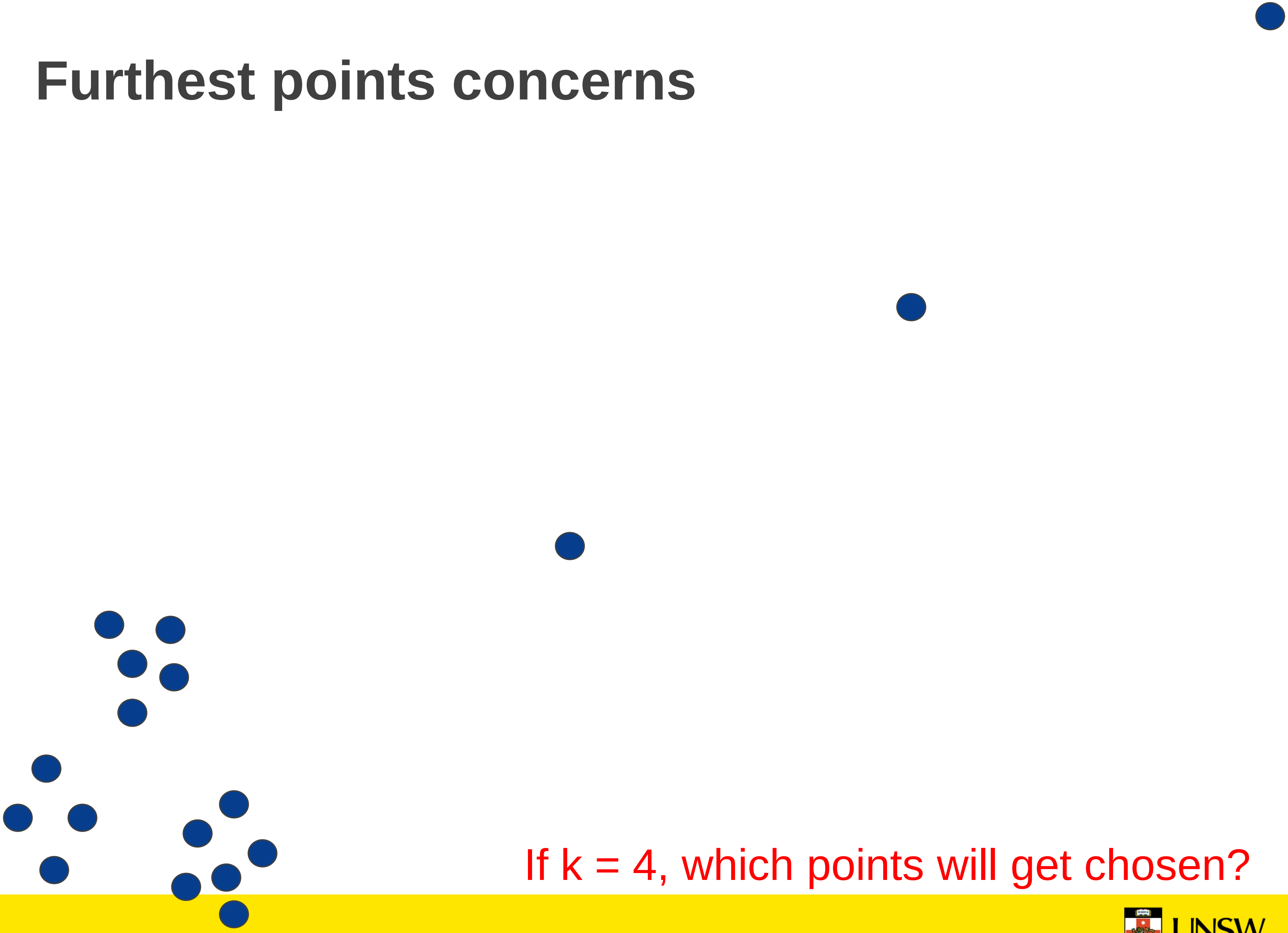
K-means: Initialize furthest from centers



