The Application of Sequential Pattern Mining Techniques on MIMIC-IV

Cecilia Mariciuc[a] and Mădălina Răschip[a]

[a] *Alexandru Ioan Cuza University of Iaşi, Bulevardul Carol I, Nr.11, Iaşi, 700506, Romania*

Abstract

The paper studies the application of sequential pattern mining techniques to medical data from MIMIC-IV, a large healthcare dataset. Sequences of prescribed drugs to a large number of patients are analyzed in order to find out if there are patterns or temporal relationships which are general or specific to a particular disease. The PrefixSpan and Spade algorithms were applied to mine sequential patterns on all sequences or on a subset of them. The extracted patterns could be used to suggest the next prescribed drug. The experimental results show that the predictions obtained have a good accuracy for some diagnoses.

Keywords 1

sequential pattern mining, next prescribed drug, MIMIC-IV

1. Introduction

The correct use of a drug is dependent upon several conditions. Each drug has some characteristics, such as indications, possible risk factors and contraindications, like the use with other drugs or the existence of certain medical conditions. The improper use of drugs and self-medication can be dangerous [1].

The advancement of technology has made it possible to digitally collect and store patient data for their subsequent use. The manipulation of this large amount of data could bring new knowledge to the medical field [2]. Medications prescribed by specialists can be used to identify the optimal treatment. The order of the prescriptions could provide important information. Frequent subsequences or predictions of the next drug can help a doctor in making a quick decision when there are too many medication options. They can be used to make automatic recommendations in routine cases, or to verify the correctness of unusual orders.

Sequential pattern mining can be a solution to this problem because it can identify patterns of ordered events [3]. A survey of the approaches proposed for sequential pattern mining is given in [4], [13]. Sequential pattern mining was applied in different areas of research, also including the medical domain. For example, to identify temporal relationships between drug prescription and medical events or between prescriptions of different drugs [5], or to identify if a person is susceptible to a future illness [6].

In this paper, we used sequential pattern mining to predict the next medication for a patient. Other existing studies in the literature are based on machine-learning methods. In [8], the prescription data

IDDM-2021: 4th International Conference on Informatics & Data-Driven Medicine, November 19–21, 2021 Valencia, Spain
EMAIL: cecilia.mariciuc27@gmail.com (A. 1); madalina.raschip@uaic.ro (A. 2);
ORCID: 0000-0003-0020-636X (A. 2);

is transformed into a stochastic time series for prediction. Various machine-learning approaches were used and analyzed in order to predict prescription patterns. A different approach is presented in [9]. The authors used neural networks and word2vec representations to predict the medication order prescribed during hospitalization, which could be used to assist pharmacists. Good results were obtained for obstetrics and gynecology patients and newborn babies. The paper [10] predicts prescriptions for the next period of time based on the disease status, laboratory results and the previous treatment of the patient through a framework of machine learning. The authors used three Long Short-Term Memory models. The experiments were performed on data from the MIMIC-III ICU and other data from hospitals in China.The results obtained reveal the effectiveness of the methods. Another study [11]uses probabilistic topic modelling to predict clinical order patterns.

A similar study to ours is presented in [7]. The authors describe an approach based on sequential pattern mining to identify the next prescribed medication for patients with diabetes. The CSPADE algorithm is used to mine sequential patterns at the drug class and generic drug level. The dataset used in our research is different from the one considered in [7]. We used a larger real-world dataset, MIMIC-IV, on which sequential pattern mining has not been applied before. The preprocessing step of identification of drugs and the construction of sequences are specific to this dataset. Two mining algorithms, PrefixSpan and SPADE, were considered. Although the predictions are made in a similar way by constructing some rules from the frequent patterns, the analysis of the mining algorithms on the MIMIC-IV dataset and the evaluation of the results on several diagnoses such as "heart attack" are two other elements that distinguish the current paper from the existing works.

The paper is organized as follows. A formal description of the problem of mining sequential patterns and the algorithms used to solve the problem is givenin Section 2. In Section 3 we present the dataset used and in Section 4 the experimental settings and results. We conclude with a summary and future improvements in Section 5.

