Machine Learning & Knowledge Extraction

DR KONSTANTINOS KARAMPIDIS

Πληροφορίες Μαθήματος

- Ωράριο:
  - Θεωρία: **Κάθε Τρίτη 09:00-13:00 – Αίθουσα 207**
  - Εργαστήριο: **13:00-14:00 ΕΡΓ6 (σύμφωνα με το πρόγραμμα που είναι αναρτημένο στο eclass)**
- Εργασίες
  - **1 project – Ομάδες έως 2 ατόμων – 80%**
  - **Εργαστηριακές ασκήσεις – 20%**
- Προαπαιτούμενα: Κανένα

Περιεχόμενο Μαθήματος

- Εισαγωγή στη Μηχανική Μάθηση - τι είναι, γιατί μας ενδιαφέρει, παραδείγματα προβλημάτων, η μηχανική μάθηση ως αναζήτηση, υπόθεση επαγωγικής μάθησης
- Επεξεργασία εισόδου – Μείωση διαστατικότητας- Αξιόλογηση
- Μέθοδοι επιβλεπόμενης μάθησης
- **Νευρωνικά Δίκτυα**
- Εξελικτική Μάθηση – Γενετικοί Αλγόριθμοι
- **Μηχανική Μάθηση Βασιζόμενη σε Κανόνες**
- Ενισχυτική Μάθηση
- Μάθηση Αναπαράστασης
- Εξόρυξη Δεδομένων

Unsupervised learning

Unsupervised learning can be compared to learning which takes place in the human brain while learning new things. It can be defined as: "Unsupervised learning is a type of machine learning in which models are trained using unlabeled dataset and are allowed to act on that data without any supervision."

Unsupervised learning cannot be directly applied to a regression or classification problem because unlike supervised learning, we have the input data but no corresponding output data.

The goal of unsupervised learning is to find the underlying structure of dataset, group that data according to similarities, and represent that dataset in a compressed format

Unsupervised learning - Example

Suppose the unsupervised learning algorithm is given an input dataset containing images of different types of cats and dogs. The algorithm is never trained upon the given dataset, which means it does not have any idea about the features of the dataset. The task of the unsupervised learning algorithm is to identify the image features on their own. Unsupervised learning algorithm will perform this task by clustering the image dataset into the groups according to similarities between images.

Unsupervised learning - Example

Why use Unsupervised Learning ?

- Unsupervised learning is helpful for finding useful insights from the data.
- Unsupervised learning is much similar as a human learns to think by their own experiences, which makes it closer to the real AI.
- In real-world, we do not always have input data with the corresponding output.

Unsupervised learning - Types

The unsupervised learning algorithm can be further categorized into two types of problems:

- Clustering
- Association

Unsupervised learning - Types

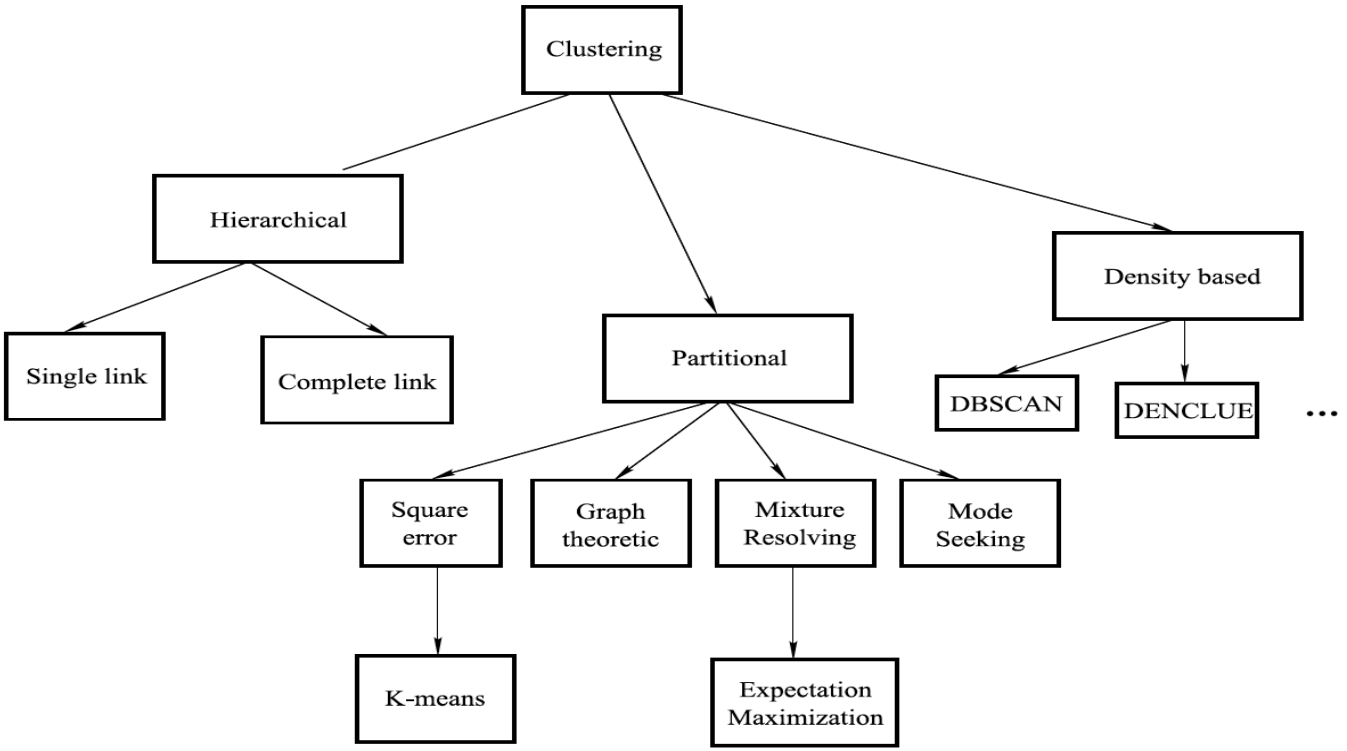
Clustering is a method of grouping the objects into clusters such that objects with most similarities remains into a group and has less or no similarities with the objects of another group. Cluster analysis finds the commonalities between the data objects and categorizes them as per the presence and absence of those commonalities.

An association rule is an unsupervised learning method which is used for finding the relationships between variables in the large database. It determines the set of items that occurs together in the dataset. Association rule makes marketing strategy more effective. Such as people who buy X item (suppose a bread) are also tend to purchase Y (Butter/Jam) item.

Unsupervised learning - Algorithms

- Partitional clustering (K-means)
- Hierarchal clustering
- Neural Networks (SOM)
- Density-based clustering (DBSCAN)
- Association Rule Learning (Apriori algorithm)

Unsupervised learning - Types



https://journal.hep.com.cn/fcs/EN/article/downloadArticleFile.do?attachType=PDF&id=25589

Unsupervised learning - Algorithms

Advantages of Unsupervised Learning

- Unsupervised learning is used for more complex tasks as compared to supervised learning because, in unsupervised learning, we don't have labeled input data.
- Unsupervised learning is preferable as it is easy to get unlabeled data in comparison to labeled data.

Disadvantages of Unsupervised Learning

- Unsupervised learning is intrinsically more difficult than supervised learning as it does not have corresponding output.
- The result of the unsupervised learning algorithm might be less accurate as input data are not labeled, and algorithms do not know the exact output in advance.

K-means

One of the most used clustering algorithm is k-means.

It allows to group the data according to the existing similarities among them in k clusters, given as input to the algorithm.

The k-means algorithm clusters data by trying to separate samples in n groups of equal variance, minimizing a criterion known as the inertia or within-cluster sum-of-squares.

This algorithm requires the number of clusters to be specified. It scales well to large number of samples and has been used across a large range of application areas in many different fields.

K-means

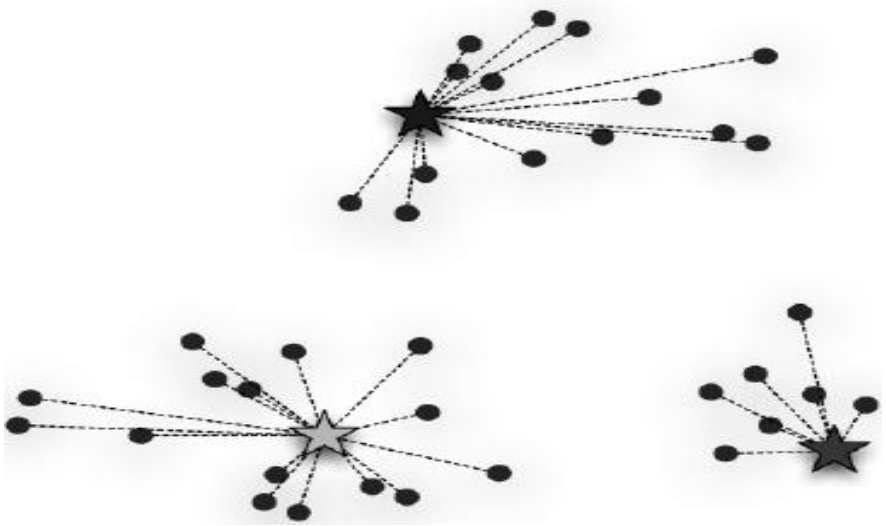
The k-means algorithm divides a set of N samples x into disjoint clusters C, each described by the mean μ_j of the samples in the cluster. The means are commonly called the cluster "centroids"; note that they are not, in general, points from X, although they live in the same space. The K-means algorithm aims to choose centroids that minimize the inertia, or within-cluster sum of squared criterion:

$$\sum_{i=0}^{n} \boldsymbol{min}(\left\|\boldsymbol{x_i} - \boldsymbol{m_j}\right\|^2)$$

K-means

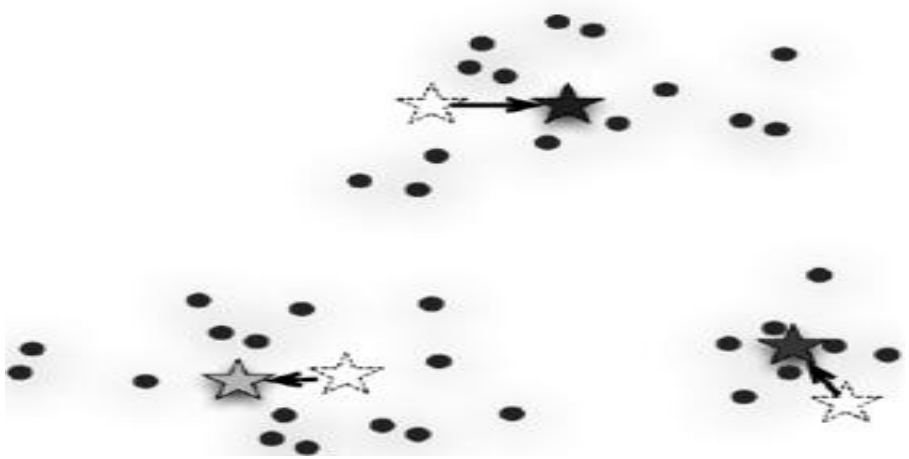
Suppose we want to divide the data into 3 clusters (k = 3). The steps to execute the algorithm are:

- Initially, the algorithm selects 3 random values and defines them as centroids for each one of the 3 clusters.



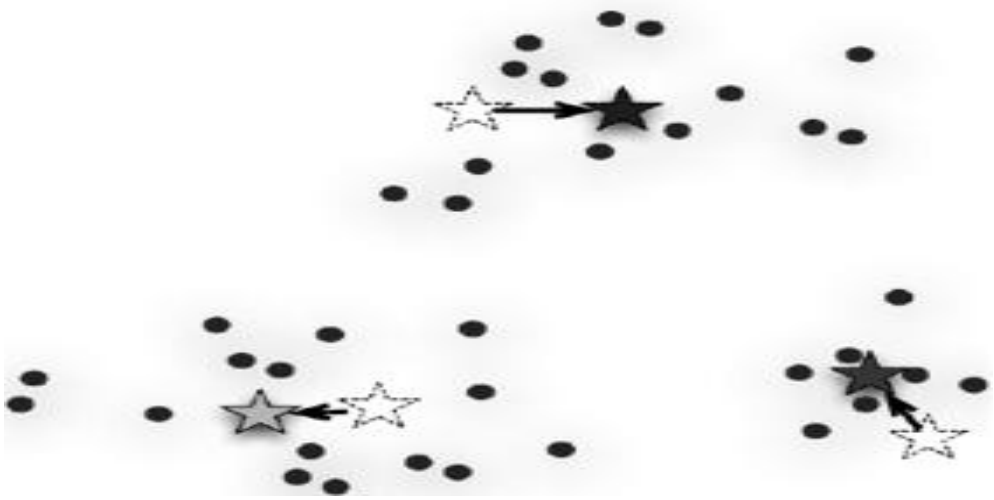
K-means

- The algorithm takes each instance from the data set, calculates its distance from the centroid of each cluster, and finally places it in the cluster with the minimum distance.
- Then for each one of the clusters the algorithm calculates again the centroid based on the average of all instances belonging to the particular cluster.



