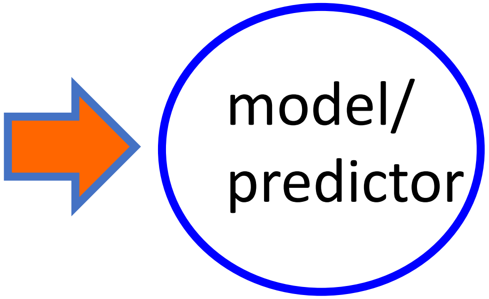
Introduction about K-Means Clustering

Supervised learning

label

label_1

label_3



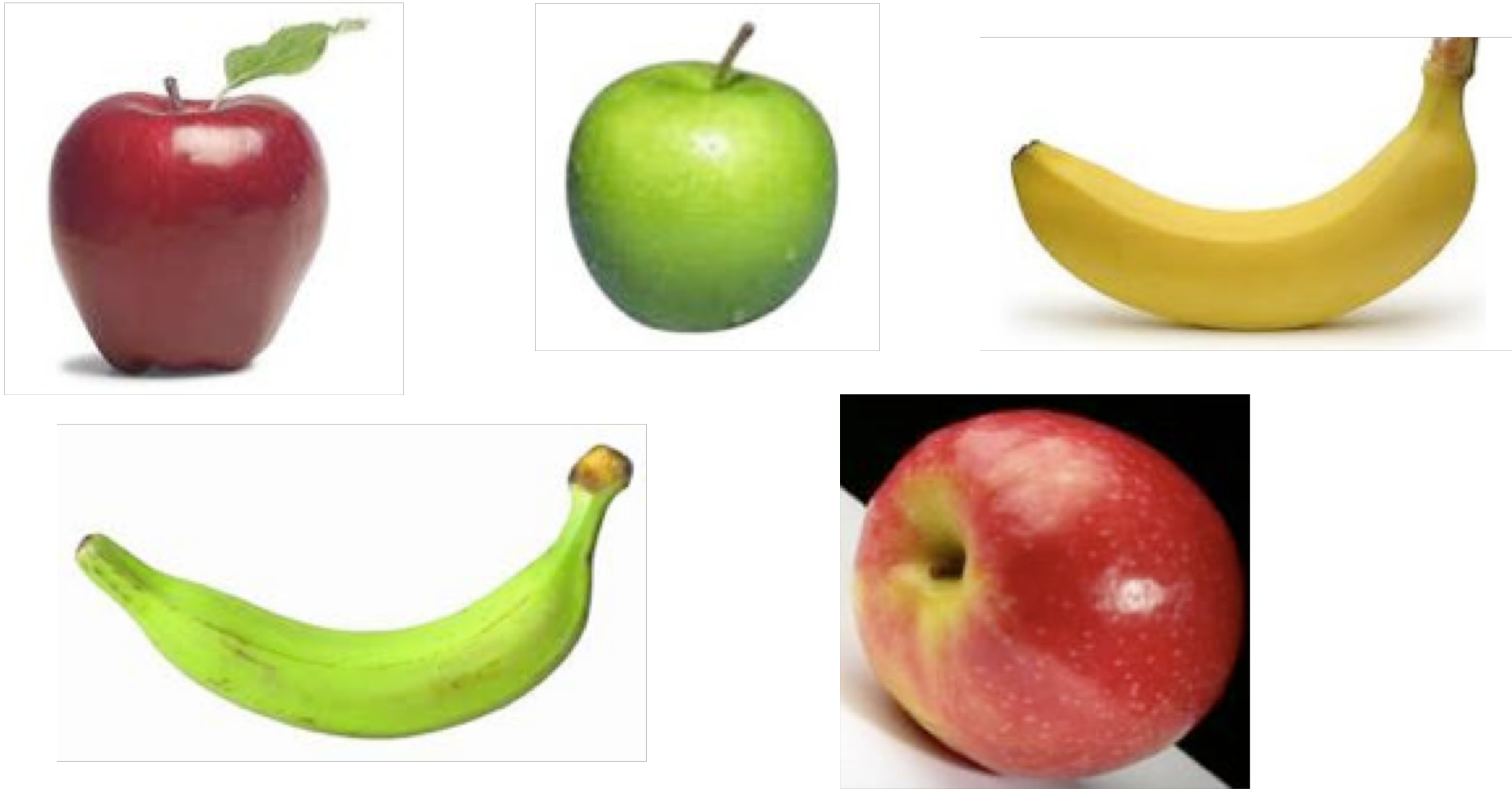
model/
predictor

label_4

label_5

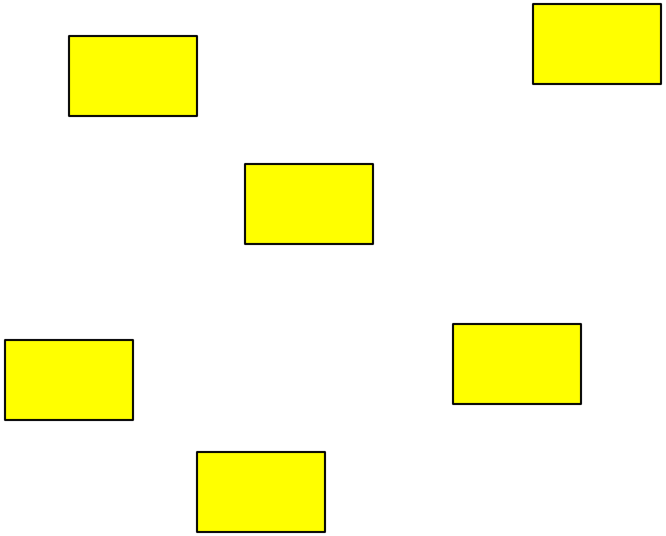
Supervised learning: given labeled examples

Unsupervised learning



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

Unsupervised learning



Given some example without labels, do something!