2. Sequential Pattern Mining

The problem that sequential pattern mining is trying to solve can be describedas follows: knowing that many events occur in time, can we learn more about this data if we analyse any ordered sequence encountered? [13]

In the following we formally describe the problem. Let $I = \{i_1, i_2, \dots, i_n\}$ be a set of elements, also called an alphabet. An event $\left(i_{x_1}, i_{x_2}, \dots, i_{x_k}\right), 1 \le x_j \le n, \forall\, j \in \{1, \dots, k\}$ is a nonempty subset of I and an unordered collection of elements. A sequence $\langle e_1, e_2, \dots, e_q \rangle$ is an ordered collection of events. A sequence that contains k elements is known as a k-sequence. A sequence $s_e = \langle e_1, e_2, \dots, e_n \rangle$ is a subsequence of the sequence $s_f = \langle f_1, f_2, \dots, f_m \rangle$ if there exist integers $1 \le i_1 < i_2 < \cdots < i_n \le m$ such that $e_1 \subseteq f_{i_1}, e_2 \subseteq f_{i_2}, \dots, e_n \subseteq f_{i_n}$. A sequence database is a set of sequences that have associated identifiers. The support of a sequence *s*, denoted *sup*(*s*), in a sequence database represents the number of sequences containing *s*, i.e., for which *s* is a subsequence. Giving a value for the minimum support, denoted $minsup$, a sequence is considered frequentin a database if its support is at least equal to the $minsup$. Sequential pattern mining aims to find these frequent sequences.

2.1. SPADE

SPADE (Sequential PAttern Discovery using Equivalence classes) [14] is an Apriori-based algorithm, making use of the Apriori property that claims that any subsequence of a frequent sequence is also a frequent sequence. SPADE works with data organized in vertical format, by

transforming the initial sequence database into a table composed of all events where a row is an event linkedwith the corresponding sequence identifier (SID) and its position in the sequence(EID).

At each step k, the algorithm searches for k-sequences that have the chance to be frequent, by generating id-lists. The first step is to find the 1-frequent sequences. Support is calculated for each element of the alphabet, counting the entries in the vertical formatted table that contains it. Those entries will be included in its id-list. Subsequently, only items that reach the minimum support are frequent and will be considered for finding 2-frequent sequences. In the general case, candidate k-sequences are found by joining the id-lists of any two frequent (*k*-1)-sequences, that have the same SID and have ordered sequential positions (EIDs). The algorithm stops when no more frequent sequences have been found or no more candidate sequences have been constructed.

2.2. PrefixSpan

PrefixSpan (Prefix-Projected Sequential Patterns Mining) [15] is a Pattern-Growth-based algorithm, because it does not generate candidate sequences, but instead uses partitioning of the data set into projections, which will be explored separately to extend the already known frequent sequences.

The PrefixSpan algorithm includes the following steps:

1. Find 1-frequent sequences in the dataset that will later be concatenated to the current frequent sequence (or the current frequent prefix) to form new frequent sequences. Initially, the current frequent sequence is an empty sequence, $s = \langle\_\rangle$.
2. The search space is partitioned according to the sequences found in the previous step. For each new, frequent sequence obtained, a projection is created, considering that sequence as a prefix.
3. For each projection, look for the elements with support at least equal to $minsup$ which will be used to extend the previous frequent sequences.

These steps are repeated recursively, the algorithm operating on a divide et impera strategy.

3. MIMIC-IV dataset

MIMIC (Medical Information Mart for Intensive Care) is a relational database, publicly accessible which documents the hospitalizations of patients at Beth Israel Deaconess Medical Center (BIDMC) in Boston, MA, USA. MIMIC-IV [12] is the latest version of the MIMIC database and represents an improvement of MIMIC-III, with a modular structure and more recent patient data from 2008 to 2019. MIMIC-IV contains five modules that reflect the origin of the data: *core*, *hosp*, *icu*, *ed* and *cxr*. We used the *hosp* module which provides information from the electronic medical records that include laboratory tests, medications, and diagnoses. From this module, the following tables were used: *prescriptions*,*diagnoses_icd* and *d_icd_diagnoses*. The prescriptions table contains information about the prescribed medications. The drug type field has three possible values: MAIN, BASE, or ADDITIVE. The *diagnoses_icd table* records the diagnosesfor which a patient was billed. Each diagnosis has associated a *seq_num* whichrepresents the importance of the diagnosis. The lower the *seq_num* is, the more significant the diagnosis is. The official name of a diagnosis can be identified using the table *d_icd_diagnoses*.

