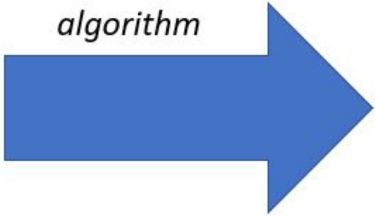
School of AI

Unsupervised Learning

February 2020

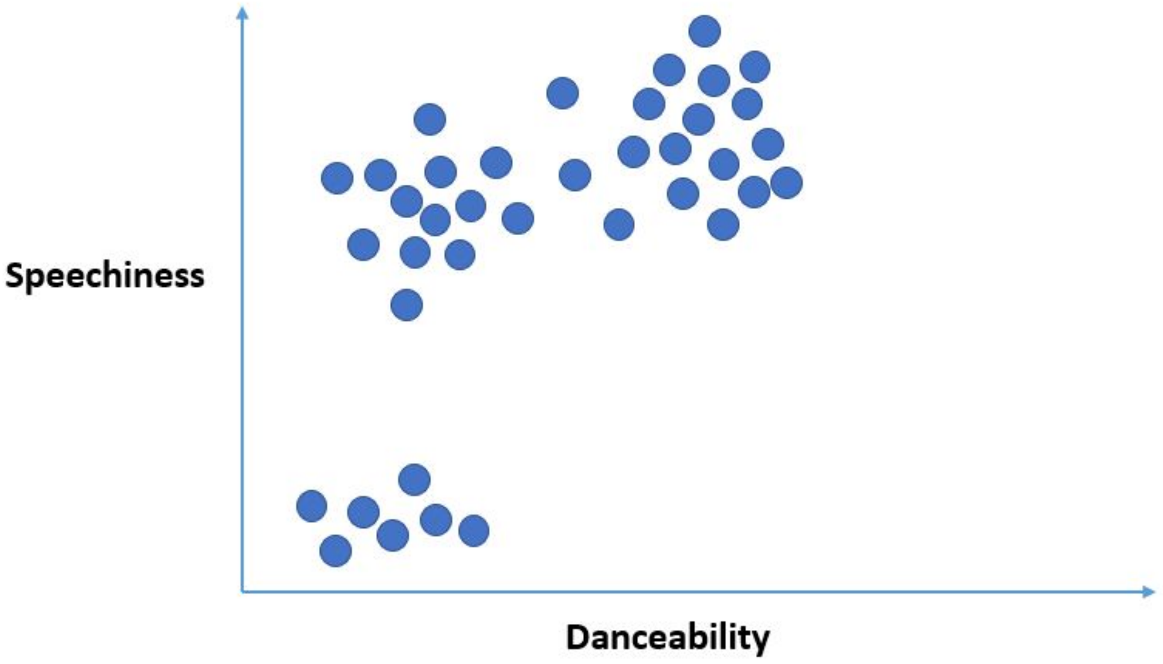
Unsupervised Learning

Selection algorithm



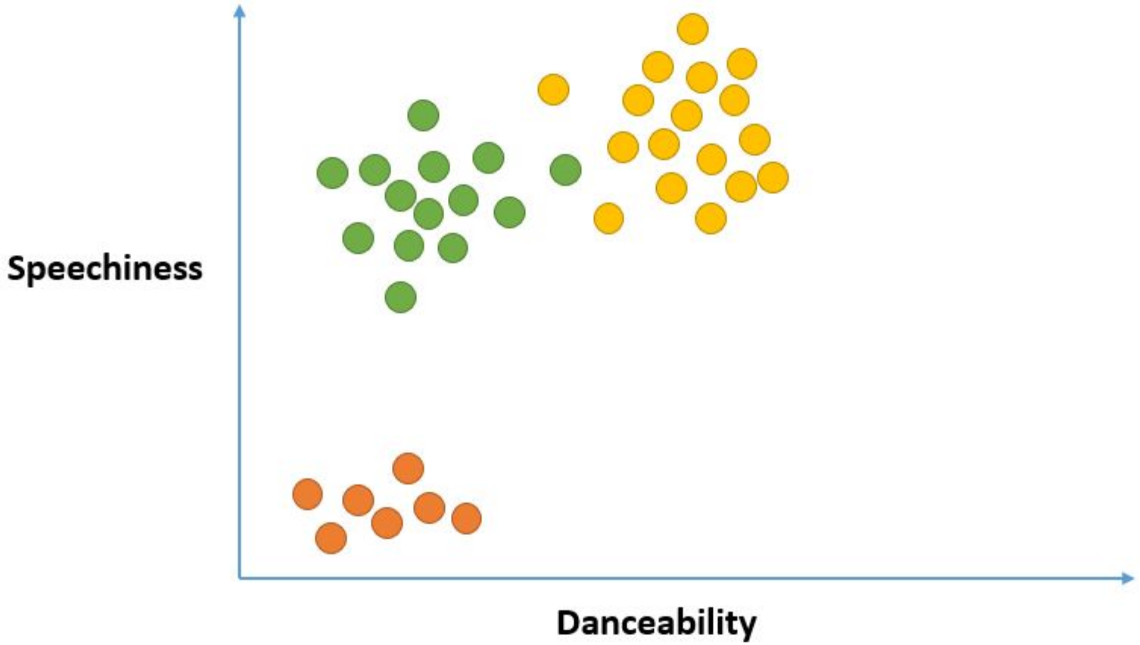
Satisfied User

Unsupervised Learning



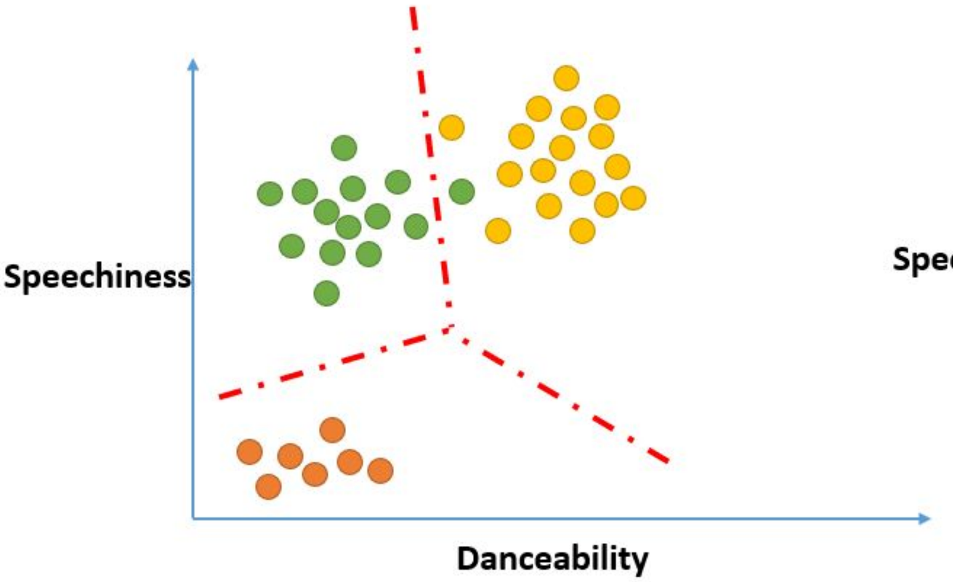
https://developer.spotify.com/documentation/web-api/reference/tracks/get-audio-features/

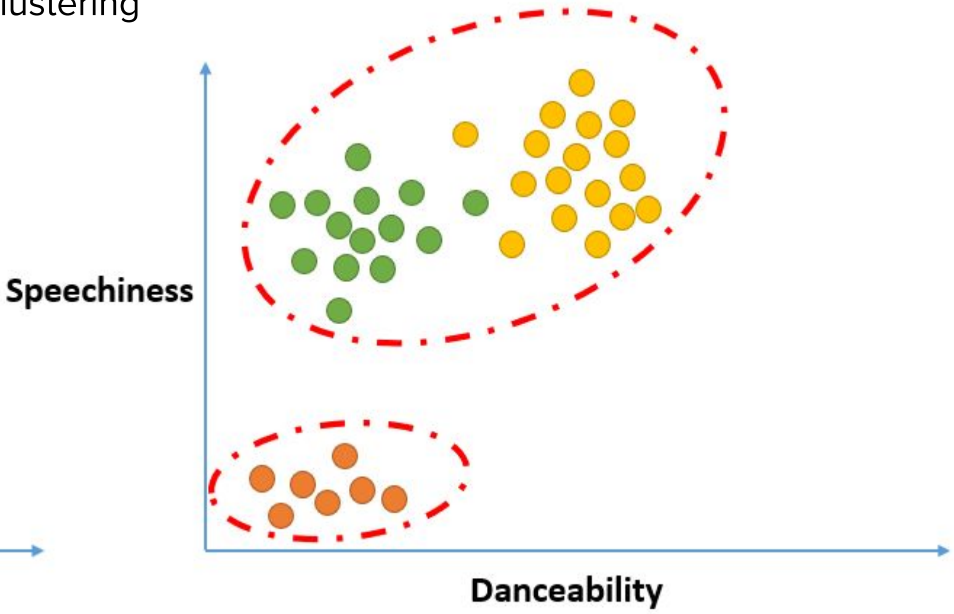
Unsupervised Learning



Unsupervised Learning

Classification vs. Clustering





Unsupervised Learning

Classification vs. Clustering

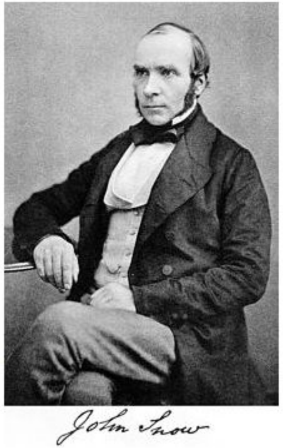
Criteria	Classification	Clustering
Prior knowledge of training data classes	Yes	No
Use case	Classify new sample into new classes	Suggest groups based on pattern similarities
Algorithms	Decision trees, Bayesian classifiers	K-means clustering, Hierarchical clustering, Expectation Maximization (EM)
Training data needs	Data from different classes with class labels	Example data without labels

Unsupervised Learning

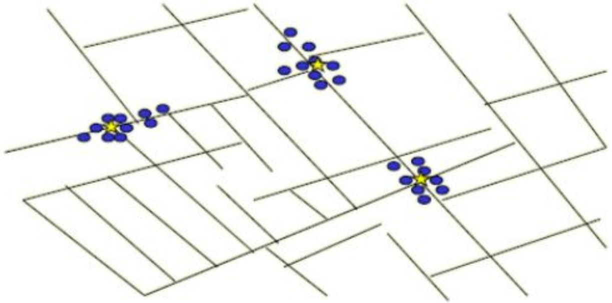
Clustering History

- John Snow, a London physician plotted the location of cholera deaths on a map during an outbreak in 1850.

- The locations indicated that cases were clustered around certain intersections where there were polluted wells, thus exposing both the problem and the solution.