K-means

The algorithm runs repeatedly until there is no change in clusters centers or a termination condition is true.



K-means - Example

Suppose we have 5 people and for each of them we know two features (height and weight). We want to group them into k=2 clusters. Our dataset will look like this:

	Height (H)	Weight (W)
Person 1	167	55
Person 2	120	32
Person 3	113	33
Person 4	175	76
Person 5	108	25

K-means

First of all, we have to initialize the value of the centroids for our clusters. For instance, let's choose Person 2 and Person 3 as the two centroids c_1 and c_2, so that c_1=(120,32) and c_2=(113,33). Now we compute the Euclidian distance between each of the two centroids and each point in the data.

	Distance of object from *c1*	**Distance of object from** *c2*
Person 1	52.3	58.3
Person 2	0	7.1
Person 3	7.1	0
Person 4	70.4	75.4
Person 5	13.9	9.4

K-means

At this point, we will assign each object to the cluster it is closer to (that is taking the minimum between the two computed distances for each object). We can then arrange the points as follows:

Person 1 → cluster 1

Person 2 → cluster 1

Person 3 → cluster 2

Person 4 → cluster 1

Person 5→ cluster 2

K-means

Let's iterate, which means to redefine the centroids by calculating the mean of the members of each of the two clusters.

So c'_1 = ((167+120+175)/3, (55+32+76)/3) = (154, 54.3) and

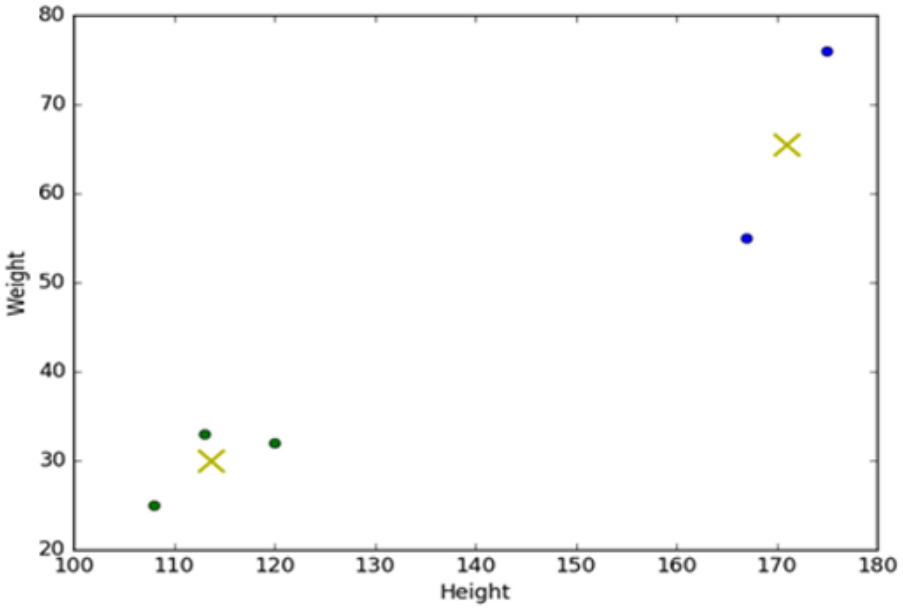
c'_2 = ((113+108)/2, (33+25)/2) = (110.5, 29)

Then, we calculate the distances again and re-assign the points to the new centroids.

We repeat this process until the centroids don't move anymore (or the difference between them is under a certain small threshold).

K-means

In our case, the result we get is given in the figure below. You can see the two different clusters labelled with two different colours and the position of the centroids, given by the crosses.



Agglomerative clustering

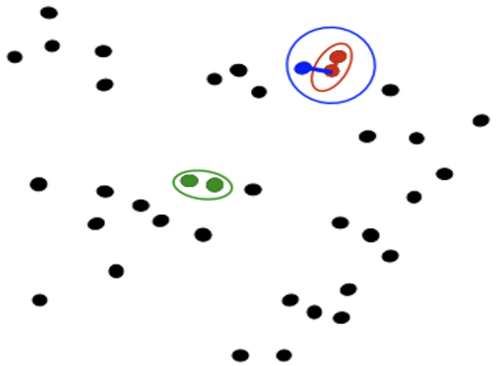
A “bottom up” approach where elements start as individual clusters and clusters are merged as one moves up the hierarchy.

