K-means Clustering

gaurav sood
http://gsood.com
twitter | github

2015

Unsupervised Learning

Unsupervised Learning

- Everything is dimension reduction

Unsupervised Learning

- Everything is dimension reduction
- In supervised learning, labels supervise dimension reduction

Unsupervised Learning

- Everything is dimension reduction
- In supervised learning, labels supervise dimension reduction
- For instance, regression is about finding a low dimensional representation of Y

Unsupervised Learning

- Everything is dimension reduction
- In supervised learning, labels supervise dimension reduction
- For instance, regression is about finding a low dimensional representation of Y
- Supervised Learning $\sim$ Given Apples and Oranges, learn traits of Apples Vs. Oranges

Unsupervised Learning

- Everything is dimension reduction
- In supervised learning, labels supervise dimension reduction
- For instance, regression is about finding a low dimensional representation of Y
- Supervised Learning $\sim$ Given Apples and Oranges, learn traits of Apples Vs. Oranges
- Given a bunch of spherical fruits, optimally describe types of fruits

Ways to Think About Unsupervised Learning

Ways to Think About Unsupervised Learning

- Learning the probability model of the data $p(x_n|x_1, ..., x_{n-1})$

Ways to Think About Unsupervised Learning

- Learning the probability model of the data $p(x_n|x_1, ..., x_{n-1})$
- **Applications:** Outlier detection, Data compression

Ways to Think About Unsupervised Learning

- Learning the probability model of the data $p(x_n|x_1, ..., x_{n-1})$
- **Applications:** Outlier detection, Data compression
- Find rows similar to each other, groups of rows dissimilar to each other

Ways to Think About Unsupervised Learning

- Learning the probability model of the data $p(x_n | x_1, ..., x_{n-1})$
- **Applications:** Outlier detection, Data compression
- Find rows similar to each other, groups of rows dissimilar to each other
- Find columns similar to each other, groups of columns dissimilar to each other

Ways to Think About Unsupervised Learning

- Learning the probability model of the data $p(x_n | x_1, ..., x_{n-1})$
- **Applications:** Outlier detection, Data compression
- Find rows similar to each other, groups of rows dissimilar to each other
- Find columns similar to each other, groups of columns dissimilar to each other
- **Applications:** Group movies by ratings, Segment shoppers

Solutions

- Two kinds of methods:

Solutions

- Two kinds of methods:
  - Principal components analysis

Solutions

- Two kinds of methods:
  - Principal components analysis
  - Clustering

Solutions

- Two kinds of methods:
  - Principal components analysis
  - Clustering
- Clustering looks to partition data into similar subgroups

Solutions

- Two kinds of methods:
  - Principal components analysis
  - Clustering
- Clustering looks to partition data into similar subgroups
- Two popular methods:

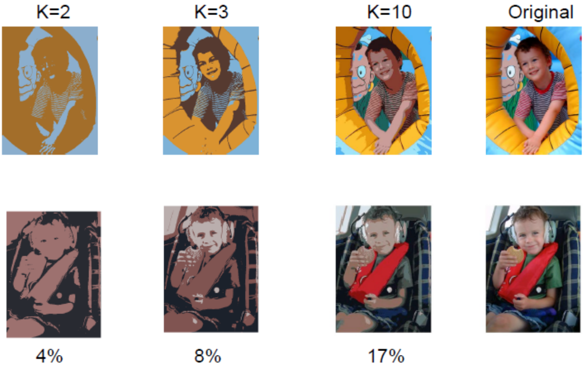
Solutions

- Two kinds of methods:
    - Principal components analysis
    - Clustering
- Clustering looks to partition data into similar subgroups
- Two popular methods:
    - Hierarchical clustering (computationally expensive)

Solutions

- Two kinds of methods:
  - Principal components analysis
  - Clustering
- Clustering looks to partition data into similar subgroups
- Two popular methods:
  - Hierarchical clustering (computationally expensive)
  - k-means clustering (pre-specify k)