Unsupervised learning applications

learn clusters/groups without any label

customer segmentation (i.e. grouping)

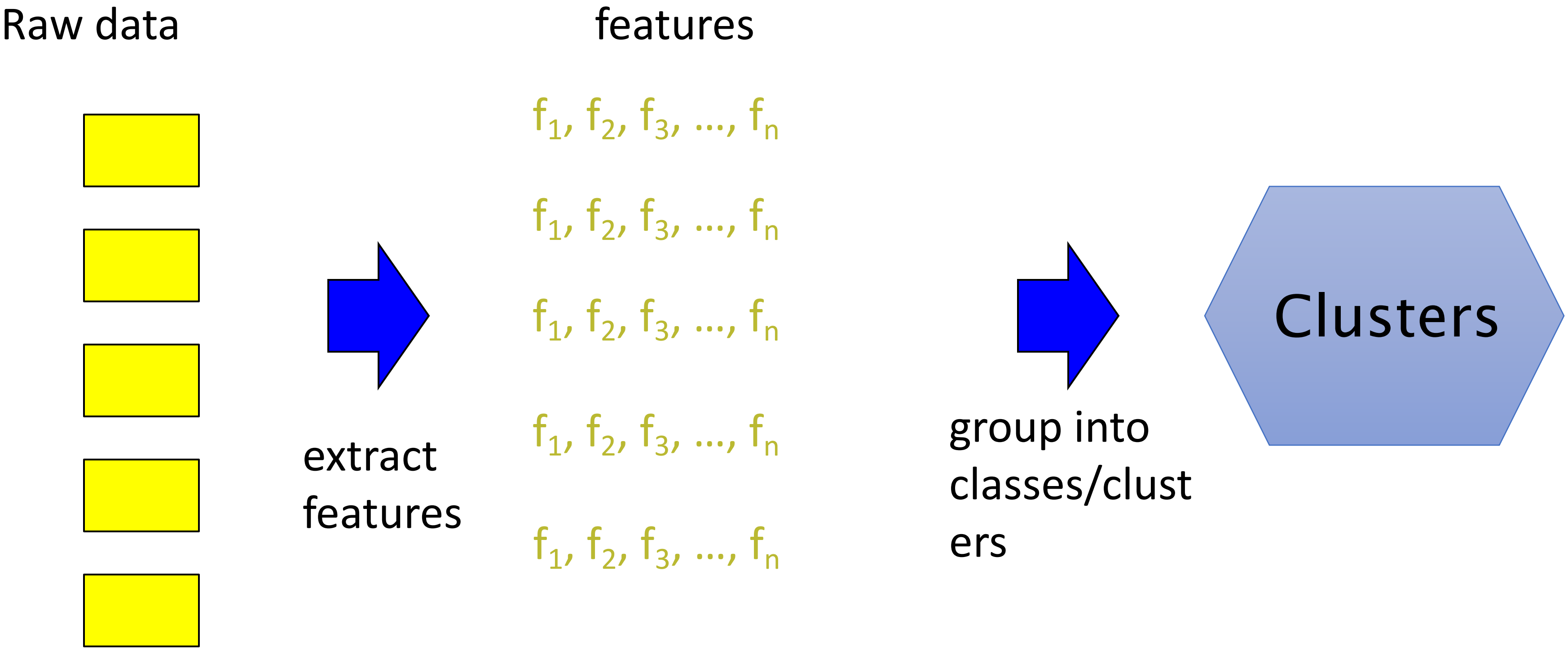
image compression

bioinformatics: learn motifs

find important features

…

Unsupervised learning: clustering



Raw data

features

$f_1, f_2, f_3, \ldots, f_n$

$f_1, f_2, f_3, \ldots, f_n$

$f_1, f_2, f_3, \ldots, f_n$

$f_1, f_2, f_3, \ldots, f_n$

$f_1, f_2, f_3, \ldots, f_n$

extract features

group into classes/clusters

Clusters

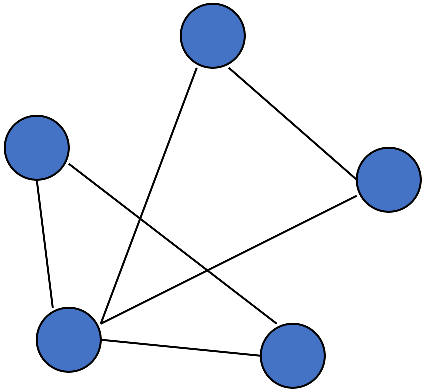
No "supervision", we're only given data and want to find natural groupings

Clustering

Clustering: the process of grouping a set of objects into classes of similar objects

Applications?

