Unsupervised learning. K-means.

Simon Šuster, University of Groningen

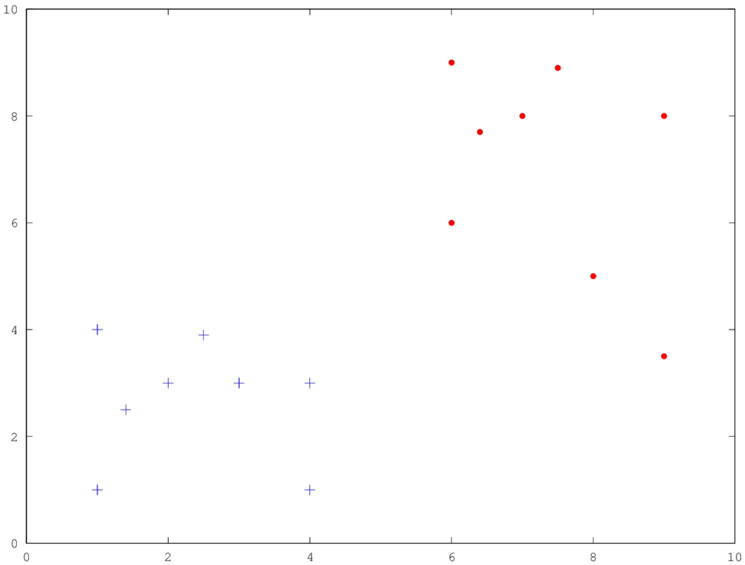
Course *Learning from data*
December 2, 2013

Some slides adapted from Andrew Ng

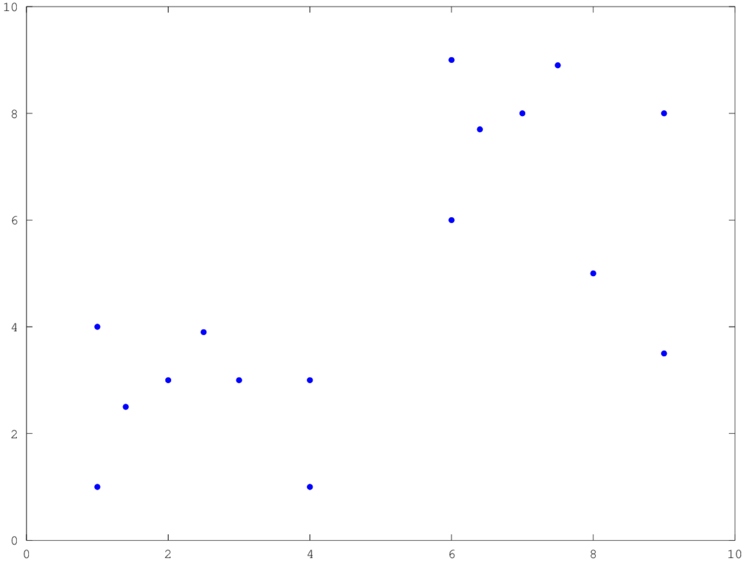
References

- Hal Daumé III: A Course in Machine Learning
  http://ciml.info
- C. D. Manning, P. Raghavan, H. Schuetze: Introduction to Information Retrieval
  http://www-nlp.stanford.edu/IR-book/

Supervised learning I



Unsupervised learning I



Supervised vs. unsupervised learning

- Labeled vs. unlabeled data

- Unsupervised learning tries to find structure in the data
- Clustering: the structure is clusters
- Clusters are groups of objects similar to each other
- What is the right set of clusters?