Source: Pattern Recognition and Machine Learning



k-Means Clustering

- k-means: Assume that we must split data into k clusters

k-Means Clustering

- k-means: Assume that we must split data into k clusters
  - Each observation belongs to one cluster

k-Means Clustering

- k-means: Assume that we must split data into k clusters
  - Each observation belongs to one cluster
  - No observation belongs to more than one cluster

k-Means Clustering

- k-means: Assume that we must split data into k clusters
    - Each observation belongs to one cluster
    - No observation belongs to more than one cluster
- Find partitioning that minimizes within cluster variation summed over all k clusters

k-Means Clustering

- k-means: Assume that we must split data into k clusters
  - Each observation belongs to one cluster
  - No observation belongs to more than one cluster
- Find partitioning that minimizes within cluster variation summed over all k clusters
- Euclidean distance between observations, sum it over all observations

$$\min._{C_1,\ldots,C_K} \sum_{k=1}^{k} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \tag{1}$$

k-Means (Lloyd's) Algorithm

k-Means (Lloyd's) Algorithm

- Randomly assign observations to 1 of k clusters

k-Means (Lloyd's) Algorithm

- Randomly assign observations to 1 of k clusters
- Iterate:

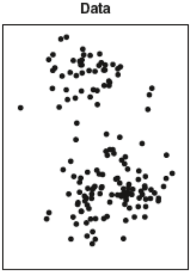
k-Means (Lloyd's) Algorithm

- Randomly assign observations to 1 of k clusters
- Iterate:
  - For each of the k clusters, compute the centroid

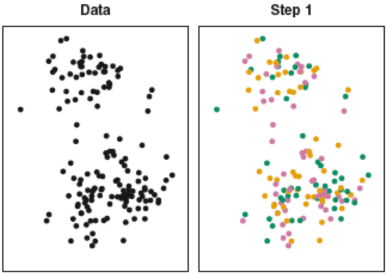
k-Means (Lloyd's) Algorithm

- Randomly assign observations to 1 of k clusters
- Iterate:
  - For each of the k clusters, compute the centroid
  - Assign each observation to cluster whose centroid is closest