Clustering applications



Find clusters of users

- Targeted advertising
- Exploratory analysis

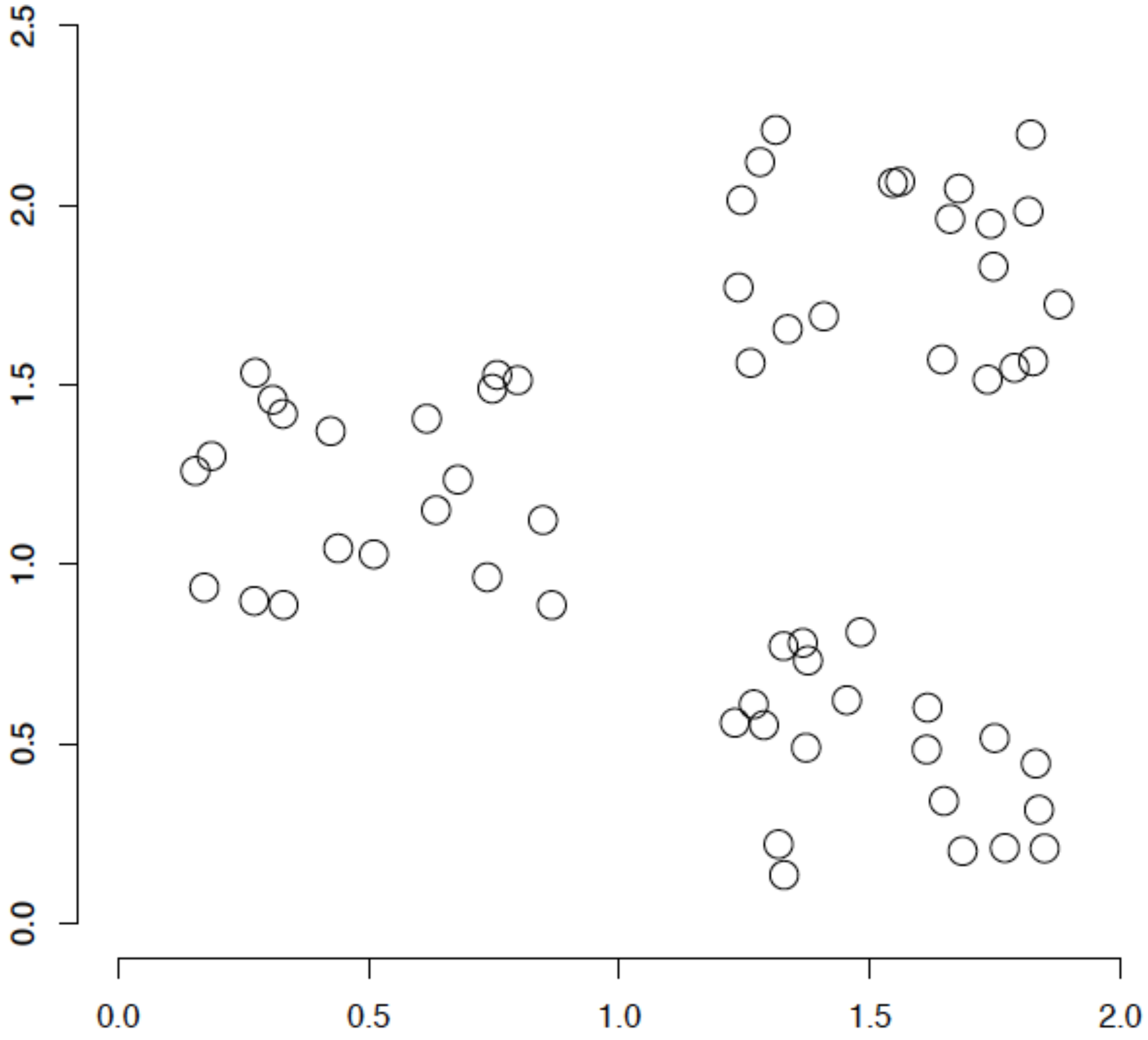
Clusters of the Web Graph

- Distributed pagerank computation

Who-messages-who IM/text/twitter graph

~100M nodes

A data set with clear cluster structure



What are some of the issues for clustering?

What clustering algorithms have you seen/used?

Issues for clustering

Representation for clustering

- How do we represent an example
  - features, etc.
- Similarity/distance between examples

Number of clusters

- Fixed a priori
- Data driven?

K-means

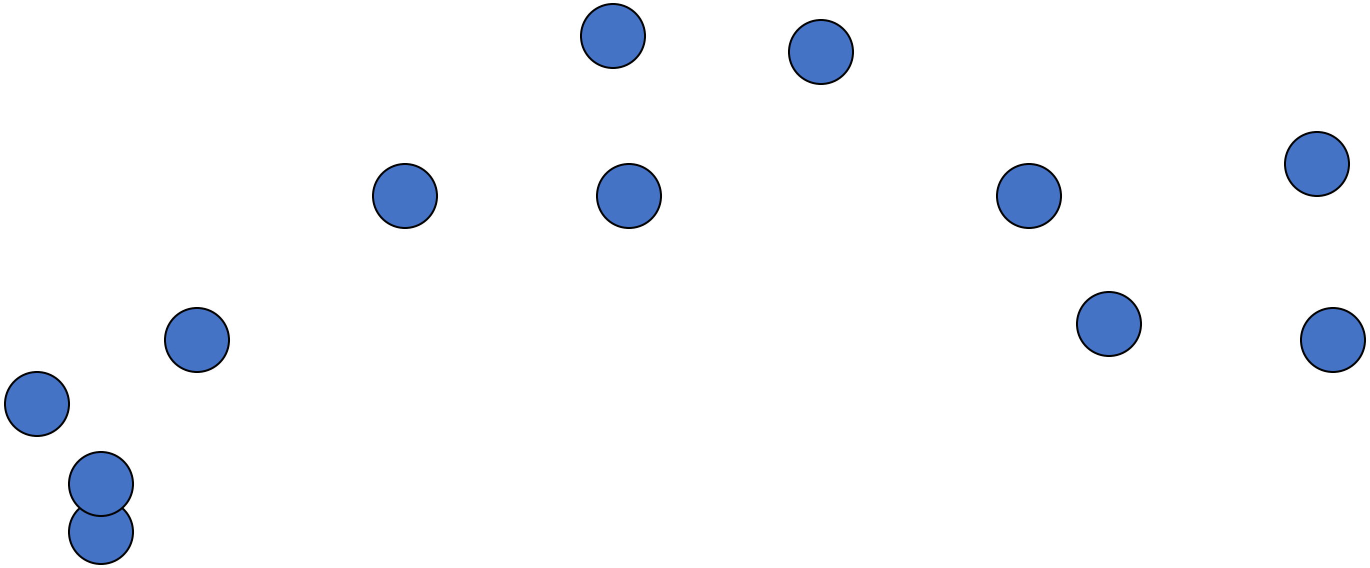
Most well-known and popular clustering algorithm:

Start with some initial cluster centers

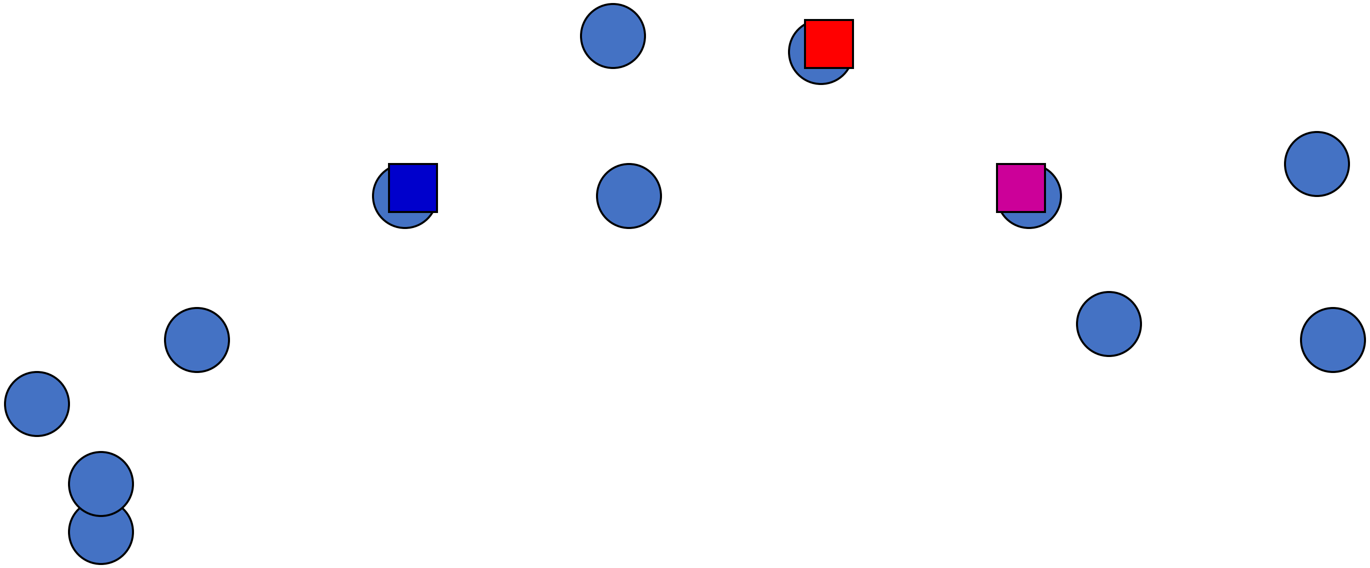
Iterate:

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

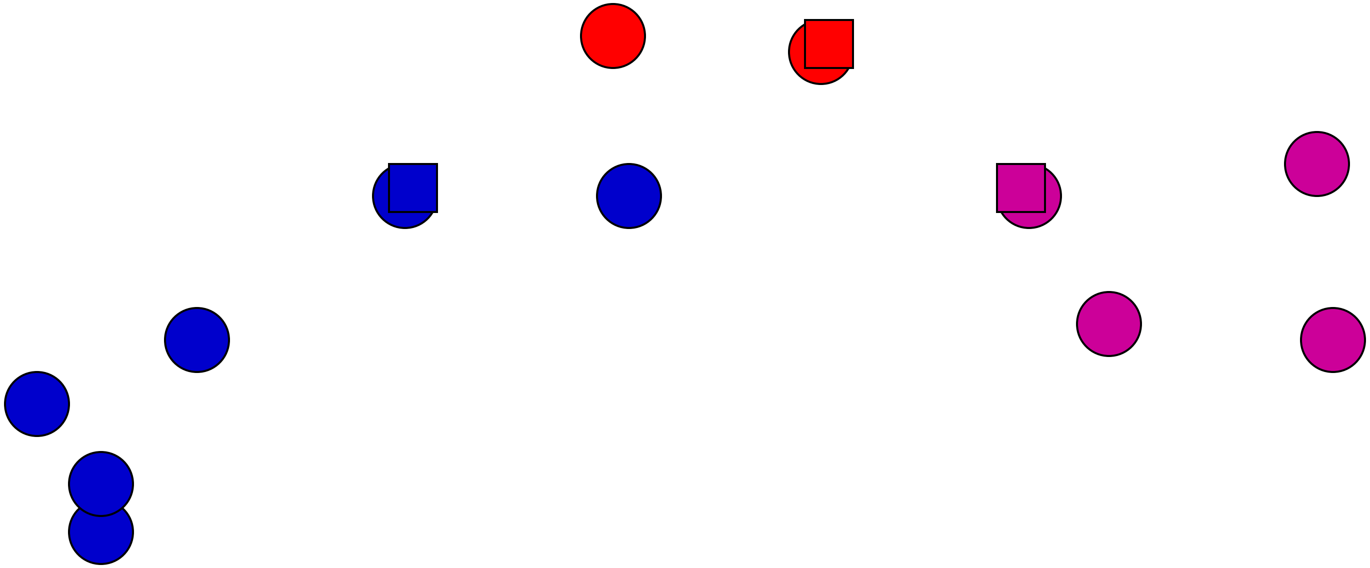
K-means: an example



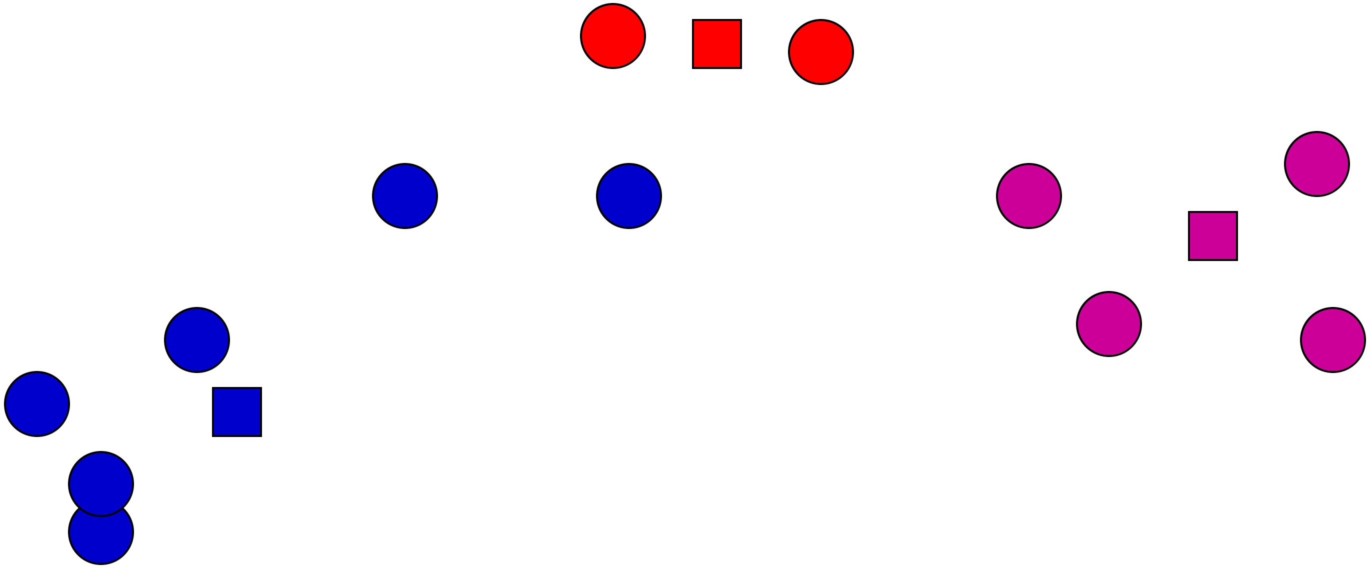
K-means: Initialize centers randomly



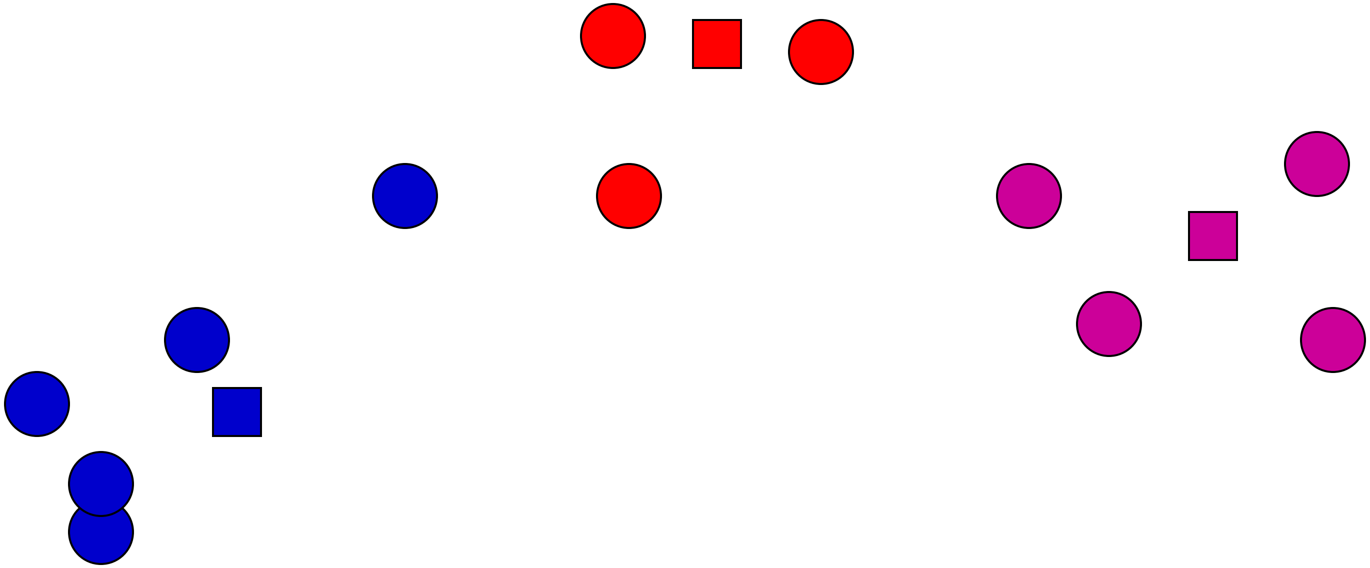
K-means: assign points to nearest center