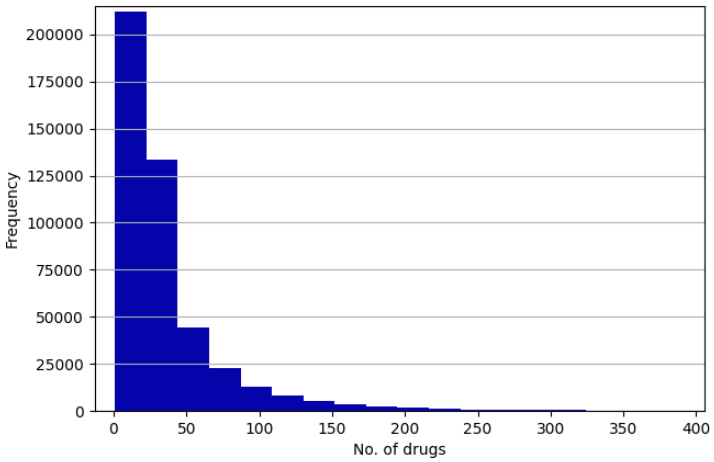

Figure 1: Distribution of the number of drugs per hospitalization

The prescriptions table contains 17008053 records, i.e., drugs that were individually prescribed. In most prescriptions, the drug type was in the MAIN category. Prescriptions were made for 232064 patients, with 452115 hospitalizations. A distribution of the number of drugs per hospitalization is available in Figure 1. In most cases, this number falls in the range [0,400], although there are also much higher values (a maximum of 2156).

There are 5280351 diagnoses in the associated table *diagnoses_icd*, established for 255106 patients who had 521111 hospitalizations. A patient may have several hospitalizations, and for each hospitalization, several diagnoses. The distribution of the number of diagnoses per hospitalization is given in Figure 2.

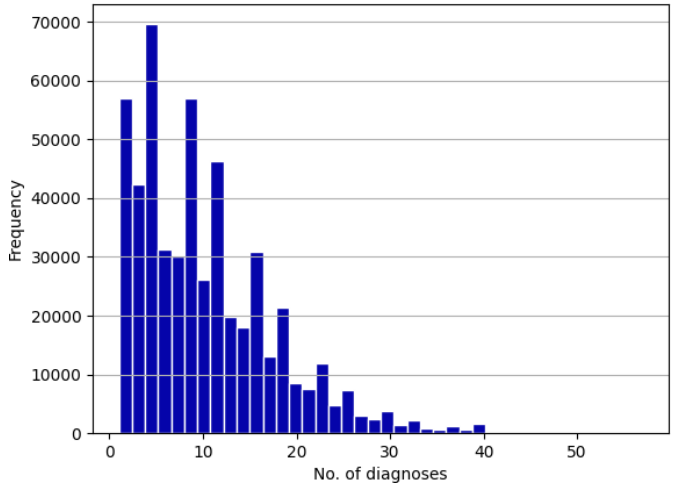

Figure 2: Distribution of the number of diagnoses per hospitalization

The *d_icd_diagnoses* table contains 109775 lines, or possible diagnoses. Table 1 shows the ranking of the most common diagnoses.

Table 1
Distribution of the number of diagnoses per hospitalization

No.	Diagnosis	No. of occurrences
1	Unspecified essential hypertension	104080
2	Other and unspecified hyperlipidemia	68215
3	Essential (primary) hypertension	54696
4	Hyperlipidemia, unspecified	51097
5	Esophageal reflux	49593
6	Diabetes mellitus without mention of complication, type II or unspecified type, not stated as uncontrolled	43705
7	Personal history of nicotine dependence	40803
8	Atrial fibrillation	37337
9	Depressive disorder, not elsewhere classified	36905
10	Congestive heart failure, unspecified	36891
11	Coronary atherosclerosis of native coronary artery	36404
12	Gastro-esophageal reflux disease without esophagitis	35610
13	Need for prophylactic vaccination and inoculation against viral hepatitis	32686
14	Personal history of tobacco use	32225
15	Major depressive disorder, single episode, unspecified	30398
16	Acute kidney failure, unspecified	29276
17	Unspecified acquired hypothyroidism	29051
18	Encounter for immunization	27146
19	Atherosclerotic heart disease of native coronary artery without angina pectoris	26706
20	Tobacco use disorder	26340