- Some other types of unsupervised learning:
    - Dimensionality reduction
    - Novelty/anomaly detection

Clustering

Applications

- Search results
- Social network analysis
- Word clusters (e.g. semantic coherence)
- Linguistic typologies

K-means

- Very popular clustering algorithm
- In its standard form $\Rightarrow$ hard clustering
- Flat $\Rightarrow$ partitioning, no hierarchy

K-means process

- Choose K (number of clusters)
- Initialize K centroids

Repeatedly perform these 2 steps:

1. Cluster assignment:
    - Points are assigned the class of the closest centroid
2. Move centroid
    - Move centroid to the average of the points of the same cluster

After a number of iterations, centroid/assignments don't change anymore

Illustration I

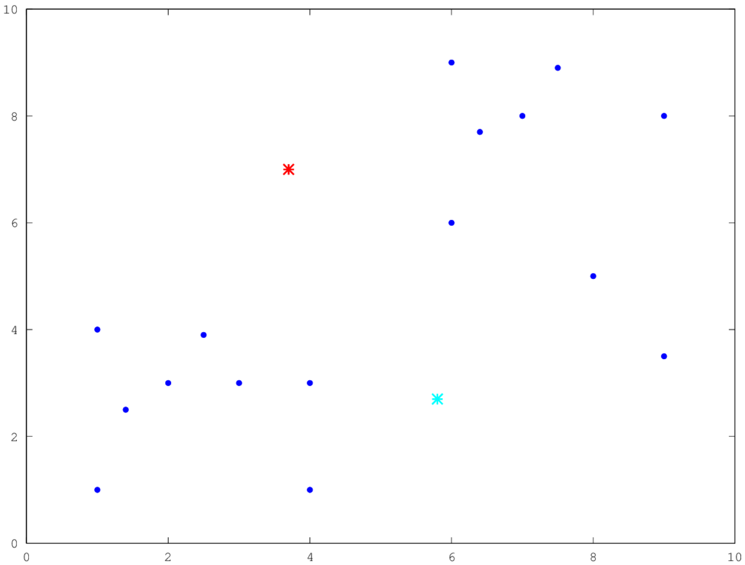


Illustration I

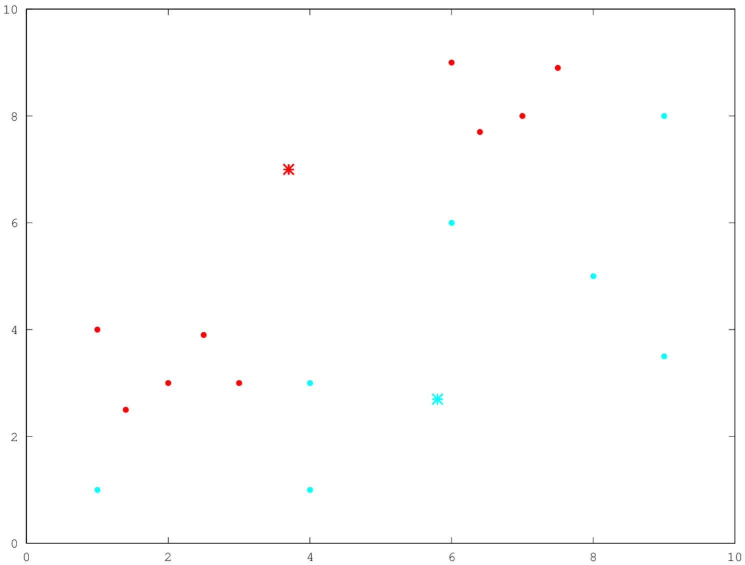


Illustration I

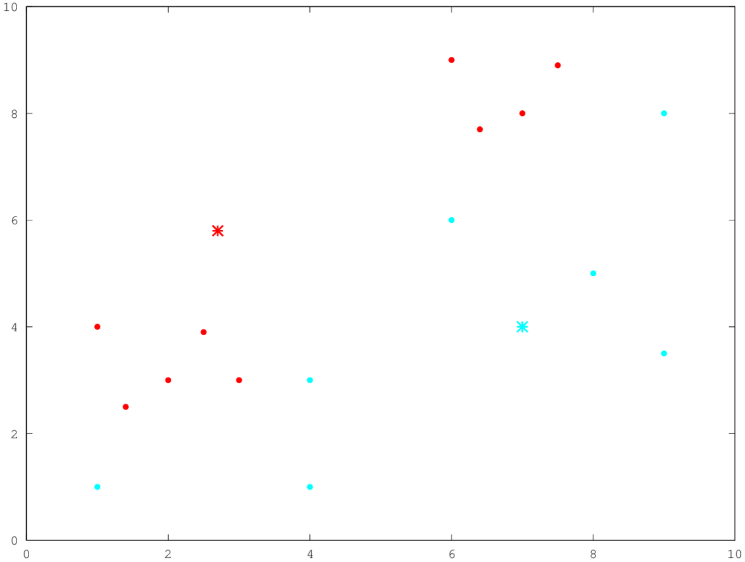


Illustration I

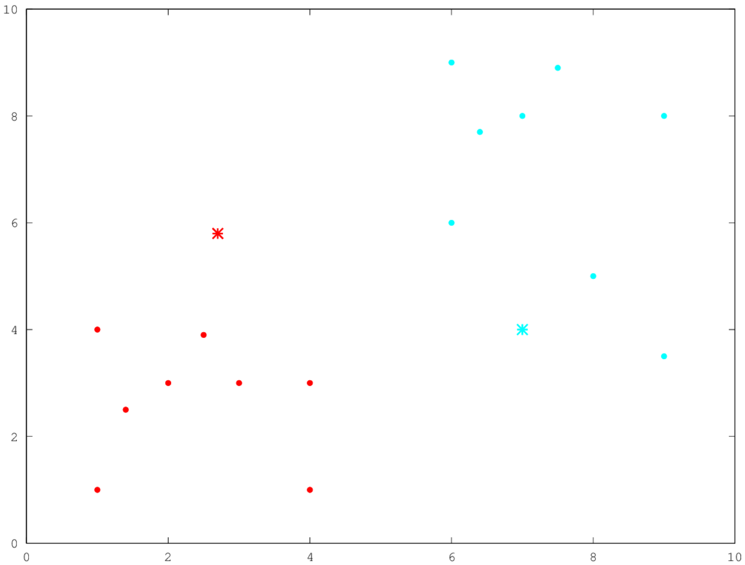
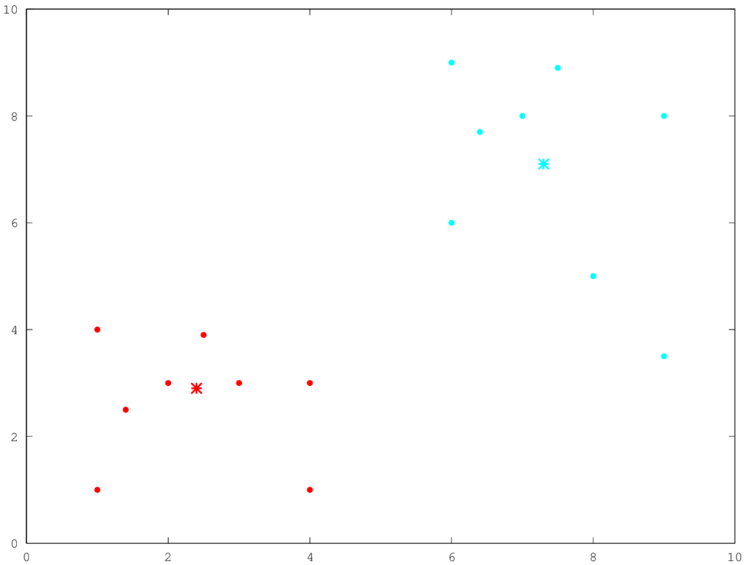


Illustration I



K-means algorithm

Initialize K cluster centroid vectors $\mu_1, \mu_2, ..., \mu_K$
Repeat:

for $i = 1$ to m
$c^{(i)} :=$ index (from 1 to K) of centroid closest to $x^{(i)}$
(distance measured by $||x^{(i)} - \mu_k||^2$)

for $k = 1$ to K
$\mu_k :=$ average of points assigned to cluster k

Practical issues:

- If you end up in an empty cluster, remove it (leaving $K - 1$)
- In case of ties when assigning examples to clusters, resolve in a consistent way, e.g. take lowest index

Cost function

The algorithm is optimizing this cost function:

$$J(c^{(1)}, ..., c^{(m)}; \mu_1, ..., \mu_K) = \frac{1}{m} \sum_{i=1}^{m} ||x^{(i)} - \mu_{c^{(i)}}||^2$$

- That's mean of squared distances between examples and their corresponding centroids

Why is it useful to know about this?

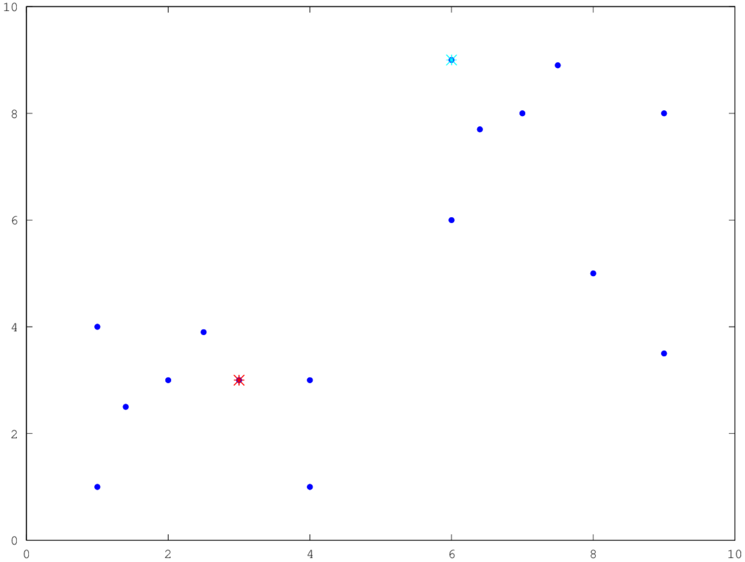
- Cost function should decrease with every iteration
- Observing convergence and avoiding local optima

Importance of initialization

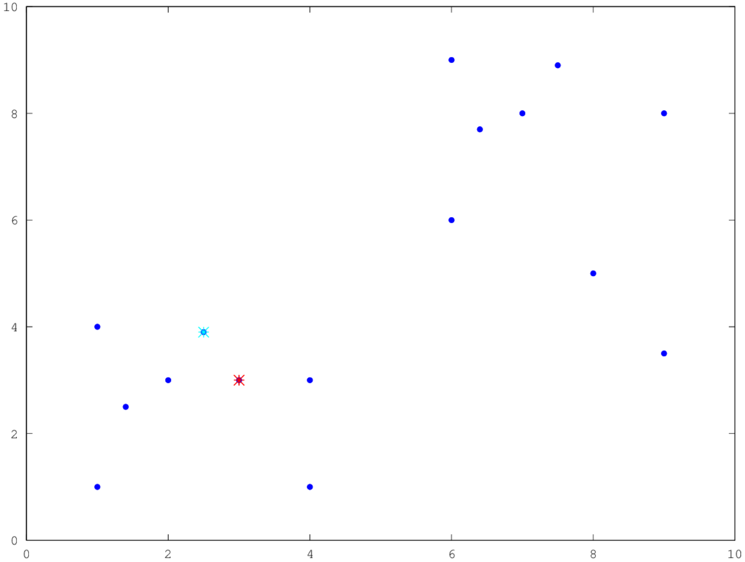
How do we initialize the centroids?

- randomly pick K examples
- set $\mu_1, ..., \mu_K$ equal to these K examples ($\mu_1 = x^{(i)}$, ...)