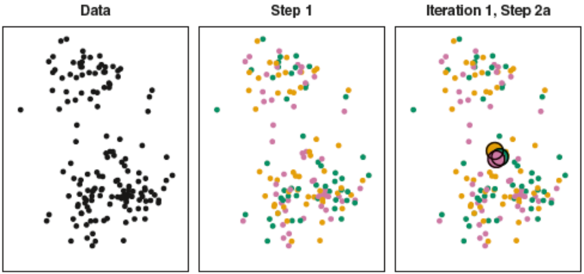
Source: James et al. 2015



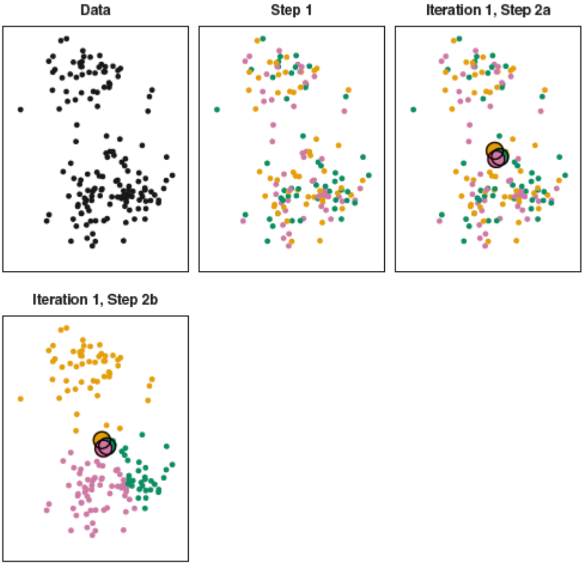
Source: James et al. 2015



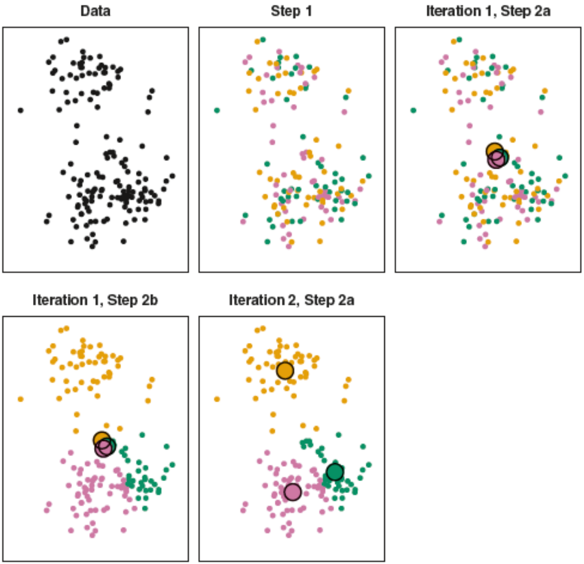
Source: James et al. 2015



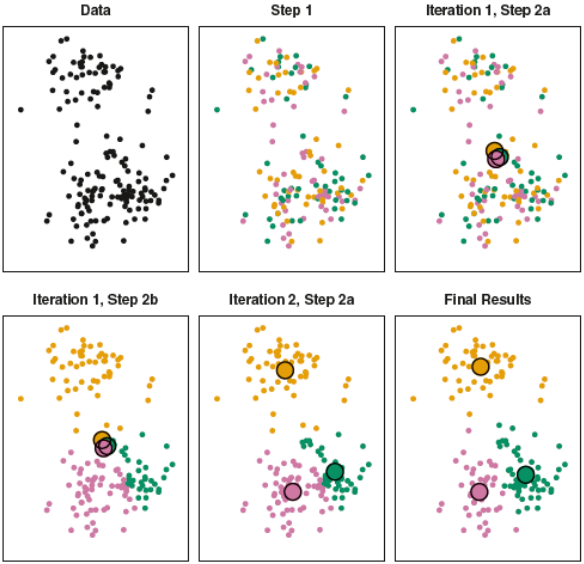
Source: James et al. 2015



Source: James et al. 2015



Source: James et al. 2015



k-Means Algorithm

- Randomly assign observations to 1 of k clusters
- Iterate:
  - For each of the k clusters, compute the centroid
  - Assign each observation to cluster whose centroid is closest

k-Means Algorithm

- Randomly assign observations to 1 of k clusters
- Iterate:
  - For each of the k clusters, compute the centroid
  - Assign each observation to cluster whose centroid is closest
- Why does it work?

k-Means Algorithm

- Randomly assign observations to 1 of k clusters
- Iterate:
  - For each of the k clusters, compute the centroid
  - Assign each observation to cluster whose centroid is closest
- Why does it work?
- It doesn't. Local minima possible.

k-Means Algorithm

- Randomly assign observations to 1 of k clusters
- Iterate:
  - For each of the k clusters, compute the centroid
  - Assign each observation to cluster whose centroid is closest
- Why does it work?
- It doesn't. Local minima possible.
- Initialization:

k-Means Algorithm

- Randomly assign observations to 1 of k clusters
- Iterate:
  - For each of the k clusters, compute the centroid
  - Assign each observation to cluster whose centroid is closest
- Why does it work?
- It doesn't. Local minima possible.
- Initialization:
  - Forgy: Randomly choose k observations and set them as centroids.

k-Means Algorithm

- Randomly assign observations to 1 of k clusters
- Iterate:
  - For each of the k clusters, compute the centroid
  - Assign each observation to cluster whose centroid is closest
- Why does it work?
- It doesn't. Local minima possible.
- Initialization:
  - Forgy: Randomly choose k observations and set them as centroids.
  - Random Partition: Assign each observation randomly to one of the clusters.

k-Means Algorithm

- Randomly assign observations to 1 of k clusters
- Iterate:
  - For each of the k clusters, compute the centroid
  - Assign each observation to cluster whose centroid is closest
- Why does it work?
- It doesn't. Local minima possible.
- Initialization:
  - Forgy: Randomly choose k observations and set them as centroids.
  - Random Partition: Assign each observation randomly to one of the clusters.
  - Run an alternate clustering algorithm on a small sample and use the clusters as initial centroids

k-Means Algorithm

- Randomly assign observations to 1 of k clusters
- Iterate:
  - For each of the k clusters, compute the centroid
  - Assign each observation to cluster whose centroid is closest
- Why does it work?
- It doesn't. Local minima possible.
- Initialization:
  - Forgy: Randomly choose k observations and set them as centroids.
  - Random Partition: Assign each observation randomly to one of the clusters.
  - Run an alternate clustering algorithm on a small sample and use the clusters as initial centroids
  - Pick dispersed points as centroids. For e.g. k-means++ and variations of it.