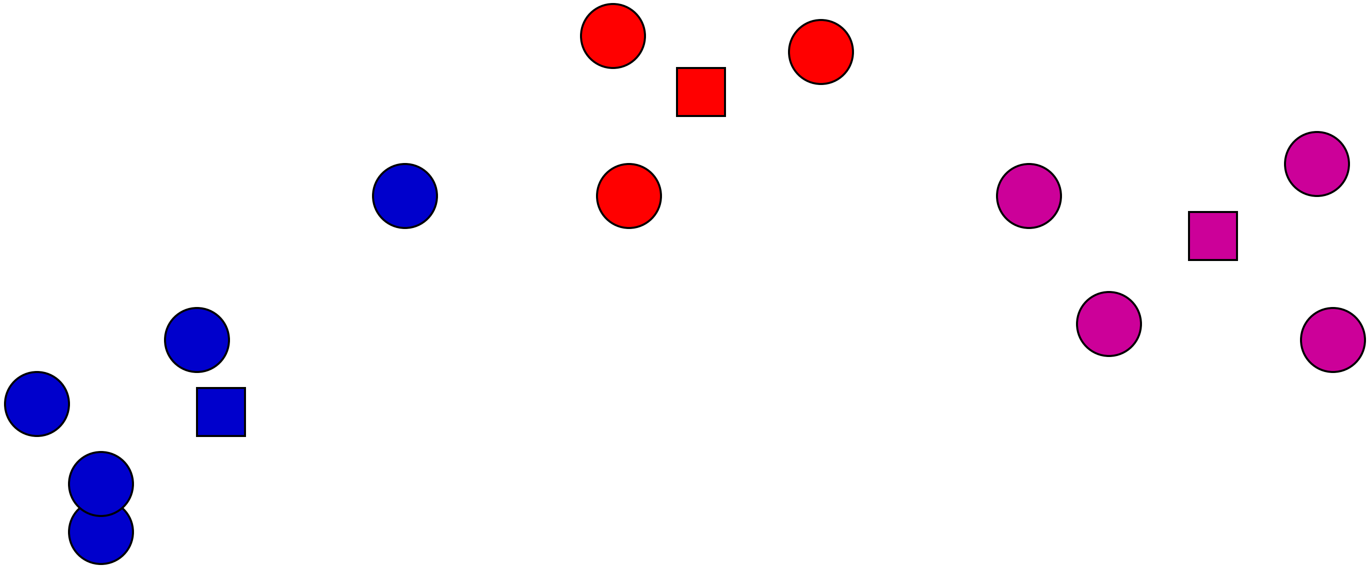
K-means: readjust centers



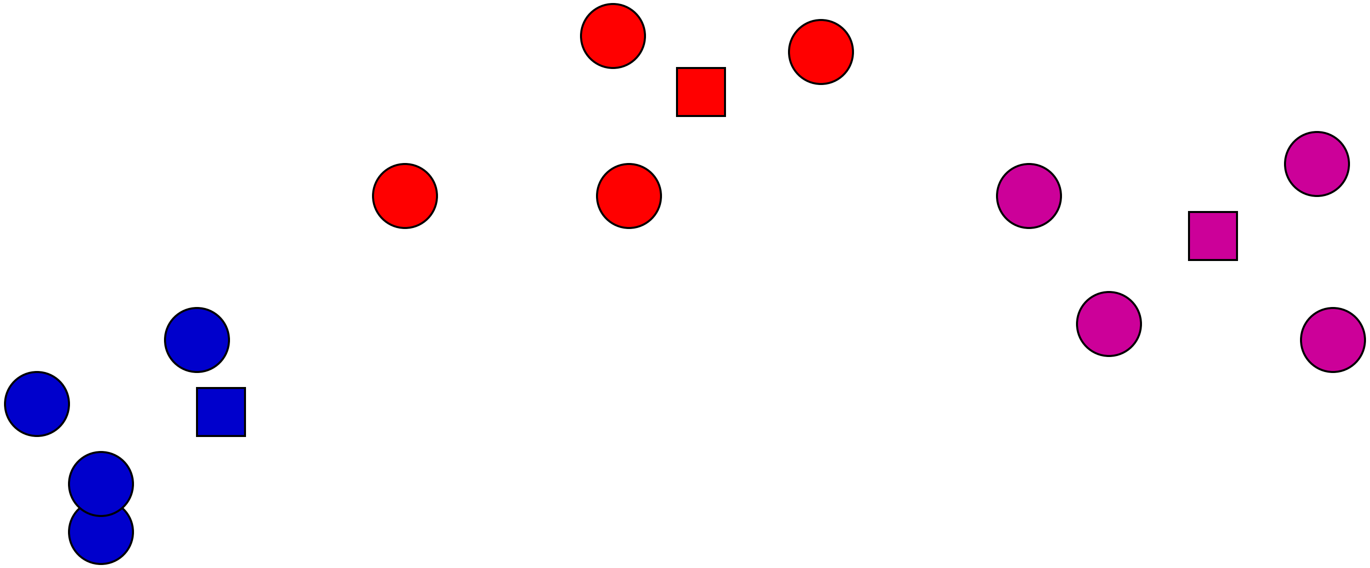
K-means: assign points to nearest center