- First merge very similar instances.
- Incrementally build larger clusters out of smaller clusters

Agglomerative clustering

Algorithm:

- Maintain a set of clusters
- Initially, each instance in its own cluster
- Repeat:
  - Pick the two "closest" clusters
  - Merge them into a new cluster
  - Stop when there's only one cluster left



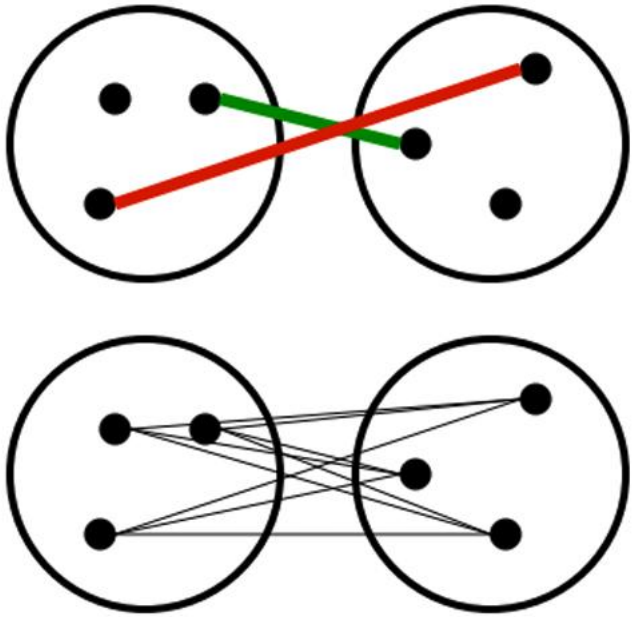
Produces not one clustering, but a family of clusterings represented by a dendrogram.

Agglomerative clustering

How should we define "closest" for clusters with multiple elements?

- Closest pair: single-link clustering
- Farthest pair: complete-link clustering

Average of all pairs



Divisive hierarchical methods

The divisive hierarchical method proceeds the opposite way of the agglomerative hierarchical method.

Initially, all the data points belong to a single cluster. The number of clusters is increased by one at each stage of the algorithm by dividing an existing cluster into two clusters according to some criteria. A divisive hierarchical method may be adopted in which a single cluster is subdivided into smaller and smaller clusters.

Divisive hierarchical methods

Divisive hierarchical clustering methods are essentially of two types:

- monothetic
- polythetic.

A monothetic method divides the data on the basis of the possession or otherwise of a single specified attribute, while a polythetic method divides the data based on the values taken by all attributes.

Divisive hierarchical methods

Algorithm Basic Divisive Hierarchical Clustering

1: Start with the root node consisting all the data points

2: **repeat**

3: Split parent node into two parts C_1 and C_2 using Bisecting K-means to maximize Ward's* distance $W(C_1,C_2)$.

4: Construct the dendrogram. Among the current, choose the cluster with the highest squared error.

5: **until** Singleton leaves are obtained.

*Distance is calculated as the increase in the total within-cluster variance after merging the clusters.