4. Experimental results

This section describes the steps followed in generating predictions using Sequential Pattern Mining algorithms on the MIMIC-IV dataset, as shown in Figure 3. The steps are the following: finding the list of distinct drugs, filtering hospitalizations by diagnoses, building sequences of drugs, running sequential pattern algorithms and building rules.

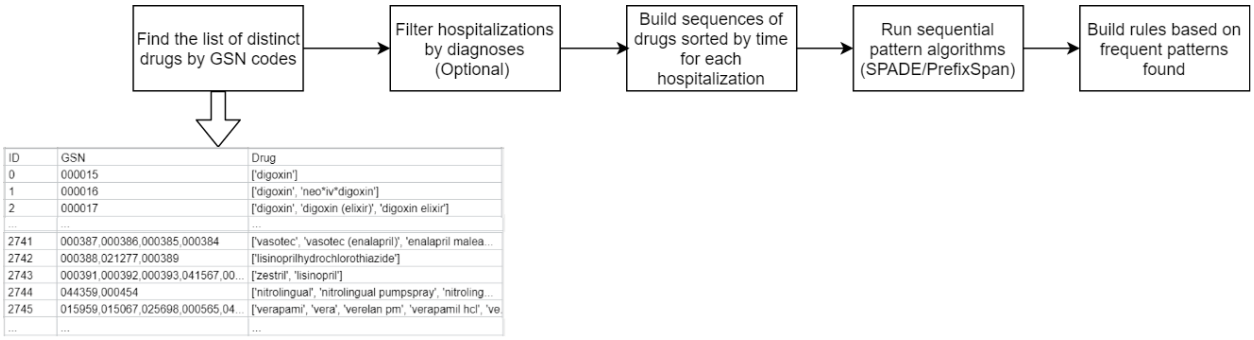

Figure 3: Pipeline for the strategy used

The cases where the predictions are relevant and the parameters that influence their accuracy are analyzed.

4.1. Preprocessing

The same drug may appear in prescriptions in several forms, such as various abbreviations (‘hepa’, ‘hepar’, ‘hepari’, ‘heparin’), some of the letters are capitalized (‘acetaZOLAMIDE’, ‘Acetazolamide’, ‘AcetaZOLamide’), more or less spaces and special characters (’Dextromethorphan-’, ’Dextromethorphan’), additional words, such as ’pain’, ’bulk’, ’extended release’ (’vancomycin’, ’vancomycin (bulk)’). Another, more complex problem, is that medicines may appear under completely different names, i.e. with the generic name, or with the name used by the brand. A solution to all these inconsistencies is the usage of the *gsn* field, which contains one or more 6 digit Generic Sequence Number (GSN) codes. GSN identifies a product based on its formula, dose, method of administration and concentration and can be used to group generally equivalent products, which may differ only through the manufacturer. In order to reduce the existence of several equivalent elements, we created a list of drugs with a unique id associated with the help of the GSN codes. Since a drug or other equivalent drugs can be associated with several GSN codes, groups of GSN codes will be established so that one group contains all codes that have been mentioned together directly or indirectly. Two drugs will be considered equivalent if at least one of their GSN codes (not necessarily identical) is found in the same group of GSN codes. Thus, starting from a list of 16970 pairs (drug, *gsn*), we obtained a list of 3398 drugs with a unique id after preprocessing.

4.2. The construction of sequences

A sequence is an ordered list of events of the form $\langle e_1, e_2, \dots, e_n \rangle$ and initially the events are empty subsets of the alphabet *I*. In our case, the alphabet is the set of drugs ids $I = \{0,1,2, \dots, 3397\}$. A sequence corresponds to a hospitalization and is represented by the list of ids of the drugs prescribed, grouped and sorted by time. For example, the sequence $\langle (2624), (2624), (2769, 539, 1100) \rangle$ specifies that in the case of a hospitalization, the drug with id 2624 was prescribed first, then again, the same drug, and then followed by a group of three drugs.