source: wikipedia



From: Nina Mishra HP Labs

Topics for today

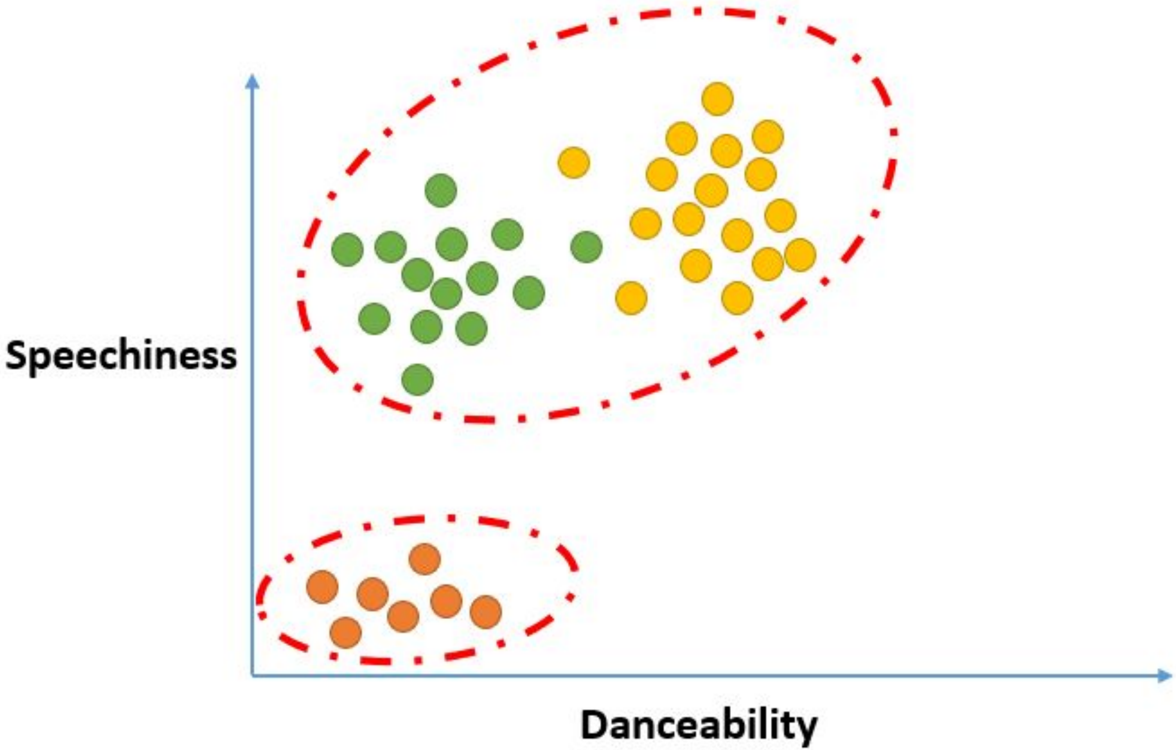
- K-means clustering
- Bag of words (dictionary learning)
- Python examples

- Hierarchical clustering
- Spectral clustering
- Self organizing maps (SOM)
- Independent component analysis (ICA)
- Unsupervised learning with neural networks

K-means clustering

- One of the oldest yet most robust ways to divide data into categories (clusters).

k = ?



K-means clustering

Given k, the **k-means algorithm** works as follows:

1. Choose k (random) data points (seeds) to be the initial centroids, cluster centers

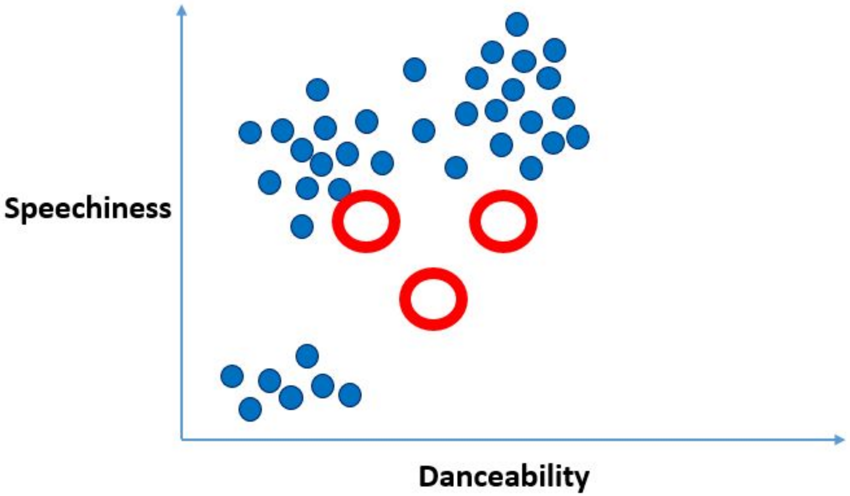
2. Assign each data point to the closest centroid

3. Re-compute the centroids using the current cluster memberships

4. If a convergence criterion is not met, repeat steps 2 and 3

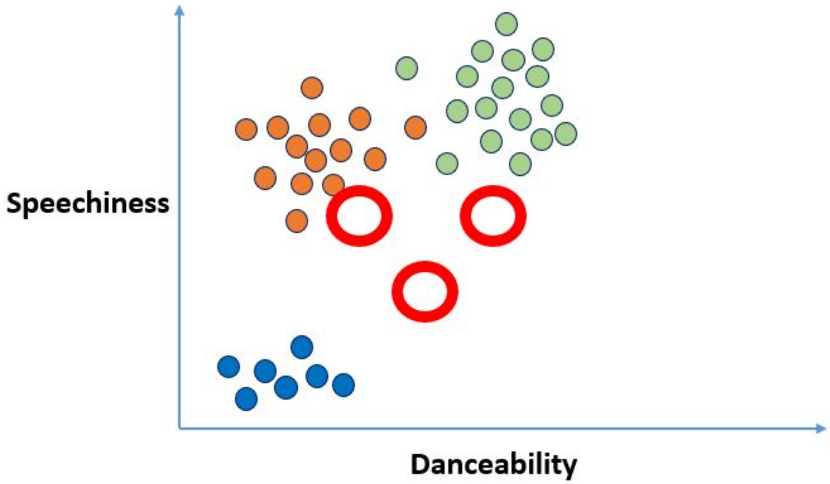
K-means clustering

Step-1: Determine random cluster centers



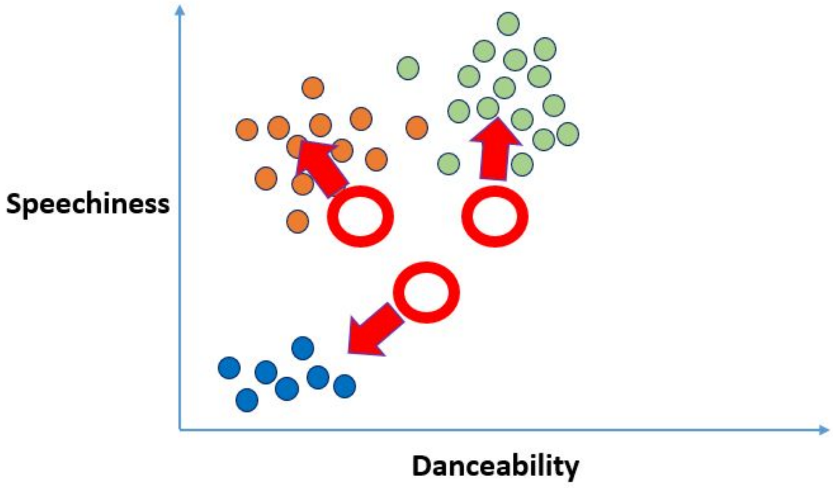
K-means clustering

Step-2: Determine the cluster membership (based on the similarity measurement)



