# Clustering

Jaime Arguello
INLS 613: Text Data Mining
jarguell@email.unc.edu

(some material taken or adapted from slides by Hinrich Schutze)

# Clustering
## objective

- Grouping documents or instances into subsets or clusters
- Documents in the same cluster should be similar
- Documents in different clusters should be dissimilar
- A common form of unsupervised learning
- Unsupervised = no human-produced labels
- The goal is to discover structure from the data

# Clustering vs. Classification

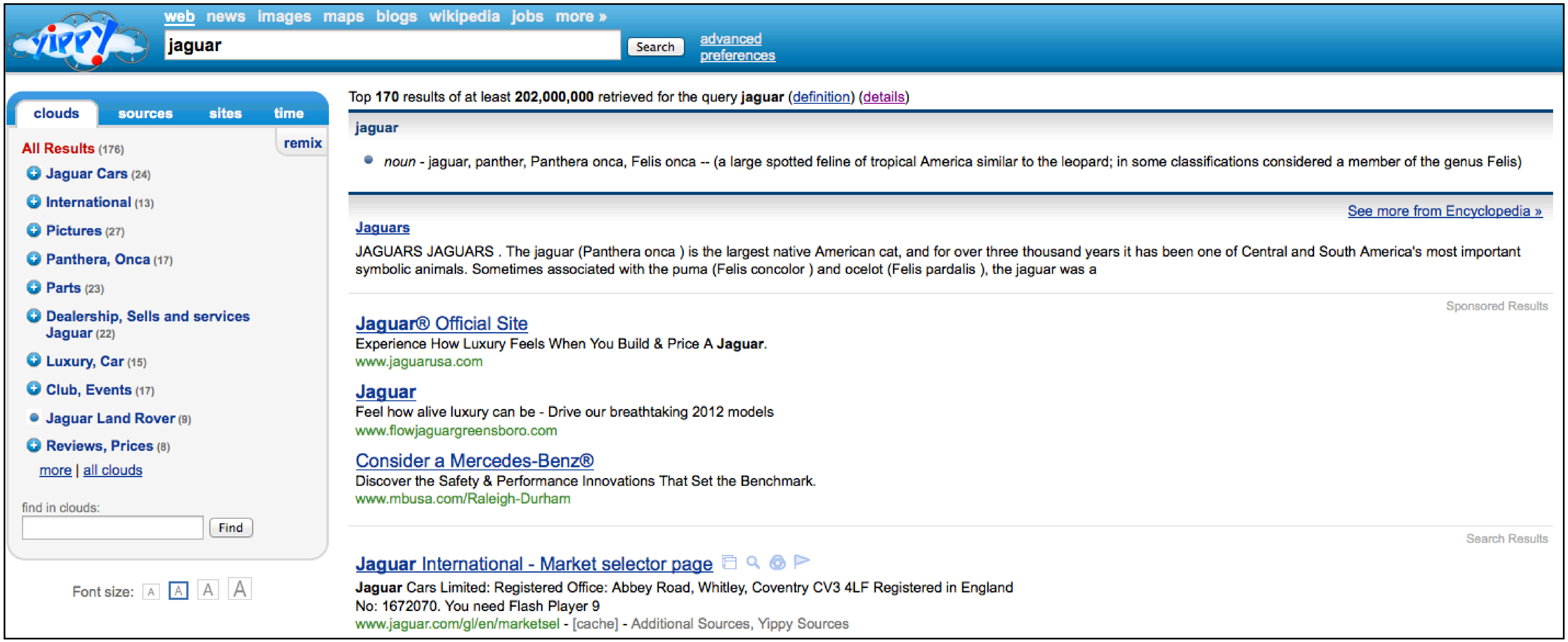
- Classification:
  - the input to the system is a set of labeled data
  - the algorithm learns a model for predicting the label on new examples
- Clustering:
  - the input to the system is a set of unlabeled data
  - the algorithm infers the labels from the data and assigns a label to each input instance

# Clustering
## applications

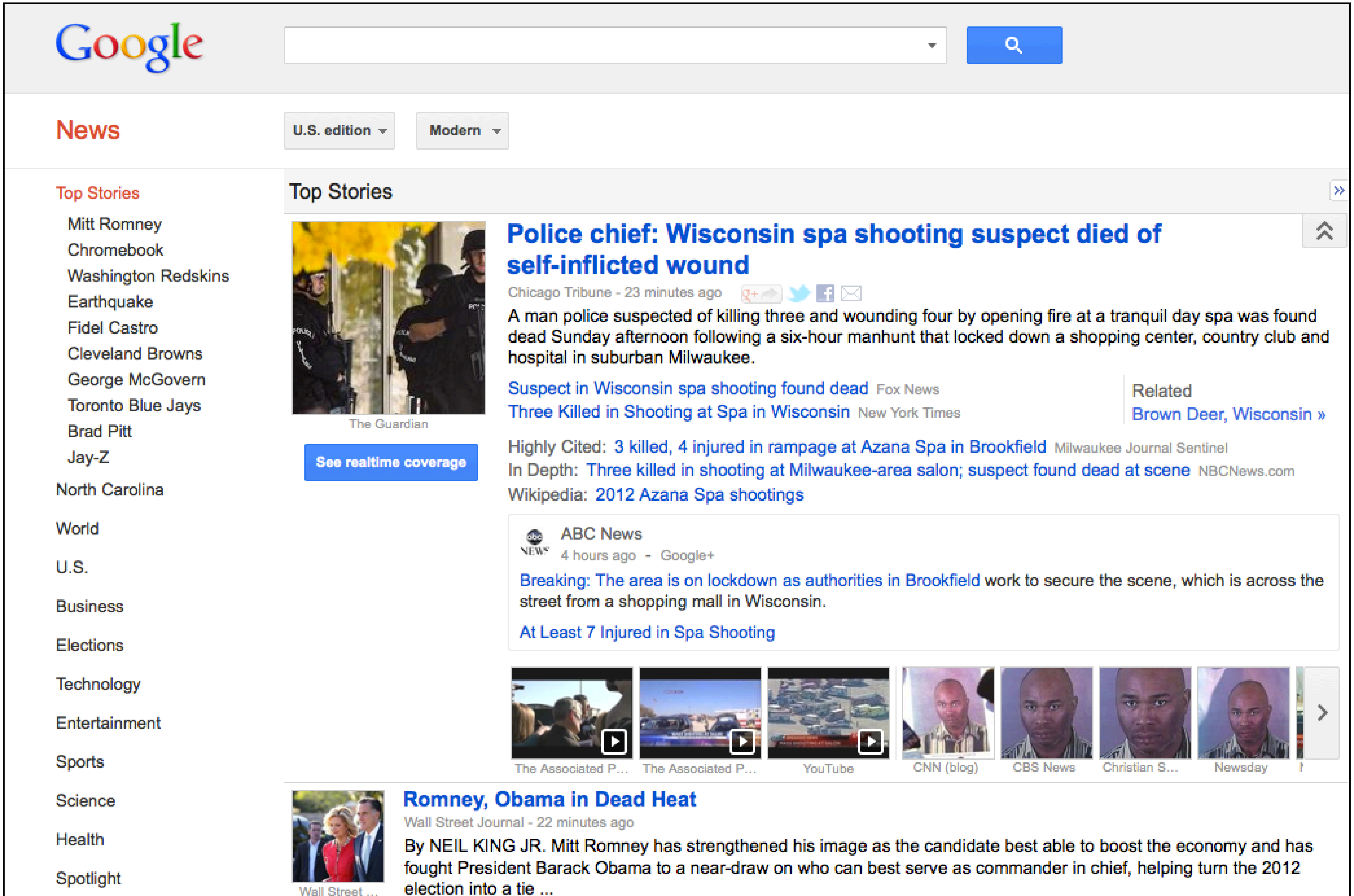
- Search engine results clustering: grouping search engine results by topic
    - the user can identify the relevant clusters and ignore the non-relevant ones
- Collection clustering: grouping documents by topic to support navigation and exploration
- Data analytics: grouping instances to identify popular trends (big clusters) and outliers (small clusters)

# Clustering Applications

## search engine results clustering



# Clustering Applications
## collection clustering



# Clustering
## objective

- Grouping documents or instances into subsets or clusters
- Documents within a the same cluster should be similar
- Documents from different clusters should be dissimilar

# Clustering
## basics

- What does it mean for documents to be “similar” or “dissimilar”?

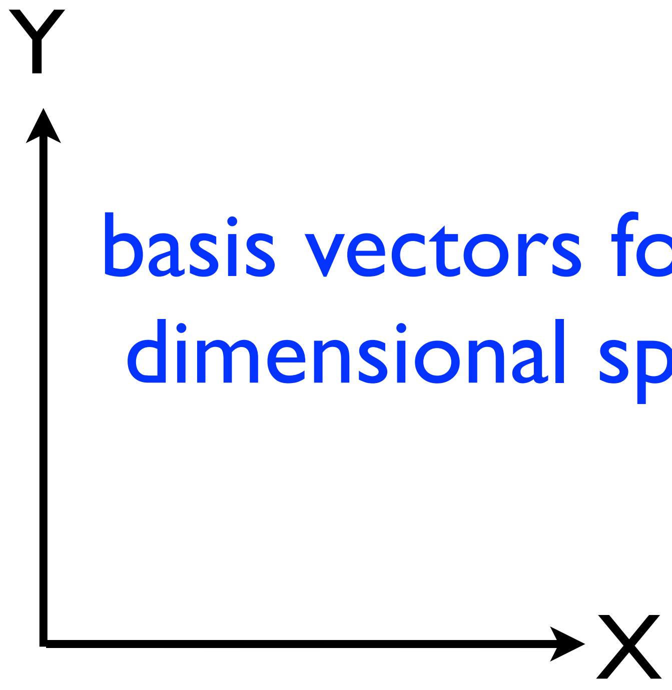
# Clustering
## basics

- What does it mean for documents to be similar or dissimilar?
- We need a computational way of modeling similarity
- One solution: model similarity using distance in a vector space representation of the collection or dataset
  - small distance = high similarity
  - long distance = low similarity

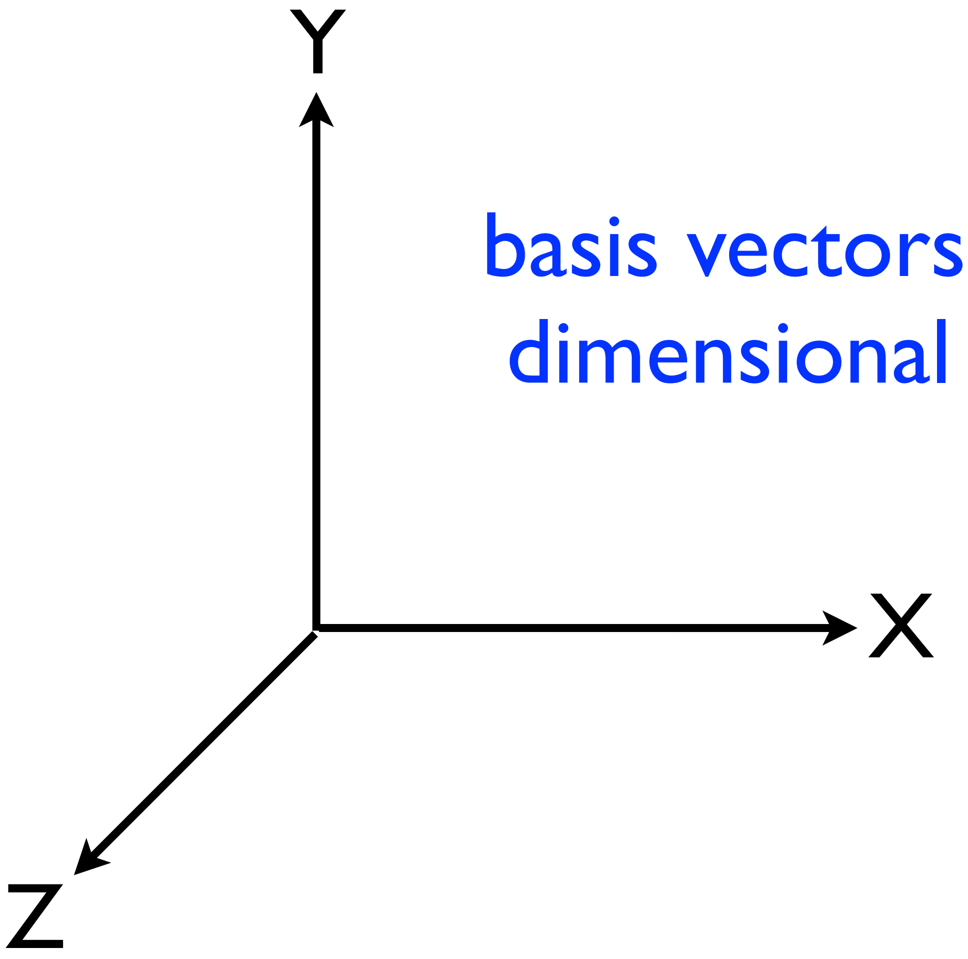
# Vector Space Representation

review

- A vector space is defined by a set of linearly independent basis vectors
- The basis vectors correspond to the dimensions or directions of the vector space