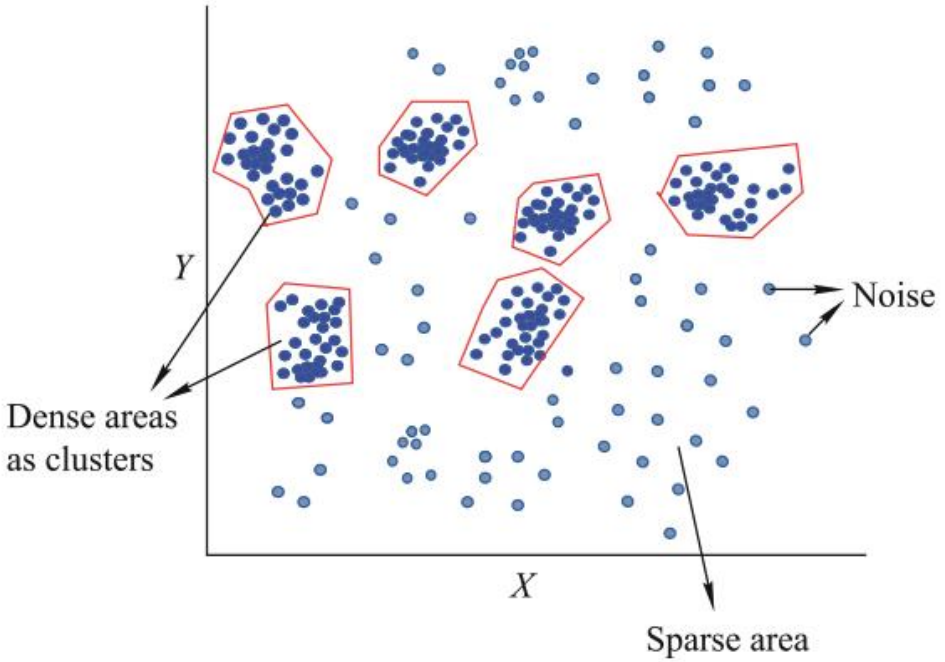
Density-based clustering

Clusters may be looked at as dense regions in the data space; where clusters are separated by a sparse region containing “relatively few” data. Given this assumption, a cluster can either be of “regular” or “arbitrary” shape.

Density-based clustering

The most common density-based clustering techniques are:

- DBSCAN
- OPTICS
- VDBSCAN
- DVBSCAN
- DBCLASD
- ST-DBSCAN
- DENCLUE



https://journal.hep.com.cn/fcs/EN/article/downloadArticleFile.do?attachType=PDF&id=25589

Density-based clustering - DBSCAN

DBSCAN (Density-based algorithm for discovering clusters in large spatial databases with noise)

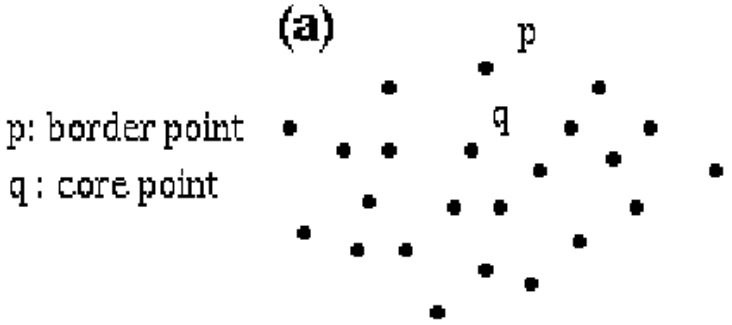
Pros: Extracts clusters of arbitrary shapes with filtration of noises.
Cons: Fails to detect clusters of variable densities. Unreliable in high-dimensional data space

Key issues addressed: Dense regions are detected as clusters. Points in sparser regions are filtered as outliers.

Density-based clustering - DBSCAN

The key idea is that for each point of a cluster, the neighbourhood of a given radius (*Eps)* has to contain at least a minimum number of points (*MinPts*), i.e. the density in the neighbourhood has to exceed some threshold.

The shape of a neighbourhood is determined by the choice of a distance function for two points p and q, denoted by dist(p,q).



Density-based clustering - DBSCAN

To find a cluster, DBSCAN starts with an arbitrary point p and retrieves all points density-reachable from p wrt. *Eps* and *MinPts*. If p is a core point, this procedure yields a cluster wrt. *Eps* and *MinPts*.
If p is a border point, no points are density-reachable from p and DBSCAN visits the next point of the database.

Association Rules

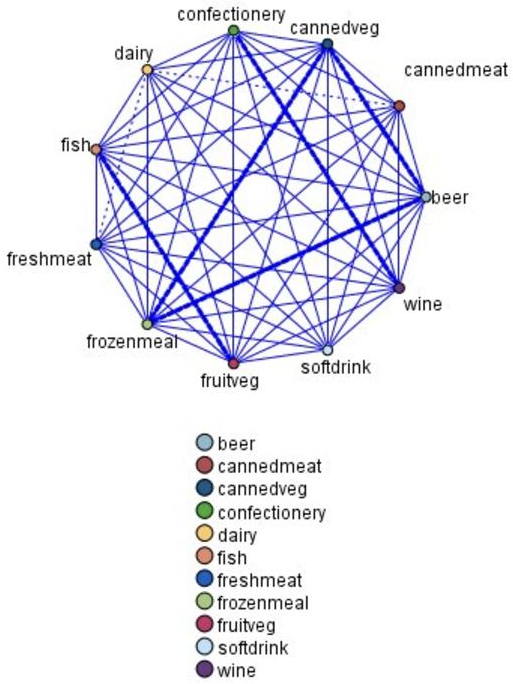
Association rules associate a particular conclusion (the purchase of a particular product) with a set of conditions (the purchase of several other products). For example, the rule

```
beer <= cannedveg & frozenmeal (173, 17.0%, 0.84)
```

states that *beer* often occurs when *cannedveg* and *frozenmeal* occur together. The rule is 84% reliable and applies to 17% of the data, or 173 records.

