LINFO2364 Mining Patterns in Data

Issambre L'Hermite Dumont

Academic year 2025-2026 - Q2

UCLouvain

Contents

Introduction

In our data-driven world, the ability to extract meaningful information from vast datasets is crucial. Understanding all the aspect of this discipline is essential to derive those meaningful information, and this is the goal of this course. First, we need to define some key concepts.

1.1 Definitions

Definition 1.1. A pattern is a recurring structure in a dataset.

Patterns can be simple or complex, relevant or irrelevant. Their advantages is that they are interpretable. When found, relevant patterns can be used to make predictions, to understand the underlying structure of the data, and to make informed decisions.

Definition 1.2. Data mining is the process of discovering interesting patterns, models, and other kinds of knowledge in large data sets.

1.1.1 Type of data

We can mine data out of various types of structure of data:

- **Tabular data**: Data is organized in rows and columns. Example: spreadsheets, databases.
- **Sequences**: Data points are ordered in a sequence. Example: DNA sequences, text data.
- **Graphs, trees, networks**: Data is represented as nodes and edges. Example: social networks, web graphs.

Those structures can be discretes, continuous, enumerable data, etc. Those structures, can be combined to form more complex data types. And they can be highly structured, semi-structured, or unstructured.

Definition 1.3. Highly structured data are relational databases, with uniform record or table-like structures, with a fixed set of well-defined attributes. This is rarely the case in real-world data.

Definition 1.4. Semi-structured data are not as structured as in relational databases, but presents some structure with clearly defined semantic meaning. For example:

- Transactional dataset: structured into transactions, but each transaction is an unstructured set of values
- Sequence data set: unstructured collection of ordered sequences of values
- Graphs: set of nodes connected by a set of edges, with edges labelled given some semantic

Definition 1.5. Unstructured data have no predefined structure or organization. For example: text documents, images, audio files, videos.

Those requires advanced techniques to extract patterns, like deep learning or domain-specific methods. We can also categorize data based on the way they are generated:

- **Stored data**: Data is collected and stored in a finite set.
- **Streamed data**: Data is continuously generated and updated over time. Example: video surveillance, etc.

1.1.2 Types of data mining

Techniques and algorithms may vary depending on the data but also on way we will use mined patterns. We can categorize data mining tasks into two main types:

- **Descriptive data mining**: finds patterns that characterize the properties of the data.
- **Predictive data mining**: finds patterns that can be used to make predictions by induction.

We can identify patterns by multiples ways:

- **Frequent patterns**: patterns that identify a recurring structure in the data.
- **Associations patterns**: patterns that show rules of implications between different attributes
- **Correlations**: patterns that has positive or negative correlation between different attributes.

We can uses patterns for predictions:

- **Classification**: assign new data to classes based on similarities with historic data.
- **Regression**: predict numerical values based on the new data and the historic data.
- **Feature selection**: identify the most relevant features.

Preprocessing

To analyse data, and retrieve meaningful patterns, data often needs to be preprocessed. This step is crucial as raw data is often incomplete, inconsistent, and noisy. Preprocessing improves the quality of the data and the efficiency of the mining algorithms.

2.1 Useful statistical values

Definition 2.1. The mean (or average) of a dataset is the sum of all values divided by the number of values.

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{2.1}$$

Definition 2.2. The midrange is the average of the largest and smallest values in the set.

Definition 2.3. The variance of a dataset measures is defined like this:

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2 \tag{2.2}$$

Definition 2.4. The standard deviation of a dataset is the square root of the variance:

$$\sigma = \sqrt{\sigma^2} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2} \tag{2.3}$$

Definition 2.5. The mode for a set of data is the value that occurs most frequently. When there are multiple values with the same highest frequency, the dataset can be unimodal, bimodal, trimodal or multimodal.

Itemset mining

3.1 Definitions

Definition 3.1. An itemset is a set of items and an itemset containing k items is called a k-itemset.

Frequent itemset mining finds associations and correlations between items in large transactional or relational datasets. It is widely used in market basket analysis for example. It is the analysis of the buying habits of customers by finding associations between the different items that customers place in their shopping basket.

Definition 3.2. A transactionnal dataset is a collection of transactions, where each transaction is a set of items.

With $\mathcal{L}$ the set of possible items, $\mathcal{T}$ the set of possible transactions, a transactional dataset $\mathcal{D}$ can be seen as the function $\mathcal{D} : \mathcal{T} \to 2^{\mathcal{L}}$. It can also be represented as a binary matrix, where rows represent transactions and columns represent items, mathematically it is like the function $\mathcal{D} : \mathcal{T} \times \mathcal{L} \to \{0, 1\}$.

