Do not distribute course material

https://mpatacchiola.github.io/blog/2020/07/31/gaussian-mixture-models.html
Chapter 9 in Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow
http://cs229.stanford.edu/notes2020spring/cs229-notes7a.pdf
http://cs229.stanford.edu/notes2020spring/cs229-notes7b.pdf

pages 179-181 in http://ciml.info/

Clustering, K-Means and EM

INTRODUCTION TO MACHINE LEARNING

PROF. LINDA SELLIE

THANKS TO PROF RANGAN FOR SOME OF THE SLIDES

Outline

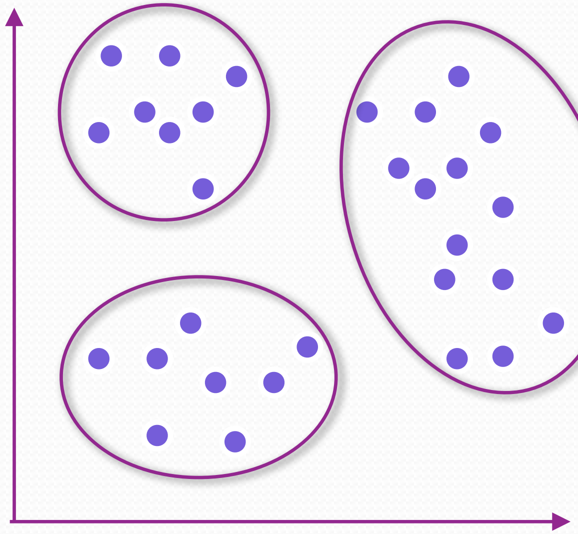
- Motivating Examples: Document clustering, image segmentation, image compression
- K-means
- K++-means (how to initialize the parameters before starting the algorithm)
- Hyperparameter K
- (On our own) K-means for document clustering

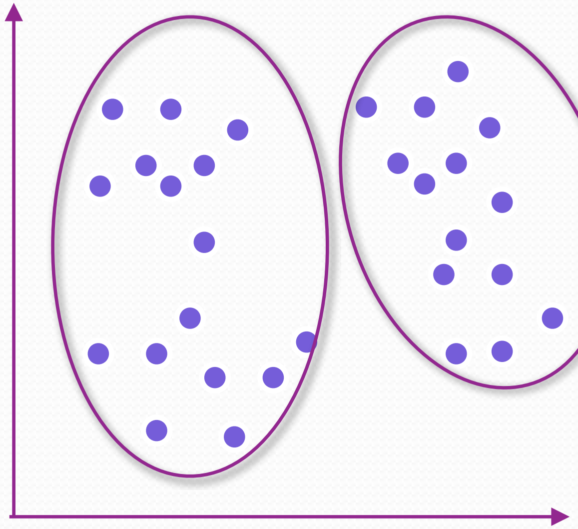
Unsupervised Machine Learning

$$\left\{(\mathbf{x}^{(1)}, \cancel{y}^{(1)}), (\mathbf{x}^{(2)}, \cancel{y}^{(2)}), \ldots, (\mathbf{x}^{(N)}, \cancel{y}^{(N)})\right\}$$

$$\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(N)}\}$$

$$\mathbf{x}^{(i)} \in \mathbb{R}^D$$





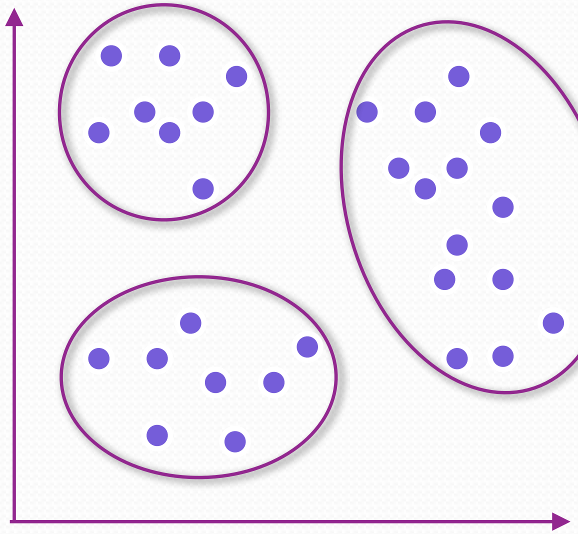
Pair share: how many clusters should we make?

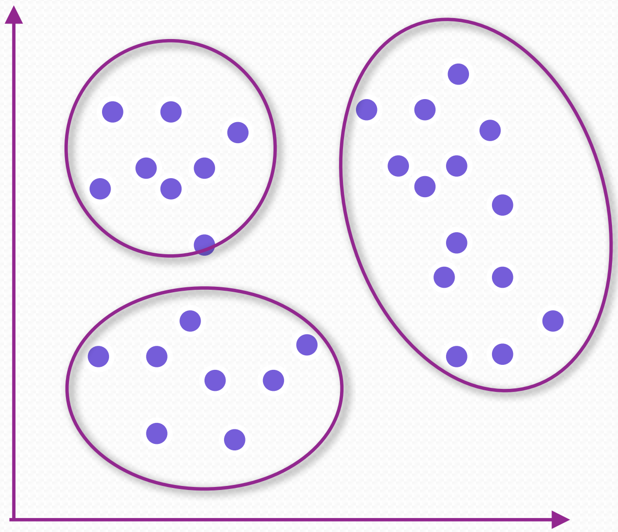
Unsupervised Machine Learning

$$\left\{(\mathbf{x}^{(1)}, \cancel{y}^{(1)}), (\mathbf{x}^{(2)}, \cancel{y}^{(2)}), \ldots, (\mathbf{x}^{(N)}, \cancel{y}^{(N)})\right\}$$

$$\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(N)}\}$$

$$\mathbf{x}^{(i)} \in \mathbb{R}^D$$





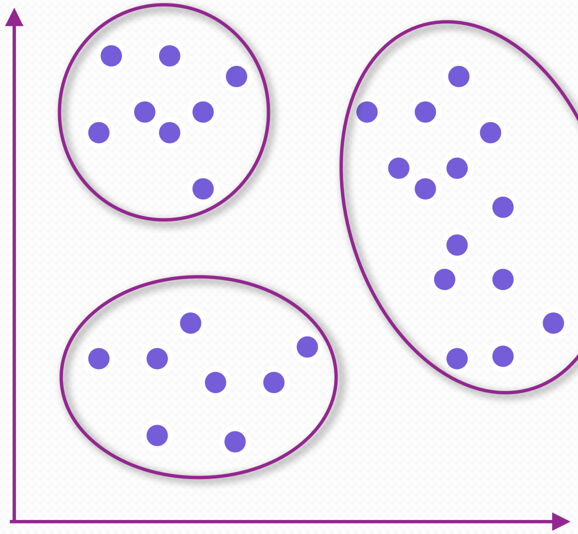
Pair share: how many clusters should we make?

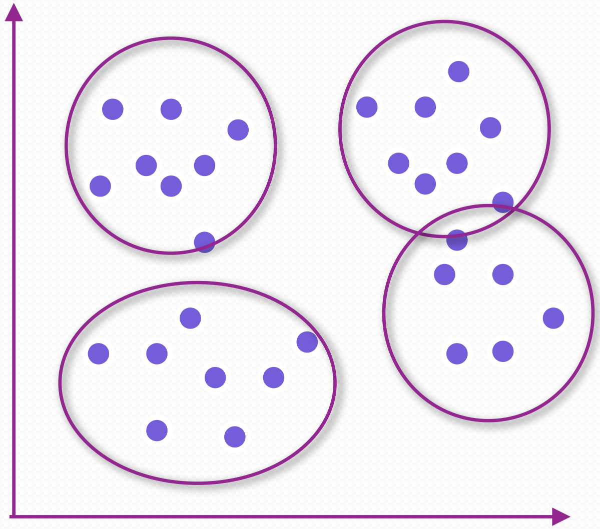
Unsupervised Machine Learning

$$\left\{(\mathbf{x}^{(1)}, \cancel{y^{(1)}}), (\mathbf{x}^{(2)}, \cancel{y^{(2)}}), \ldots, (\mathbf{x}^{(N)}, \cancel{y^{(N)}})\right\}$$

$$\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(N)}\}$$

$$\mathbf{x}^{(i)} \in \mathbb{R}^D$$





Pair share: how many clusters should we make?