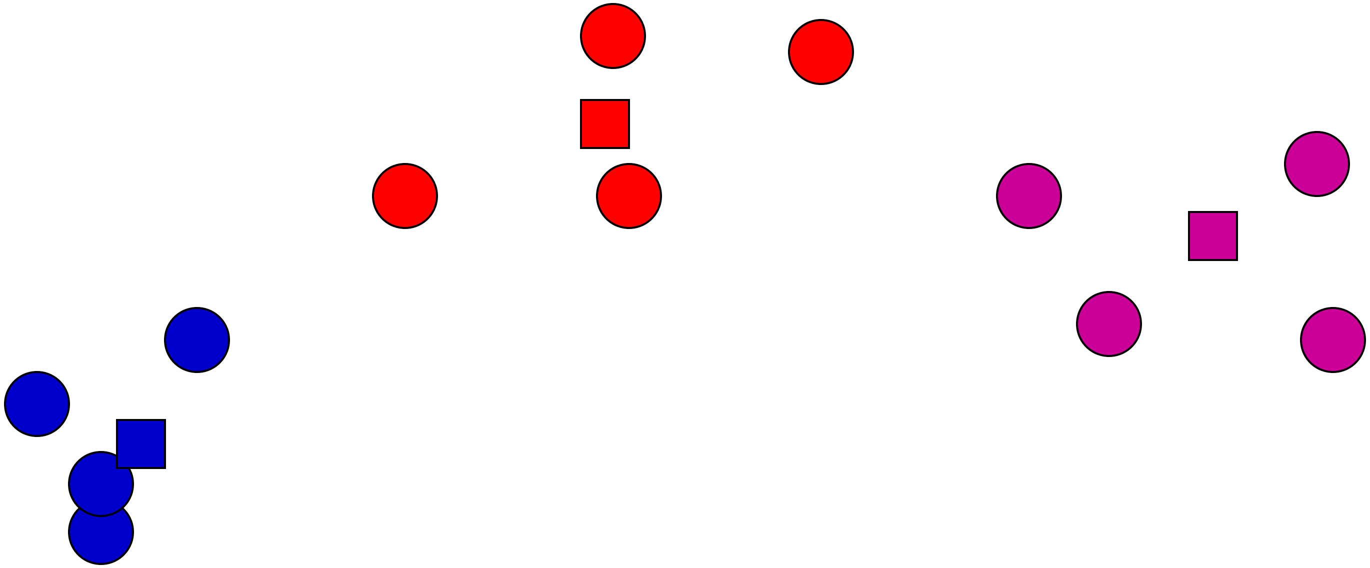
K-means: readjust centers



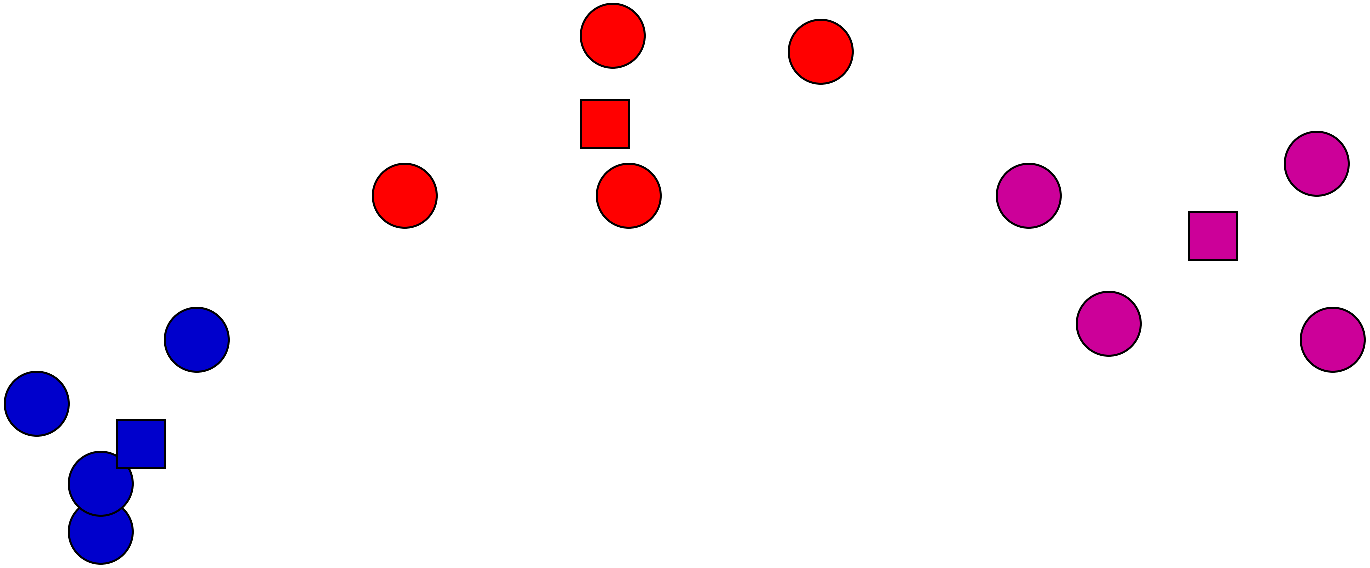
K-means: assign points to nearest center



