TECHNISCHE UNIVERSITÄT DRESDEN

SEMANTIC COMPUTING

Lecture 6: Unsupervised Machine Learning

Dagmar Gromann

International Center For Computational Logic

TU Dresden, 23 November 2018

Overview

- Unsupervised Machine Learning overview
- Association Rules
- Hierarchical Clustering
- K-means
- Spectral Clustering

Unsupervised Machine Learning

Definition: Unsupervised Machine Learning

Definition

Unsupervised machine learning is a set of algorithms that attempt to **directly infer the distributional properties of input data (X)** without the availability of corresponding output variables, that is, no correct answer or degree of error for each observation is available.

Challenge

Since there is no direct measure of success with such algorithms, it is **difficult to estimate the validity of the inferences drawn** from the output. In other words, the effectiveness of the algorithm is difficult to verify directly (as we can with supervised methods).

Common Unsupervised Learning Problems

- **Clustering**: problem of attempting to find the natural partitions of patterns in the data (also called “data segmentation” or “instance-based learning”)
- **Association**: problem of discovering association rules that describe large proportions of the data or in other words discover simple rules that describe regions of high density, e.g. people that buy potatoes and onions tend to buy hamburger meat (“market basket” analysis)

Association Rules

Association Rule Mining

Goal

The goal is to find rules using the following concepts:

- coverage = number of instances that are predicted correctly
- accuracy = coverage expressed as proportion of the number of instances to which the rule applies
- item = attribute-value pairs, such as $outlook = sunny$
- item sets = attribute-value pairs with a prespecified minimum coverage

Two major steps for generating rules:

1. generating item sets with the specified minimum coverage
2. determining the rules with the specified minimum accuracy for each item set

Step 1: Generating Item Sets

1. generate all one-item sets with the given minimum coverage
2. generate candidate two-item sets based on the one-item sets
3. continue “Second” for all further item sets
4. Candidate sets with less than minimum coverage are removed from the storage, e.g. hash table

Given five three-item sets $(ABC), (ABD), (ACD), (ACE), (BCD)$
where $A : outlook = sunny$ for the weather
then: $(ABCD)$ is the union of the first two sets and is a candidate four-item set

Example Data Set

	One-item sets	Two-item sets	Three-item sets	Four-item sets
1	outlook = sunny (5)	outlook = sunny temperature = mild (2)	outlook = sunny temperature = hot humidity = high	outlook = sunny temperature = hot humidity = high play = no (2)
...				
3	outlook = rainy (5)	outlook = sunny humidity = normal (2)	outlook = sunny temperature = hot play = yes (2)	outlook = overcast temperature = hot windy = false play = yes (2)
...				
6	temperature = hot (4)	outlook = sunny windy = false (3)	outlook = sunny windy = false play = no (2)	temperature = cool humidity = normal windy = false play = yes (2)

Table: Decision on playing tennis given a certain weather

Source: Witten, I. H., Frank, E., Hall, M. A., and Pal, C. J. (2016). Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann.

Step 2: Generating Rules

Generate rules for each item set based on the specified minimum accuracy. Some basics to speed up computation:

Idea

If we have two items in the antecedent (left side) and two items in the consequent (right side), this double-consequent rule holds with the given minimum coverage and accuracy, then both single consequent rules from the same item set must also hold.

Double-consequent rule:
If windy = false and play = no then outlook = sunny and humidity = high
Single-consequent rules:
If humidity = high and windy = false and play = no then outlook = sunny
If outlook = sunny and windy = false and play = no then humidity = high

Example Association Rules

After item sets with the predefined coverage have been specified, they are turned into association rules with a predefined minimum accuracy.

Some item sets produce several rules, others produce none. For instance, the item set humidy = normal, windy = false, play = yes can produce the following rules:

	Association Rule	Coverage	Accuracy
1	If humidity = normal and windy = false and play = yes	4/4	100%
2	If humidity = normal and play = yes then windy = false	4/6	67%
3	If windy = false and play = yes then humidity = normal	4/6	67%
4	If humidity = normal then windy = false and play = yes	4/7	57%
...			