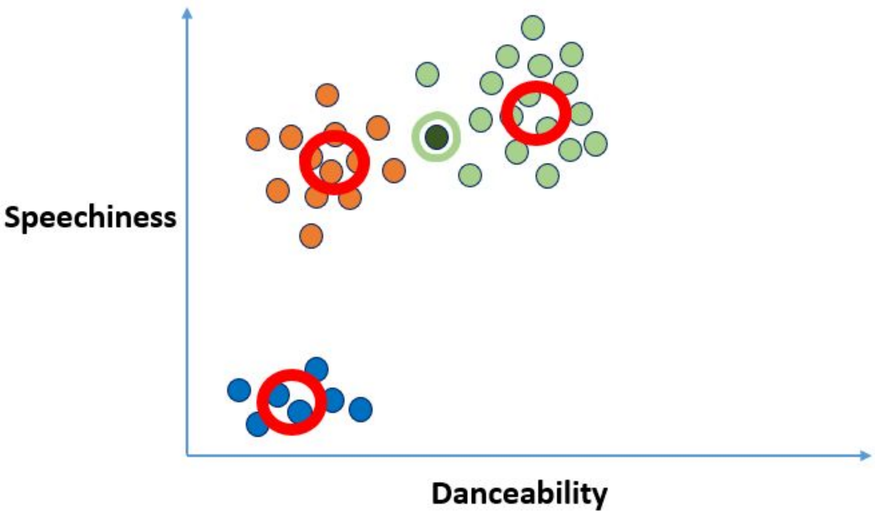
K-means clustering

Step-3: Re-estimate the cluster centers



K-means clustering

Repeat Step-2: Determine the cluster membership (with the renewed cluster center positions)



K-means clustering

The k-means algorithm **stopping (convergence) criterion** is as follows:

- no (or minimum) re-assignments of data points to different clusters, or
- no (or minimum) change of centroids, or
- minimum decrease in the sum of squared error (SSE)

$$SSE = \sum_{j=1}^{k} \sum\nolimits_{\mathbf{x} \in C_j} d(\mathbf{x}, \mathbf{m}_j)^2$$

– C_j is the jth cluster,
– $\mathbf{m}_j$ is the centroid of cluster C_j (the mean vector of all the data points in C_j),
– $d(\mathbf{x}, \mathbf{m}_j)$ is the (Eucledian) distance between data point $\mathbf{x}$ and centroid $\mathbf{m}_j$.

K-means clustering

Strengths:

– **Simple:** easy to understand and to implement

– **Efficient:** Time complexity: O(tkn),

where n is the number of data points, k is the number of clusters, and t is the number of iterations.

– Since both k and t are small. k-means is considered a linear algorithm.

• *K-means is the most popular clustering algorithm*.

• *Note that: it terminates at a local optimum if SSE is used. The global optimum is hard to find due to complexity.*

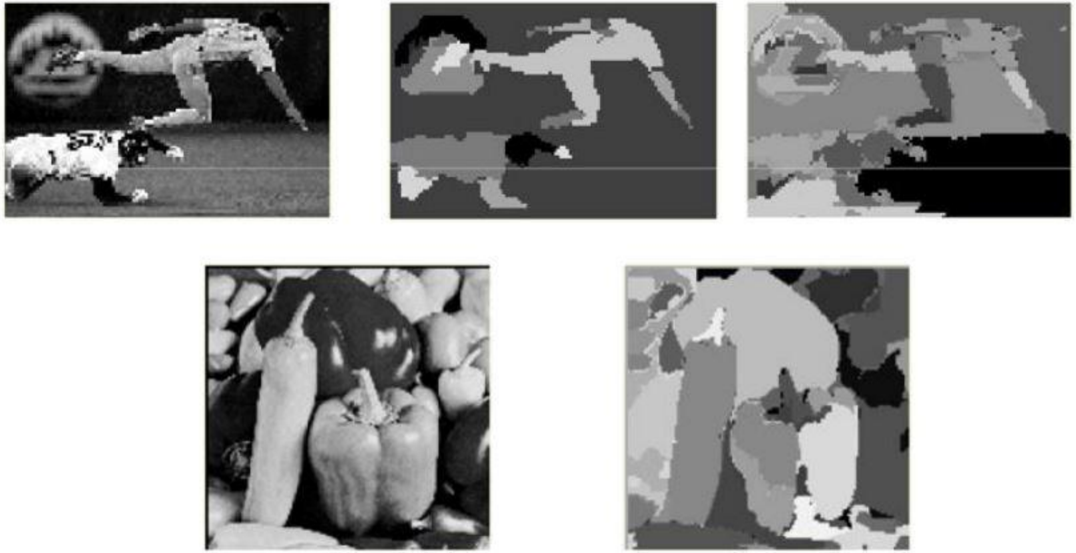
K-means clustering

Weakness:

- The algorithm is only applicable if the mean is defined.
- The user needs to specify k.
- The algorithm is sensitive to outliers
  - Outliers are data points that are very far away from other data points.
  - Outliers could be errors in the data recording or some special data points with very different values.

K-means clustering

Image processing (**fully automated segmentation**) example:



From: Image Segmentation by Nested Cuts, O. Veksler, CVPR2000

K-means clustering

Choosing a good similarity measure to group data points:

Proximity measure, either:

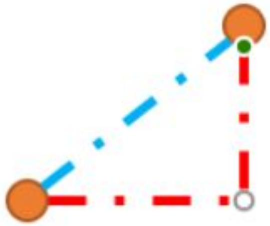
* similarity measure $s(x_i, x_k)$ (large if x_i and x_k are similar)
* dissimilarity (or spatial distance) measure $d(x_i, x_k)$ (small if x_i and x_k are similar).