K-means: readjust centers



K-means: assign points to nearest center

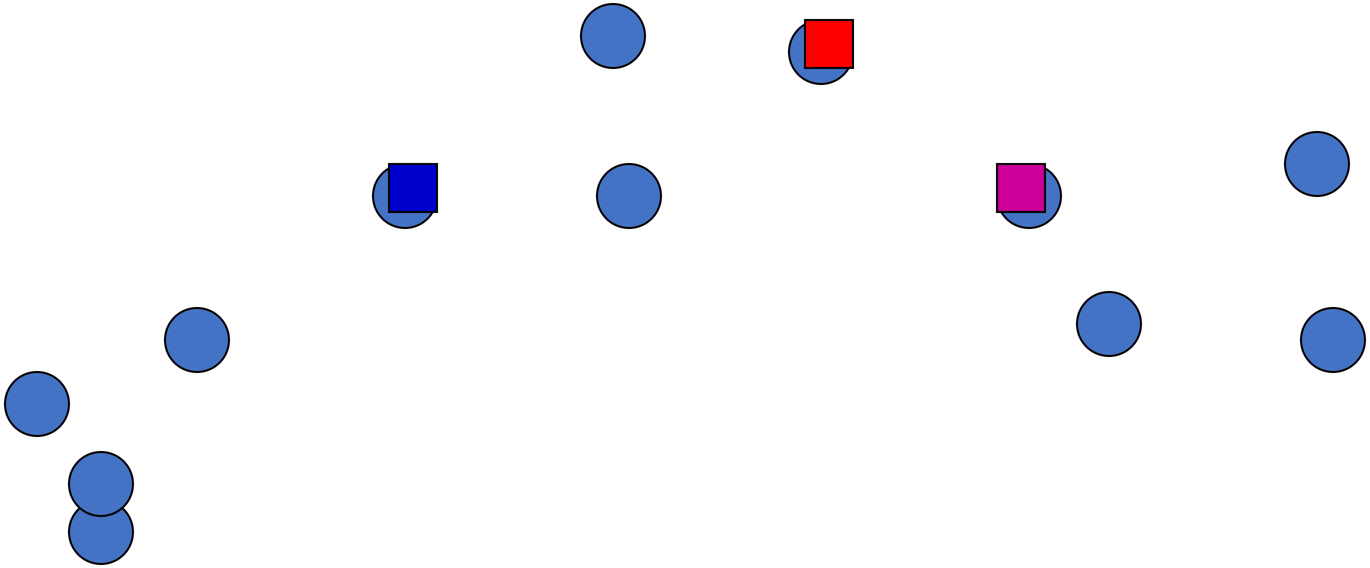


No changes: Done

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

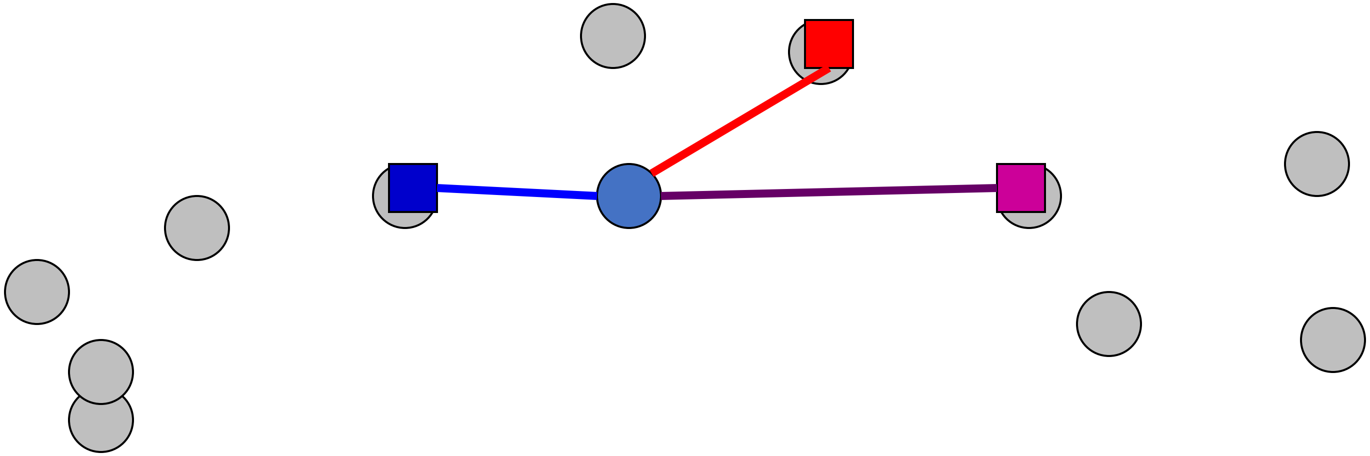
Iterate:

- **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

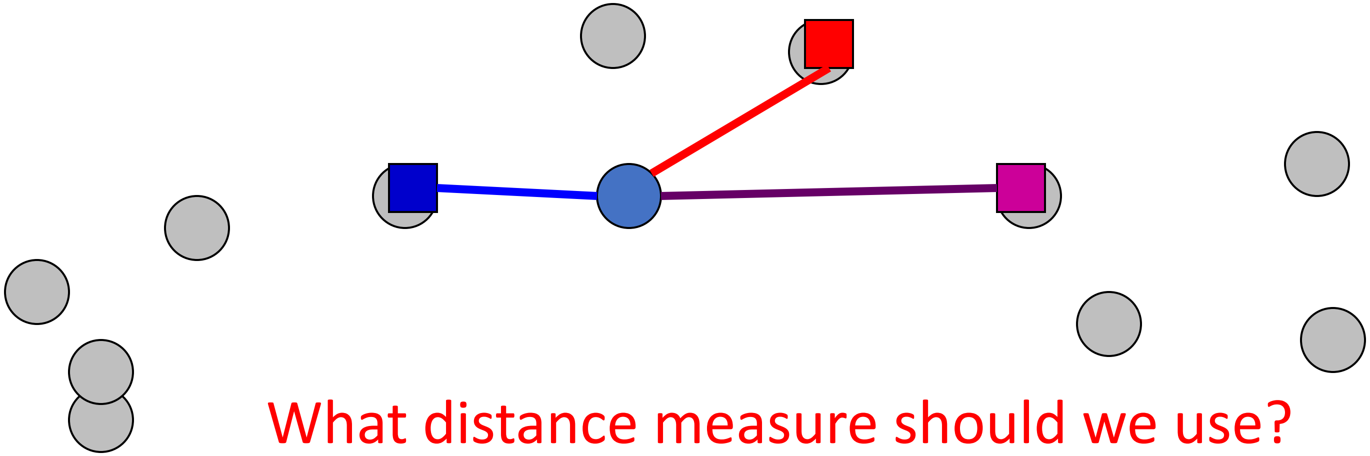
Iterate:

- **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:

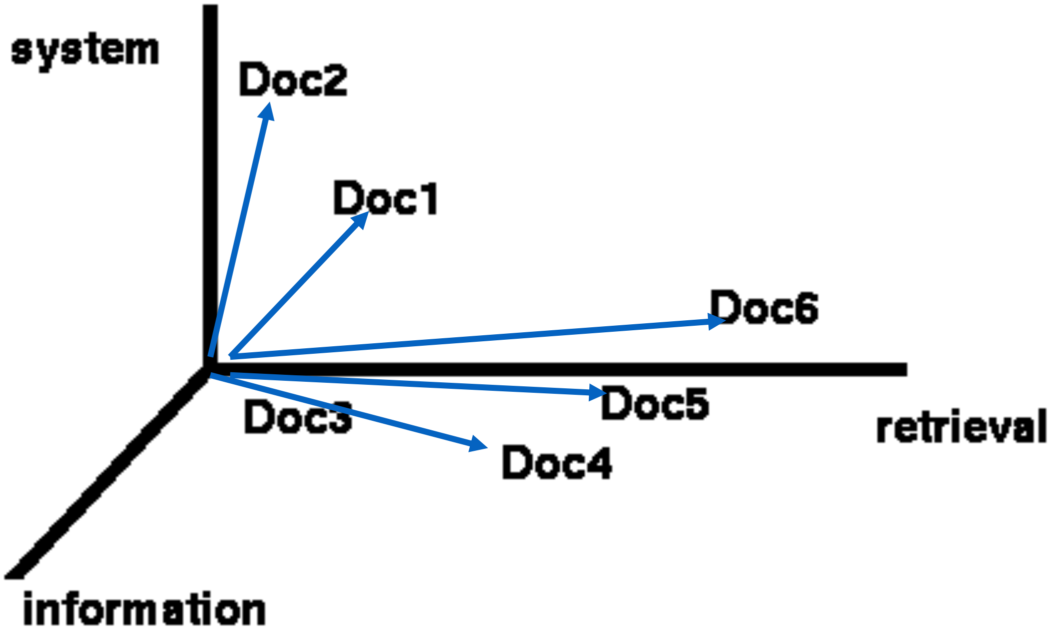
$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

good for spatial data

Clustering documents (e.g. wine data)

One feature for each word. The value is the number of times that word occurs.

Documents are points or vectors in this space



cosine similarity

$$sim(x,y) = \frac{x \bullet y}{|x||y|} = \frac{x}{|x|} \bullet \frac{y}{|y|} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}$$

correlated with the
angle between two vectors