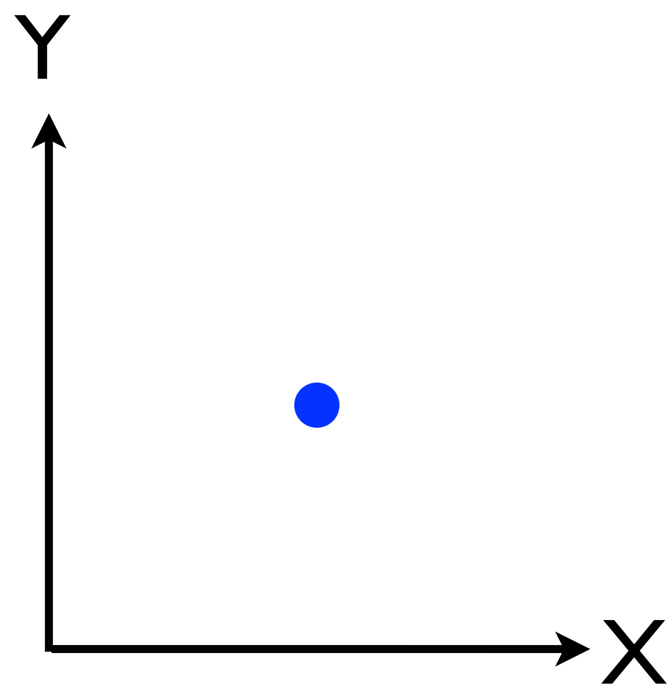

basis vectors for 2-dimensional space

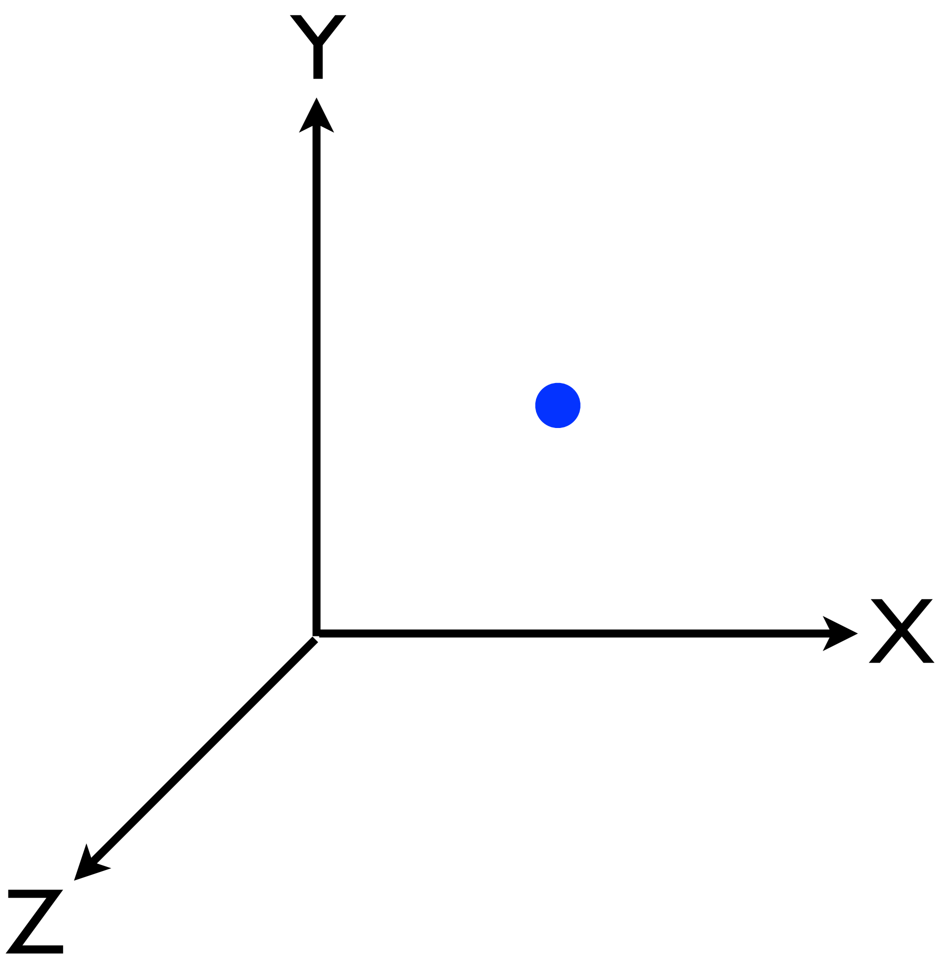



basis vectors for 3-dimensional space

# Vector Space Representation
review

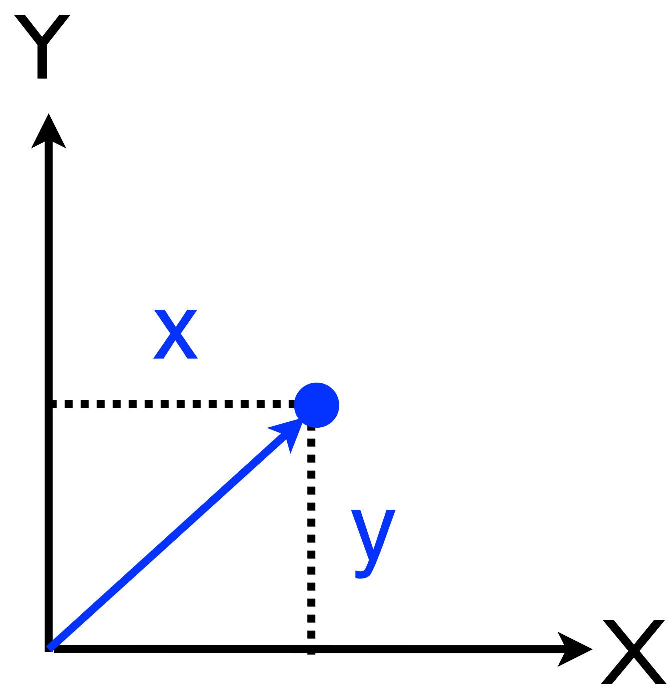
- A vector is a point in a vector space

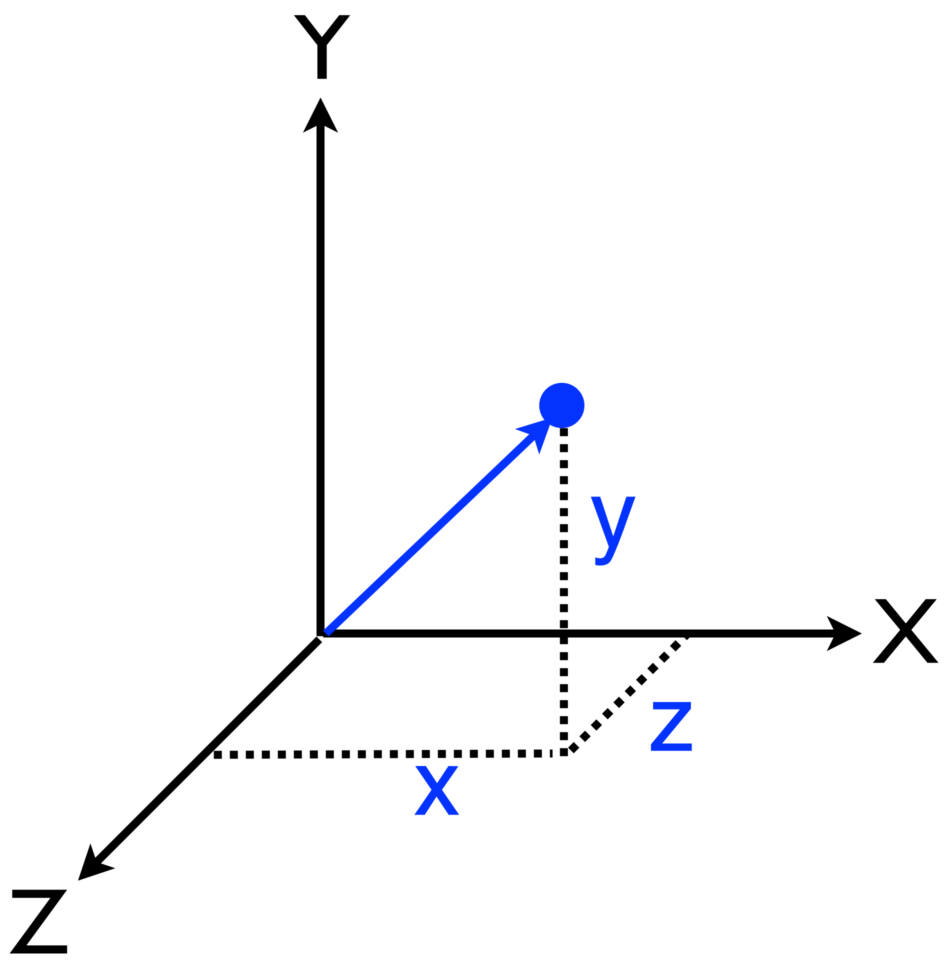


# Vector Space Representation
review

- A 2-dimensional vector can be written as [x,y]
- A 3-dimensional vector can be written as [x,y,z]



# Vector Space Representation

## review

| w_1 | w_2 | w_3 | w_4 | w_5 | w_6 | w_7 | w_8 | w_9 | w_10 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |

# Vector Space Representation

## review

| w_1 | w_2 | w_3 | w_4 | w_5 | w_6 | w_7 | w_8 | w_9 | w_10 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |

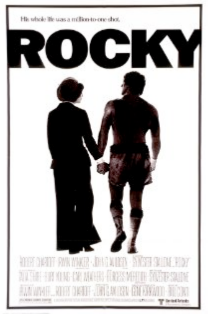
- We can represent this document as a vector in a 10-dimensional vector space

# Vector Space Representation

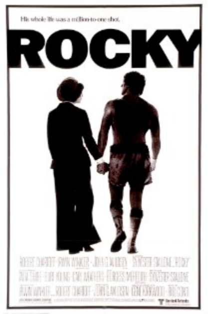
## review

| w_1 | w_2 | w_3 | w_4 | w_5 | w_6 | w_7 | w_8 | w_9 | w_10 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |

- This representation assumes binary term-weights.
- Are there other term-weighting schemes?

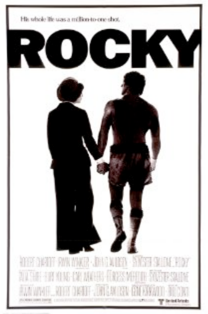



# TF, IDF, or TF.IDF?




# TF, IDF, or TF.IDF?

ability adrain adrian already apollo aspiring balboa
beat befriended befriends better boxer boxes boxing
canvas cash champion checks chooses collecting
collector creed current deadbeats debt debts
distance doesn downtown earns ease easily
exhibition explains extra extremely factory far forgot
gazzo gear giving gotten heavyweight idea interested
italian jergens keep living loan lot lovers managers match meat
mickey nobody odds packing paulie pennsylvania pet
philadelphia pittance promoter prove publicity
ready rocky sells shark sharp shop shy skills somebody spends
stallion struggling stunt supplies supposed surprised
thanksgiving think thrilled title touting trainer training
triumph unknown ve viciousness visits want willing win
won




# TF, IDF, or TF.IDF?

ability adrain adrian already apartment apollo aspiring balboa become
befriended befriends big boxer boxes boxing canvas champion chance checks
chooses collecting collector creed current deadbeats debt debts distance doesn downtown
earns ease easily exhibition extra extremely factory fight forgot gazzo gear gotten
heavyweight his is jergens later loan lot lovers managers match meat mickey named
nobody odds packing paulie pennsylvania pet philadelphia pittance promoter
publicity ready rocky sells set shark sharp shot shy somebody someone stallion store
struggling stunt supplies supposed surprised thanksgiving think thrilled time title touting trainer training
triumph up ve viciousness visits where who willing won works

# Vector Space Representation

review

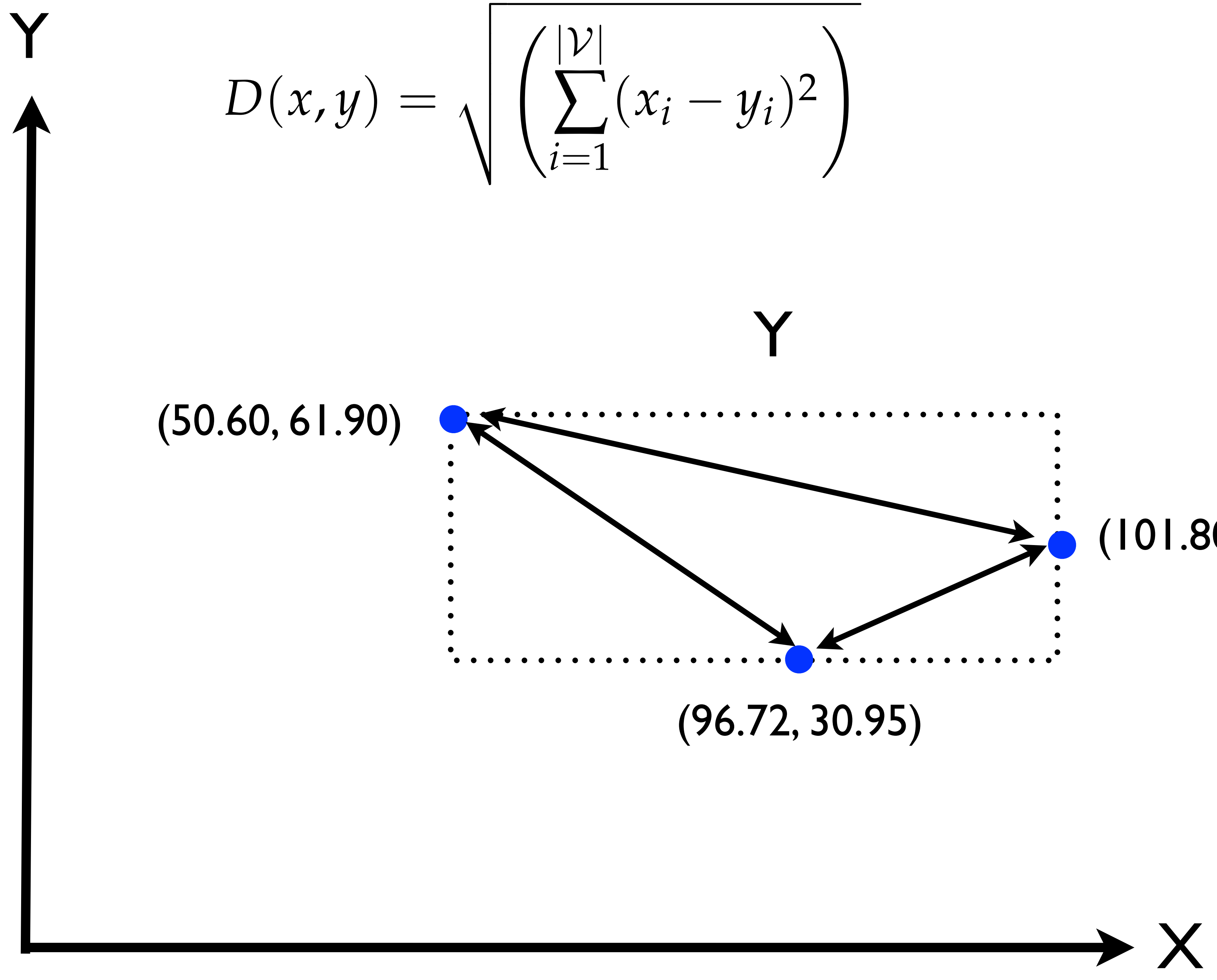
- Similarity = Euclidean Distance:

$$D(x, y) = \sqrt{\left( \sum_{i=1}^{|\mathcal{V}|} (x_i - y_i)^2 \right)}$$

# Vector Space Representation

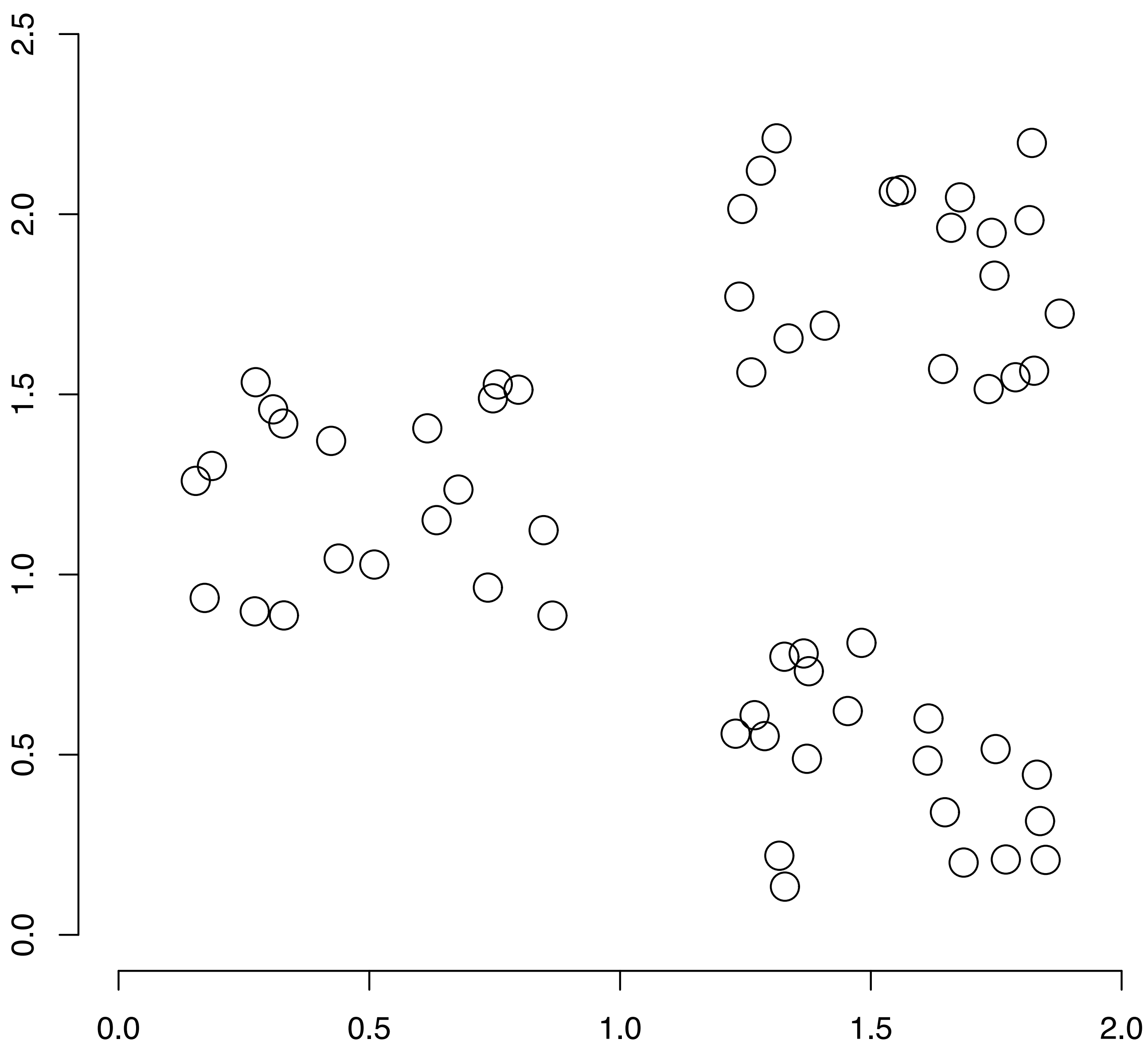
## review

$$D(x,y) = \sqrt{\left(\sum_{i=1}^{|\mathcal{V}|}(x_i - y_i)^2\right)}$$



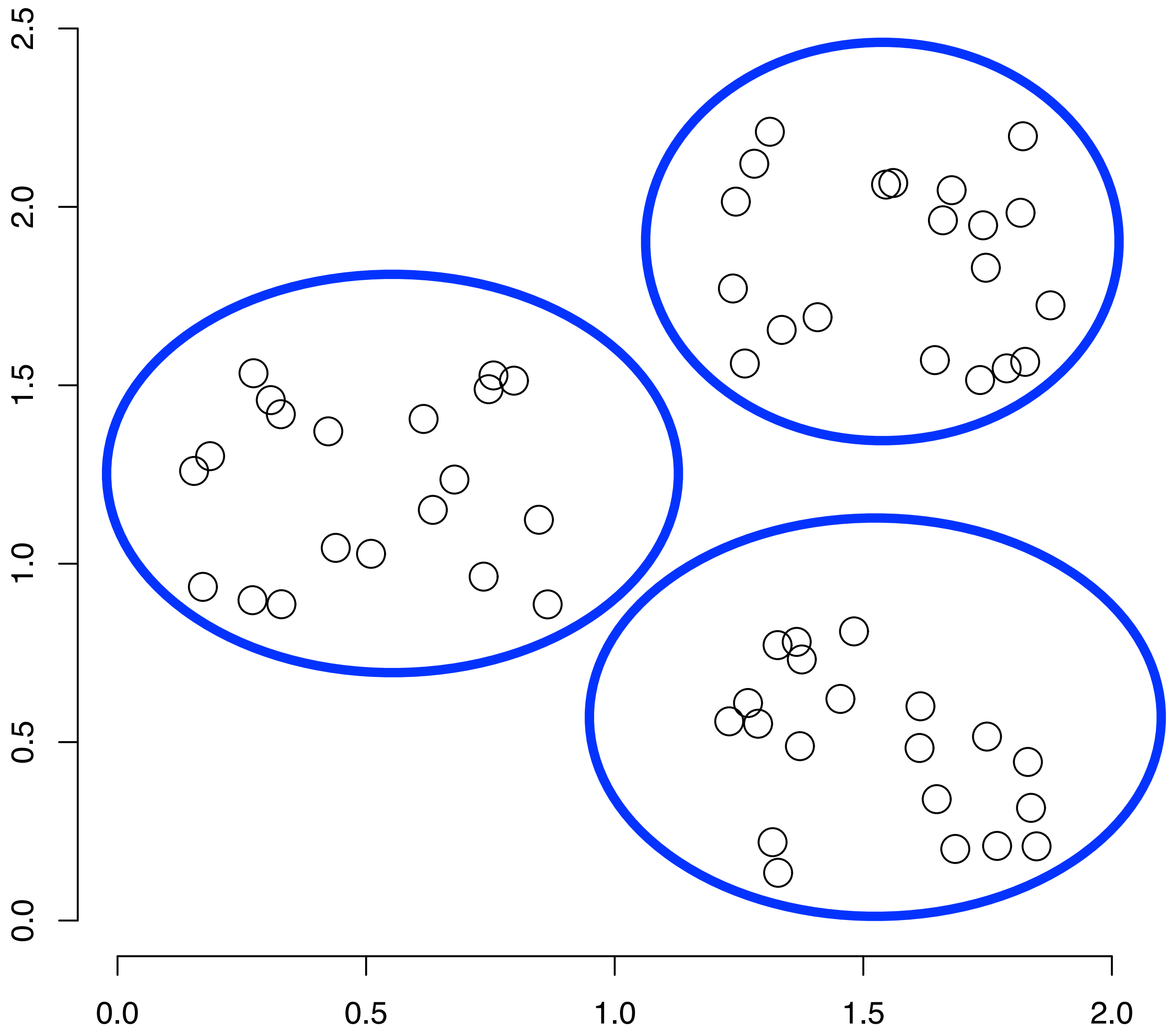
# Clustering

- What would we expect a clustering algorithm to do with this dataset?