Furthest point from center

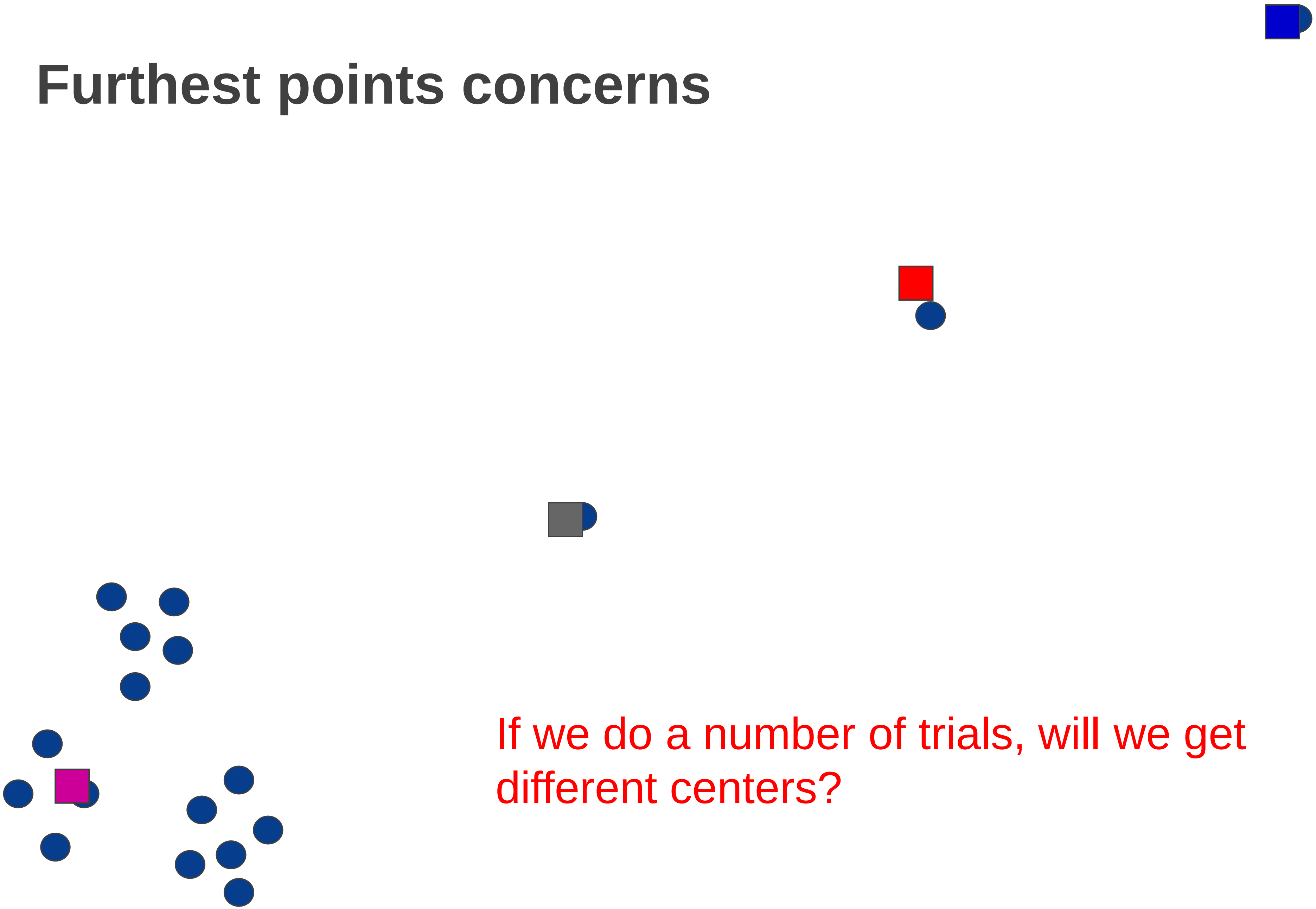
Any issues/concerns with this approach?

Furthest points concerns



If k = 4, which points will get chosen?

Furthest points concerns



If we do a number of trials, will we get different centers?

K-means++

```
$\mu_1$ = pick random point

for k = 2 to K:
    for i = 1 to N:
        $s_i = \min d(x_i, \mu_{1\ldots k-1})$ // smallest distance to any center

    $\mu_k$ = randomly pick point proportionate to s
```

How does this help?

K-means++

```
$\mu_1$ = pick random point
for k = 2 to $\mathbf{K}$:
    for i = 1 to $\mathbf{N}$:
        $s_i$ = min d($x_i$, $\mu_{1\ldots k-1}$) // smallest distance to any center
    $\mu_k$ = randomly pick point proportionate to $s$
```

- Makes it possible to select other points
  - if #points >> #outliers, we will pick good points
- Makes it non-deterministic, which will help with random runs
- Nice theoretical guarantees!

What Is A Good Clustering?

Internal criterion: A good clustering will produce high quality clusters in which:

- the intra-class (that is, intra-cluster) similarity is high
- the inter-class similarity is low
- The measured quality of a clustering depends on both the document representation and the similarity measure used

Clustering Evaluation

- Intra-cluster cohesion (compactness):

– Cohesion measures how near the data points in a cluster are to the cluster centroid.