Definition 3.3. An itemset $l \subset \mathcal{L}$ covers (or matches) a transaction $t \subset \mathcal{T}$ if every item from l is in t. Consider the match function, with $D(t, l)$ the function representing the transactional dataset:

$$\texttt{match}(l, t) = \begin{cases} 1 & \text{if } \forall i \in l, D(t, i) = 1 \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

Definition 3.4. The occurence frequency of an itemset (also called frequency, support count, count or absolute support) is the number of transactions that contain the itemset (i.e., the size of the cover)

$$\texttt{support}_{\texttt{count}}(l) = \sum_{t \in \mathcal{T}} \texttt{match}(l, t) \tag{3.2}$$

Definition 3.5. The relative support of an itemset l in a database $\mathcal{D}$ is the percentage of transactions containing the itemset:

$$\texttt{support}_{\mathcal{D}}(l) = \frac{\texttt{support}_{\texttt{count}}(l)}{|\mathcal{T}|} \tag{3.3}$$

Definition 3.6. An association rule represents an implication of the form $X \Rightarrow Y$, where X and Y are itemsets.

Definition 3.7. The rule support is the support of the itemset containing all the elements of the rule $\texttt{support}_{\text{count}}(X \Rightarrow Y) = \texttt{support}_{\text{count}}(X \cup Y)$.

We consider an association rule or an itemset as frequent if its support is greater than a user-defined minimum support threshold θ.

Definition 3.8. The rule confidence is the percentage of transactions containing X that contains Y too

$$\texttt{confidence}(X \Rightarrow Y) = P(Y|X) = \frac{\texttt{support}_{\text{count}}(X \cup Y)}{\texttt{support}_{\text{count}}(X)} \tag{3.4}$$

3.2 Apriori algorithm

The key idea behind the Apriori algorithm is that all nonempty subsets of a frequent itemset must also be frequent.

Algorithm 1 Apriori algorithm

```
1: **Input:** Transactional dataset $\mathcal{D}$, minimum support threshold $\theta$
2: **Output:** Set $F$ of frequent itemsets
3: $F = \emptyset$
4: $C_1 = \{\{i\} : i \in \mathcal{L}\}$
5: $L_1 = \{c \in C_1 : \texttt{support}_{\text{count}}(c) \geq \theta\}$
6: $F = F \cup L_1$
7: $k = 2$
8: while $L_k$ is not empty do
9:     $C_k = \texttt{apriori\_gen}(L_{k-1})$
10:    $L_k = \{c \in C_k : \texttt{support}_{\text{count}}(c) \geq \theta\}$
11:    $F = F \cup L_k$
12:    $k = k + 1$
13: end while
14: return $F$
```

The $\texttt{apriori\_gen}(L_{k-1})$ function generates candidate k-itemsets from the frequent (k-1)-itemsets L_{k-1} by joining them and pruning those that have infrequent subsets.

Algorithm 2 apriori_gen(L_{k-1})

```
1:  Input: Set $L_{k-1}$ of frequent (k-1)-itemsets
2:  Output: Set $C_k$ of candidate k-itemsets
3:  $C_k = \emptyset$
4:  for each $l_1 \in L_{k-1}$ do
5:      for each $l_2 \in L_{k-1}$ do
6:          if $l_1[: k-2] = l_2[: k-2]$ and $l_1[k-1] < l_2[k-1]$ then
7:              $c = l_1 \cup l_2$
8:              if all (k-1)-subsets of $c$ are in $L_{k-1}$ then
9:                  $C_k = C_k \cup \{c\}$
10:             end if
11:         end if
12:     end for
13: end for
14: return $C_k$
```

The number of candidate itemsets is bounded by $\mathcal{O}\left(\binom{|\mathcal{L}|}{k}\right)$.

The two mains weaknesses of the Apriori algorithm are:

- Potential generation of huge candidate sets
- Repeated scan of the database and checks for pattern matching to the transactions

The Apriori algorithm can be improved using:

- **Hash-based techniques**: Instead of building L_2 from the join operation, build it similarly to L_1. When scanning the transactions, generate the 2-itemset subset of the transaction, hash them and map them to corresponding buckets containings counters to be increased. L_2 is the set of the 2-itemsets associated to buckets of a count higher than θ
- **Transaction reduction**: A transaction that does not contains any frequent k-itemsets cannot contain any frequent $(k+1)$-itemsets. They can be flagged and avoided at next steps.
- **Partitioning**: The dataset can be partitioned in N non-overlapping sub-database. Given θ, the minimum support count threshold for the whole database, $\theta' = \lfloor \frac{\theta}{N} \rfloor$ is the threshold for each partition. The candidate set C_k is the union of all local frequent k-itemset of each partition as any itemset that is potentially frequent with respect to D must occur as a frequent itemset in at least one of the partitions.
- **Sampling**: The frequent itemsets of a sample of the database are computed to serve as C_k. Some degree of accuracy is traded for eciency, thus a lower value than θ' is used to counterbalanced the effect.