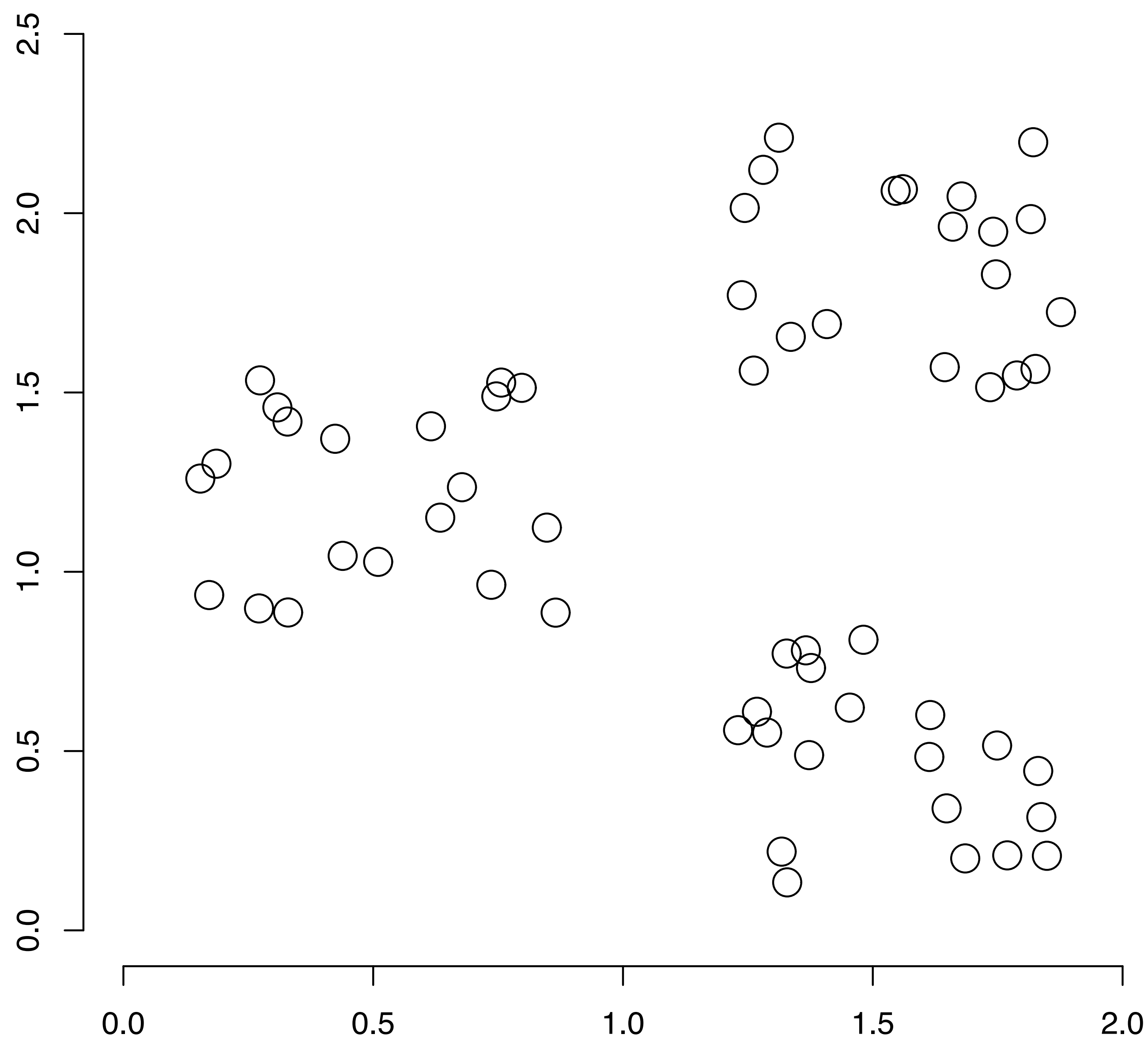
# Clustering

- What would we expect a clustering algorithm to do with this dataset?



# Clustering

- Propose an algorithm that might be able to do this!

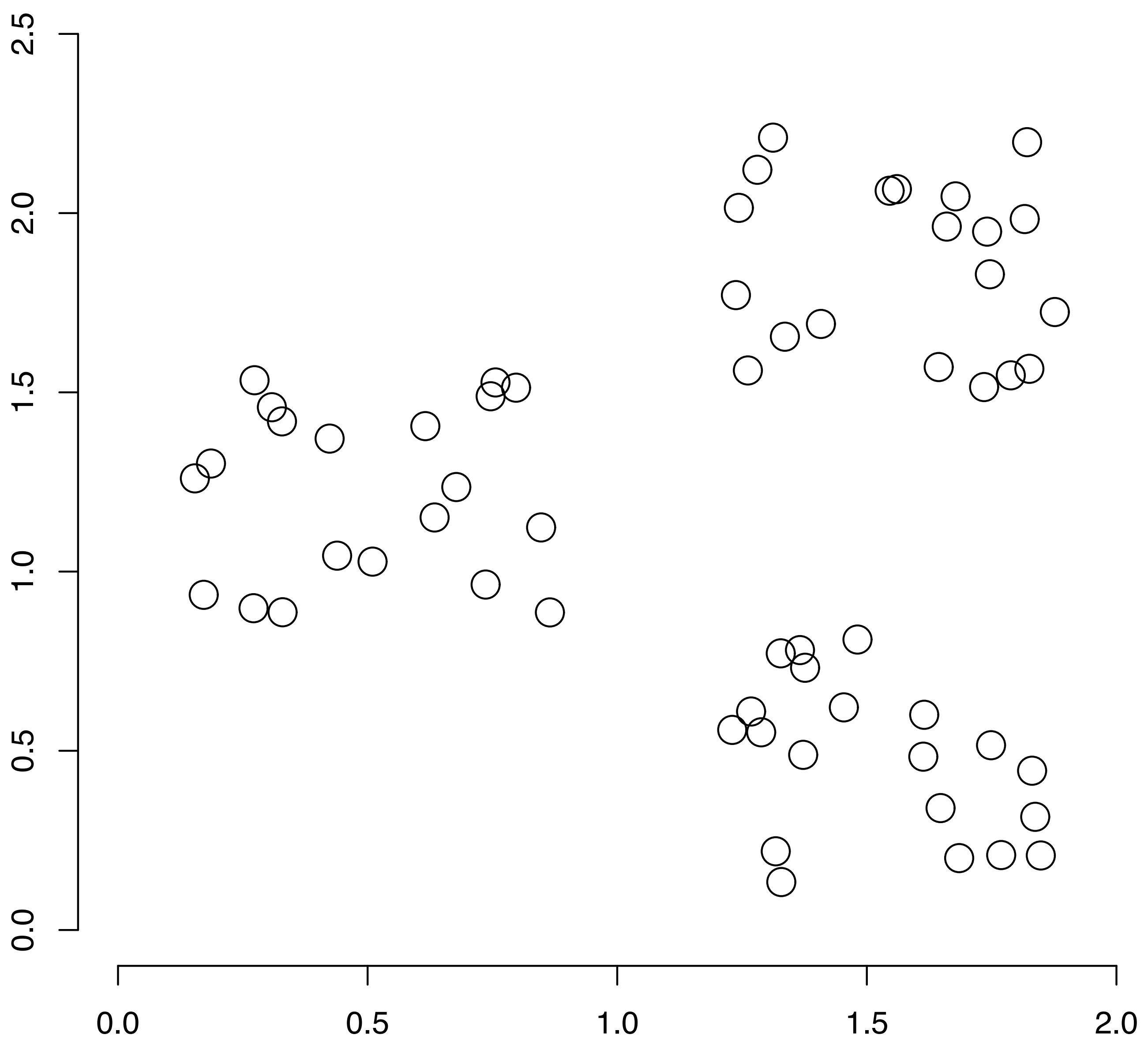


# Clustering

- Input: number of desired clusters K
- Output: assignment of documents to K clusters
- Algorithm:
  - randomly select K documents (seeds)
  - assign each remaining document to its nearest seed

# Clustering

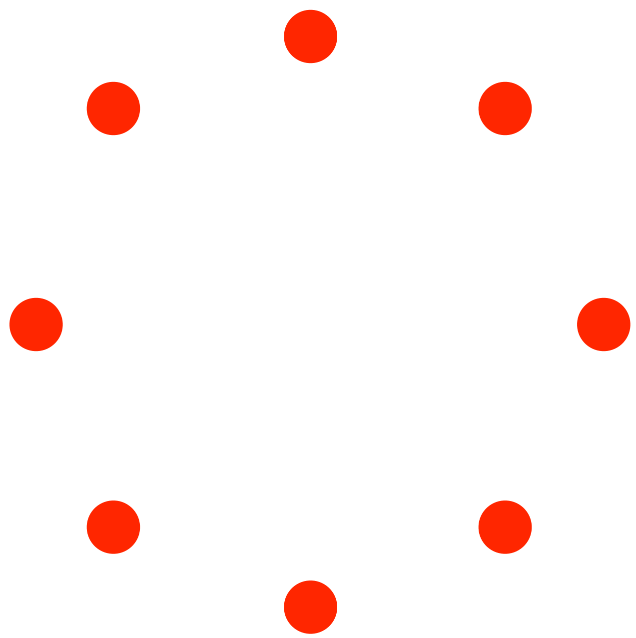
- Could this work?



# K-Means Clustering

# K-means Clustering
## cluster centroid

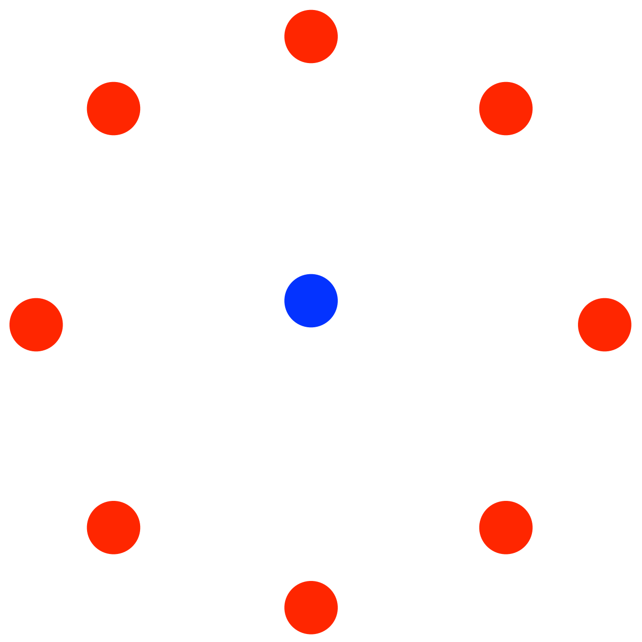
- The key to understanding K-means clustering is to understand the concept of a cluster centroid
- Given a cluster, you can think of its centroid as a point (or vector) that corresponds to its "center of mass"



# K-means Clustering

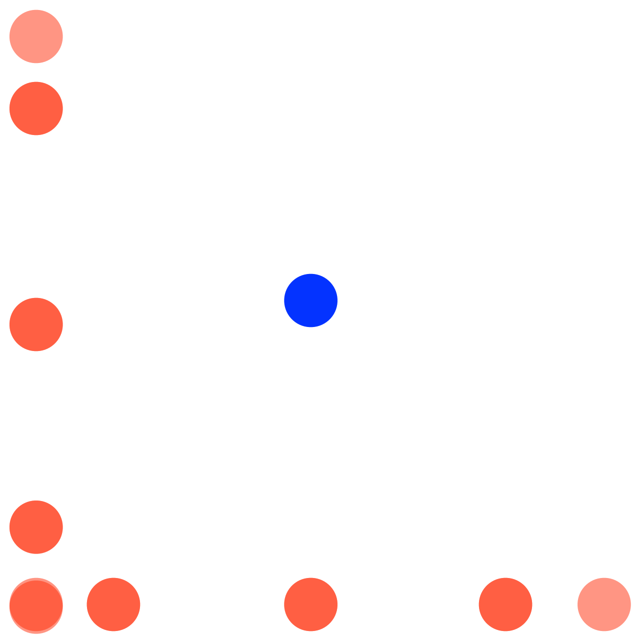
## cluster centroid

- The key to understanding K-means clustering is to understand the concept of a cluster centroid
- Given a cluster, you can think of its centroid as a point (or vector) that corresponds to its "center of mass"



# K-means Clustering
## cluster centroid

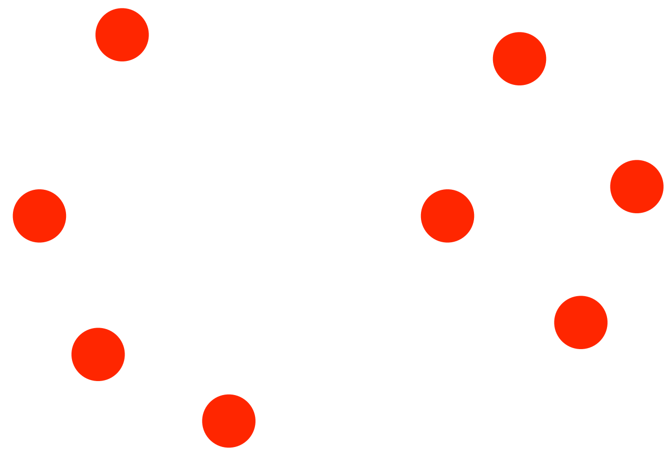
- The key to understanding K-means clustering is to understand the concept of a cluster centroid
- Given a cluster, you can think of its centroid as a point (or vector) that corresponds to its “center of mass”



# K-means Clustering

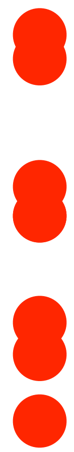
## cluster centroid

- The key to understanding K-means clustering is to understand the concept of a cluster centroid
- Given a cluster, you can think of its centroid as a point (or vector) that corresponds to its "center of mass"



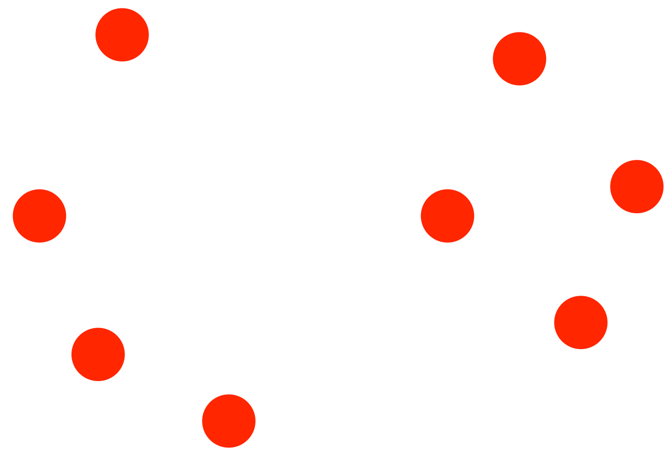
# K-means Clustering
## cluster centroid

- The key to understanding K-means clustering is to understand the concept of a cluster centroid
- Given a cluster, you can think of its centroid as a point (or vector) that corresponds to its "center of mass"



# K-means Clustering
## cluster centroid

- The key to understanding K-means clustering is to understand the concept of a cluster centroid
- Given a cluster, you can think of its centroid as a point (or vector) that corresponds to its "center of mass"