Association Rules

Association rule algorithms automatically find the associations that you could find manually using visualization techniques, such as the Web node.



Association Rules

The advantage of association rule algorithms over the more standard decision tree algorithms (C5.0 and C&R Trees) is that associations can exist between *any* of the attributes. A decision tree algorithm will build rules with only a single conclusion, whereas association algorithms attempt to find many rules, each of which may have a different conclusion.

The disadvantage of association algorithms is that they are trying to find patterns within a potentially very large search space and, hence, can require much more time to run than a decision tree algorithm.

Association Rules

The algorithms use a **generate and test** method for finding rules. Simple rules are generated initially, and these are validated against the dataset. The good rules are stored and all rules, subject to various constraints, are then specialized.

Specialization is the process of adding conditions to a rule. These new rules are then validated against the data, and the process iteratively stores the best or most interesting rules found. The user usually supplies some limit to the possible number of antecedents to allow in a rule, and various techniques based on information theory or efficient indexing schemes are used to reduce the potentially large search space.

At the end of the processing, a table of the best rules is presented.

Association Rules – Apriori algorithm

The strength of a given association rule is measured by two main parameters: support and confidence. Support refers to how often a given rule appears in the database being mined. Confidence refers to the number of times a given rule turns out to be true in practice.

Support of itemset A, denoted support(A)= frequency an Itemset A appears in the dataset T

$$\text{support}(\text{A}) = \frac{|\{t \in T : A \subseteq t\}|}{|T|} \text{ , t is an itemset.}$$

Let A and B be two itemsets. An association rule A → B asserts that if a transaction contains A, it is also likely to contain B

The Confidence is the percentage of all transactions satisfying A that also satisfy B.

$$\text{conf}(A \Rightarrow B) = \frac{supp(A \cap B)}{supp(A)} = \frac{\text{number of transactions containing A and B}}{\text{number of transactions containing A}}$$

Association Rules – Apriori algorithm

Define Minimum Support and Confidence: Set thresholds for minimum support and confidence to filter out less significant itemsets and rules.
•**Minimum Support**: The minimum frequency an itemset must have to be considered relevant.
•**Minimum Confidence**: The minimum probability that a rule is correct.
Generate Frequent Itemsets: Use the Apriori algorithm to identify itemsets that meet the minimum support threshold.

Step 1: Calculate the support for each individual item.
Step 2: Prune items with support less than the minimum support.
Step 3: Generate candidate itemsets by combining frequent items.
Step 4: Calculate the support for each candidate itemset and prune those below the minimum support.
Step 5: Repeat the process until no more frequent itemsets can be generated.

Association Rules – Apriori algorithm

Generate Association Rules: From the frequent itemsets, generate association rules that meet the minimum confidence threshold.

•**Example Rule**: If a customer buys bread and butter (itemset), they are likely to buy jam (consequent) with a certain confidence level.

Analyze and Use the Results: Use the generated rules to make business decisions, such as product placement, promotions, and inventory management.

Association Rules – Apriori algorithm

Apriori Algorithm uses frequent itemsets to generate association rules and it is based on the Apriori principle mentioned above. Apriori algorithms are designed to process databases containing transactional information content (e.g., lists of items purchased by customers). Frequent itemset is an itemset whose support value is greater than threshold value i.e. support(X)>= minsup.

Consider the given dataset with given transactions.

Transaction ID	Items
1	{A,C,D}
2	{B,C,E}
3	{A,B,C,E}
4	{B,E}
5	{A,B,C,E}

Association Rules – Apriori algorithm

We can try to calculate the support value of different itemsets.
support(A)=$\frac{3}{5}$, support(D)=$\frac{1}{5}$, ...
Let minsup=$\frac{2}{5}$, then we have :

Itemset	Support	F/I	Itemset	Support	F/I
A	3/5	F	ABD	0/5	I
B	4/5	F	ABE	2/5	F
C	4/5	F	ACD	1/5	I
D	1/5	I	ACE	2/5	F
E	4/5	F	ADE	0/5	I
AB	2/5	F	BCD	0/5	I
AC	3/5	F	BCE	3/5	F
AD	1/5	I	BDE	0/5	I
AE	2/5	F	CDE	0/5	I
BC	3/5	F	ABCD	0/5	I
BD	0/5	I	ABCE	2/5	F
BE	4/5	F	ABDE	0/5	I
CD	1/5	I	ACDE	0/5	I
CE	3/5	F	BCDE	0/5	I
DE	0/5	I	ABCDE	0/5	I
ABC	2/5	F			

Total itemsets: 31 (= 2^5 - 1), 5 is the amount of single item, which means A,B,C,D,E in this example.
Amount of Frequent itemsets: 15
Amount of Infrequent itemsets: 16

Association Rules – Apriori algorithm

We have a dataset T with 4 transactions, let minsup $= \frac{2}{4} = \frac{1}{2}$, mincof $= \frac{1}{2}$.