The goal is to have examples in the same cluster be “close” to each other

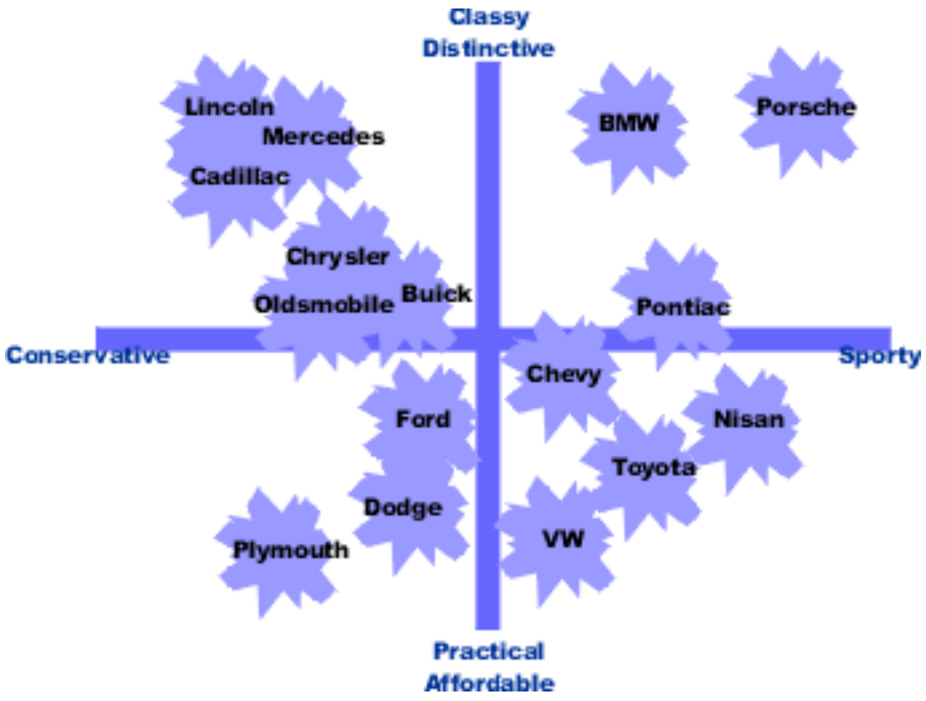
Some clustering applications

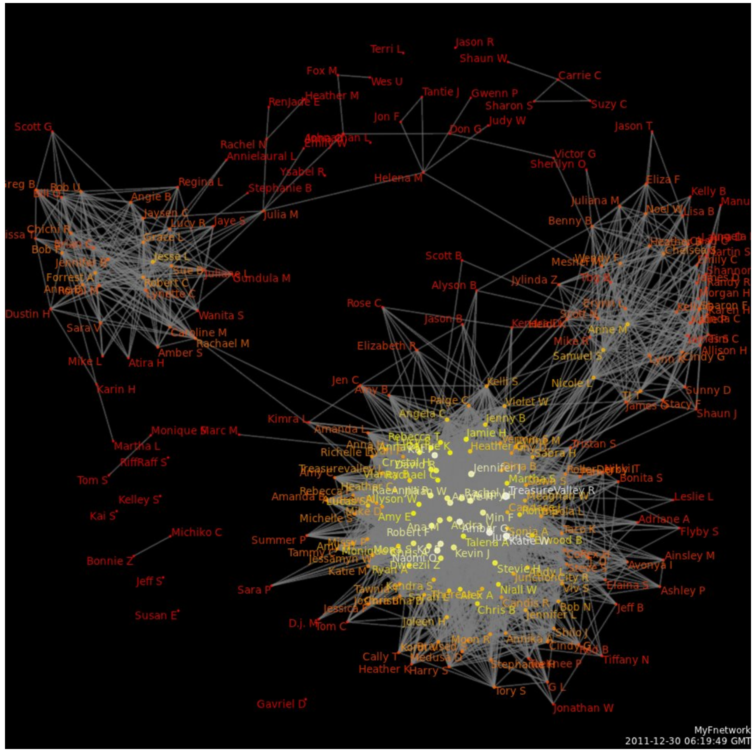
- Customer segmentation based on their purchases and activities. Allows targeted marketing for different clusters
- Dimensionality reduction: If there are k cluster each example will have k new features. Each feature is a measure of how well the example fits into a cluster
- Impute missing values
- Anomaly detection (aka outlier detection)
- Semi-supervised learning (you receive a small amount of labeled data). Label the unlabeled data in the cluster according to the labeled data
- Search engines
- Segmentation

Clustering

- Clustering is a classic unsupervised learning task.
  - .Organizing data
- There are many algorithms for clustering high-dimensional data

Clustering





https://en.wikipedia.org/wiki/Market_segmentation

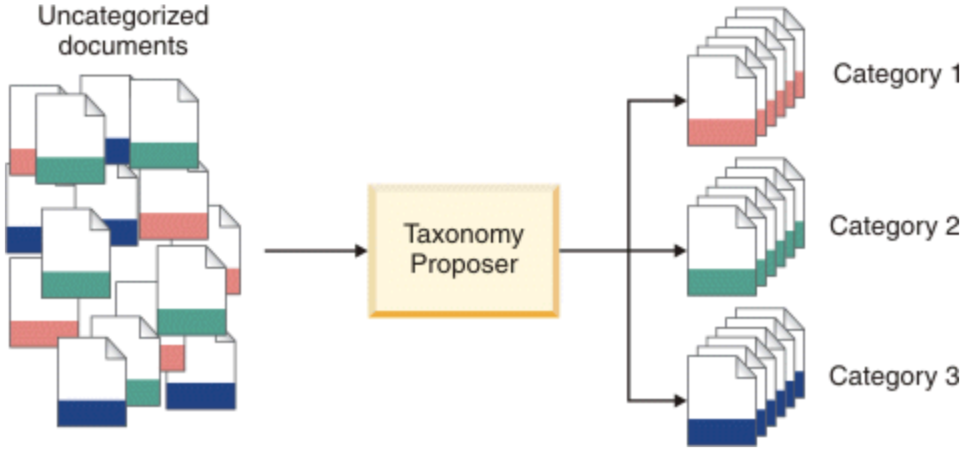
Document Clustering

IBM

IBM Knowledge Center

Content Classification > Content Classification 8.8.0 > Configuring > Cat
Using the Taxonomy Proposer to discover new categories

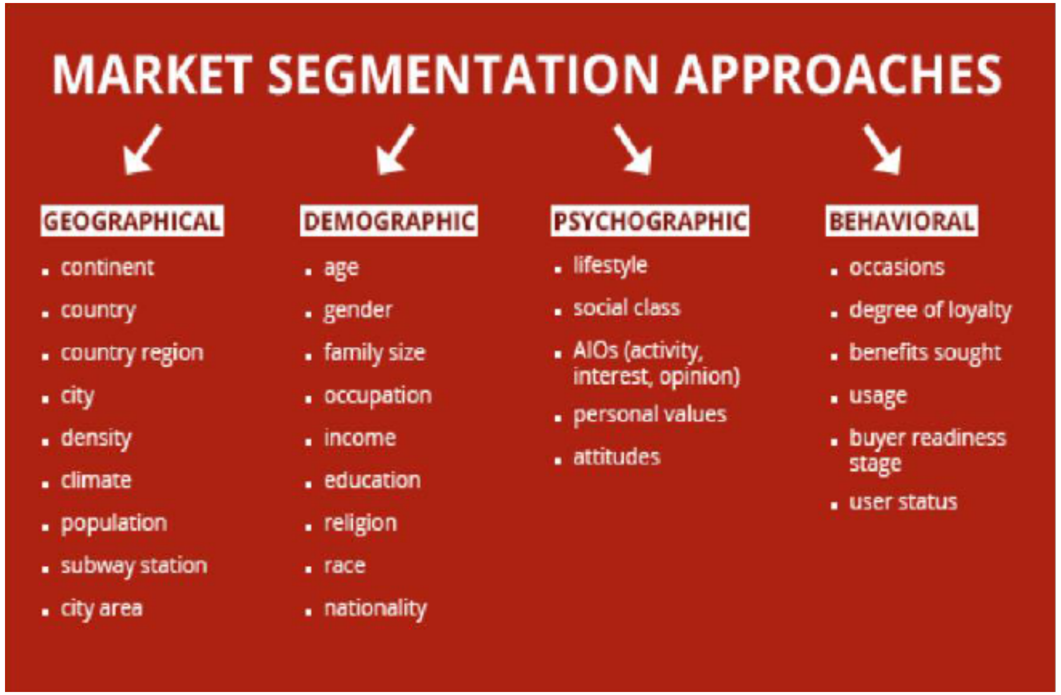
Using the Taxonomy Proposer to discover new categories



- ❑Data mining
- ❑Often have huge numbers of documents
- ❑How can we organize this?
- ❑Key idea: documents are often in clusters
- ❑Can we detect these clusters?
- ❑Can be a lucrative service
  - See IBM service to left

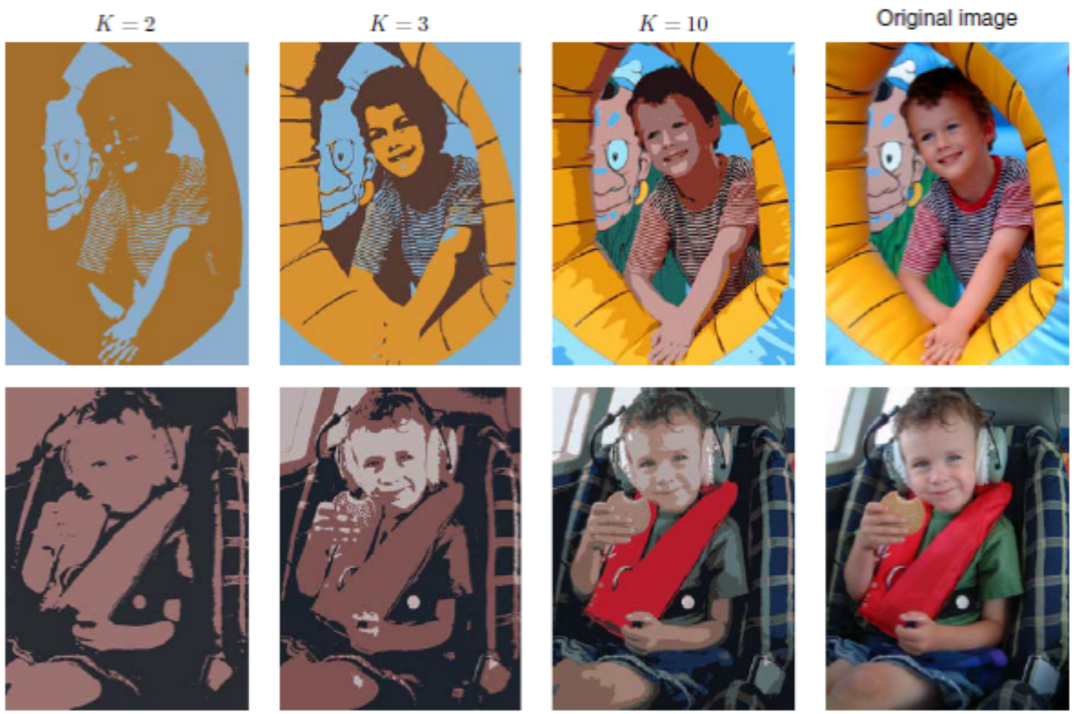
Clustering

- Clustering has many applications
  - Any time you want to segment data
  - Uncovering latent discrete variables
- Examples:
  - Segmenting sections of an image
  - Segmenting customers in market data