(Dis-)Advantages of Association Rules

Advantages:

- efficient on large datasets
- especially performant on binary attributes (present or absent)

Disadvantages:

- makes one pass through the entire dataset for each different size of item sets - reads it to main memory
- memory use depends on minimum coverage specified (affects the number of passes needed)

Clustering

Introduction to Clustering

Clustering

- places objects that are more closely related (according to a certain similarity measure) in one group (cluster) and assign dissimilar objects to another group
  - objects can be related to one another by some kind of measurements
  - objects can be related by some kind of relation (grammatical, semantic, lexical, etc.)
- is used for exploratory data analysis
- is used for automatically organizing data
- can be used to grasp of hidden structures in data

Similarity metrics

The decision of merging clusters is based on the measure of their closeness, the similarity measure of data points int he cluster. For instance;

- **Manhattan distance**: $d(a,b) = \sum_{i=1}^{n} |a_i - b_i|$
- **Euclidean distance**: $||a-b||_2 = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2}$
- **Maximum distance**: $d(a,b) = max|a_i - b_i|$
- **Jaccard distance**: $d_J(a,b) = 1 - J(a,b) = \frac{|a \cup b| - |a \cap b|}{|a \cup b|}$
- **Cosine similarity**: $cos(a,b) = \frac{a \cdot b}{||a|| \cdot ||b||} = \frac{\sum_{i=1}^{n} a_i b_i}{\sqrt{\sum_{i=1}^{n} a_i^2}\sqrt{\sum_{i=1}^{n} b_i^2}}$
- **Edit distance**: $d(a,b) = |a| + |b| - 2|LCS(x,y)|$
  LCS = Longest Common Subsequence

How to choose the metric?

Higher powers increase the influence of large difference at the expense of small differences. Think of actual instances in the dataset and would it means if they are separated by a certain distance.

- Euclidean distance: effects of large distance can dwarf effect of small distance; solution: normalize all attribute values to lie between 0 and 1
- Manhattan distance: every attribute might be off a little

Applications

- grouping search results by topic
- clustering news articles
- clustering social media users by interest
- grouping protein sequences by their function
- clustering customers by their purchasing habits
- many more