Transaction ID	Itemsets
1	A,C,D
2	B,C,E
3	A,B,C,E
4	B,E

Step 1: Scan T, calculate the support for each candidates, then we can get C_1 (the candidate set of level k) and L_1 (the frequent dataset).

Itemset	Support
A	2/4
B	3/4
C	3/4
D	1/4
E	3/4

C1

Itemset	Support
A	2/4
B	3/4
C	3/4
E	3/4

L1

Association Rules – Apriori algorithm

Step 2: Combine the itemsets in L_1 to generate C_2.

Itemset
A,B
A,C
A,E
B,C
B,E
C,E

Association Rules – Apriori algorithm

Step 3: Scan C_2 in order to get L_2.

Itemset	Support
A,B	1/4
A,C	2/4
A,E	1/4
B,C	2/4
B,E	3/4
C,E	2/4

C2

Itemset	Support
A,C	2/4
B,C	2/4
B,E	3/4
C,E	2/4

L2

Association Rules – Apriori algorithm

Step 4: Combine itemsets in L_2 to get C_3:

Itemset
A,B,C
B,C,E

Step 5: Scan C_3.

Itemset	Support
A,B,C	1/4
B,C,E	2/4

C3

Itemset	Support
B,C,E	2/4

L3

Association Rules – Apriori algorithm

$\Rightarrow$ STOP, the algorithm can't run anymore because no new frequent itemsets are identified.
$\Rightarrow \{B, C, E\}$ is the final frequent itemset.
$\Rightarrow$ Calculate the confidence and compare with the mincof to find out which association rules are possible.

When $\{B, C, E\}$ is the frequent itemset. We have the following possibilities for the association rules:
(1) $B \wedge C \rightarrow E$, $E \rightarrow B \wedge C$
(2) $B \wedge E \rightarrow C$, $C \rightarrow B \wedge E$
(3) $C \wedge E \rightarrow B$, $B \rightarrow C \wedge E$
Then we need to find out which possibilities are acceptable.

Association Rules – Apriori algorithm

(1) $\text{conf}(B \wedge C \rightarrow E) = \frac{2}{2} = 1$, $\text{conf}(E \rightarrow B \wedge C) = \frac{2}{3}$
(2) $\text{conf}(B \wedge E \rightarrow C) = \frac{2}{3}$, $\text{conf}(C \rightarrow B \wedge E) = \frac{2}{3}$
(3) $\text{conf}(C \wedge E \rightarrow B) = \frac{2}{2} = 1$, $\text{conf}(\text{is } B \rightarrow C \wedge E) = \frac{2}{3}$
All confidences are higher than minconf, so all of the possibilities are acceptable.

Μηχανική Μάθηση & Εξόρυξη Γνώσης

Ερωτήσεις

?

Βιβλιογραφία

- Ι. Βλαχάβας, Π. Κεφαλάς, Ν. Βασιλειάδης, Φ. Κόκκορας, Η. Σακελλαρίου, Τεχνητή Νοημοσύνη - Γ' Έκδοση, ISBN: 978-960-8396-64-7, Έκδοση/Διάθεση: Εκδόσεις Πανεπιστημίου Μακεδονίας, 2011
- Ian H. Witten and Eibe Frank. 2005. Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- Κ. Διαμαντάρας, Ι. Μπότσης, Μηχανική Μάθηση – Α' Έκδοση, ISBN: 978-960-461-955-5, Εκδόσεις Κλειδάριθμος, 2019
- P.-N. Tan, M.Steinbach, V. Kumar, «Introduction to Data Mining», Addison Wesley, 2006
- An Introduction to Genetic Algorithms Jenna Carr https://www.whitman.edu/Documents/Academics/Mathematics/2014/carrjk.pdf
- Unsupervised learning – Wei Wu https://na.uni-tuebingen.de/ex/ml_seminar_ss2022/Unsupervised_Learning%20Final.pdf