K-means clustering

Some spatial distance measures:

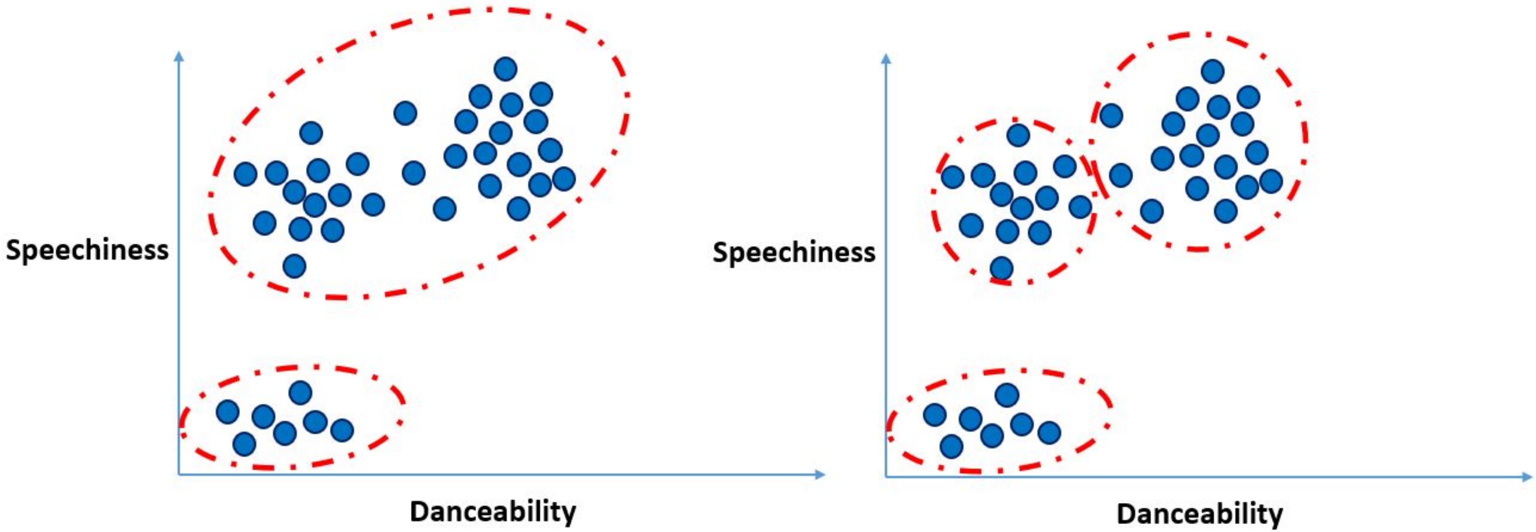
* Euclidean distance $d(x_i, x_j) = \sqrt{\sum_k^N (x_i^k - x_j^k)^2}$

* Manhattan (city block) distance $d(x_i, x_j) = \sum_k^N |(x_i^k - x_j^k)|$



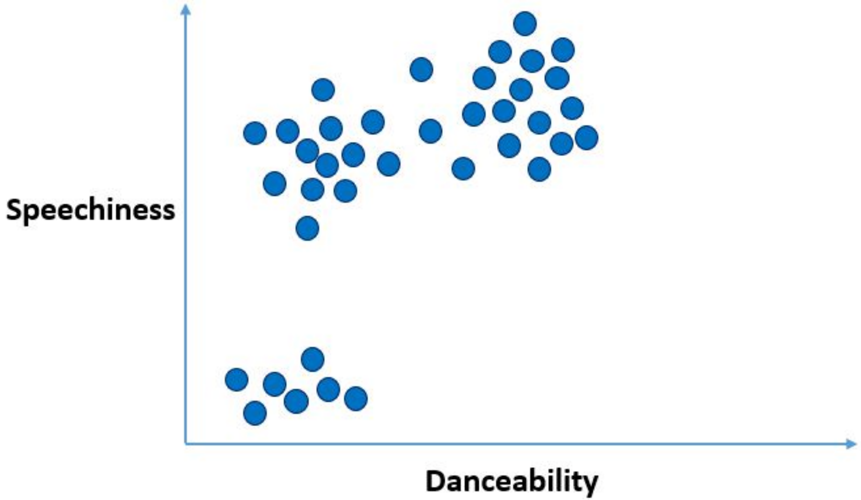
K-means clustering

How many clusters?



K-means clustering

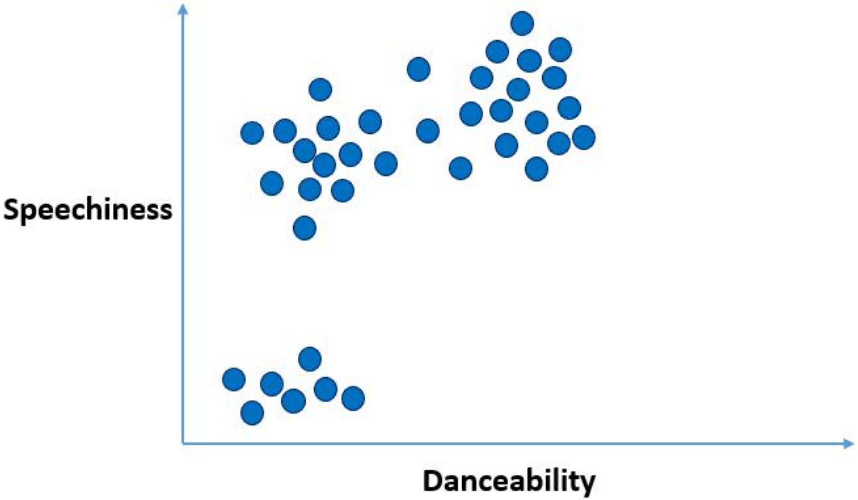
Possible solutions to find a good k (when it is unknown):



2 classes or **3 classes**?