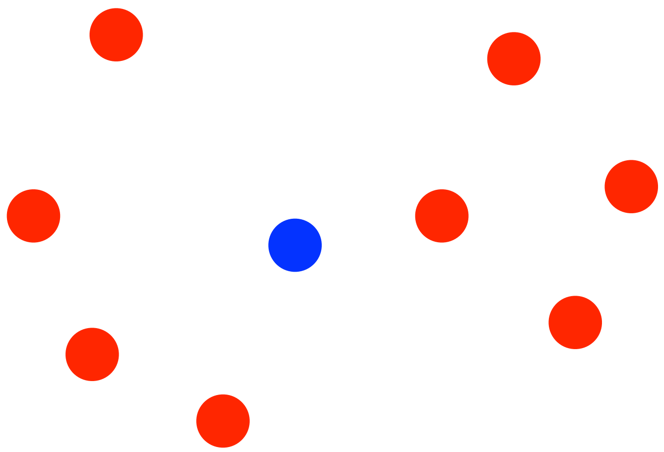


# K-means Clustering
## cluster centroid

- The key to understanding K-means clustering is to understand the concept of a cluster centroid
- Given a cluster, you can think of its centroid as a point (or vector) that corresponds to its “center of mass”

# K-means Clustering
## cluster centroid

- The key to understanding K-means clustering is to understand the concept of a cluster centroid
- Given a cluster, you can think of its centroid as a point (or vector) that corresponds to its "center of mass"



# K-means Clustering

## cluster centroid

docs assigned to cluster 1

| w_1 | w_2 | w_3 | w_4 | w_5 | w_6 | w_7 | w_8 | w_9 | w_10 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |

cluster 1 centroid

| w_1 | w_2 | w_3 | w_4 | w_5 | w_6 | w_7 | w_8 | w_9 | w_10 |
|---|---|---|---|---|---|---|---|---|---|
| ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |

# K-means Clustering

## cluster centroid

docs assigned to cluster 1

| w_1 | w_2 | w_3 | w_4 | w_5 | w_6 | w_7 | w_8 | w_9 | w_10 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |

cluster 1 centroid (average!)

| w_1 | w_2 | w_3 | w_4 | w_5 | w_6 | w_7 | w_8 | w_9 | w_10 |
|---|---|---|---|---|---|---|---|---|---|
| 0.33 | 0.5 | 0.5 | 0.5 | 1 | 0.33 | 0.33 | 0.83 | 0.5 | 0.5 |

# K-means Clustering

## cluster centroid

- For each dimension $i$ , set:

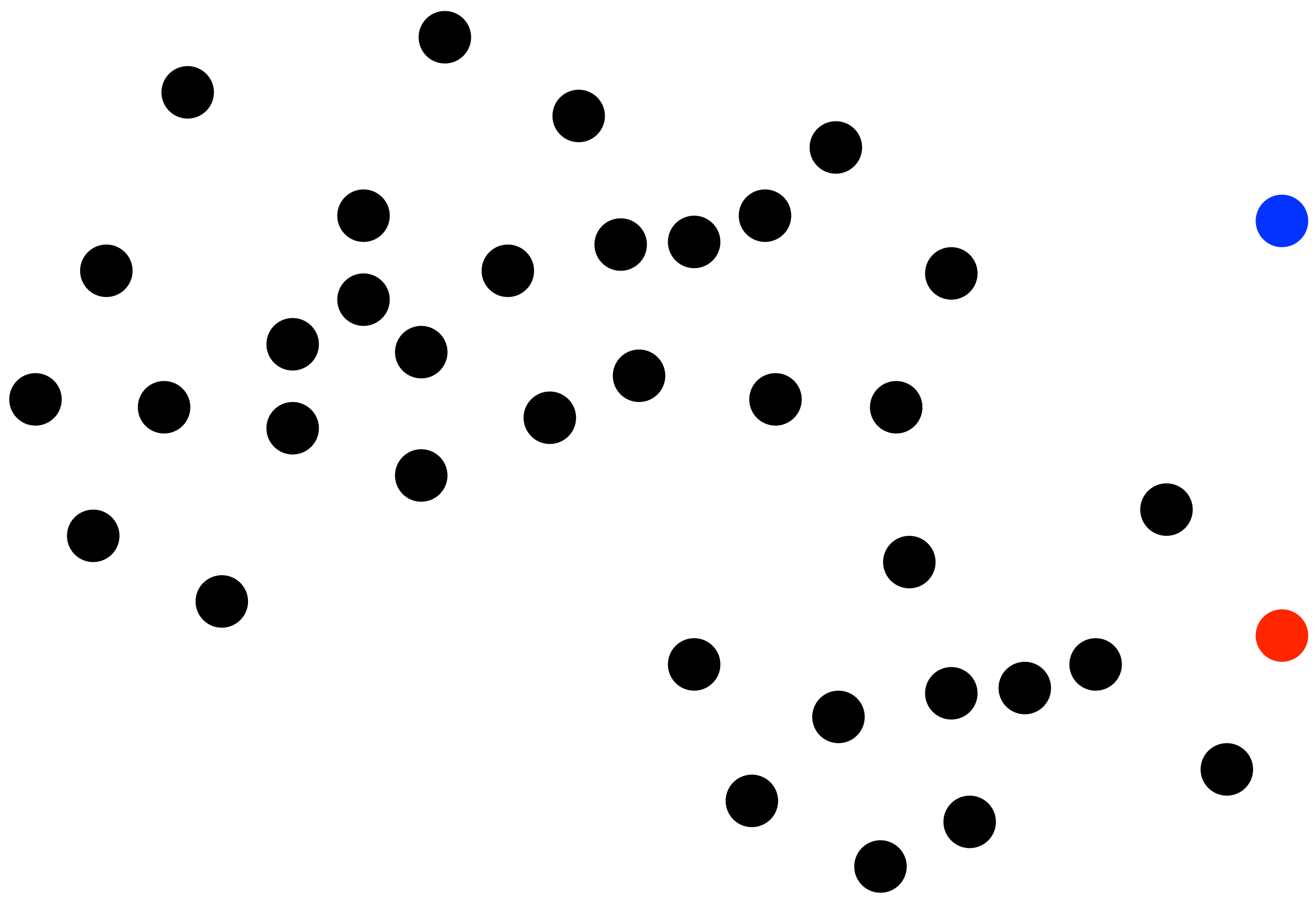
$$c_i = \frac{1}{|C|} \sum_{d \in C} d_i$$

# K-means Clustering

- Input: number of desired clusters K
- Output: assignment of documents to K clusters
- Algorithm:
  - Step 1: randomly select K documents (seeds)
  - Step 2: assign each document to its nearest seed
  - Step 3: compute all K cluster centroids
  - Step 4: re-assign each document to its nearest centroid
  - Step 5: re-compute all K cluster centroids
  - Step 6: repeat steps 4 and 5 until <u>terminating condition</u>

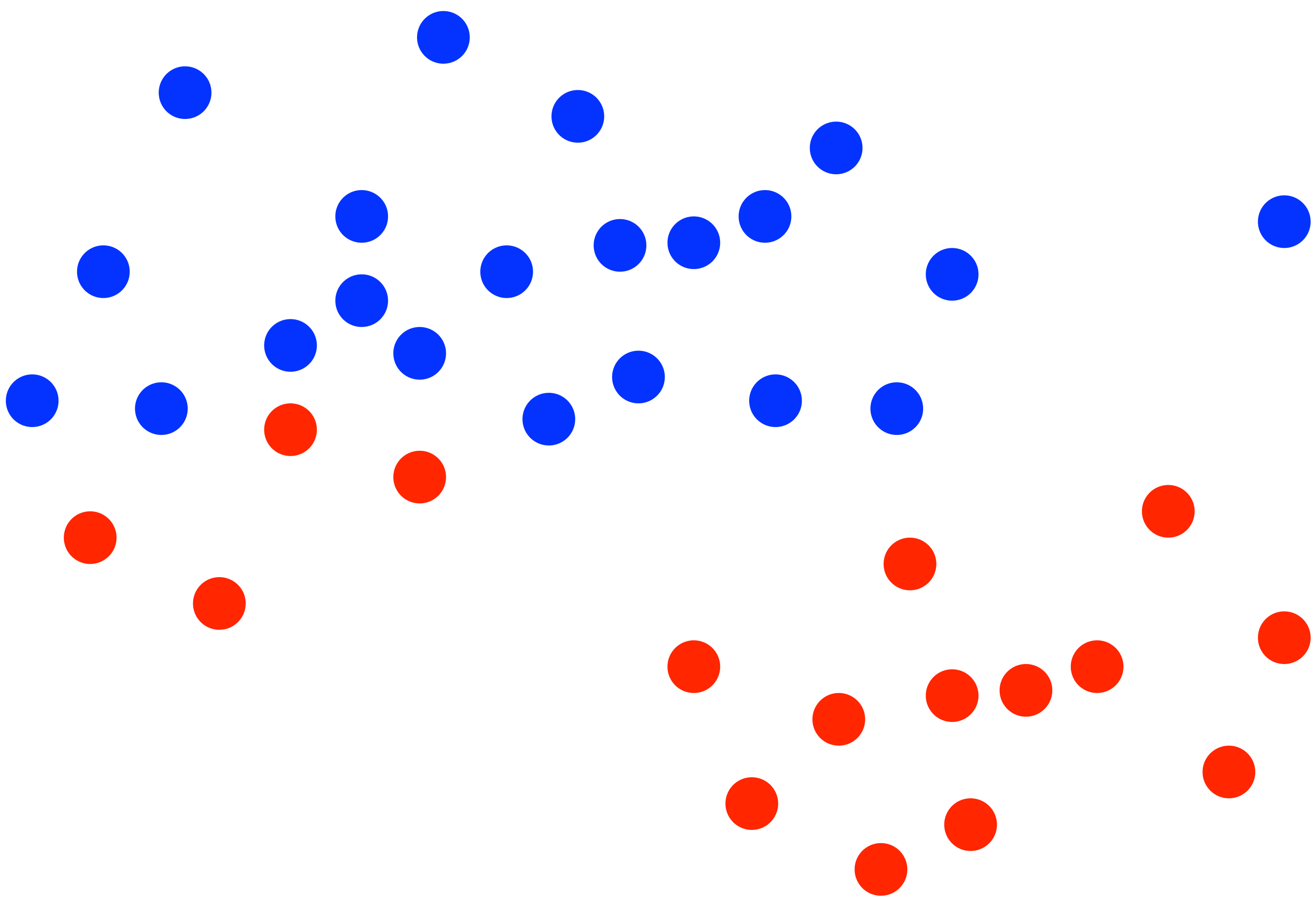
# K-means Clustering

- Step 1: randomly select K documents (seeds)