We considered two cases for the generation of sequences: the sequences are built for all hospitalizations, or only on a subset of hospitalizations. In the first case, the list of distinct hospitalizations that have at least one prescription can be easily found by querying the prescriptions table. For each element of the list, the events of the corresponding sequence are considered. Given that there are 452 115 distinct hospitalizations, the number of generated sequences is high, fact which limits the competence of mining algorithms. Consequently, for the second case, we considered filtering the hospitalizations after one or more diagnoses. Given a set of keywords, we will search for hospitalizations that have diagnoses that contain all the keywords. For example, for the words ‘heart’ and ‘pneumonia’, hospitalization with the following diagnoses ‘Pneumonia due to adenovirus’, ‘Aneurysm of heart’, ‘Other and unspecified hyperlipidemia’ will be selected. In addition to this filtering, when constructing sequences, only prescriptions with a drug type equal to MAIN will be considered.

This filtering is meant to facilitate the use of fewer resources (time and memory) by algorithms and to obtain better results, because the selection of hospitalizations by diagnoses can increase the chance of finding more common patterns.

4.3. Sequence pattern mining

The frequent sequences of prescribed drugs were extracted using two sequential pattern mining algorithms, SPADE and PrefixSpan, available in the open-source Java library SPMF [16]. We run the algorithms on an instance based on Windows 10 Pro that has an Intel(R) Core (TM) i7-8550U CPU @ 1.80GHz processor with8 GiB of memory.

SPADE cannot be applied to the entire dataset due to the additional memory the algorithm requires to transform the sequence database into a vertical format. SPADE is suitable to be used for a subset of hospitalizations. The results of SPADE are given in Table 2.

Table 2
The results of SPADE

Diagnosis	diagnoses	*minsup*	sequences (hospitalizations)	Avg no. of events	frequent sequences	Time (s)	Memory (mb)
Heart failure	73	0.025	52086	21.04	80983	86.94	1556.81
Born in hospital	21	0.008	37113	2.77	44020	41.21	407.844
Acute kidney failure	15	0.025	48255	24.92	68860	162.21	906.58
Need for prophylactic vaccination and inoculation against viral hepatitis	1	0.0001	29177	1.44	96546	44.98	329.158
Circumcision	2	0.0001	13269	2.05	262264	45.61	329.158
Encounter for immunization	1	0.0011	20149	1.88	73801	51.04	415.74

We considered six use cases, i.e., hospitalizations that had the following diagnoses: *Heart failure*, *Born in hospital*, *Acute kidney failure*, *Need for prophylactic vaccination and inoculation against viral hepatitis*, *circumcision* and *Encounter for immunization*. We selected the hospitalizations for a diagnosis based on some terms. The chosen terms are contained in or represent the names for the most common diagnoses. We consider diagnoses with $seq\_num \geq 5$ because of the higher chance that they will be the main reason for the hospitalization. For example, there are 73 different diagnoses containing the term 'heart failure' and for which this diagnosis is important ($seq\_num \geq 5$). One of the most common diagnoses is *Congestive heart failure, unspecified*, according to **Error! Reference source not found.**. The number of resulted sequences is 52086, with an average of 21,04 events. A number of 80983 frequent sequences were found by applying the algorithm on the sequences. The selected values for the minimum support are specified in Table 2. The value of $minsup$ is empirically chosen for practical time limits. A lower number of events means that fewer medications are prescribed for those hospitalizations, which allows the choice of a lower minimum support.

PrefixSpan The parameters of the algorithm are the value of the minimumsupport and, optionally, the maximum length of the sequences.

We tested the algorithm for all hospitalizations, using a minimum supportof 0.025 (10891 sequences) and a maximum length of a sequence of 20. The number of frequently found sequences is equal to 9771. Some of the most frequently used drugs are: *lactated ringers (Ringer's Lactate Solution)*, *hydralazine (Hydralazine)*, *tylenol (Paracetamol)*, 0.9% *sodium chloride*, *potassium chloride*, *heparin flush*, etc.