From: Market segmentation possibilities in the tourism market context of South Africa

Image Segmentation



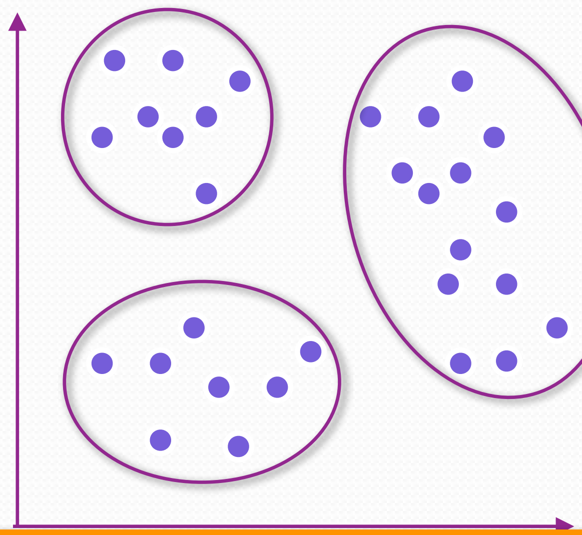
- ❑Also from Bishop.
- ❑Use K-means on the RGB values (dimension = 3)

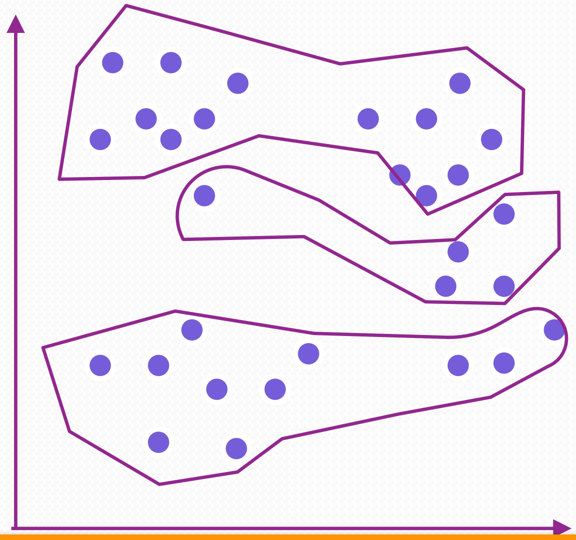
How can we find clusters in the data?

What makes a “good” cluster?

$$\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(N)}\}$$

$$c^{(1)}, c^{(2)}, \ldots, c^{(N)} \qquad 1 \le c^{(i)} \le K$$





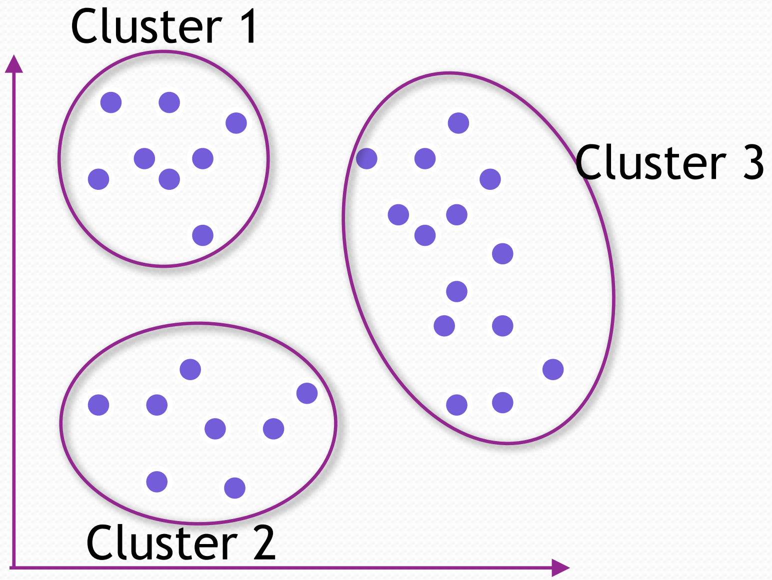
Pair share: which clustering do you like better? Why?

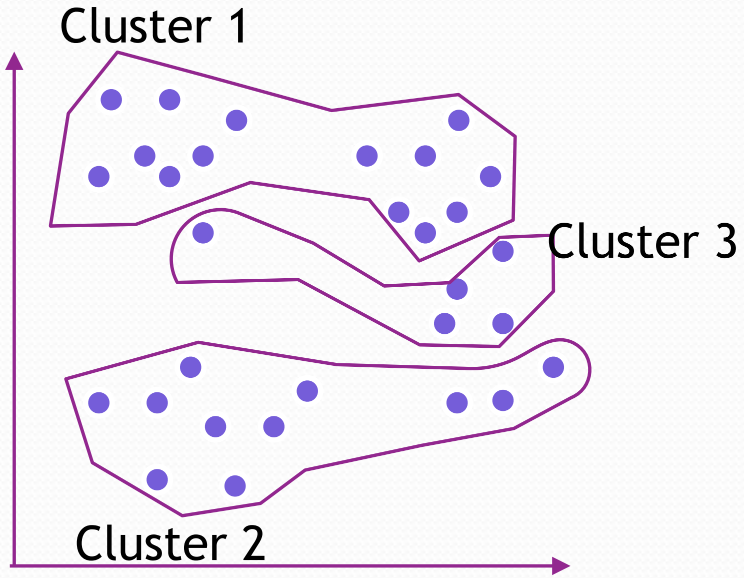
Mathematically what makes one clustering assignment better than another?

“Goodness” Metric

$$\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(N)}\}$$

$$c^{(1)}, c^{(2)}, \dots, c^{(N)} \qquad 1 \le c^{(i)} \le K$$





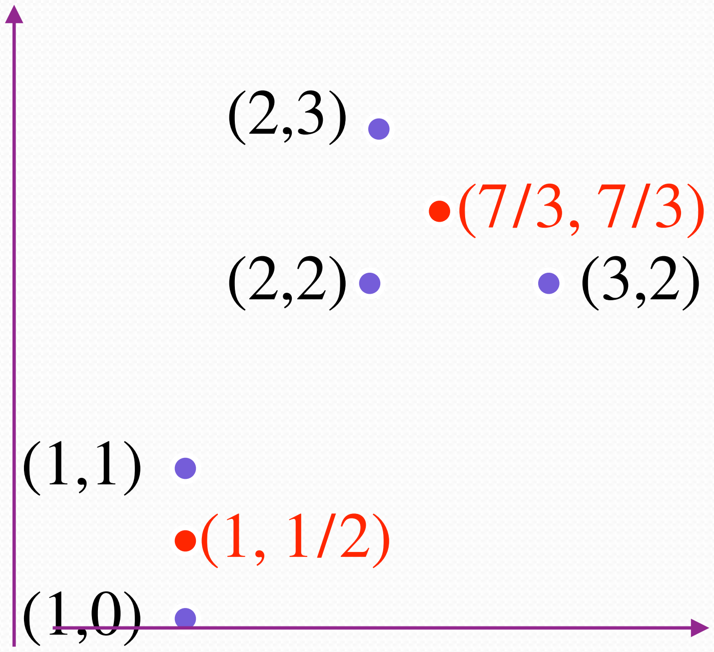
$$\sum_{\mathbf{x} \in \text{ cluster 1}} \|\mathbf{x} - \mu_1\|_2^2 + \sum_{\mathbf{x} \in \text{ cluster 2}} \|\mathbf{x} - \mu_2\|_2^2 + \sum_{\mathbf{x} \in \text{ cluster 3}} \|\mathbf{x} - \mu_3\|_2^2$$

$$= \sum_{i=1}^{3} \sum_{\mathbf{x} \in \text{cluster i}} \|\mathbf{x} - \mu_i\|_2^2 \qquad = \sum_{j=1}^{N} \|\mathbf{x}^{(j)} - \mu_{c^{(j)}}\|_2^2$$

Minimizes distortion function, J

Goal: minimize our objective function

$$J(c,\mu) = \sum_{j=1}^{N} \|\mathbf{x}^{(j)} - \mu_{c^{(j)}}\|_2^2 \quad = \sum_{i=1}^{2} \sum_{\mathrm{x} \in \text{cluster i}} \|\mathbf{x} - \mu_i\|_2^2$$



Pair share: Let K=2.
Where would you make the cluster centers: μ_1, μ_2?

Pair share: For each point, which cluster would you assign it to?

Pair share What is $J(c,\mu)$ for this cluster assignment?

$$J(c,\mu) = (1/2)^2 + (1/2)^2 + 2/9 + 2/9 + 5/9$$

Outline

- Motivating Examples: Document clustering, image segmentation, image compression
- K-means
- K++-means (how to initialize the parameters before starting the algorithm)
- (On our own) K-means for document clustering

One clustering method is K-means clustering.