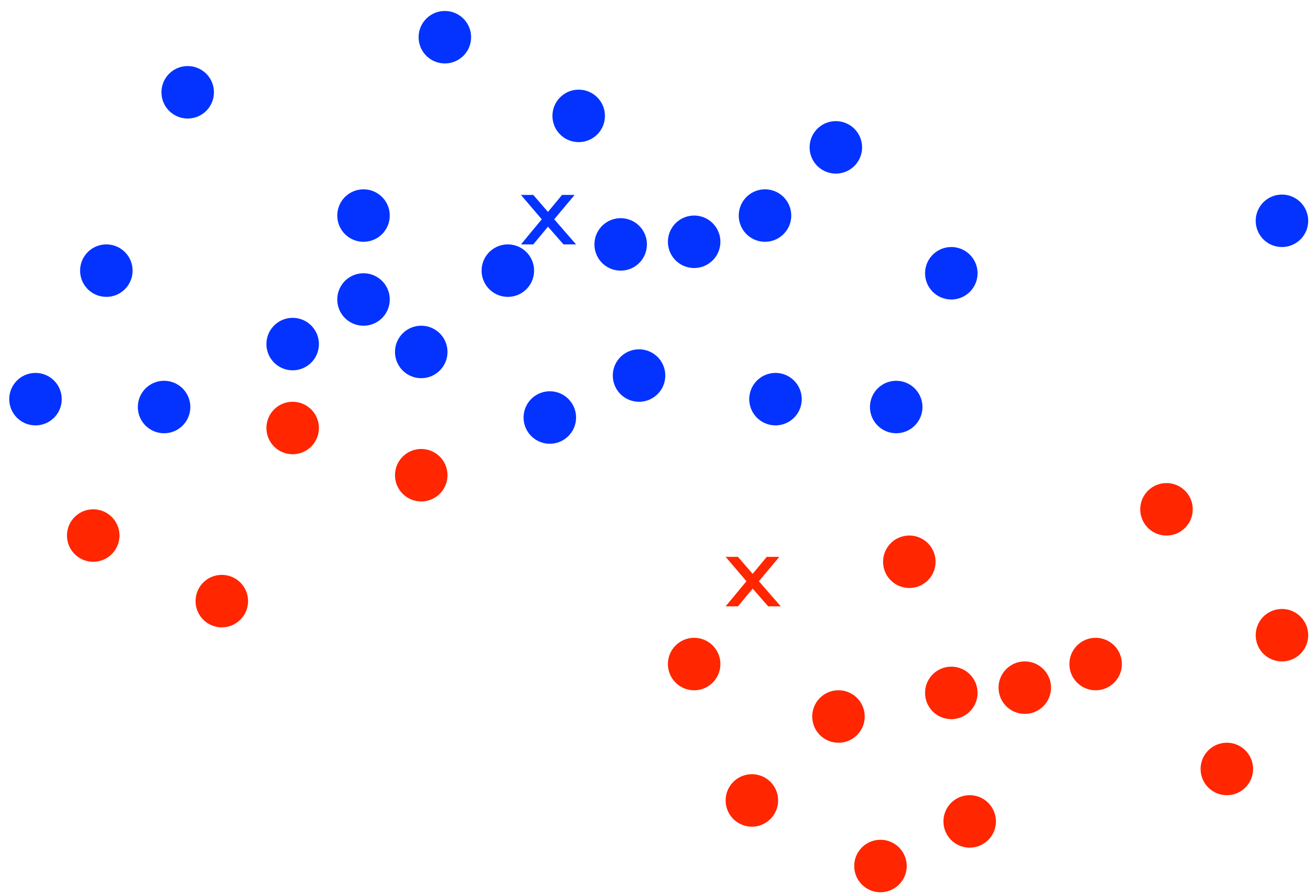
# K-means Clustering

- Step 2: assign each document to its nearest seed



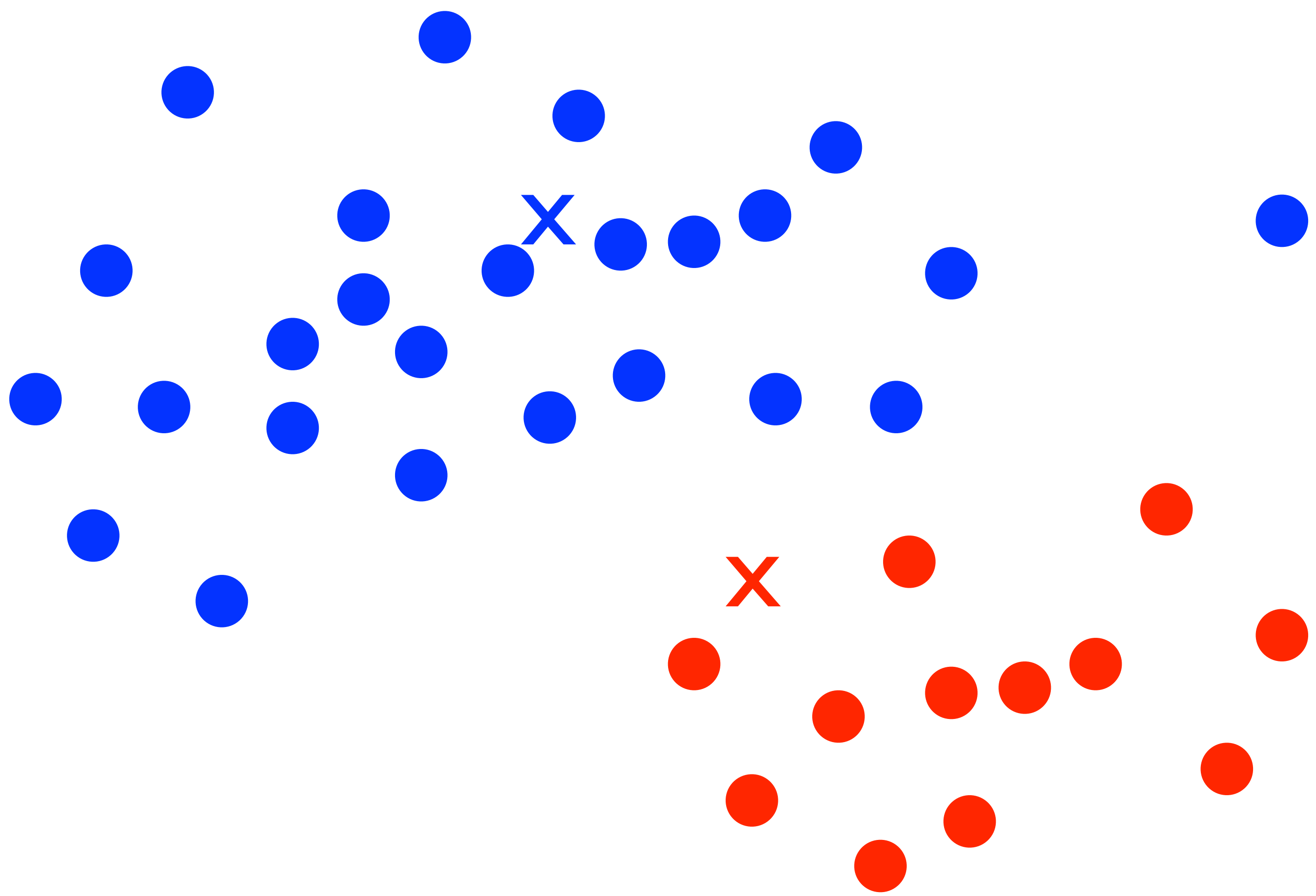
# K-means Clustering

- Step 3: compute all K cluster centroids



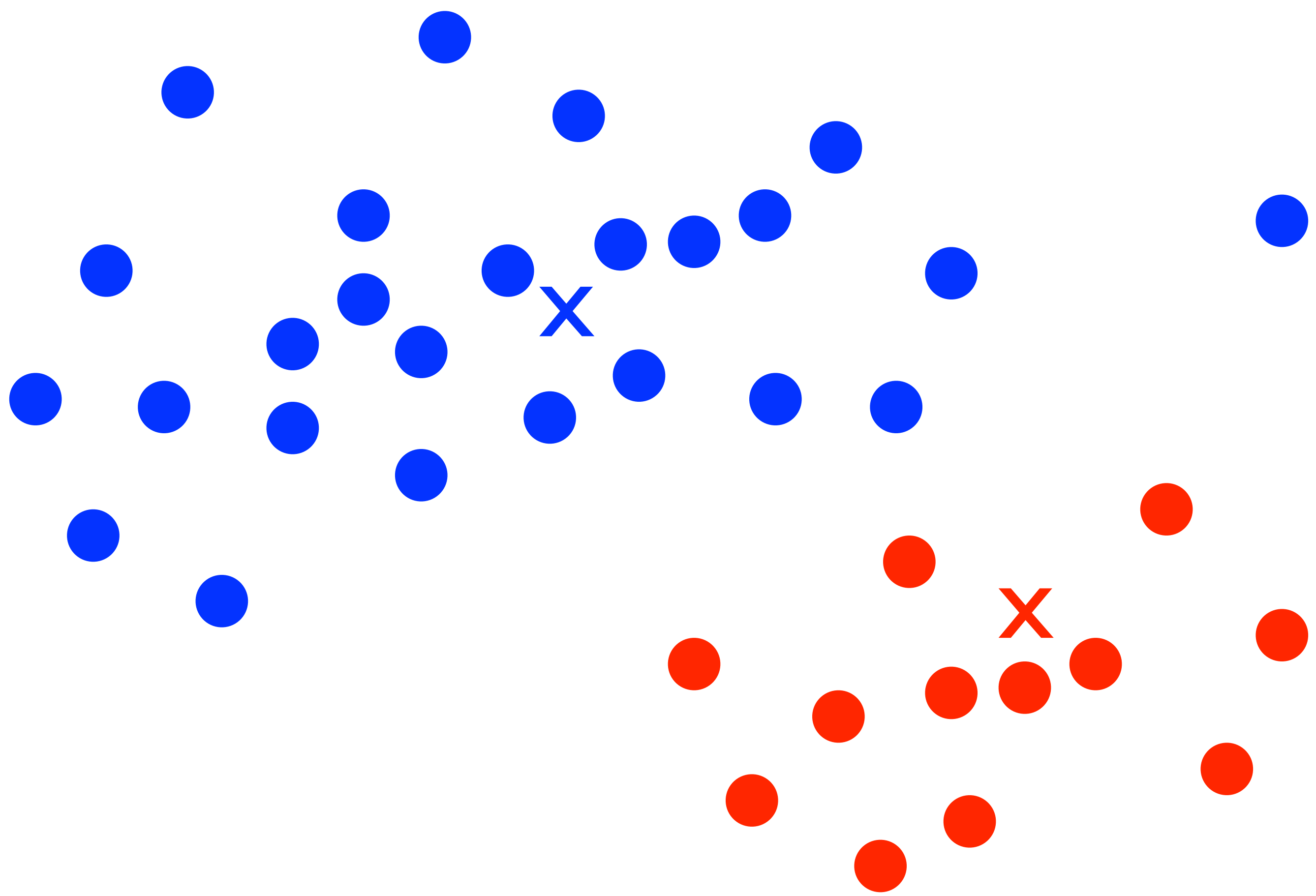
# K-means Clustering

- Step 4: re-assign each document to its nearest centroid



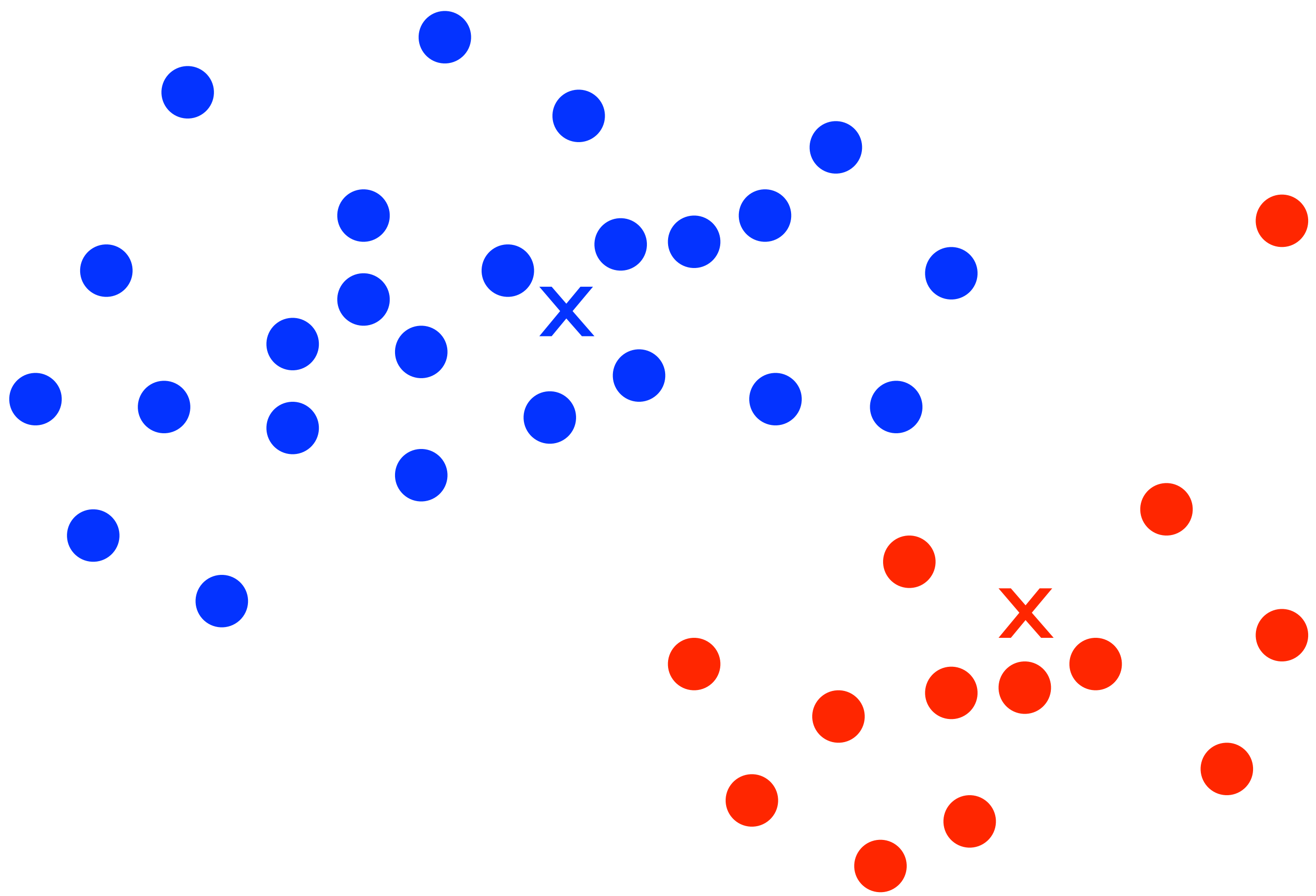
# K-means Clustering

- Step 4: re-compute all K cluster centroids



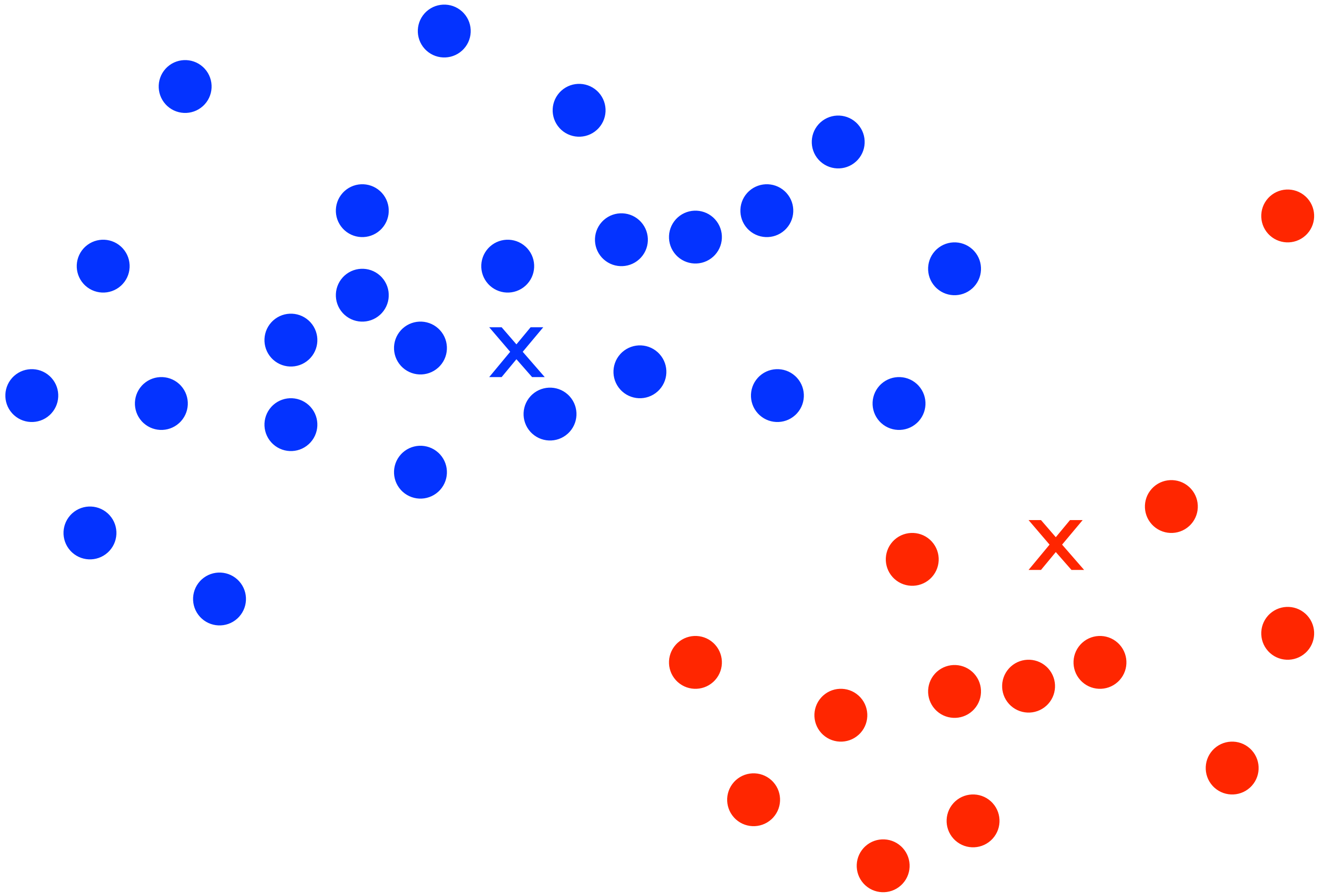
# K-means Clustering

- Step 5: re-assign each document to its nearest centroid



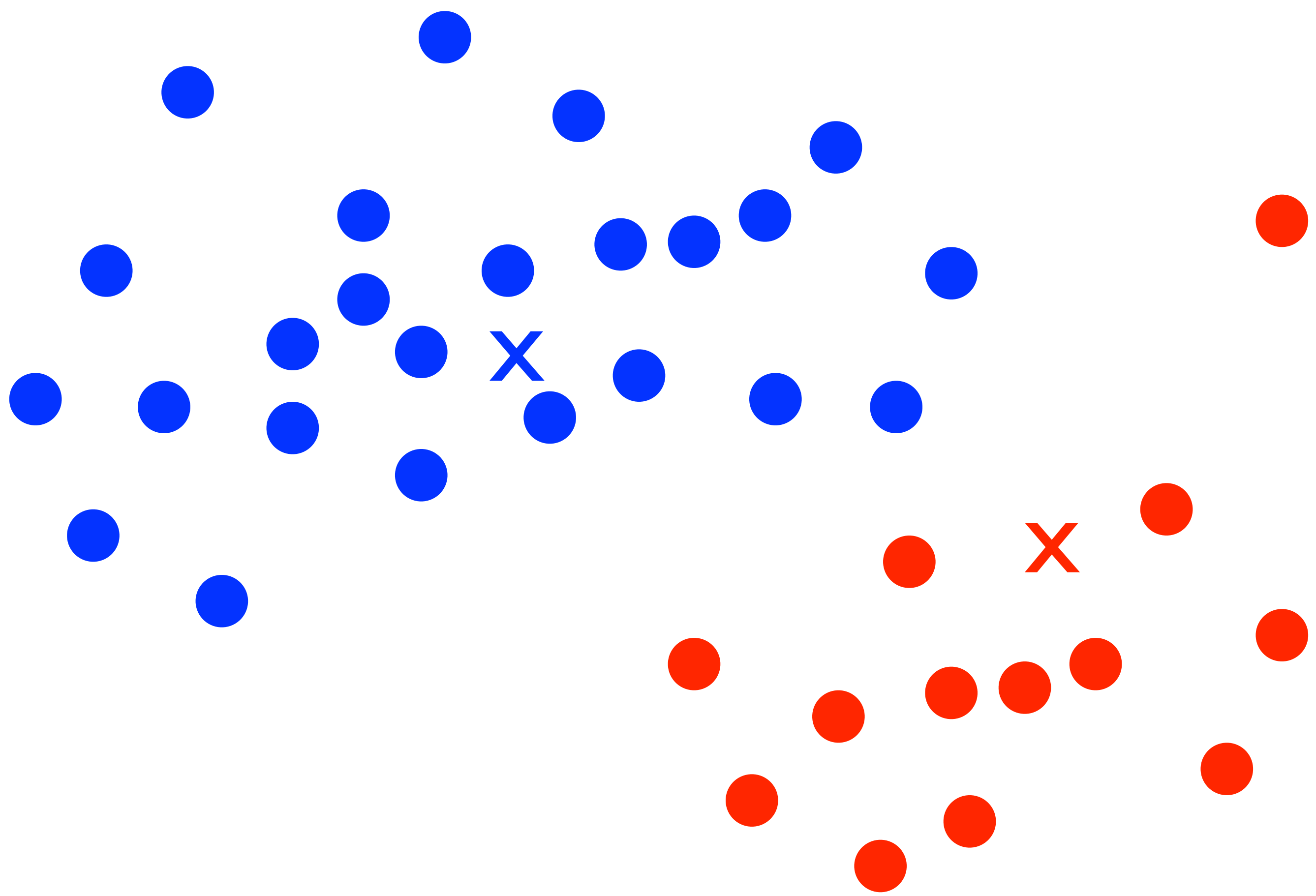
# K-means Clustering

- Step 4: re-compute all K cluster centroids



# K-means Clustering

- Step 5: re-assign each document to its nearest centroid



# K-means Clustering

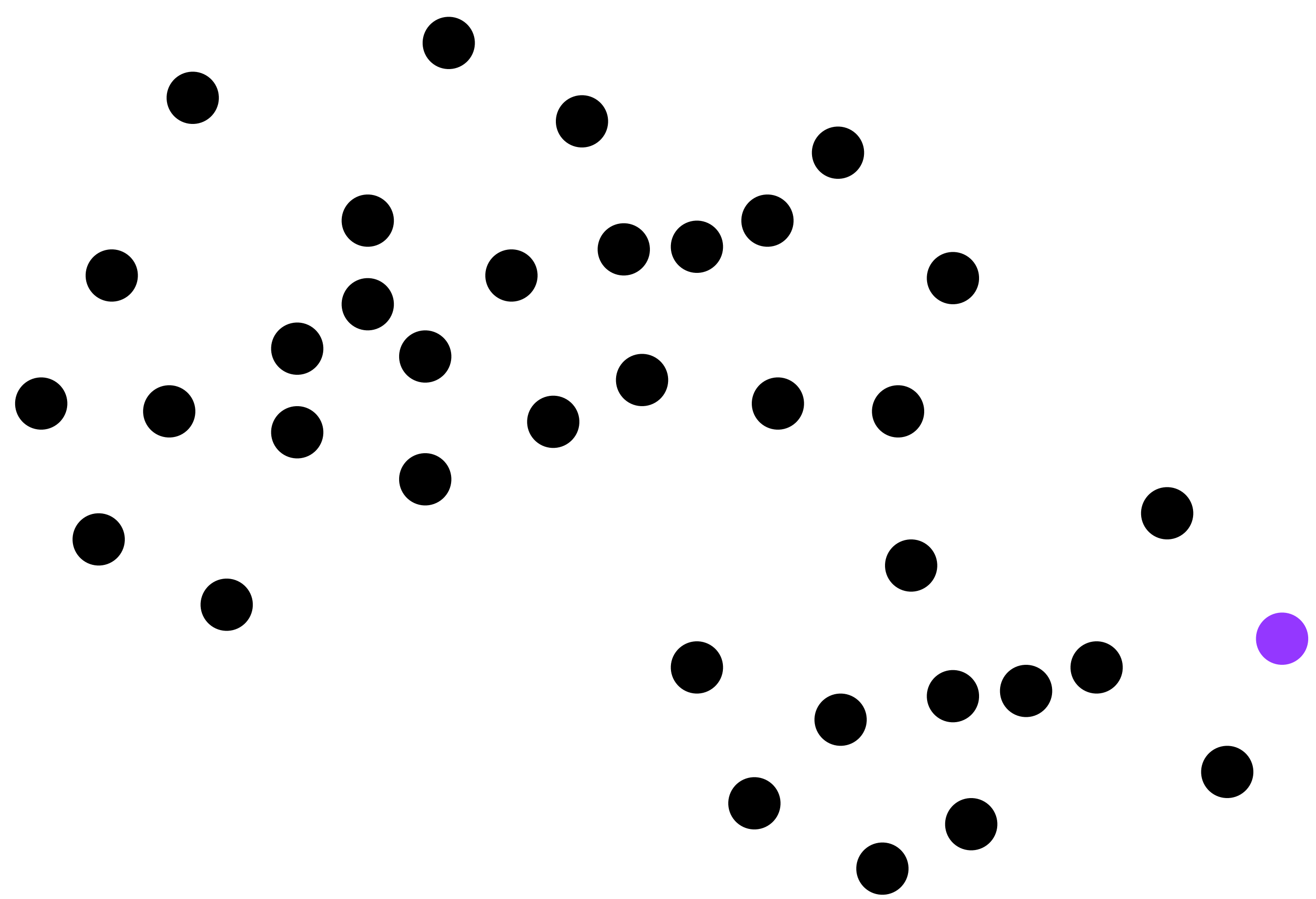
- Input: number of desired clusters K
- Output: assignment of documents to K clusters
- Algorithm:
    - Step 1: randomly select K documents (seeds)
    - Step 2: assign each document to its nearest seed
    - Step 3: compute all K cluster centroids
    - Step 4: re-assign each document to its nearest centroid
    - Step 5: re-compute all K cluster centroids
    - Step 6: repeat steps 4 and 5 until <u>terminating condition</u>