Distance between clusters

- Complete Linkage
  Farthest distance between points in clusters

Distance between clusters

- Complete Linkage
  Farthest distance between points in clusters

- Single
  Closest pair

Distance between clusters

- Complete Linkage
  Farthest distance between points in clusters
- Single
  Closest pair
- Average
  All pairs, and then take the average

Distance between clusters

- Complete Linkage
  Farthest distance between points in clusters
- Single
  Closest pair
- Average
  All pairs, and then take the average
- Centroid
  Has problems called inversions
  Used in Genomics

Distance between clusters

- Complete Linkage
  Farthest distance between points in clusters
- Single
  Closest pair
- Average
  All pairs, and then take the average
- Centroid
  Has problems called inversions
  Used in Genomics
- Complete and Average most commonly used

Practical Issues

- Choice of Similarity Measure

Practical Issues

- Choice of Similarity Measure
  - Scaling Matters

Practical Issues

- Choice of Similarity Measure
  - Scaling Matters
  - Jaccard — can be gotten quickly by minhashing via LSH
    Distance

Practical Issues

- Choice of Similarity Measure
  - Scaling Matters
  - Jaccard — can be gotten quickly by minhashing via LSH Distance
  - Correlation based measures (+/- may matter)

Practical Issues

- Choice of Similarity Measure
  - Scaling Matters
  - Jaccard — can be gotten quickly by minhashing via LSH
    Distance
  - Correlation based measures (+/- may matter)

- High dimensional data. Solutions e.g. DANN

Practical Issues

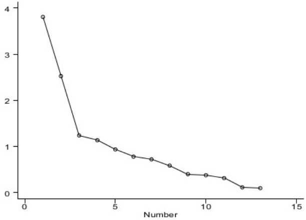
- Choice of Similarity Measure
    - Scaling Matters
    - Jaccard — can be gotten quickly by minhashing via LSH Distance
    - Correlation based measures (+/- may matter)
- High dimensional data. Solutions e.g. DANN
- Choosing k:

Practical Issues

- Choice of Similarity Measure
  - Scaling Matters
  - Jaccard — can be gotten quickly by minhashing via LSH Distance
  - Correlation based measures (+/- may matter)
- High dimensional data. Solutions e.g. DANN
- Choosing k:
  - Calculate average distance to centroid for multiple k

Practical Issues

- Choice of Similarity Measure
  - Scaling Matters
  - Jaccard — can be gotten quickly by minhashing via LSH Distance
  - Correlation based measures (+/- may matter)
- High dimensional data. Solutions e.g. DANN
- Choosing k:
  - Calculate average distance to centroid for multiple k
  - Plot them, look for the *knee*



A(N)alyst choose k contd.

- Calinski-Harabasz (CH) Index:

A(N)alyst choose k contd.

- Calinski-Harabasz (CH) Index:
  - Between Cluster, $B = \sum_1^k n_k \|X_k - \bar{X}\|^2$

A(N)alyst choose k contd.

- Calinski-Harabasz (CH) Index:
  - Between Cluster, $B = \sum_1^k n_k \|X_k - \bar{X}\|^2$
  - Within Cluster, $W = \sum_1^k \|X_i - \bar{X}_k\|^2$

A(N)alyst choose k contd.

- Calinski-Harabasz (CH) Index:
  - Between Cluster, $B = \sum_1^k n_k \|X_k - \bar{X}\|^2$
  - Within Cluster, $W = \sum_1^k \|X_i - \bar{X}_k\|^2$
  - Maximize Between Cluster Variation, Minimize Within Cluster Variation

A(N)alyst choose k contd.