It finds a predetermined (K) number of clusters in an unlabeled dataset

K-Means

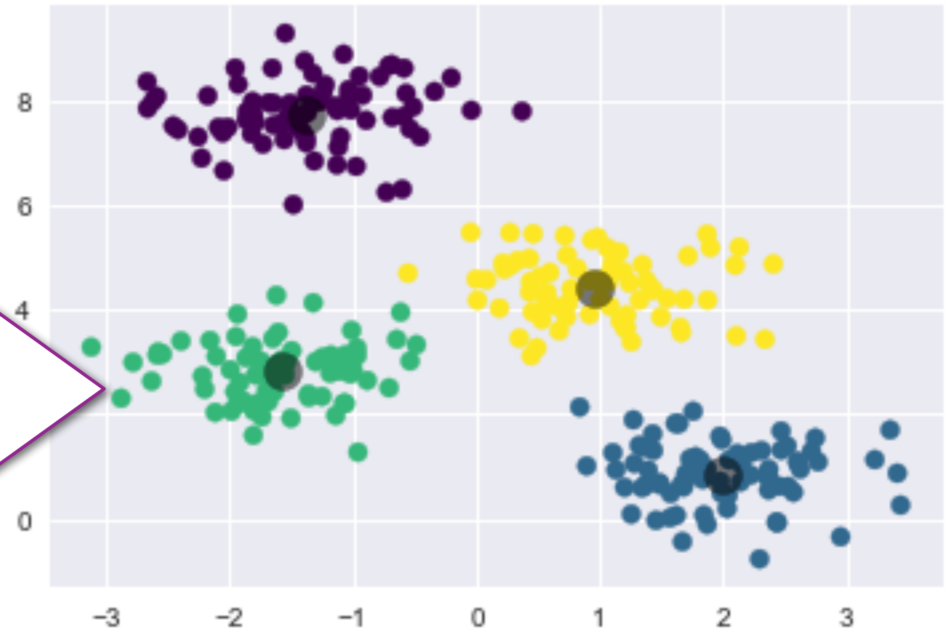
Assigns each example examples one of K clusters, where $\boldsymbol{\mu}_j$ is the center of cluster j (i.e., the *mean* of its cluster)

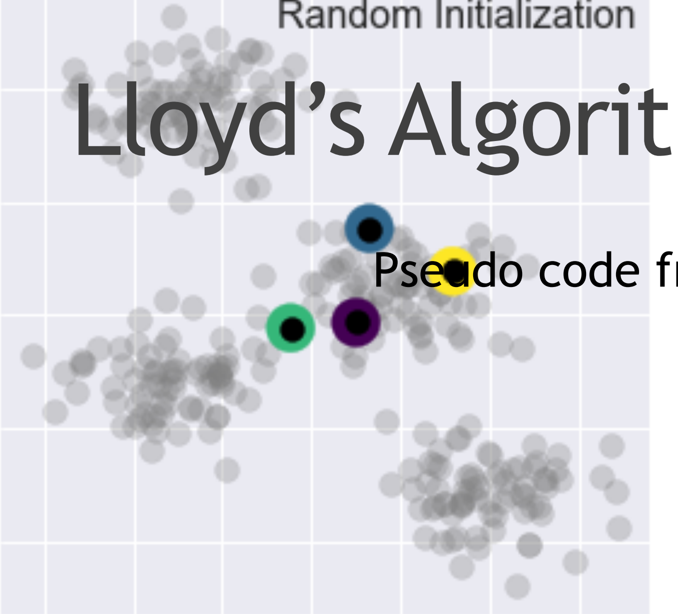
$c^{(i)}$ is the cluster $\mathbf{x}^{(i)}$ belongs to

$$J(c,\mu) = \sum_{i=1}^{N} \left\| \boldsymbol{x}^{(i)} - \boldsymbol{\mu}_{c^{(i)}} \right\|^2$$

NP hard to solve this problem!

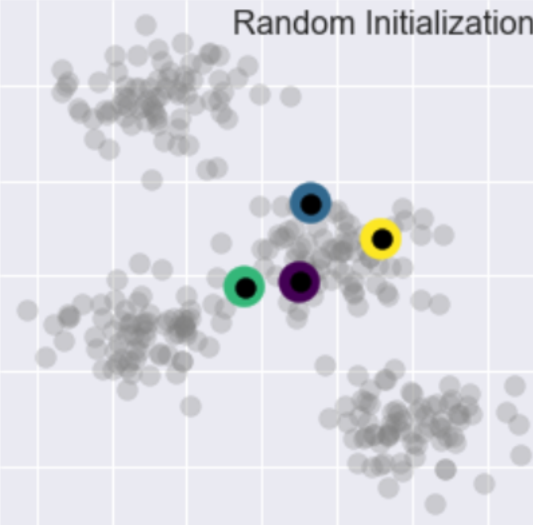
There is an exponential number of ways to assign points to clusters





Lloyd's Algorithm (Stuart Lloyd, 1957)

Pseudo code from CS229 Lecture notes



1. Initialize cluster *centroids* $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_K \in \mathbb{R}^d$ randomly
2. Repeat until convergence:

   For every i, set

$$c^{(i)} := \arg\min_{j} \left\| \boldsymbol{x}^{(i)} - \boldsymbol{\mu}_j \right\|^2$$

Update cluster membership of every example. Every example belongs to the cluster it is closest to.

   For every j $\in$ {1,..,K}, set

$$\mu_j := \frac{\sum_{i=1}^{N} 1\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{N} 1\{c^{(i)} = j\}}$$

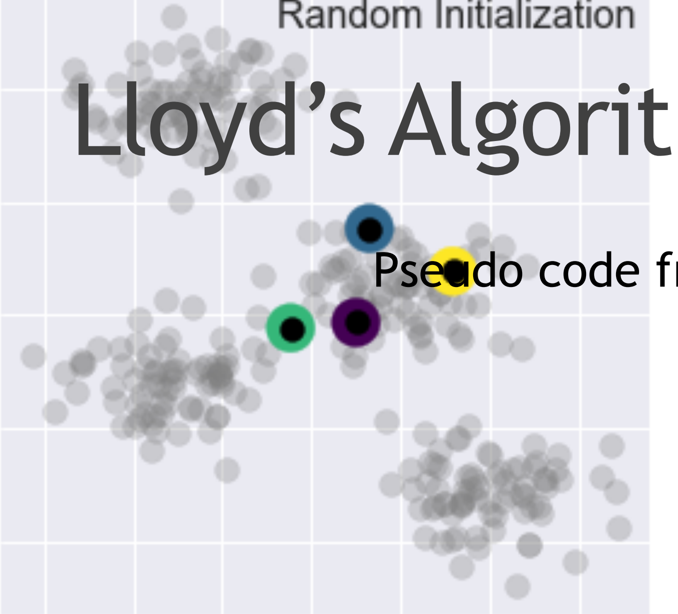
}

Suppose $\mathbf{x}^{(2)}, \mathbf{x}^{(9)}, \mathbf{x}^{(21)}$ were assigned to cluster 1 then $\mu_1 = (\mathbf{x}^{(2)} + \mathbf{x}^{(9)} + \mathbf{x}^{(21)})/3$

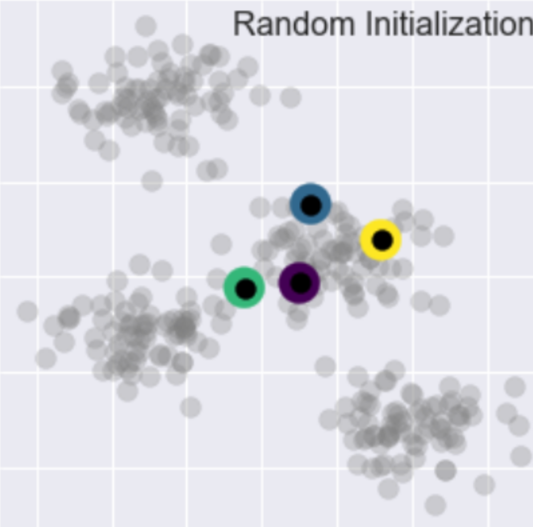
Definition:

$$1\{c^{(i)} = j\} = \begin{cases} 1 & c^{(i)} = j \\ 0 & c^{(i)} \neq j \end{cases}$$

Lloyd's Algorithm (Stuart Lloyd, 1957)



Pseudo code from CS229 Lecture notes



1. Initialize cluster *centroids* $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_K \in \mathbb{R}^d$ randomly
2. Repeat until convergence:

For every i, set

$$c^{(i)} := \arg\min_{\mathbf{j}} \left\| \boldsymbol{x}^{(i)} - \boldsymbol{\mu}_j \right\|^2$$

Update cluster membership of every example. Every example belongs to the cluster it is closest to.

For every j ∈ {1,..,K}, set

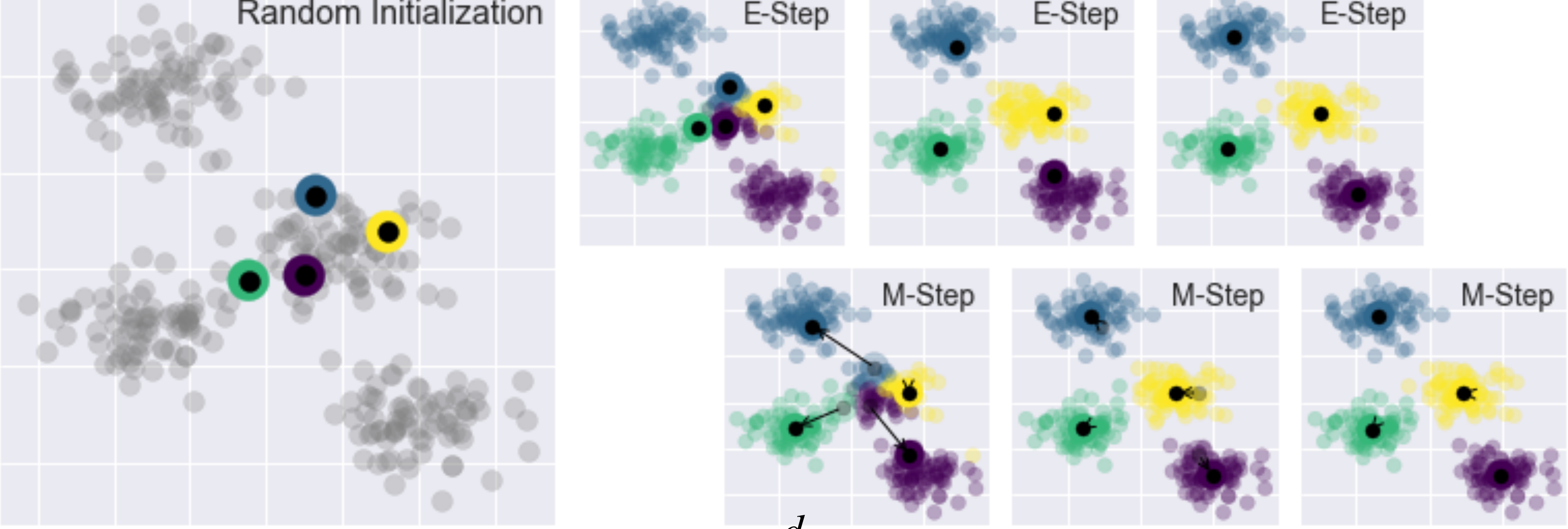
$$\mu_j := \frac{\sum_{i=1}^{N} 1\left\{c^{(i)} = j\right\} x^{(i)}}{\sum_{i=1}^{N} 1\left\{c^{(i)} = j\right\}}$$