# K-means Clustering
## potential drawback

- The quality of the output clustering depends on the choice of K and on the initial seeds
- In many cases, the choice of K is pre-determined by the application
  - Search engine results clustering: grouping search engine results by topic
  - Collection clustering: grouping documents by topic to support navigation and exploration
- Later we'll see ways of setting K dynamically

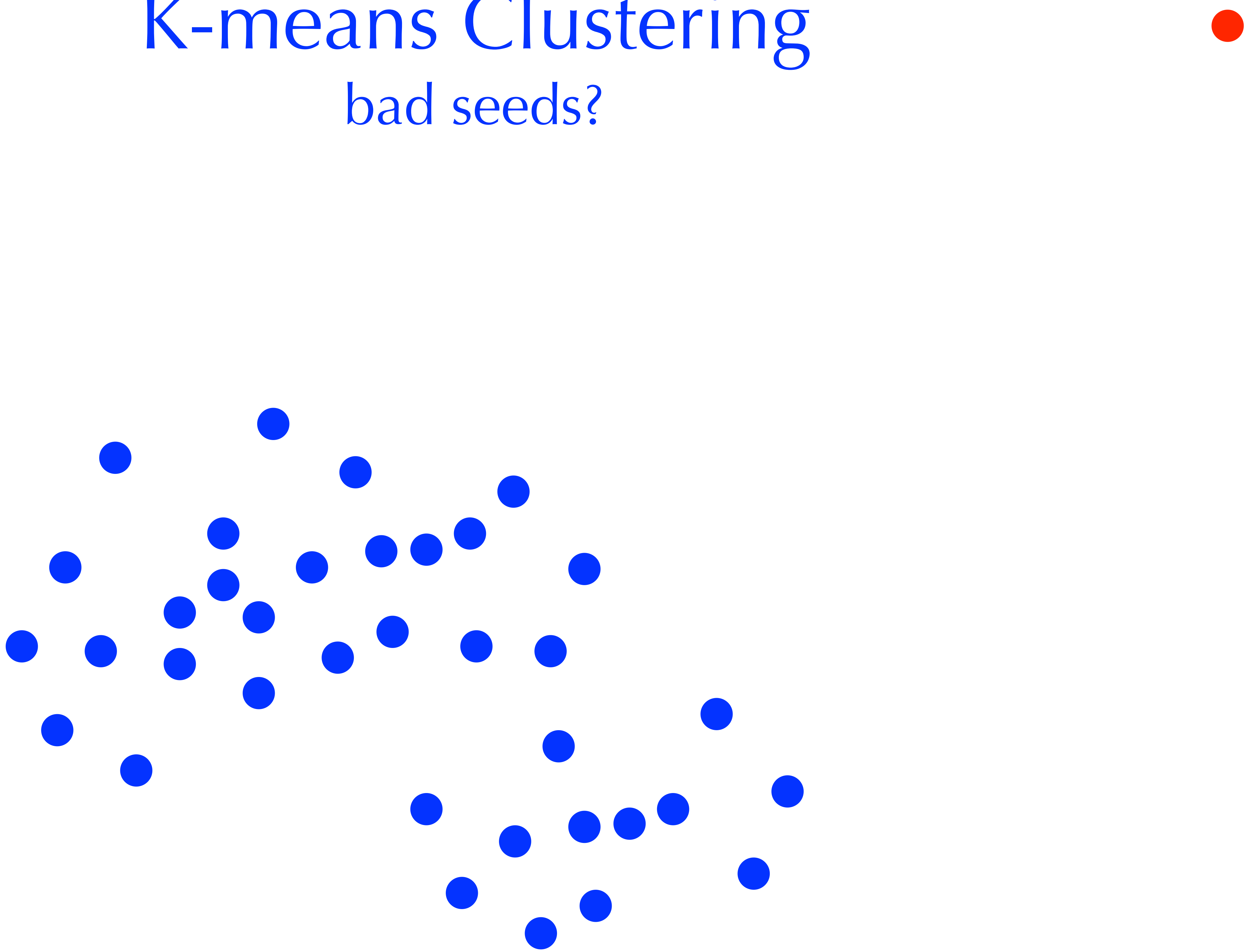
# K-means Clustering

## bad seeds?

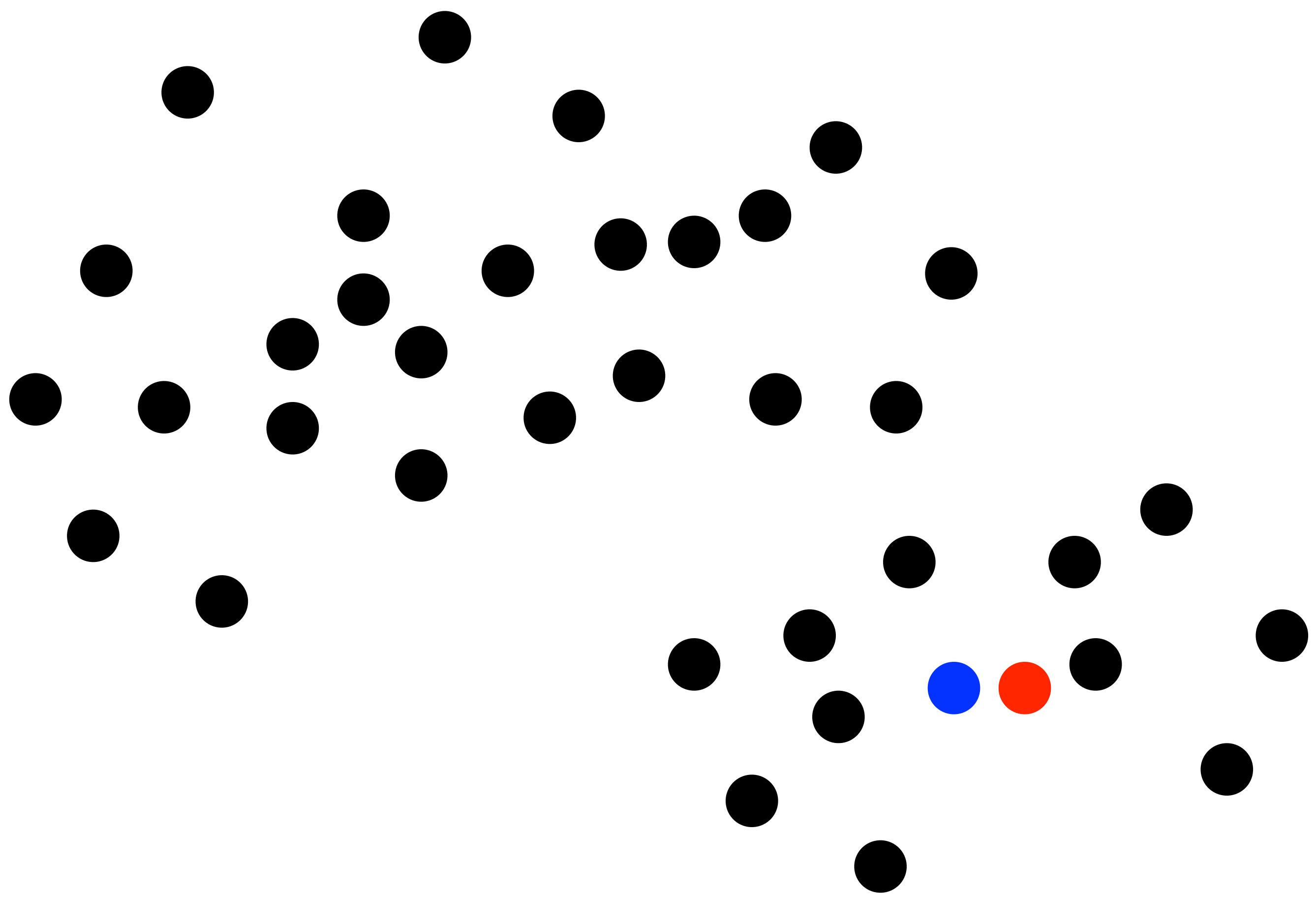


# K-means Clustering

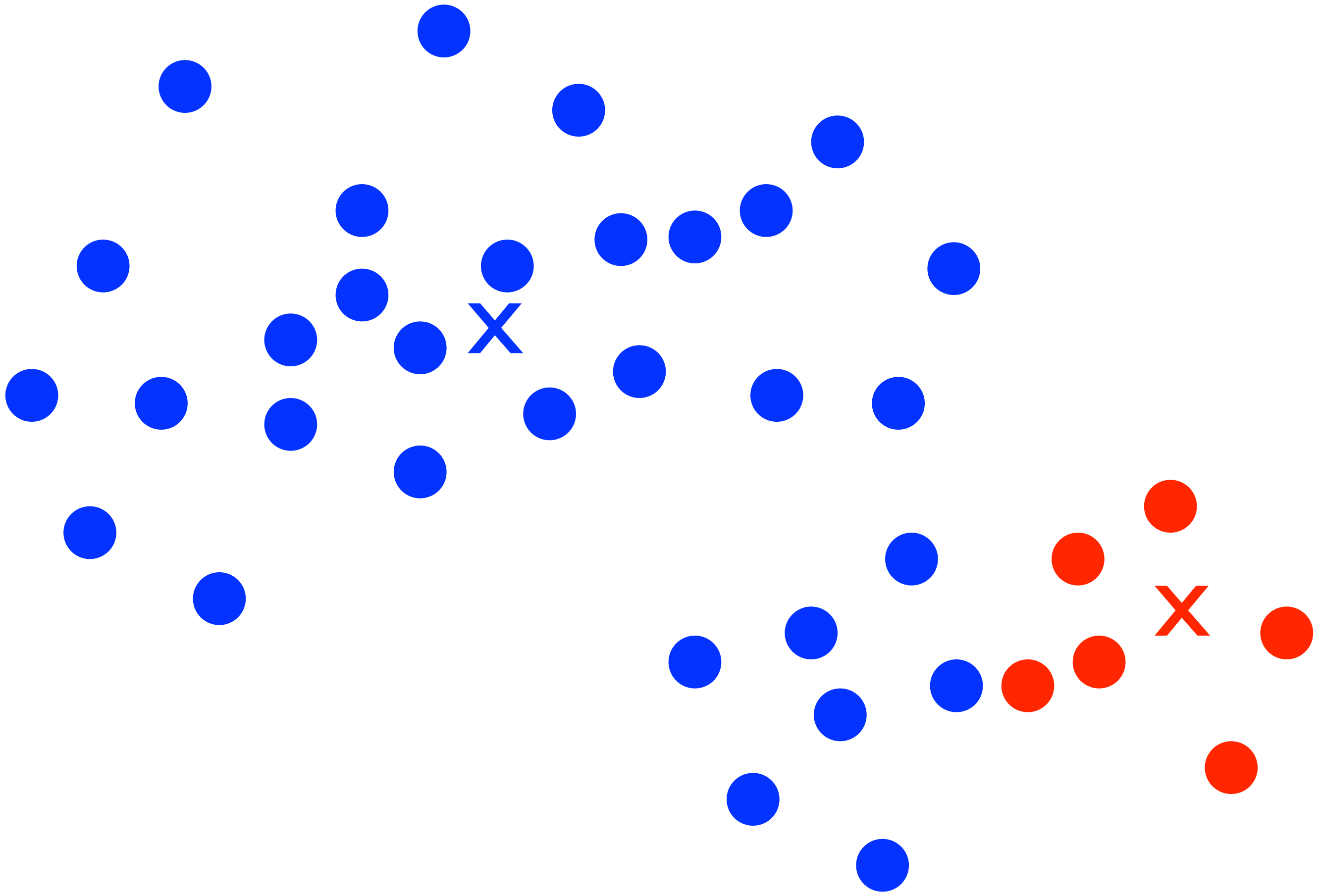
## bad seeds?