Basic Distinction

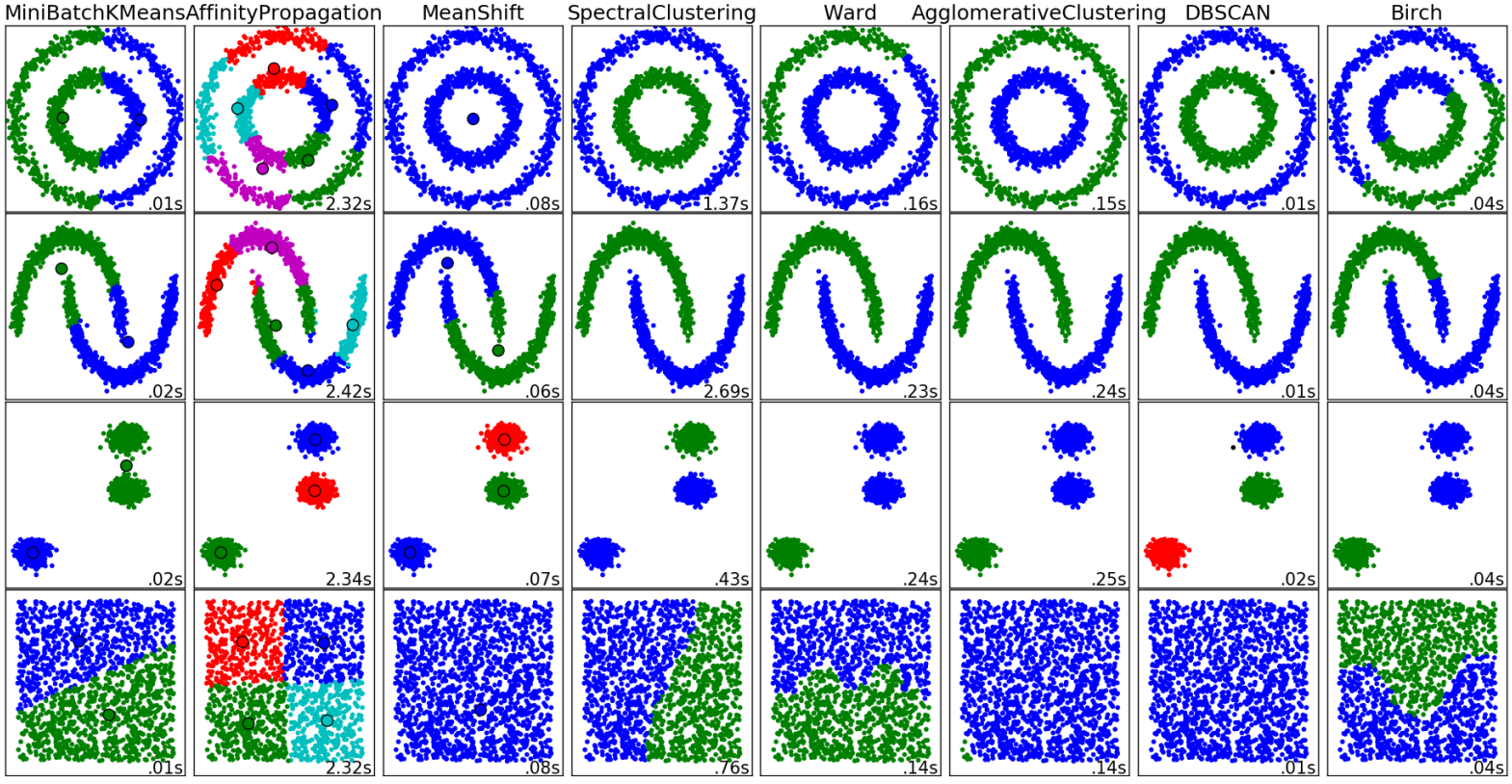
- **Hard clustering**: each data point belongs to exactly one cluster or no cluster at all
- **Soft clustering**: each data point is assigned a probability of belonging to one cluster and degrees of membership/multiple memberships are possible

Overview of Clustering Algorithms

Name	Definition	Main Example	Characteristics
Connectivity Models	data points closer in data space are more similar than those far away	hierarchical clustering	easy to interpret but do not scale well
Centroid Models	iterative where similarity is interpreted as proximity of data point to centroid	k-means	provide final number of clusters
Distribution Models	based on probability of data points in a cluster belonging to the same distribution (e.g. Gaussian)	Expectation-maximization (EM) algorithm	frequent problems of overfitting
Density Models	isolate different density regions in data as basis for clustering	Density-Based Spatial Clustering of Applications with Noise (DBSCAN)	not good on high-dimensional data or clusters with varying densities

Visualization

First intuition on differences; last row shows a homogeneous dataset without any good clustering for comparison.



Source: https://scikit-learn.org/0.18/auto_examples/cluster/plot_cluster_comparison.html

Cluster Organization

- **Hierarchical**: produces a tree of clusters where each node represents a subgroup of the parent node (usually only with hard clustering)
- **Non-hierarchical**: relations between clusters are usually left undetermined; most algorithms iteratively improve clusters using a reallocation scheme (also called flat clustering); (can be hard or soft)