}

Update *centroid* of each cluster to be the *average(mean)* of examples assigned to cluster j

Definition:

$$1\left\{c^{(i)} = j\right\} = \begin{cases} 1 & c^{(i)} = j \\ 0 & c^{(i)} \neq j \end{cases}$$



1. Initialize cluster *centroids* $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_K \in \mathbb{R}^d$ randomly
2. Repeat until convergence:

   For every i, set

$$c^{(i)} := \arg\min_{j} \left\| \boldsymbol{x}^{(i)} - \boldsymbol{\mu}_j \right\|^2$$

   For every j $\in$ {1,..,K}, set

$$\mu_j := \frac{\sum_{i=1}^{N} 1\left\{c^{(i)} = j\right\} x^{(i)}}{\sum_{i=1}^{N} 1\left\{c^{(i)} = j\right\}}$$

}

Definition:

$$1\left\{c^{(i)} = j\right\} = \begin{cases} 1 & c^{(i)} = j \\ 0 & c^{(i)} \neq j \end{cases}$$

Slide not covered in class. Material on this slide is not on the quiz

Centroid is Minimizer $\mu_j = \frac{1}{|S_j|}\sum_{\mathbf{x}^{(i)}\in S_j}\mathbf{x}^{(i)}$

We show that our choice of μ_j is better than any other point $\mathbf{p}$.

To show this we need to prove that:

$$\sum_{\mathbf{x}^{(i)}\in S_j}\left\| \mathbf{x}^{(i)} - \frac{1}{|S_j|}\sum_{\mathbf{x}^{(i)}\in S_j}\mathbf{x}^{(i)} \right\|^2 \le \sum_{\mathbf{x}^{(i)}\in S_j}\left\| \mathbf{x}^{(i)} - \mathbf{p} \right\|^2$$

Adding 0=$-\mu_j + \mu_j$

Proof:

$$\sum_{\mathbf{x}^{(i)}\in S_j}\left\| \mathbf{x}^{(i)} - \mathbf{p} \right\|^2 = \sum_{\mathbf{x}^{(i)}\in S_j}\left\| \underbrace{\mathbf{x}^{(i)} - \mu_j}_{\mathbf{a}} + \underbrace{\mu_j - \mathbf{p}}_{\mathbf{b}} \right\|^2$$

Here $\mathbf{a}, \mathbf{b}$ are vectors. Notice that: $\|\mathbf{a}+\mathbf{b}\|^2 = \|\mathbf{a}\|^2 + \|\mathbf{b}\|^2 + 2\mathbf{a}^T\mathbf{b}$

$$= \sum_{\mathbf{x}^{(i)}\in S_j}\left\| \mathbf{x}^{(i)} - \mu_j \right\|^2 + \sum_{\mathbf{x}^{(i)}\in S_j}\left\| \mu_j - \mathbf{p} \right\|^2 + 2\sum_{\mathbf{x}^{(i)}\in S_j}(\mathbf{x}^{(i)} - \mu_j)^T(\mu_j - \mathbf{p})$$

We can move $(\mu_j - \mathbf{p})$ in front of the sum: $2(\mu_j - \mathbf{p})^T\sum_{\mathbf{x}^{(i)}\in S_j}(\mathbf{x}^{(i)} - \mu_j)$

We can rewrite this as:

$$= 2(\mu_j - \mathbf{p})^T\left(\left(\sum_{\mathbf{x}^{(i)}\in S_j}\mathbf{x}^{(i)}\right) - |S_j|\mu_j\right)$$

Now notice that: $|S_j|\mu_j = \sum_{\mathbf{x}^{(i)}\in S_j}\mathbf{x}^{(i)}$

Thus $\left(\sum_{\mathbf{x}^{(i)}\in S_j}\mathbf{x}^{(i)}\right) - |S_j|\mu_j = 0$

$$= \sum_{\mathbf{x}^{(i)}\in S_j}\left\| \mathbf{x}^{(i)} - \mu_j \right\|^2 + \sum_{\mathbf{x}^{(i)}\in S_j}\left\| \mu_j - \mathbf{p} \right\|^2$$

$$\ge \sum_{\mathbf{x}^{(i)}\in S_j}\left\| \mathbf{x}^{(i)} - \mu_j \right\|^2$$

Slide not covered in class. Material on this slide is not on the quiz

Centroid is Minimizer $\mu_j = \frac{1}{|S_j|} \sum_{\mathbf{x}^{(i)} \in S_j} \mathbf{x}^{(i)}$

We show that our choice of μ_j is better than any other point $\mathbf{p}$.

To show this we need to prove that:

$$\sum_{\mathbf{x}^{(i)} \in S_j} \left\| \mathbf{x}^{(i)} - \boxed{\mu_j} \right\|^2 \leq \sum_{\mathbf{x}^{(i)} \in S_j} \left\| \mathbf{x}^{(i)} - \mathbf{p} \right\|^2$$

Proof:

Adding 0=$-\mu_j + \mu_j$

$$\sum_{\mathbf{x}^{(i)} \in S_j} \left\| \mathbf{x}^{(i)} - \mathbf{p} \right\|^2 = \sum_{\mathbf{x}^{(i)} \in S_j} \left\| \underbrace{\mathbf{x}^{(i)} - \mu_j}_{\mathbf{a}} + \underbrace{\mu_j - \mathbf{p}}_{\mathbf{b}} \right\|^2$$

Here $\mathbf{a}, \mathbf{b}$ are vectors. Notice that: $\|\mathbf{a} + \mathbf{b}\|^2 = \|\mathbf{a}\|^2 + \|\mathbf{b}\|^2 + 2\mathbf{a}^T\mathbf{b}$

$$= \sum_{\mathbf{x}^{(i)} \in S_j} \left\| \mathbf{x}^{(i)} - \mu_j \right\|^2 + \sum_{\mathbf{x}^{(i)} \in S_j} \left\| \mu_j - \mathbf{p} \right\|^2 + 2 \sum_{\mathbf{x}^{(i)} \in S_j} (\mathbf{x}^{(i)} - \mu_j)^T (\mu_j - \mathbf{p})$$

$$= \sum_{\mathbf{x}^{(i)} \in S_j} \left\| \mathbf{x}^{(i)} - \mu_j \right\|^2 + \sum_{\mathbf{x}^{(i)} \in S_j} \left\| \mu_j - \mathbf{p} \right\|^2$$

$$\geq \sum_{\mathbf{x}^{(i)} \in S_j} \left\| \mathbf{x}^{(i)} - \mu_j \right\|^2$$

We can move $(\mu_j - \mathbf{p})$ in front of the sum: $2(\mu_j - \mathbf{p})^T \sum_{\mathbf{x}^{(i)} \in S_j} (\mathbf{x}^{(i)} - \mu_j)$

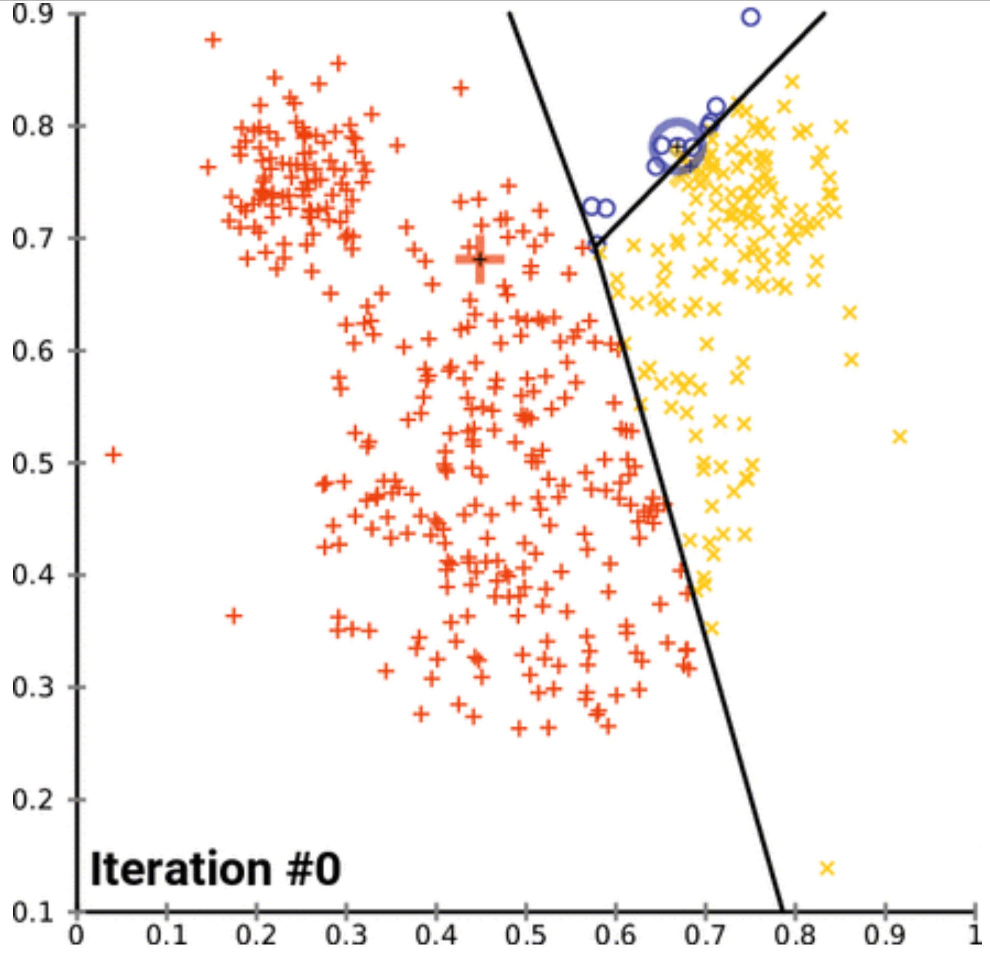
We can rewrite this as:

$$= 2(\mu_j - \mathbf{p})^T \left(\left(\sum_{\mathbf{x}^{(i)} \in S_j} \mathbf{x}^{(i)} \right) - |S_j| \mu_j \right)$$