K-means clustering

Possible solutions to find a good k (when it is unknown):

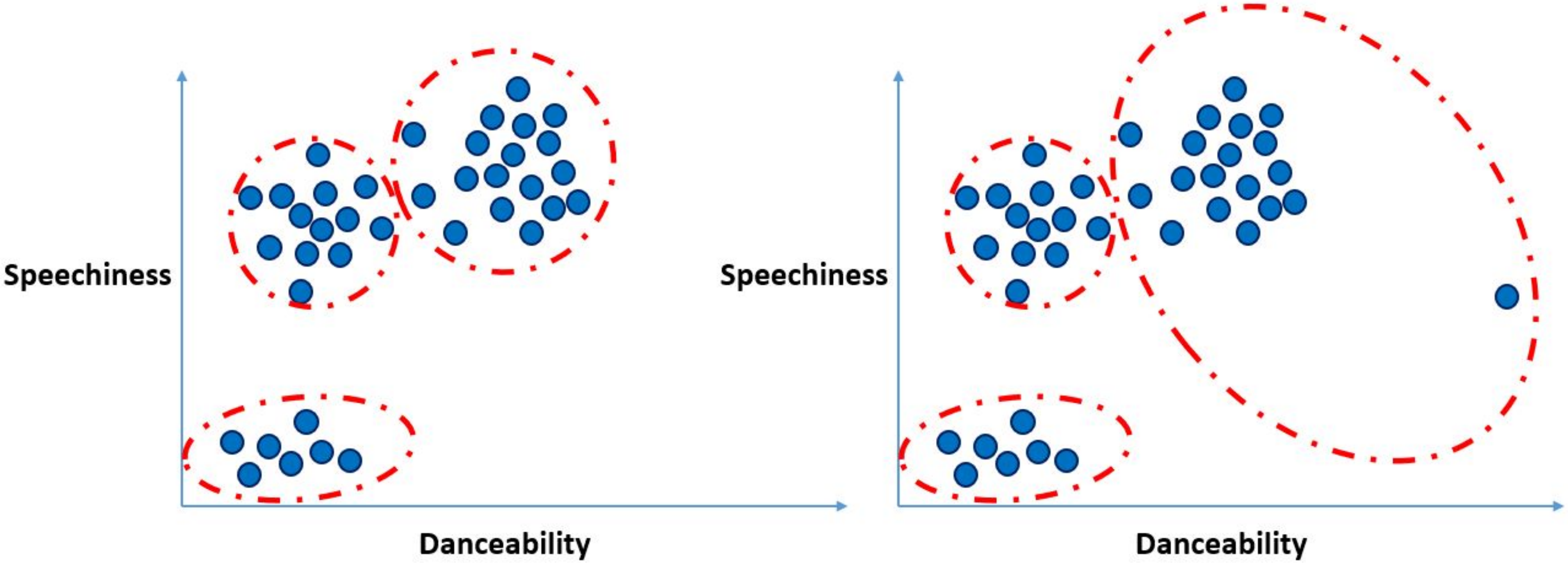


1. iterate number k

2. find the optimal cluster number looking at the best satisfied "criterion"

K-means clustering

Dealing with outliers:



K-means clustering

Dealing with outliers:

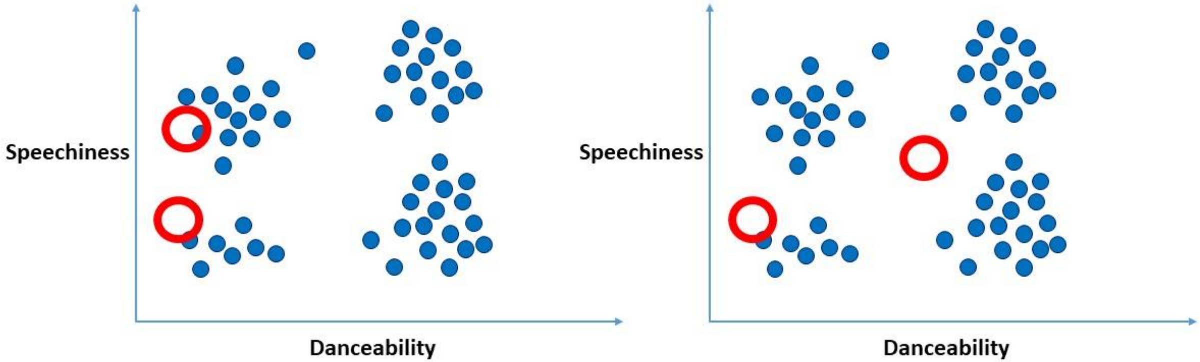
(Some ideas)

- within each cluster, remove the data points which are far away from centers and find new centers afterwards

- perform random sampling and find the cluster centers again (assuming that most of the outliers will be removed)

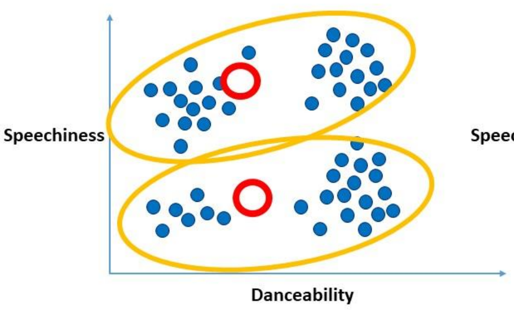
K-means clustering

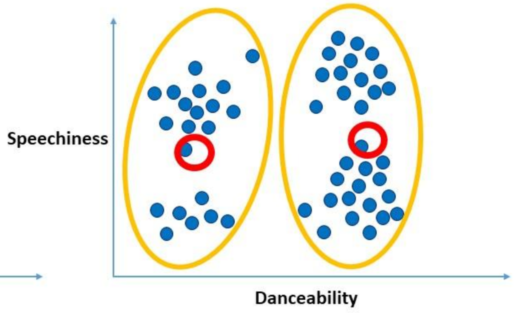
Dealing with random initialization:



K-means clustering

Dealing with random initialization:

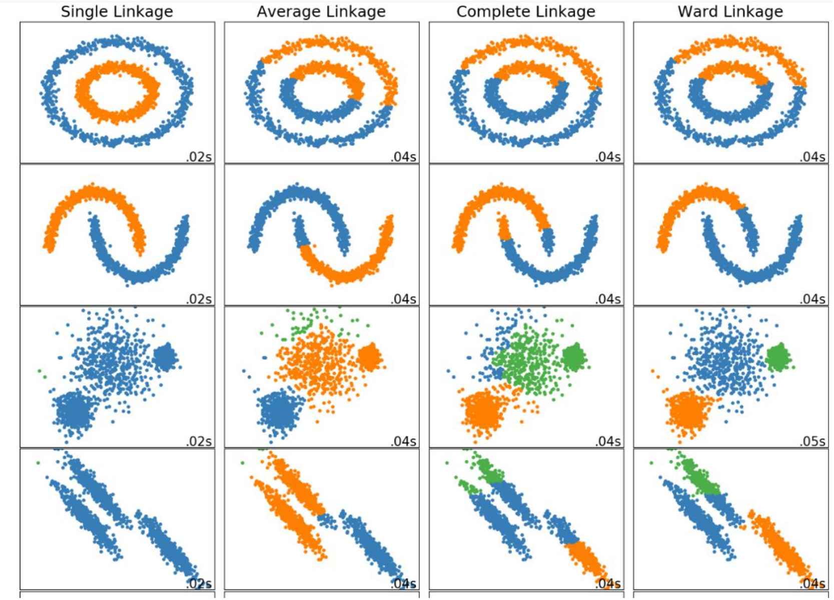




K-means clustering

Dealing with special data structures:

K-means algorithm is not suitable for discovering clusters that are not hyper-ellipsoids (or hyper-spheres).



K-means clustering

Some final reflections:

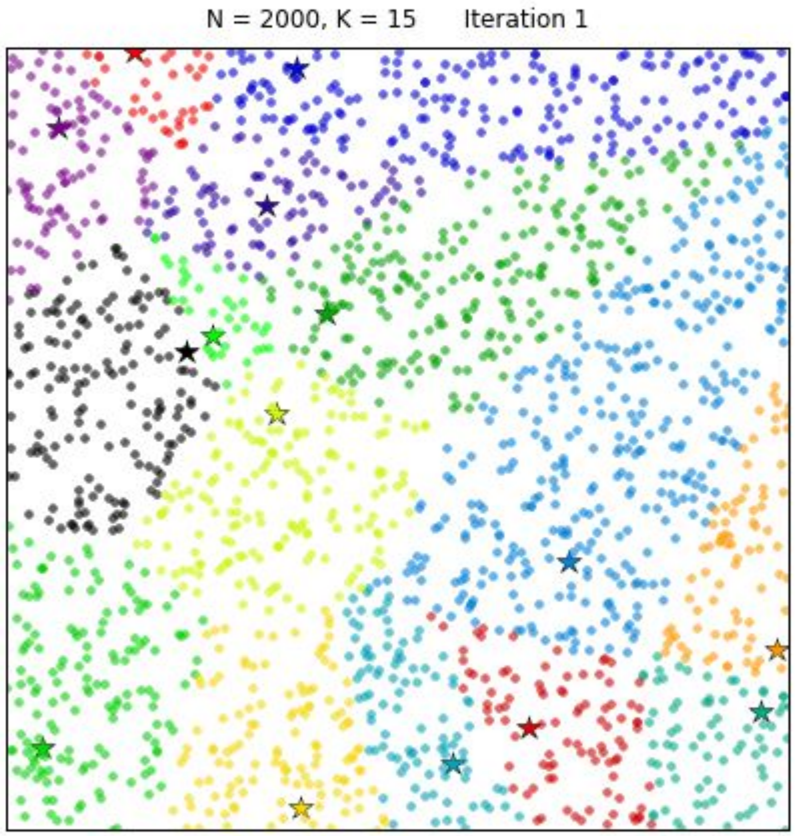
- Despite weaknesses, k-means is the most popular algorithm due to its simplicity and efficiency

- No clear evidence that any other clustering algorithm performs better in general

  (Comparing different clustering algorithms is a difficult task, since no one knows the correct clusters.)

K-means clustering

In action...



Bag of words

History:

> *It was the best of times,*
> *it was the worst of times,*
> *it was the age of wisdom,*
> *it was the age of foolishness,*

Bag of words

History:

- “it” = 1
- “was” = 1
- “the” = 1
- “best” = 1
- “of” = 1
- “times” = 1
- “worst” = 0
- “age” = 0
- “wisdom” = 0
- “foolishness” = 0

As a binary vector, this would look as follows:

```
[1, 1, 1, 1, 1, 1, 0, 0, 0, 0]
```

The other three documents would look as follows:

```
"it was the worst of times" = [1, 1, 1, 0, 1, 1, 1, 0, 0, 0]
"it was the age of wisdom" = [1, 1, 1, 0, 1, 0, 0, 1, 1, 0]
"it was the age of foolishness" = [1, 1, 1, 0, 1, 0, 0, 1, 0, 1]
```