- Calinski-Harabasz (CH) Index:
  - Between Cluster, $B = \sum_1^k n_k \|X_k - \bar{X}\|^2$
  - Within Cluster, $W = \sum_1^k \|X_i - \bar{X}_k\|^2$
  - Maximize Between Cluster Variation, Minimize Within Cluster Variation
  - CH(K) $= \frac{B(K)}{(K-1)} \frac{n-K}{W(K)}$

A(N)alyst choose k contd.

- Calinski-Harabasz (CH) Index:
  - Between Cluster, $B = \sum_1^k n_k \|X_k - \bar{X}\|^2$
  - Within Cluster, $W = \sum_1^k \|X_i - \bar{X}_k\|^2$
  - Maximize Between Cluster Variation, Minimize Within Cluster Variation
  - CH(K) $= \frac{B(K)}{(K-1)} \frac{n-K}{W(K)}$
- Gap Statistic (Tibshirani):
  - Compare observed $W(K)$ to $W_{\text{unif}}(K)$

A(N)alyst choose k contd.

- Calinski-Harabasz (CH) Index:
  - Between Cluster, $B = \sum_1^k n_k \|X_k - \bar{X}\|^2$
  - Within Cluster, $W = \sum_1^k \|X_i - \bar{X}_k\|^2$
  - Maximize Between Cluster Variation, Minimize Within Cluster Variation
  - CH(K) $= \frac{B(K)}{(K-1)} \frac{n-K}{W(K)}$
- Gap Statistic (Tibshirani):
  - Compare observed $W(K)$ to $W_{\text{unif}}(K)$
  - $\text{GAP}(K) = \log W(K) - \log W_{\text{unif}}(K)$

A(N)alyst choose k contd.

- Calinski-Harabasz (CH) Index:
  - Between Cluster, $B = \sum_1^k n_k \|X_k - \bar{X}\|^2$
  - Within Cluster, $W = \sum_1^k \|X_i - \bar{X}_k\|^2$
  - Maximize Between Cluster Variation, Minimize Within Cluster Variation
  - CH(K) $= \frac{B(K)}{(K-1)} \frac{n-K}{W(K)}$
- Gap Statistic (Tibshirani):
  - Compare observed $W(K)$ to $W_{\text{unif}}(K)$
  - $\text{GAP}(K) = \log W(K) - \log W_{\text{unif}}(K)$
  - Calculate $W_{\text{unif}}(K)$ by simulation.

Running Time

- $O(kn)$ for each iteration.

Running Time

- $O(kn)$ for each iteration.
- But total iterations can be a lot, and not bounded.

Running Time

- $O(kn)$ for each iteration.
- But total iterations can be a lot, and not bounded.
- But in practice, polynomial running time.

Running Time

- $O(kn)$ for each iteration.
- But total iterations can be a lot, and not bounded.
- But in practice, polynomial running time.
- Big (Long) Data Solutions:

Running Time

- $O(kn)$ for each iteration.
- But total iterations can be a lot, and not bounded.
- But in practice, polynomial running time.
- Big (Long) Data Solutions:
  - Bradley-Fayyad-Reina (BFR)

Running Time

- $O(kn)$ for each iteration.
- But total iterations can be a lot, and not bounded.
- But in practice, polynomial running time.
- Big (Long) Data Solutions:
    - Bradley-Fayyad-Reina (BFR)
    - CURE

Running Time

- $O(kn)$ for each iteration.
- But total iterations can be a lot, and not bounded.
- But in practice, polynomial running time.
- Big (Long) Data Solutions:
  - Bradley-Fayyad-Reina (BFR)
  - CURE
- BFR
  - Assumes clusters are normally distributed around a centroid in Euclidean space.

Running Time

- $O(kn)$ for each iteration.
- But total iterations can be a lot, and not bounded.
- But in practice, polynomial running time.
- Big (Long) Data Solutions:
  - Bradley-Fayyad-Reina (BFR)
  - CURE
- BFR
  - Assumes clusters are normally distributed around a centroid in Euclidean space.
  - Exploit that to quantify likelihood point belongs to a cluster