Now notice that: $|S_j|\mu_j = \sum_{\mathbf{x}^{(i)} \in S_j} \mathbf{x}^{(i)}$

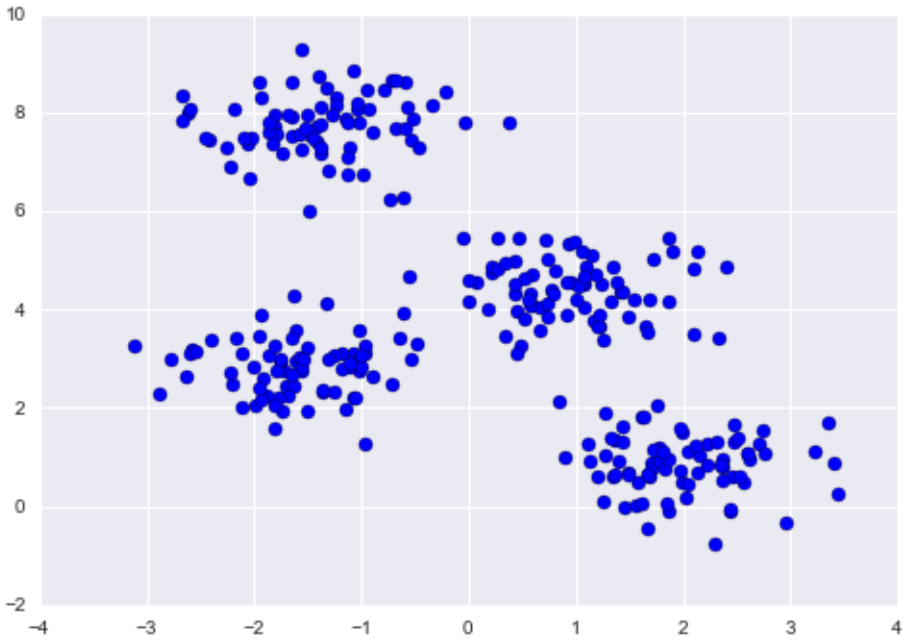
Thus $\left(\sum_{\mathbf{x}^{(i)} \in S_j} \mathbf{x}^{(i)} \right) - |S_j|\mu_j = 0$

K-Means illustrated



Uh Oh…

- The K-means clustering algorithm is guaranteed to improve the result on each step…and converge - but not to a globally optimal solution.
- However, K-means is not guaranteed to find a global minimum - only a local minimum.
- Finding the global minimum K-means error is NP-hard…
- Run the algorithm with many initial configurations and keep the one that performs best



E-M Algorithm

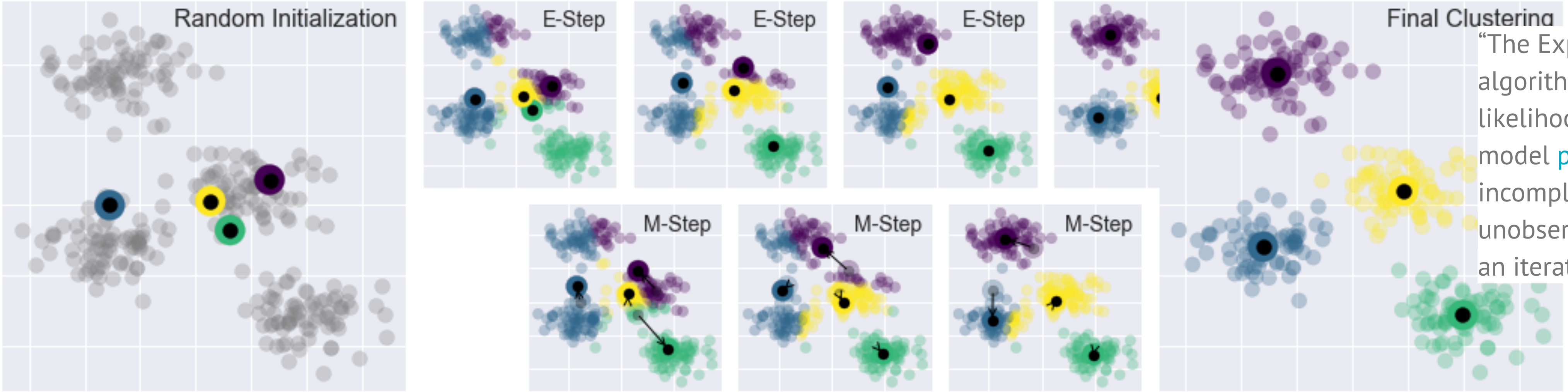
- The K-means algorithm is a variant of the E-M algorithm.
  - The E step (Expectation step) involves updating our expectation of what cluster each example belongs to.
  - The M step (Maximization step) involves maximizing the best location of the cluster centers.
- The algorithm works by minimizing a complex error function by separating the data into two steps: If one step is known, it is easy to optimize the other step

Given an assignment of points to clusters, could we find a better cluster center than taking the average of the points in a cluster to be its center?

A) Yes
B) No
C) Maybe

K-Means Converges

- The algorithm converges to a partition that is "locally optimal."
  - Given the cluster centers μ_j, we cannot find a better assignment of the examples to clusters.
  - Given the cluster assignments ($c^{(i)}$ for all $i \in 1...N$), we cannot find better centers.



"The Expectation-Maximization (EM) algorithm is a way to find maximum-likelihood estimates for model parameters when your data is incomplete, has missing data points, or has unobserved (hidden) latent variables. It is an iterative way"

Proof of convergence (to a local min)

- **Theorem** (K - Means Convergence Theorem)

We update **μ** and we update c. For each update we show that they never increase the value of

$$J(c,\mu)=\sum_{i=1}^{N}\left\|\boldsymbol{x}^{(i)}-\boldsymbol{\mu}_{c^{(i)}}\right\|^2$$

There are only a finite number of values that can be assigned to **μ** and c. (**μ** is is the mean of a subset of the examples and c ϵ {1, 2, … , K}).

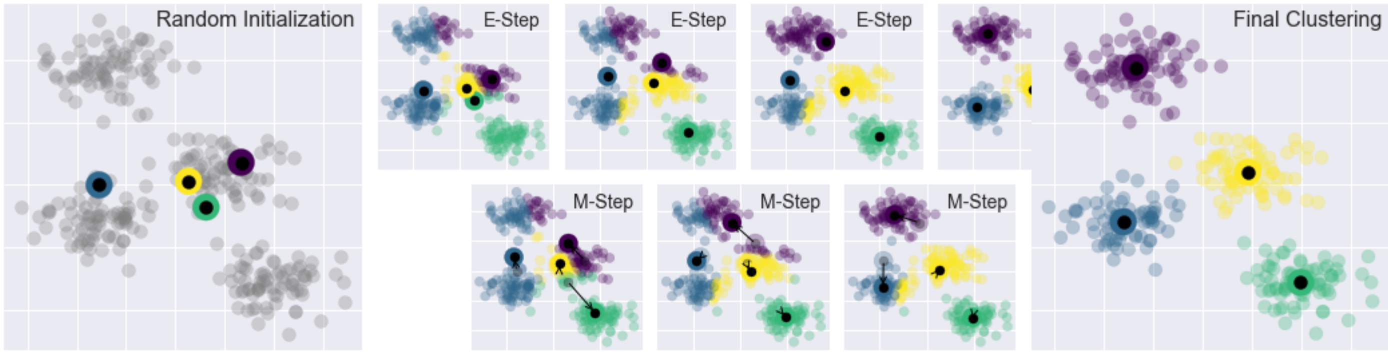
We also know that J(c, **μ**) ≥ 0.

Thus J(c, **μ**) can only decrease a finite number of times. When it stops decreasing the algorithm has converged (to a local minimum)

When we update $c^{(i)}$, it must be that $\left\|\boldsymbol{x}^{(i)}-\boldsymbol{\mu}_{c^{(i\text{ new})}}\right\|^2 \le \left\|\boldsymbol{x}^{(i)}-\boldsymbol{\mu}_{c^{(i)}}\right\|^2$

When we update $\boldsymbol{\mu}_j$ as the mean of the points which are in this cluster - it directly minimizes $\sum_{c^{(i)}=j}(\boldsymbol{x}^{(i)}-\boldsymbol{\mu}_j)^2$

Thus every iteration decreases the cost function



Since it is possible to converge to a local minimum instead of a global minimum, you should run the algorithm 10 times and choose the clustering with the lowest $J(c, \boldsymbol{\mu})$

Outline

- Motivating Examples: Document clustering, image segmentation, image compression
- K-means
- K++-means (how to initialize the parameters before starting the algorithm)
- K-means for document clustering

The big concern is poor initialization at the start of the algorithm.