# K-means Clustering
## bad seeds?

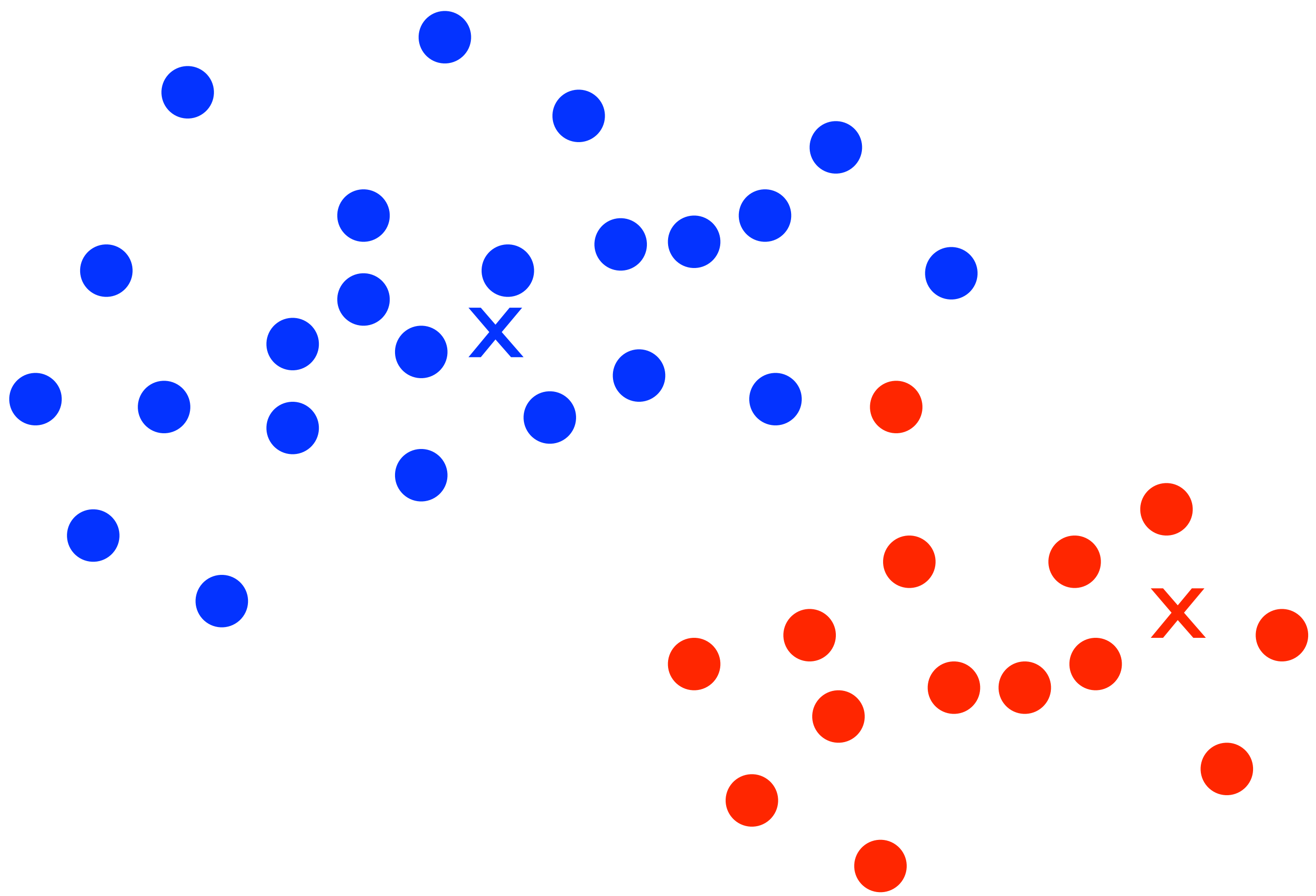


# K-means Clustering
## bad seeds?



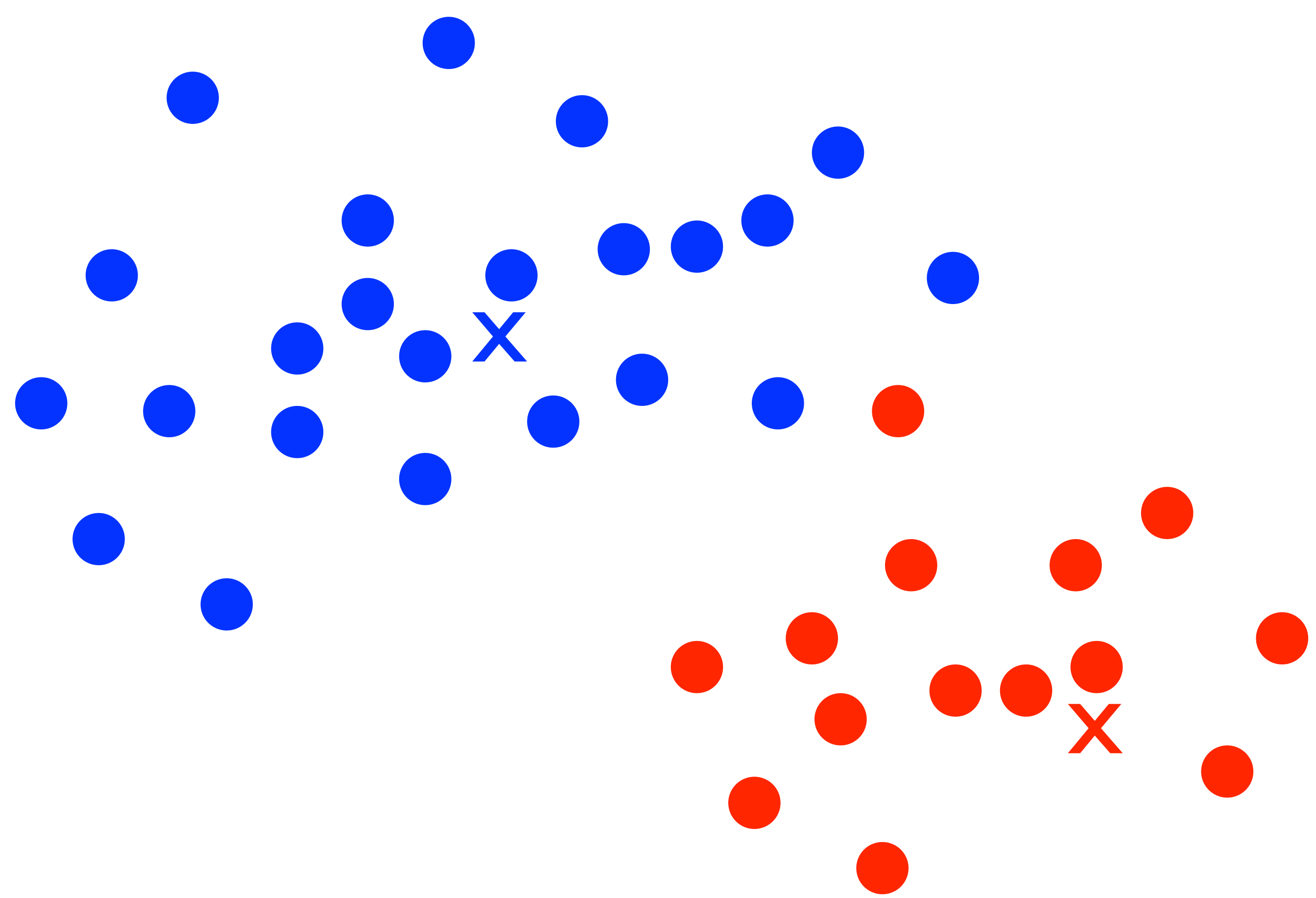
# K-means Clustering

## bad seeds?



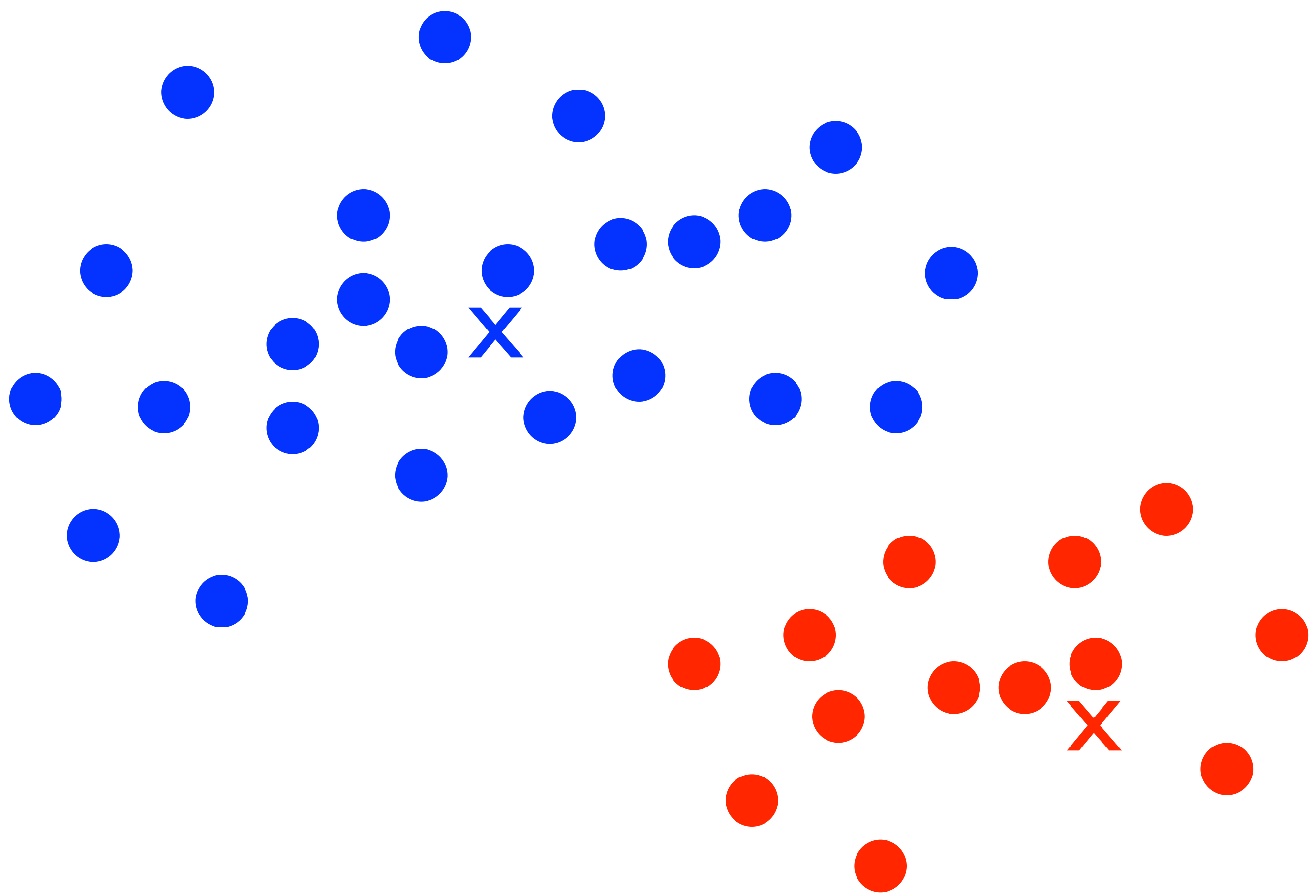
# K-means Clustering

## bad seeds?



# K-means Clustering

## bad seeds?



# K-means Clustering

## bad seeds

- It's difficult to know which seeds will yield a high-quality clustering
- However, it's usually a good idea to avoid seeds that are outliers
- How would you detect outliers?

# K-means Clustering
## clustering evaluation

- What does it mean for a clustering to be high quality anyway?
- What is the goal of clustering again?

# K-means Clustering
## internal evaluation

- In theory, a good clustering should have:
  - Similar documents in the same clusters
  - Different documents in different clusters

# K-means Clustering
## internal evaluation

$$\text{Clustering Quality} = \left( \text{Average distance between all pairs of documents in } \underline{\text{different clusters}} \right) - \left( \text{Average distance between all pairs of documents in the } \underline{\text{same cluster}} \right)$$

# K-means Clustering

## improved k-means

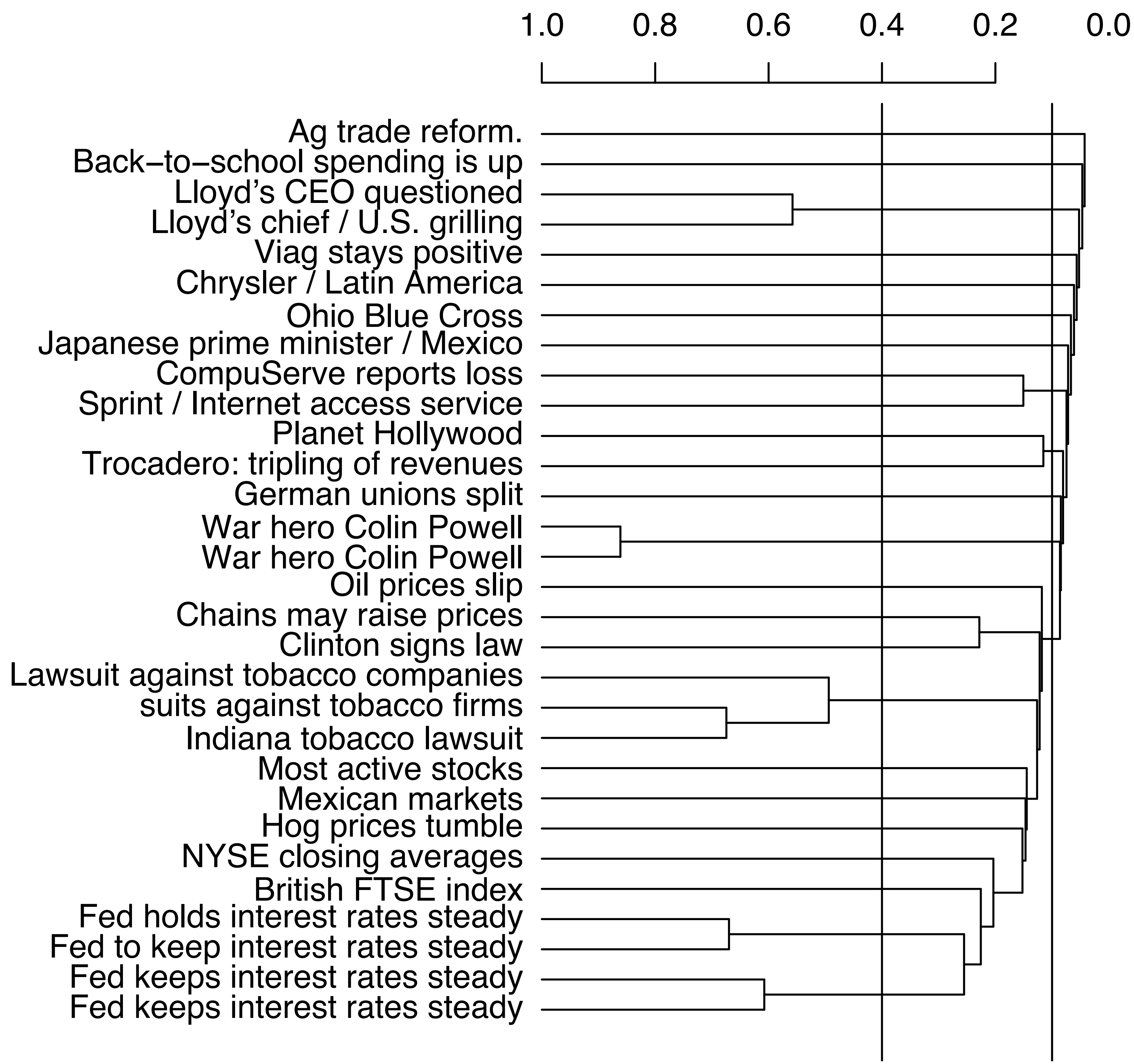
- Given a set of documents and a value K,  run K-means clustering N times and keep the clustering that produces the greatest difference between the inter-cluster distance and the intra-cluster distance

# Bottom-up Agglomerative Clustering

# Bottom-up Clustering

- While K-means requires setting K, bottom-up clustering groups the data in a hierarchical fashion
- We can then set K after the clustering is done or use a distance threshold to set K dynamically (more on this later)

# Bottom-up Clustering



# Bottom-up Clustering

- Input: data
- Output: cluster hierarchy
- Algorithm:
  - Step 1: consider every document its own cluster
  - Step 2: compute the distance between all cluster pairs
  - Step 3: merge/combine the nearest two clusters into one
  - Step 4: repeat steps 2 and 3 until every document is in one cluster

# Bottom-up Clustering

- Input: data
- Output: cluster hierarchy
- Algorithm:
  - Step 1: consider every document its own cluster
  - Step 2: compute the distance between all cluster pairs
  - Step 3: merge/combine the nearest two clusters into one
  - Step 4: repeat steps 2 and 3 until every document is in one cluster

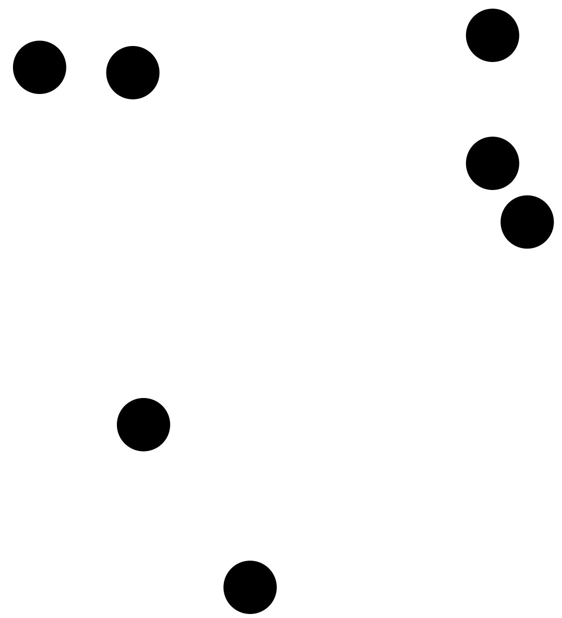
# Bottom-up Clustering

- Computing the distance between two clusters
- Single-Link: the distance between the two nearest documents
- Complete-Link: the distance between the two documents that are farthest apart
- Average-Link: the average distance between all document pairs in the two different clusters
    - this is equivalent to using the distance between the two cluster centroids

# Bottom-up Clustering

## single-link

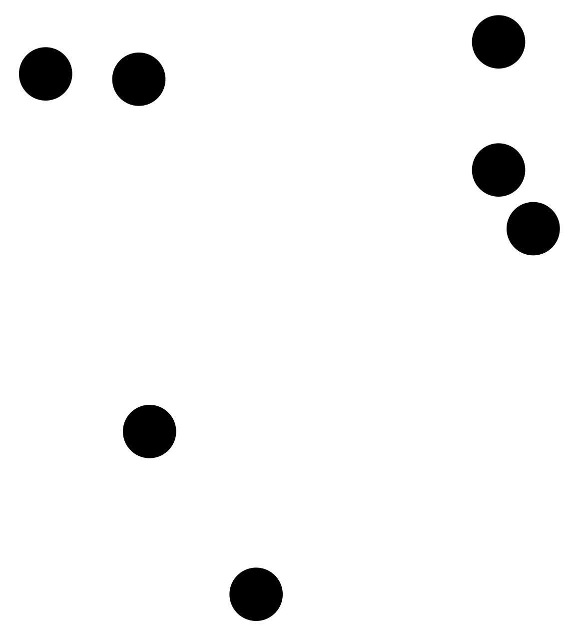
- Step 1: consider each document its own cluster



# Bottom-up Clustering

## single-link

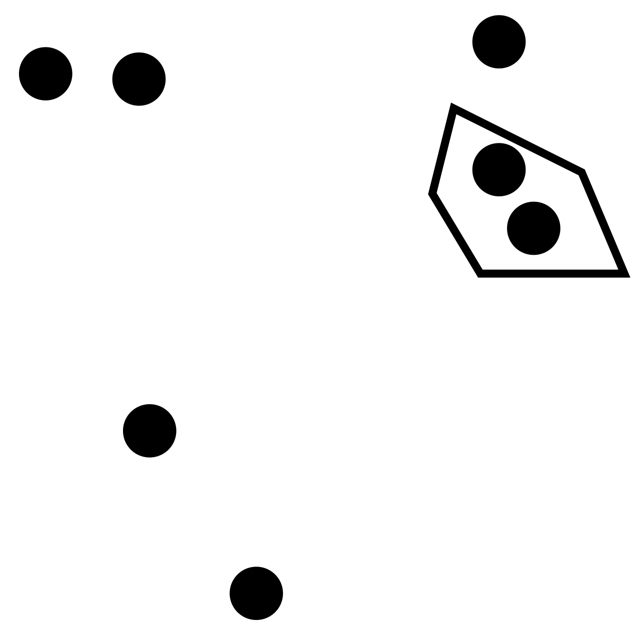
- Step 2: compute the distance between all cluster pairs
- Step 3: merge/combine the nearest two clusters into one