Hierarchical Clustering

Two Types of Hierarchical Clustering

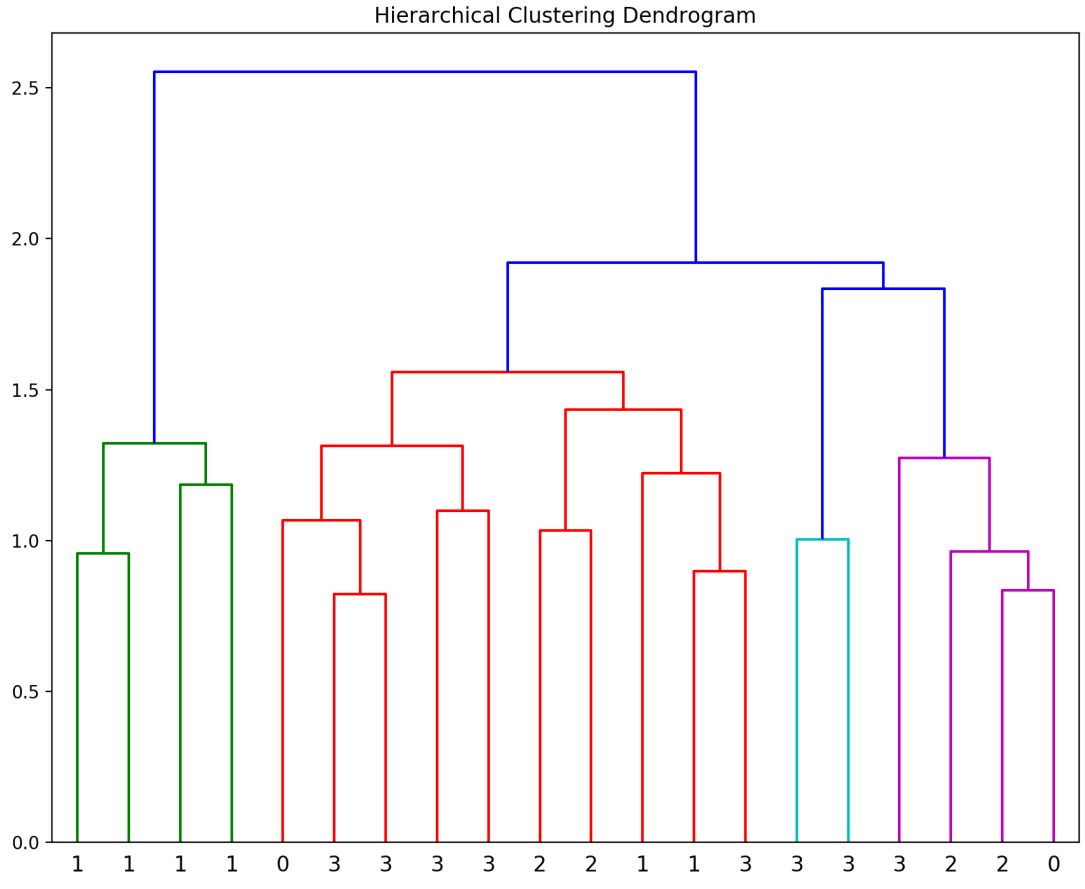
Bottom-up (Agglomerative) Clustering

Start with the individual objects; greedily put objects with maximum similarity (minimum distance) into a cluster; continue until each individual objects is assigned to one specific cluster

Top-down (Divisive) Clustering

Start with all objects in one big cluster; greedily split the cluster into two; assign objects to each group in a way to maximize within-group similarity; continue splitting the clusters until there is a cluster with only one object or the desired numbers of clusters is achieved.

Visualization: Dendrogram



Small subset of the 20 newsgroups dataset of sklearn:
0 = alt.atheism
1 = comp.graphics
2 = sci.med
3 = soc.religion.christian

k-means

Euclidean k-means Clustering

Definition

Centroid model that tries to divide a set of N samples X into K disjoint clusters C of equal variance, minimizing the pairwise squared deviations of points in the same cluster. Each cluster is described by the mean μ_i of the samples in the cluster, commonly called the cluster centroids.