How to choose the initial values…

- One heuristic (we will refine it on the next slide) is to use the *furthest-first* algorithm

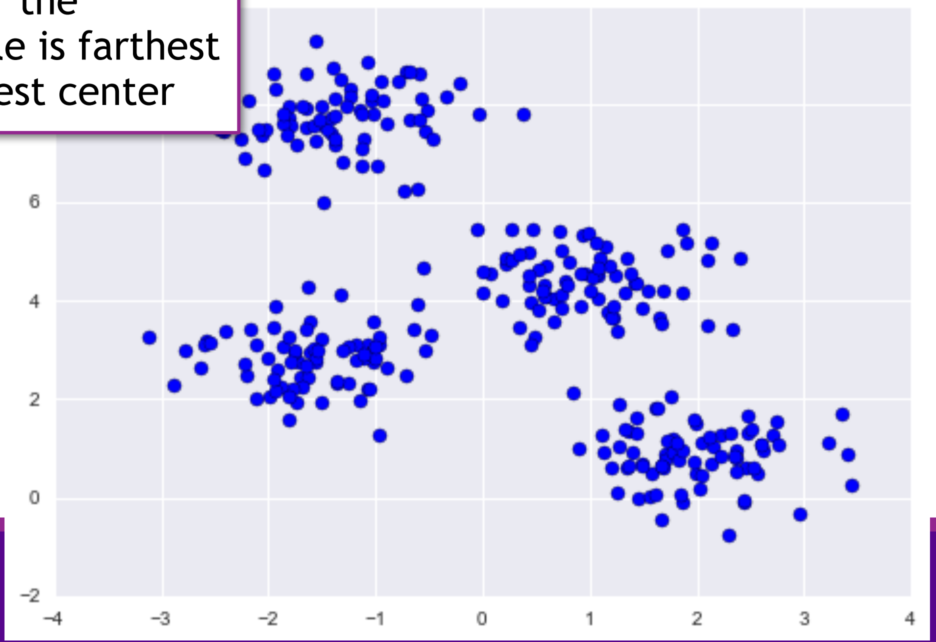
1. Pick a random example j and set $\boldsymbol{\mu}_1 = \mathbf{x}^{(j)}$
2. For k'' = 2..K:

   Find the example j that is as far as possible from all previously selected means; namely:

$$j = \arg\max_j \min_{k'<k''} \left\| \boldsymbol{x}^{(j)} - \boldsymbol{\mu}_{k'} \right\|^2$$

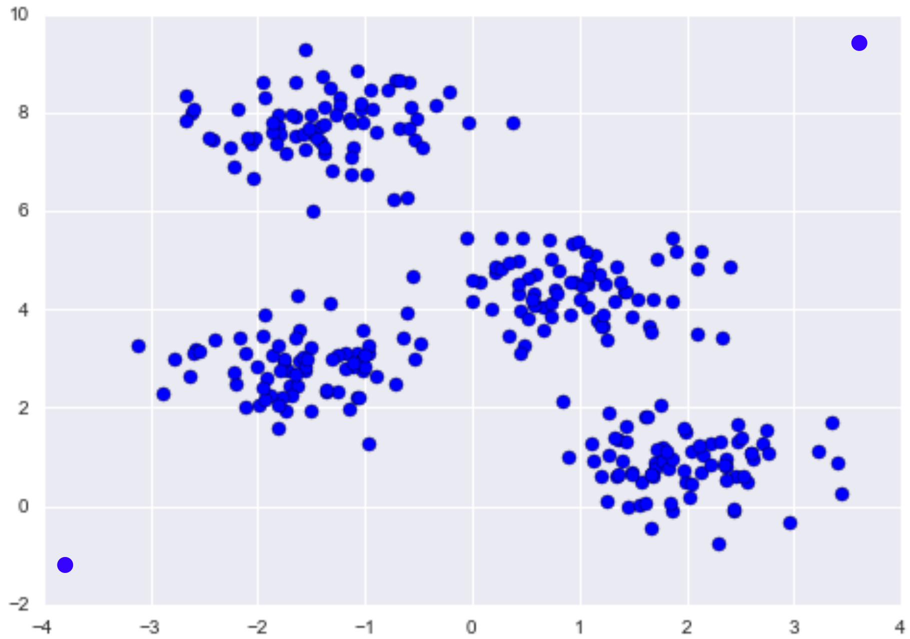
Find index of the training example is farthest from its closest center

   and set $\boldsymbol{\mu}_{k''} = \mathbf{x}^{(j)}$

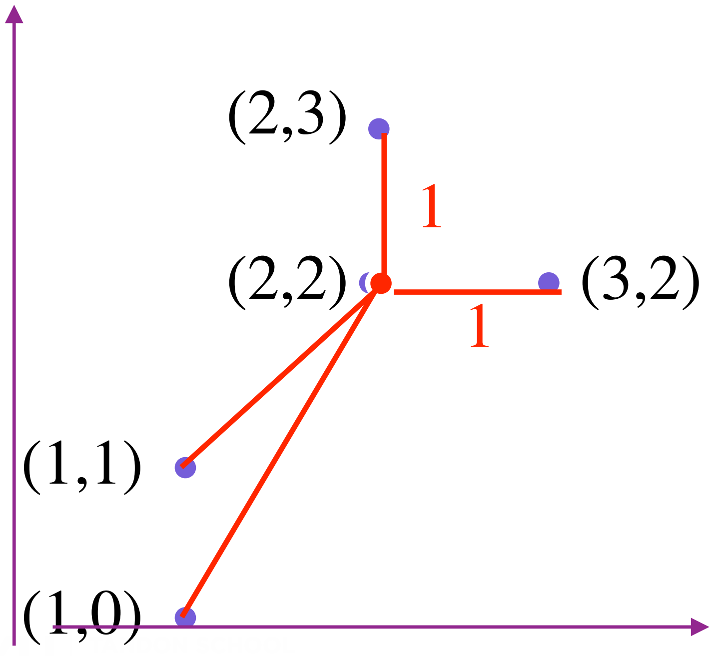


The problem is that this algorithm is sensitive to outliers.

Outliers...



Instead of choosing the furthest example from your existing clusters, select the next center randomly with probability proportional to its distance squared.



Pair share: What are the distances?
Compute one of the probabilities.

Material modified from pages 180-182 in http://ciml.info/

K-means++ algorithm

- Algorithm k-means++

  $\mu_1 = \mathbf{x}^{(j)}$ for j chosen uniformly at random *// randomly initialize first point*

  for *k"*=2 **to** *K* **do**

  $$d_j = \min_{k'<k''} \left\| \mathbf{x}^{(j)} - \boldsymbol{\mu}_{k'} \right\|, \forall j$$ *// compute distances*

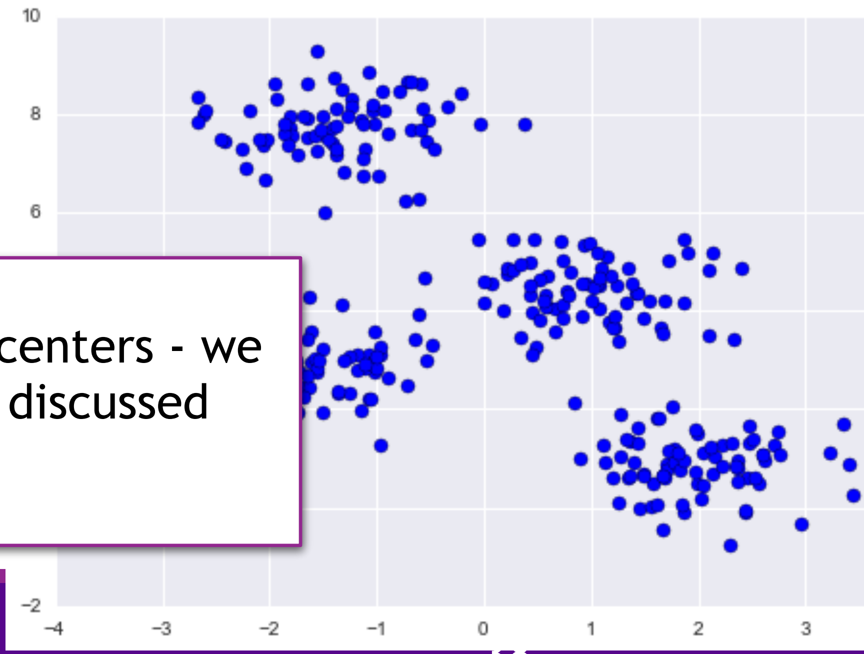
  $$p_j = \frac{d_j^2}{\sum_{i=1}^{m} d_i^2}, \forall j$$ *// normalize to probability distribution*

  j = random chosen with probability p_j

  $\mu_{k''} = \mathbf{x}^{(j)}$

  Next run k-means using μ as initial centers

After we find the initial centers - we run the K-means algorithm discussed earlier.