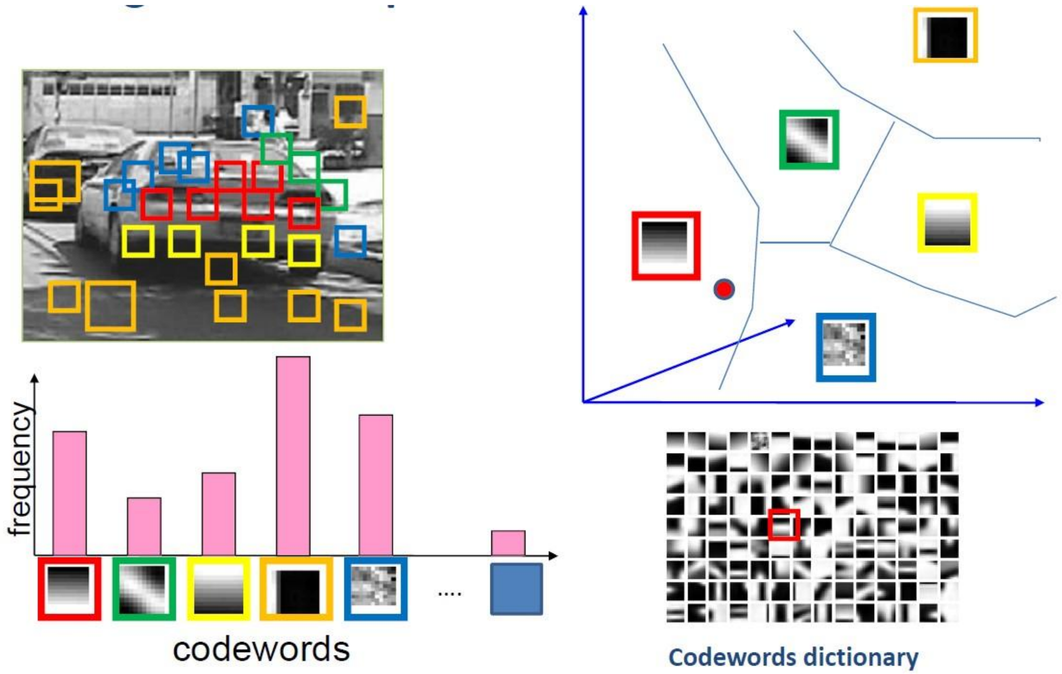
Bag of words

Constructing visual vocabulary:

“Recognizing and Learning Object Categories”, 2007, by Prof L. Fei-Fei, A. Torralba, and R. Fergus

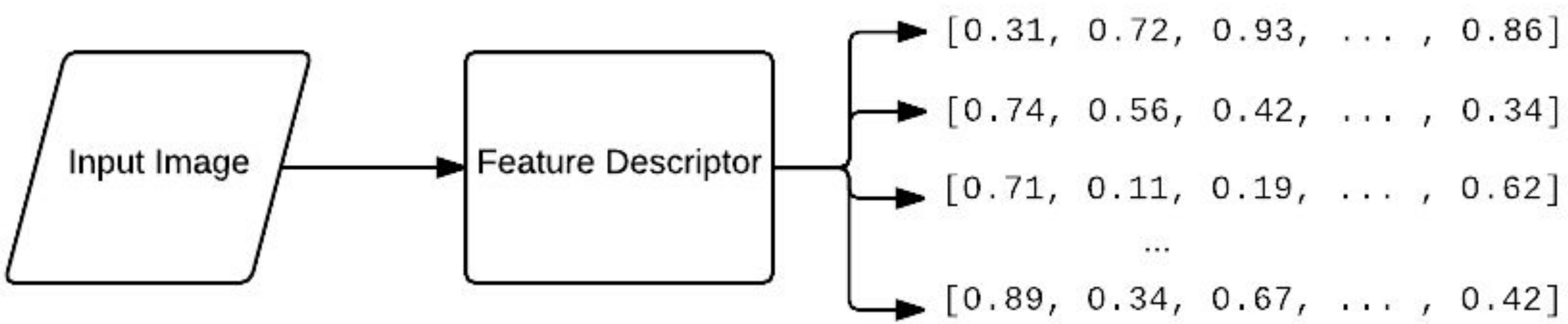
Bag of words

Clustering visual vocabulary:



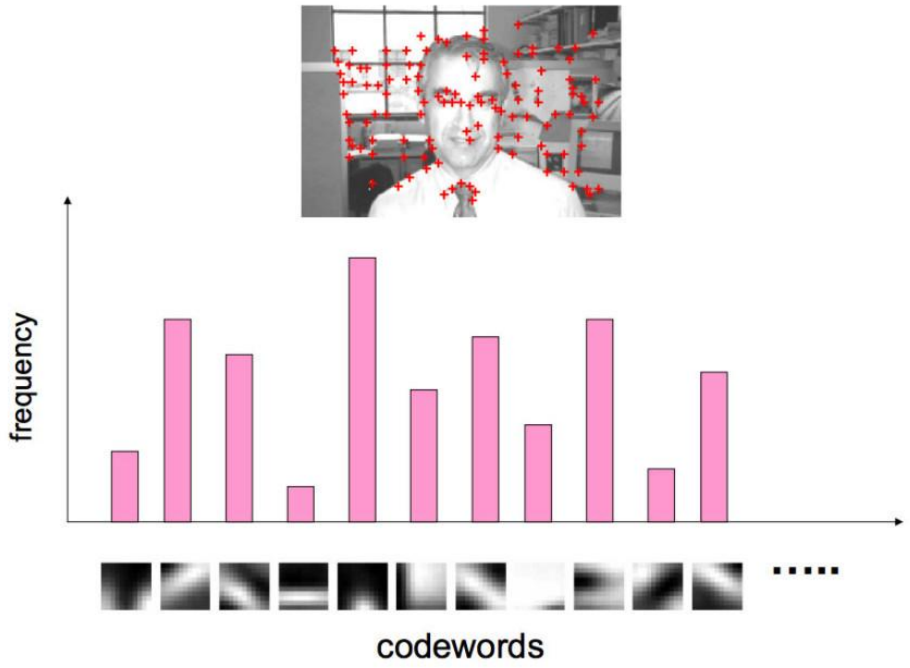
Bag of words

Clustering visual vocabulary:



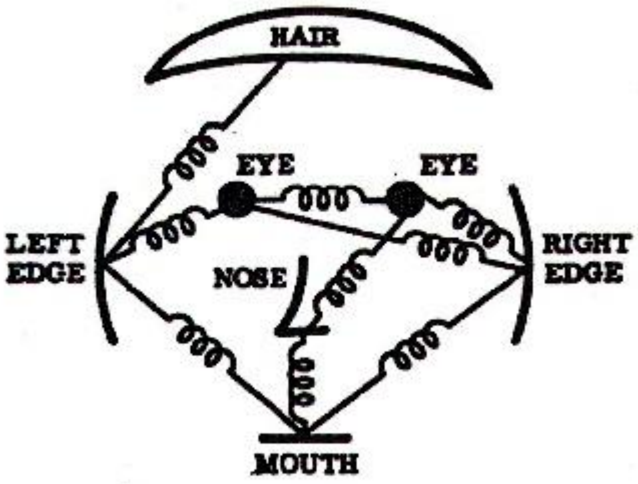
Bag of words

Clustering visual vocabulary:



Bag of words

Visual vocabulary can also lead to recognition (not a topic of today's course):



Python examples