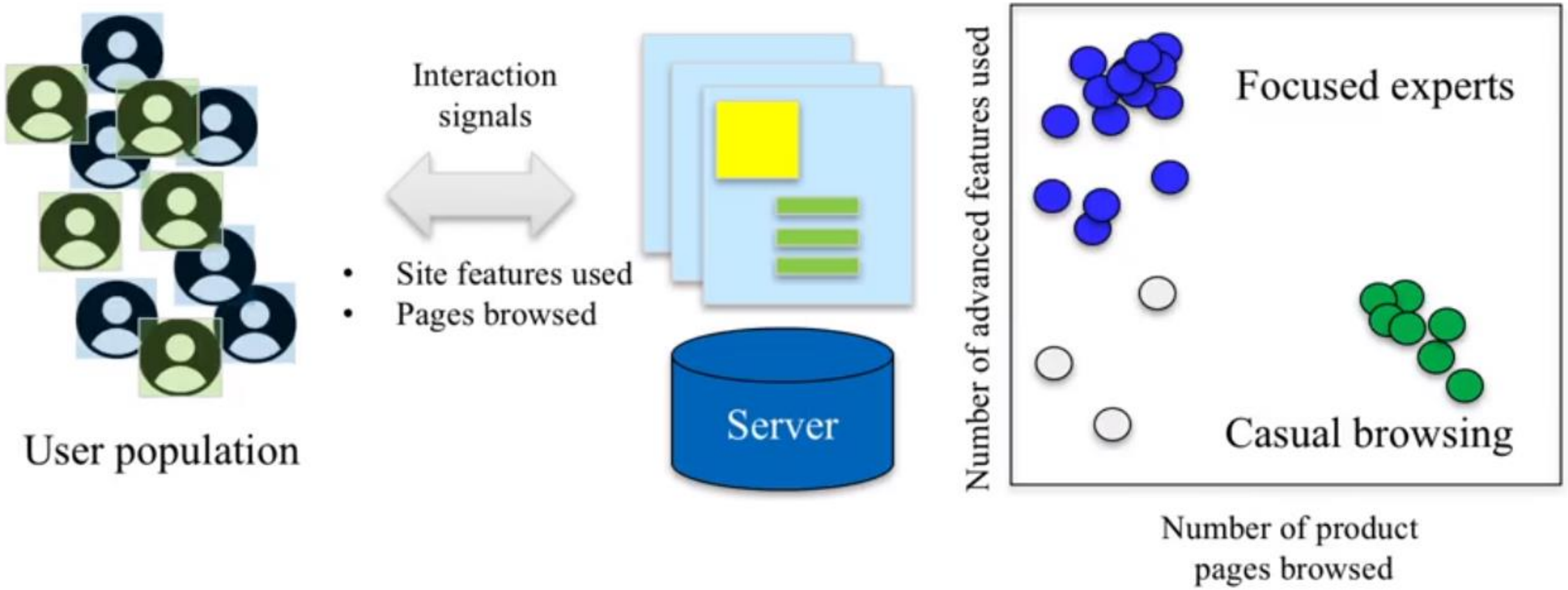
– Sum of squared error (SSE) is a commonly used measure.

- Inter-cluster separation (isolation):

– Separation means that different cluster centroids should be far away from one another.

- In most applications, expert judgments are still the key

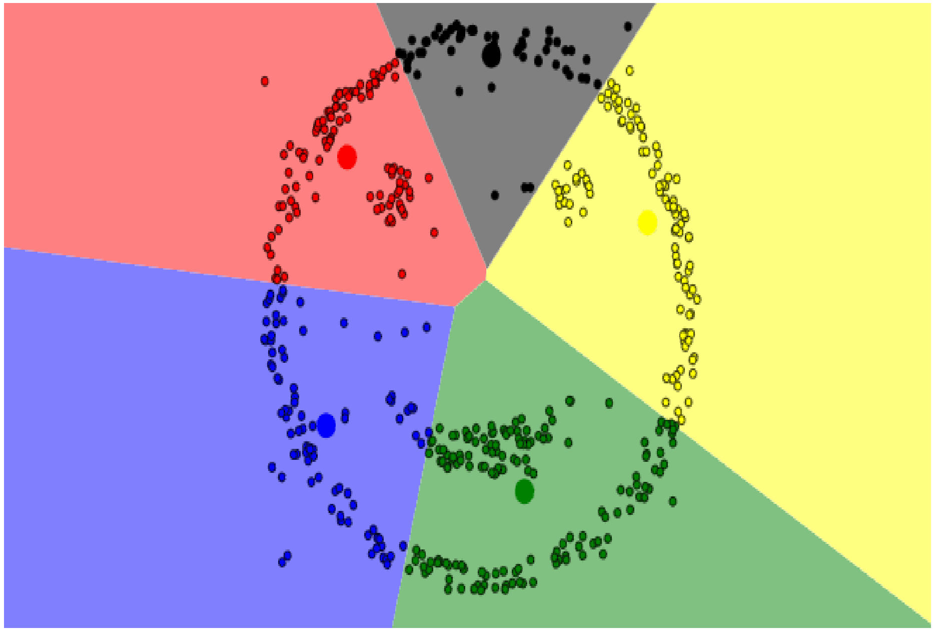
Web Clustering Examples



https://www.coursera.org/learn/python-machine-learning

Limitations of k-means

- Sometime the number of clusters is difficult to determine
- Does not do well with irregular or complex clusters.
- Has a problem with data containing outliers



http://arogozhnikov.github.io/2017/07/10/opera-clustering.html

Q&A