Input: a set of N data points $x_1, x_2, ..., x_n \in R^d$; target number of K clusters

Output: K representatives of $c_1, c_2, ..., c_k \in R^d$

Objective: choose $c_1, c_2, ..., c_k \in R^d$ to minimize $\sum_{i=1}^{n} min_{j \in 1,...,k} \|x^i - c_j\|^2$

Challenge of k-means

- Computational complexity: NP hard even for k=2 and d=2 (easy cases: k=1, d=1)
- in high-dimensional space Euclidean distance becomes inflated (best to run dimensionality reduction algorithm first, such as PCA)
- assumes clusters to be convex and isotropic

Lloyd's method

Publication: Lloyd, S. (1982). Least squares quantization in PCM. IEEE transactions on information theory, 28(2), 129-137.

Input: a set of N data points $x_1, x_2, ..., x_n \in R^d$; target number of K clusters

Initialize: initialize $c_1, c_2, ..., c_k \in R^d$ and assign data points to clusters $C_1, C_2, ..., C_k$

Repeat: repeat the following two steps until there is no further change in the cost or a certain threshold is met (until the centroids do not move significantly):

- re-assign each data point to its nearest cluster centroid
- re-compute the centroid (taking mean value of all samples assigned to the cluster)

K-means++ Initialization

Initialize centers to be generally distant from each other; let the distance between a data point and its center be $D(x)$ than k-means++ chooses the center proportional to $D^2(x)$

- Choose one center c_1 at random
- For $j = 2, ..., k$
  - Pick c_j among $x_1, x_2, ..., x_n$, choosing $x \in X$ with a probability of $\frac{D(x)^2}{\sum_{x \in X} D(x)^2}$ (this "weighting" is called D^2 weighting)
- Once the centroids have been placed proceed as with normal k-means

How to find k?

- use hierarchical clustering for data exploration
- elbow method; compare percentage of variance explained by the clusters with the number of clusters - choose the number of clusters so that adding one more cluster does not improve the modeling of the data
- cross-validation: set one partition aside as test set, compute clustering for other v-1 training sets, v values are averaged for each alternative number of clusters and the one is chosen where the reduction in the objective function is small
- ...

K-means vs. Hierarchical Clustering

- k-means works well on large number of samples and has been used across applications (even for Big Data), while the computational complexity of hierarchical clustering ($O(n^2)$) does not allow for Big Data applications
- k-means might produce different results each time you run it, while hierarchical clustering results are reproducible
- k-means requires number of clusters, while this can be read from the dendrogram with hierarchical clustering

Spectral Clustering

Spectral Clustering

Spectral clustering is a graph-based clustering method that is very promising due to its high efficiency and generally good performance. The algorithm requires the following steps:

1. Build a similarity graph between our x input objects to cluster
2. Compute the first k eigenvectors of its graph Laplacian matrix to define a feature vector for each object
3. Run k-means on these features to separate objects into classes

Full tutorial: `https://arxiv.org/pdf/0711.0189.pdf`

Build a similarity graph

There are different ways to build a similarity graph:

- choose a similarity metric to weight the edges
- ϵ-neighborhood graph: connection inside a ball of radios ϵ (a real value)
- k-nearest neighbor graph: connection to k-nearest neighbors where k is an integer number
- turn graph into weighted adjacency matrix
  $W = (w_{ij})\ i, j = 1, \ldots, n$