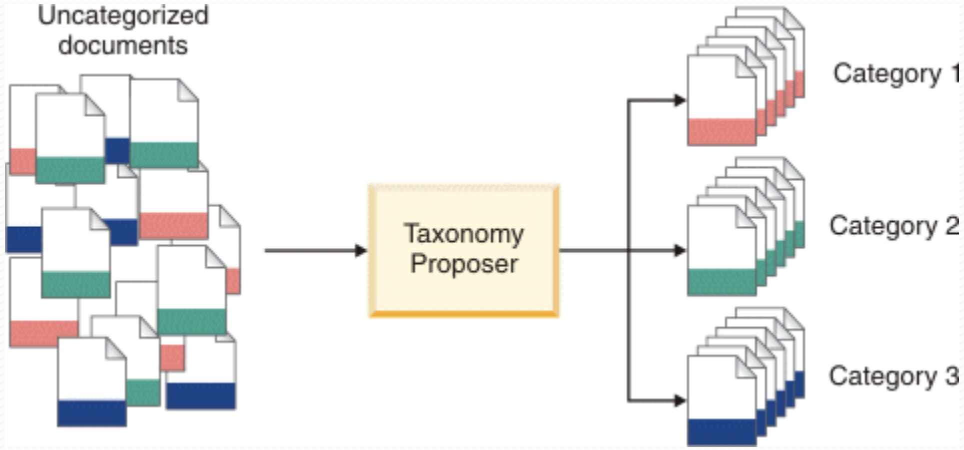


It can be proven that the expected value of the $J(c, \boldsymbol{\mu})$ when running K-means++ is never more than $O(\log K)$ times optimal $J(c, \boldsymbol{\mu})$

Outline

- Motivating Examples: Document clustering, image segmentation, image compression
- K-means
- K++-means (how to initialize the parameters before starting the algorithm)
- (On your own) K-means for document clustering

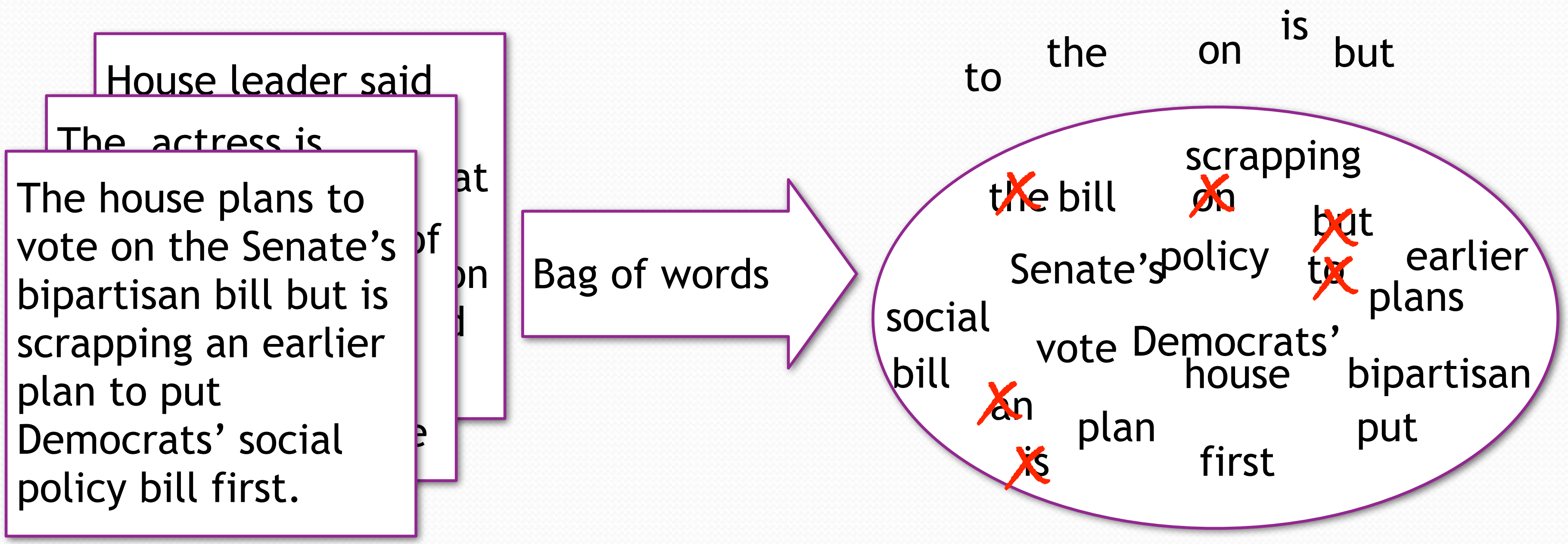
Feature Extraction:



If we want to use K-means into cluster documents,
we first need to convert text into a set of numerical values.

How can we do this?

Documents as feature vectors



$$\begin{array}{cl} \begin{bmatrix} 0 \\ 1 \\ 2 \\ 0 \\ 0 \\ 1 \\ \vdots \end{bmatrix} & \begin{array}{l} \text{politics} \\ \text{house} \\ \text{bill} \\ \text{president} \\ \text{leader} \\ \text{social} \\ \vdots \end{array} \end{array}$$

(NYTimes Nov. 5, 2021)

$\mathbf{x}$

$\Phi(\mathbf{x})$

Approach from mit.edu/6.034

Transform the feature vectors to emphasize more “relevant” words

Approach from mit.edu/6.034

Turning text into a feature vector

Document 1

The quick brown fox jumped over the lazy dog's back.

Document 2

Now is the time for all good men to come to the aid of their party.

- Document is natively text
- Must represent as a numeric vector
- Represent by word counts
  - Enumerate all words
  - Each document is count of frequencies
- Stopwords

Discussion Questions

- Is the absolute number of times a word appears the correct metric?
- What about the length of the document?
- What about the frequency of the word?
- What words "matter"?

the, for, a, in

convolutional, gradient

- Perhaps:
  - if a word appears frequently, it is important (give it a high score)
  - If a word appears in many documents, it is not important (give it a low score)

Ideas:

$$TF_{\text{"this"},d_1} = \frac{1}{5} = 0.2 \qquad TF_{\text{"this"},d_2} = \frac{1}{7} \approx 0.14 \qquad IDF_{\text{"this"}} = \log\left(\frac{2}{2}\right) = 0$$

$$TF_{\text{"example"},d_1} = \frac{0}{5} = 0 \qquad TF_{\text{"example"},d_2} = \frac{3}{7} \approx 0.429 \qquad IDF_{\text{"example"}} = \log\left(\frac{2}{1}\right) = 1$$

Example modified from https://en.wikipedia.org/wiki/Tf%E2%80%93idf

- How can we categorize how important a word is in a document?
- Perhaps:
  - if a word appears frequently, it is important (give it a high score) convolutional, gradie
  - except if the word appears in many documents, it is not important (give it a low score)
- Steps: the, for, a, in
  - Count the frequency of every word in the document

Term frequency

$$TF_{i,n} = \frac{\text{num times word } i \text{ in doc } n}{\text{total num words in doc } n}$$

  - Determine how much information a word provides: Inverse Document Frequency (IDF)

The more common a word is the lower its IDF score

Inverse doc frequency

$$IDF_i = \log\left[\frac{\text{Total num docs in corpus}}{\text{Num docs with word } i}\right]$$

Document 1

Term	Term Count
this	1
Is	1
a	2
sample	1

Document 2

Term	Term Count
this	1
Is	1
another	2
example	3

Ideas:

$$TF_{\text{"this"},d_1} = \frac{1}{5} = 0.2$$

$$TF_{\text{"this"},d_2} = \frac{1}{7} \approx 0.14$$

$$IDF_{\text{"this"}} = \log\left(\frac{2}{2}\right) = 0$$

$$TF_{\text{"example"},d_1} = \frac{0}{5} = 0$$

$$TF_{\text{"example"},d_2} = \frac{3}{7} \approx 0.429$$

$$IDF_{\text{"example"}} = \log\left(\frac{2}{1}\right) = 1$$

Example modifi... dia.org/wiki/Tf%E2%80%93idf

Frequency is relative to the size of the document

- How can we categorize how ... word is in a document?
- Perhaps:
  - if a word appears frequently, it is important (give it a high score)
  - except if the word appears in many documents, it is not important (give it a low score)
- Steps:
  - Count the frequency of every word in the document
  - Determine how much information a word provides: Inverse Document Frequency (IDF)

convolutional, gradie

the, for, a, in

Term frequency

$$TF_{i,n} = \frac{\text{num times word } i \text{ in doc } n}{\text{total num words in doc } n}$$

Inverse doc frequency

$$IDF_i = \log\left[\frac{\text{Total num docs in corpus}}{\text{Num docs with word } i}\right]$$

The more common a word is the lower its IDF score

Document 1

Term	Term Count
this	1
Is	1
a	2
sample	1

Document 2

Term	Term Count
this	1
Is	1
another	2
example	3

Term Frequency – Inverse Document Frequency

- Use TF-IDF weight for vectors:

$$X[n, i] = TF_{i,n} \times IDF_i$$

Document weight vector

Term frequency

$$TF_{i,n} = \frac{\text{num times word } i \text{ in doc } n}{\text{total num words in doc } n}$$

Inverse doc frequency

$$IDF_i = \log\left[\frac{\text{Total num docs in corpus}}{\text{Num docs with word } i}\right]$$

$$\text{TF}_{\text{"this"},d_1} = \frac{1}{5} = 0.2$$

$$\text{TF}_{\text{"this"},d_2} = \frac{1}{7} \approx 0.14$$

$$\text{IDF}_{\text{"this"}} = \log\left(\frac{2}{2}\right) = 0$$

$$\text{TF}_{\text{"example"},d_1} = \frac{0}{5} = 0$$

$$\text{TF}_{\text{"example"},d_2} = \frac{3}{7} \approx 0.429$$

$$\text{IDF}_{\text{"example"}} = \log\left(\frac{2}{1}\right) = 1$$

Example modified from https://en.wikipedia.org/wiki/Tf%E2%80%93idf

$$\text{TF-IDF}_{\text{"this"},d_1} = 0.2 \times 0 = 0$$

$$\text{TF-IDF}_{\text{"this"},d_2} = 0.14 \times 0 = 0$$

$$\text{TF-IDF}_{\text{"example"},d_1,D} = 0. \times 1 = 0$$

$$\text{TF-IDF}_{\text{"example"},d_2} = 0.429 \times 1 = 0.429$$