# Bottom-up Clustering

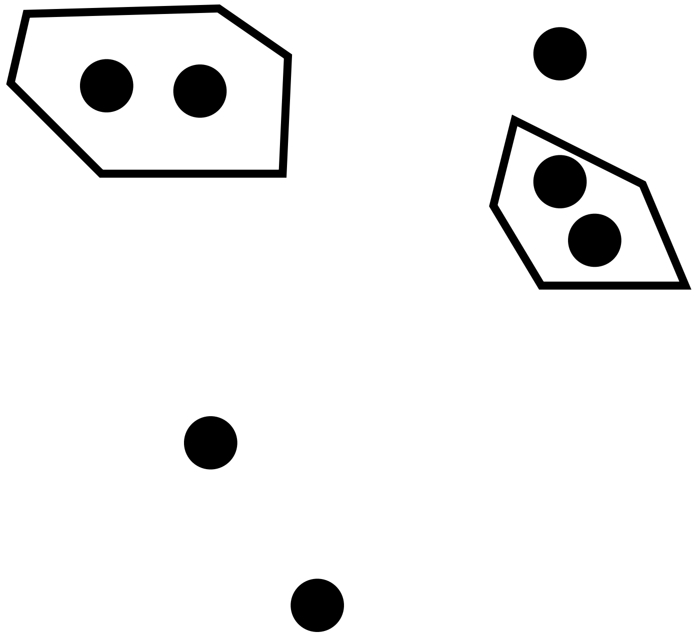
## single-link

- Step 2: compute the distance between all cluster pairs
- Step 3: merge/combine the nearest two clusters into one



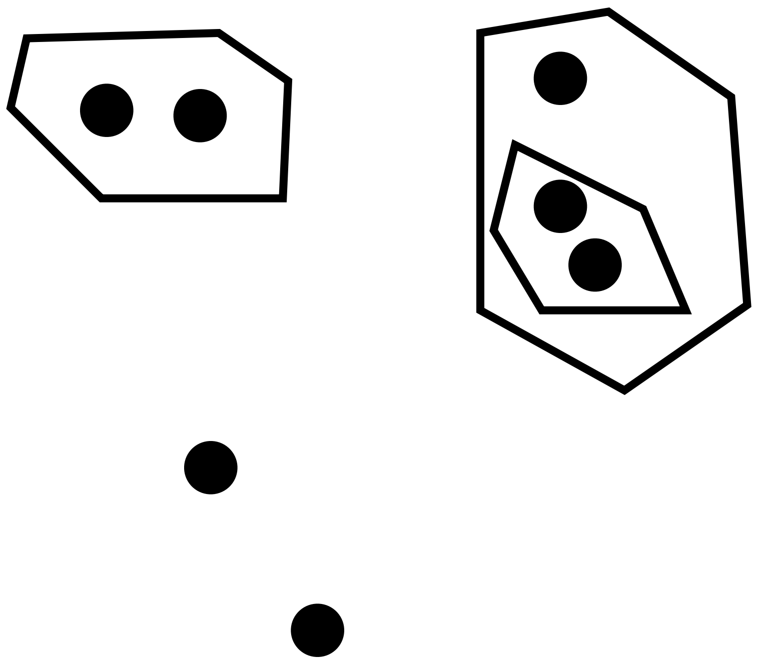
# Bottom-up Clustering
## single-link

- Step 2: compute the distance between all cluster pairs
- Step 3: merge/combine the nearest two clusters into one



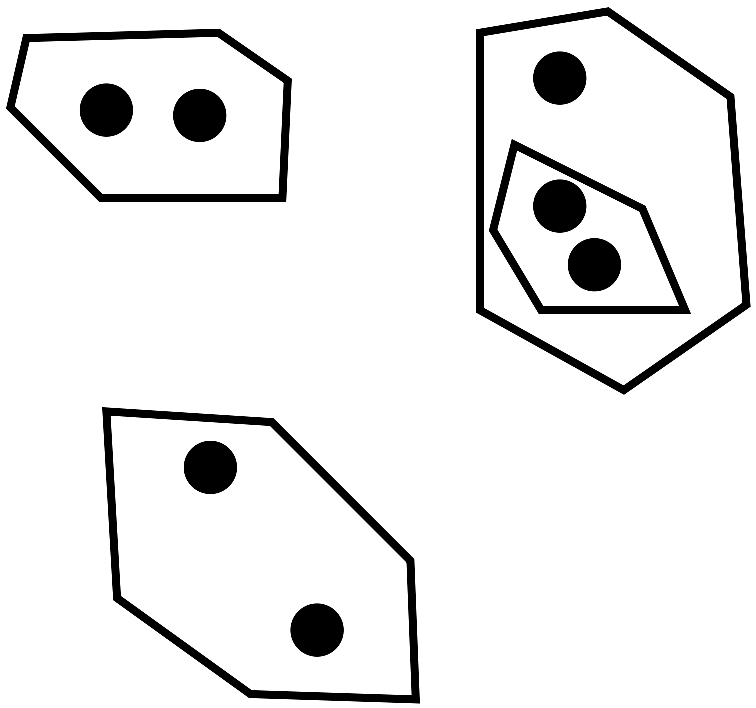
# Bottom-up Clustering
## single-link

- Step 2: compute the distance between all cluster pairs
- Step 3: merge/combine the nearest two clusters into one



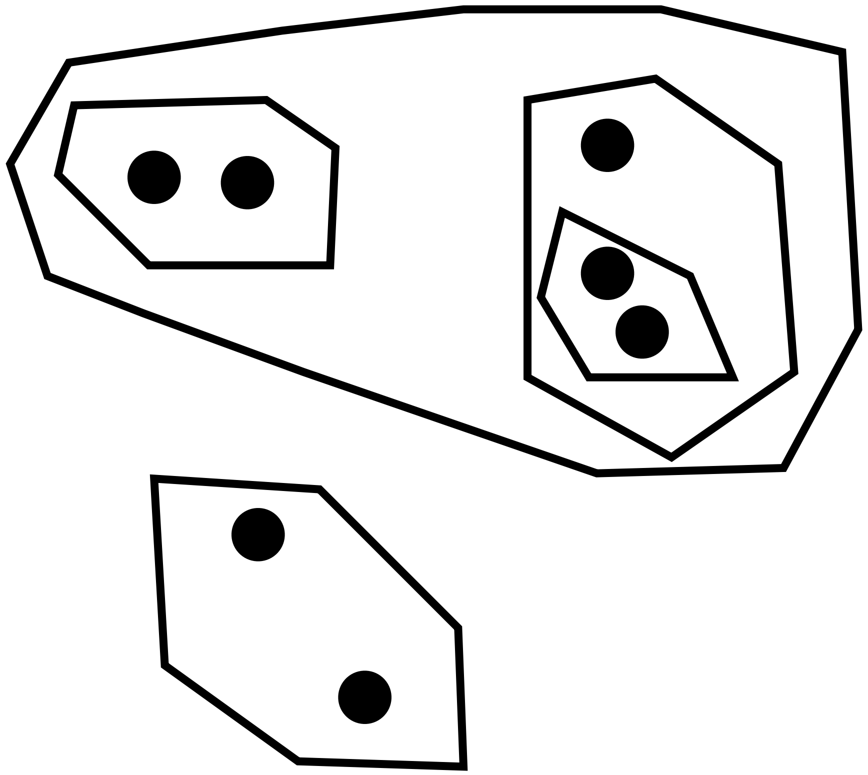
# Bottom-up Clustering
## single-link

- Step 2: compute the distance between all cluster pairs
- Step 3: merge/combine the nearest two clusters into one



# Bottom-up Clustering
## single-link

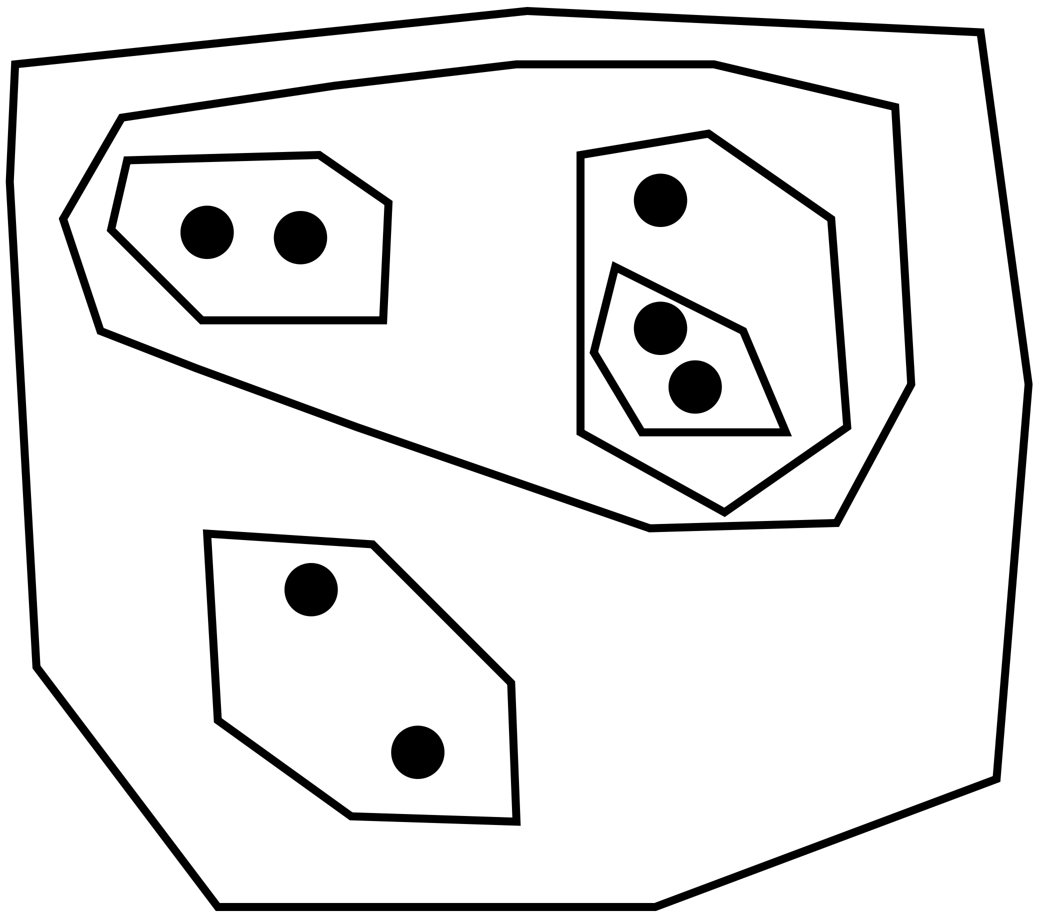
- Step 2: compute the distance between all cluster pairs
- Step 3: merge/combine the nearest two clusters into one



# Bottom-up Clustering

## single-link

- Step 2: compute the distance between all cluster pairs
- Step 3: merge/combine the nearest two clusters into one



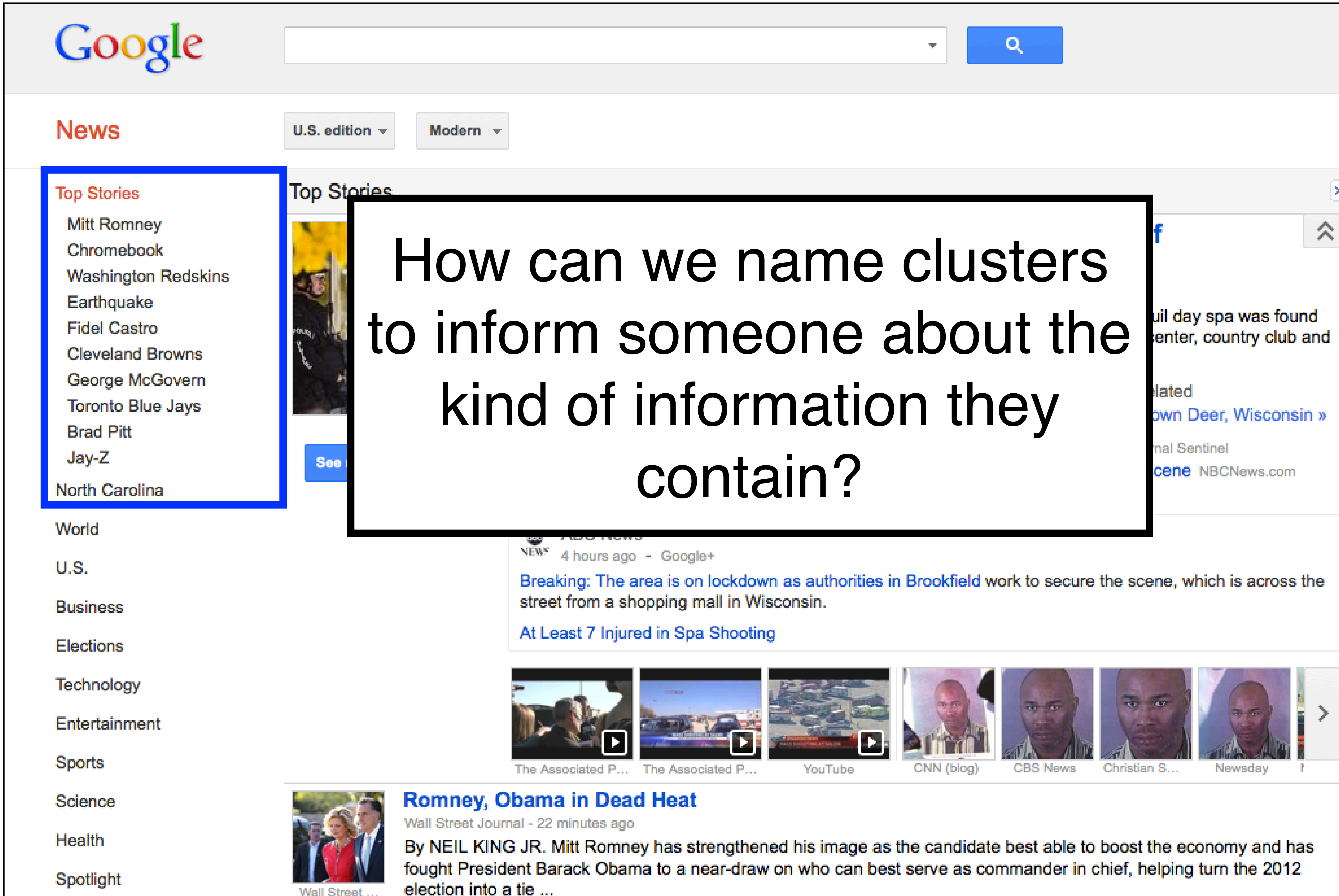
# Bottom-up Clustering

- Setting K dynamically
- Instead of setting K, we could set a distance threshold T
- Stop merging/combining clusters when the distance between the two nearest clusters > T
- Using a distance threshold can help prevent “concept drift” (especially with single-link clustering)
    - text mining --> inls 613 --> unc --> basketball

# Labeling Clusters

# Clustering Applications

## collection clustering




How can we name clusters to inform someone about the kind of information they contain?

# Labeling Clusters

## A simple solution

- Construct a vocabulary of terms and/or phrases (n-grams) that are frequent in the data
- Assign each cluster the term(s) or phrase(s) with the highest mutual information

# Mutual Information

$$\mathrm{MI}(w, c) = \log\left(\frac{P(w, c)}{P(w)P(c)}\right)$$

- **P(w,c)**: the probability that a document contains word **w** and belongs to cluster **c**
- **P(w)**: the probability that word **w** occurs in a document from any cluster
- **P(c)**: the probability that a document belongs to cluster **c**

# Mutual Information

$$\mathrm{MI}(w,c) = \log\left(\frac{P(w,c)}{P(w)P(c)}\right)$$

- If **P(w,c) = P(w) P(c)**, it means that the word **w** is independent of cluster **c**
- If **P(w,c) > P(w) P(c)**, it means that the word **w** is not independent of of cluster **c**

# Mutual Information

- Every document falls under one of these quadrants

| | belongs to cluster **c** | does not belong to cluster **c** |
|---|---|---|
| contains word **w** | a | b |
| does not contains word **w** | c | d |

total # of instances N = a + b + c + d

**P(w,c) = ???**

**P(c) = ???**

**P(w) = ???**

$$\mathrm{MI}(w,c) = \log\left(\frac{P(w,c)}{P(w)P(c)}\right)$$

# Mutual Information

- Every document falls under one of these quadrants

| | belongs to cluster **c** | does not belong to cluster **c** |
|---|---|---|
| contains word **w** | a | b |
| does not contains word **w** | c | d |

total # of instances N = a + b + c + d

**P(w,c) = a / N**

**P(c) = (a + c) / N**

**P(w) = (a + b) / N**

$$\mathrm{MI}(w,c) = \log\left(\frac{P(w,c)}{P(w)P(c)}\right)$$

# Summary

- Clustering: grouping similar documents (or instances) into subsets
- Exploratory analysis: the goal is to discover common and uncommon properties of the data
- K-means and Agglomerative Bottom-up Clustering (there are many, many others)
- Labeling clusters