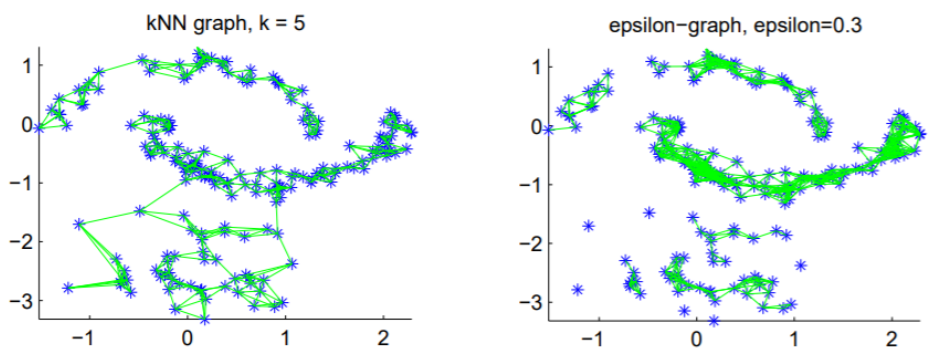


Image source:
https://towardsdatascience.com/spectral-clustering-for-beginners-d08b7d25b4d8

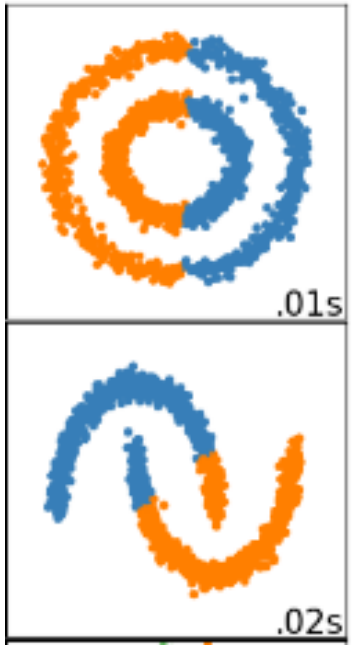
Build a graph Laplacian

- Unnormalized Graph Laplacian L is calculated as $L = D - W$ where D is the diagonal degree matrix (degree $d_i = \sum_{i=1}^{n} w_{ij}$) and W is the weighted adjacency matrix
- Normalized graph Laplacin L_{rw} is calculated as $L_{rw} = D^{-}1L = I - D^{-}1W$ where rw stands for random walk and I for identity matrix
- Compute the first k eigenvectors $v_1, \ldots, v_k$ of L; because the multiplicity k of the eigenvalue 0 of L equals the number of connected components in the adjacency matrix
- Generate a matrix of the eigenvectors, of which the rows will be clustered using k-means

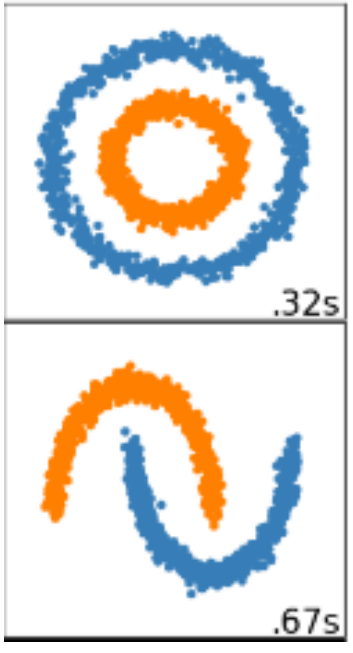
Partition the Laplacian graph

Apply k-means to Laplacian eigenvectors to find cluster; spectral clustering is superior

k-means



Spectral clustering



Source: https://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html

Spectral Clustering: Challenges

- Choice of k
  - k can be chosen in many different ways (or tested
  - for spectral clustering: most stable if k maximizes the eige gap (difference between consecutive eigenvalues)

$$\delta_k = |\lambda - \lambda_{k-1}|$$

- Choice of similarity measure can have a strong impact
- Normalize vs. unnormalized graph Laplacian matrices can affect final outcome

Review of Lecture 6

- What is unsupervised machine learning?
- What are typical problems addressed by unsupervised learning?
- Which hierarchical clustering method do you know? How do they differ?
- On which similarity metric is standard k-means based?
- How does k-means work as opposed to k-means++?
- What is spectral clustering?
- How can the number of clusters be found? In spectral clustering?
- Which algorithm would you prefer if reproducibility of results is important?