Successful initialization



Unsuccessful initialization



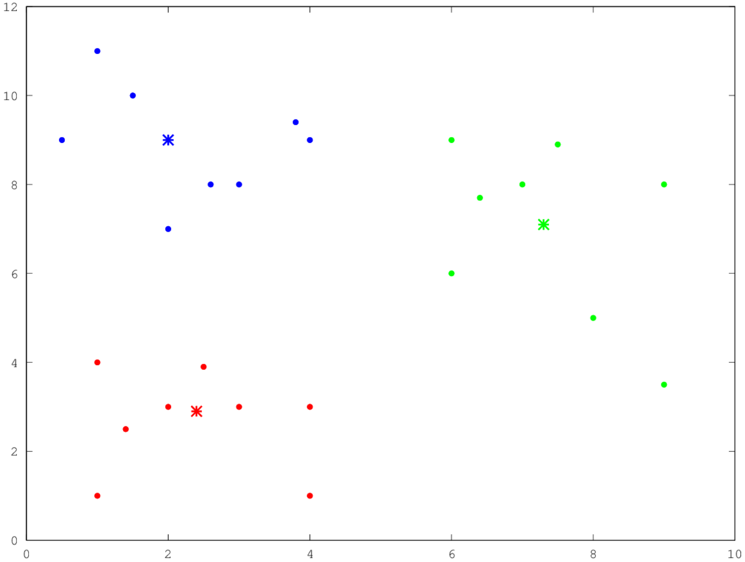
Multiple random initialization

To increase the chance that K-means finds the best possible clustering:

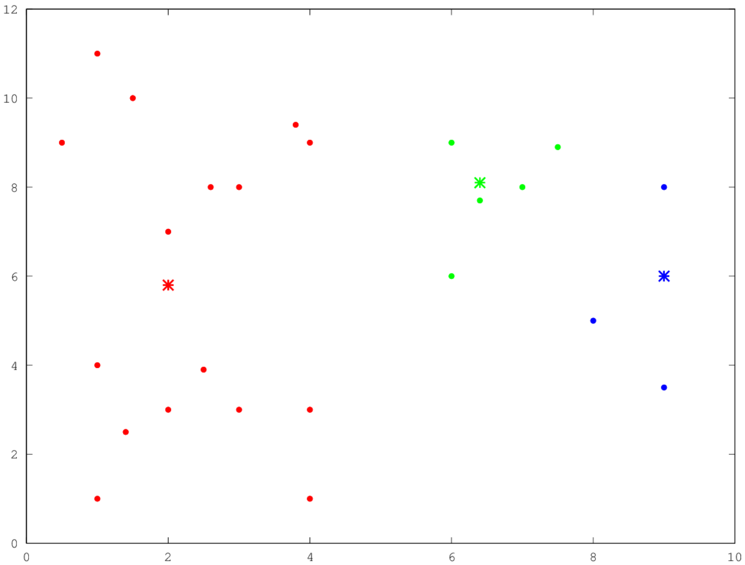
- multiple initialization
- run K-means many times
- choose the best clustering by computing cost function (lowest J)

For large K (around $>$100), multiple random intialization does not make huge differences

Global optimum



Local optimum

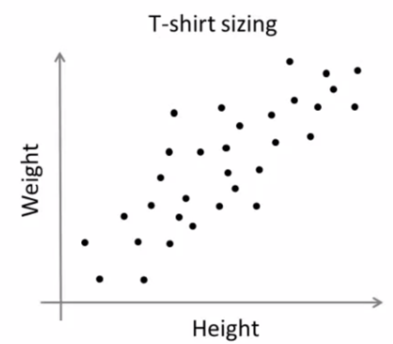


Choosing K I

- No principled way of choosing K
- For that reason, sometimes other clustering techniques are preferred
- K mostly determined manually
- Looking at clusters, often no single truth to the number of clusters

Choosing K II

- Application-motivated



Choosing K III

- "Elbow" / "Knee" method
  - Using cost function J
  - Observe the decrease of J as a function of K
  - Plot
  - Choose K at "elbow"