A small decrease in the minimum support can significantly increase the number of sequences and thus the execution time. For example, for a minimum support of 0.02 (8713 sequences), the number of frequent sequences increases to 20968.

Next, we repeated the tests made with SPADE, but using the PrefixSpan algorithm instead. The results are given in Table 3.

Table 3
The results of PrefixSpan

Diagnosis	*minsup*	Frequent sequences	Time (s)	Memory (mb)
Heart failure	0.025	101517	319	662.53
Born in hospital	0.008	44601	29.59	241.61
Acute kidney failure	0.025	86110	726.23	555.51
Need for prophylactic vaccination and inoculation against viral hepatitis	0.0001	99975	0.77	100.57
Circumcision	0.0001	268516	1.68	95.39
Encounter for immunization	0.0011	93614	6.4	114.12

PrefixSpan finds more frequent sequences and uses less memory than SPADE. However, the execution time is significantly shorter for SPADE when we have long sequences, and shorter for PrefixSpan in case of short sequences.

We next analyzed the frequent sequences that resulted from the application of the algorithms. We considered two cases: with a high minimum support (for example, hospitalizations with *heart failure* diagnosis) and with a low minimum support (for example, hospitalizations with *Need for prophylactic vaccinationand inoculation against viral hepatitis* diagnosis).

Some frequent sequences found for hospitalizations with *Need for prophylactic vaccination and inoculation against viral hepatitis* diagnosis are given in Table 4.

Table 4
Sequential patterns

Sequential patterns	Support
'erythromycin ophthalmic'	27346
'erythromycin ophthalmic' 'phytonadione (vitamin k1)' → 'erythromycin ophthalmic' 'hepatitis b vaccine' 'phytonadione (vitamin k1)'	11
'triple dye' 'erythromycin ophthalmic' 'hepatitis b immune globulin'	37

'phytonadione (vitamin k1)'	27347
'hepatitis b vaccine'	7876
'phytonadione (vitamin k1)' → 'gentamicin'	1324
'phytonadione (vitamin k1)' → 'acetaminophen'	2356
'lidocaine' 'acetaminophen' → 'hepatitis b vaccine'	189
'triple dye' → 'hepatitis b vaccine'	2525
'triple dye' 'erythromycin ophthalmic' 'hepatitis b immune globulin'	37
'erythromycin ophthalmic' 'phytonadione (vitamin k1)'→'phytonadione (vitamink1)'	7428

We analyzed the sequential patterns in order to identify the most commonly used medications. For *heart failure* diagnoses, the most common drugs among the frequent sequences are: *tylenol*, *senna laxative*, *aspirin*, *docusate sodium*, *dextrose*, *furosemide*, *metoprolol tartrate*, *glucagon*, etc. Most of these drugs are also common in all hospitalizations, making difficult to say whether they are specific to these types of hospitalizations or not. Consequently, we manually searched for drugs known to be common for the treatment of heart failure[2]. We give next some of the results:

- from the class Angiotensin-Converting Enzyme (ACE) Inhibitors: *lisinopril* is often found, being contained in over 400 sequences, *captopropyl* is found in 6 sequences, with support in the range [1000-2200]
- from the Beta Blockers class: carvedilol appears in over 100 sequences; *metoprolol* is one of the most frequently found drug
- from the class of *Vasodilators*: *hydralazine* is found in many different forms, *nitroglycerin* is found in over 500 sequences
- from the class of *Diuretics*: *furosemide* is one of the most common drugs, *torsemide* is found in over 500 sequences, *metallozone* is found only individually

For patients diagnosed with *Need for prophylactic vaccination and inoculation against viral hepatitis*, the drugs prescribed are less varied, most of them being *hepatitis b immune globulin (bayhep b)*, *hepatitis b vaccine*, *vitamin k*, *gentamicin*, *erythromycin ophthalmic*, *tylenol*, *heparin*, *triple dye*. In addition to the vaccine itself (current diagnosis indicates the need for hepatitis vaccination), usual drugs are found, or drugs specific to newborns, because the hepatitis B vaccine is administered to them immediately after birth.

4.4. Make predictions using frequent sequences

As we previously specified, the frequent sequences can be utilised to identify drugs used for different diagnoses. But when the number of sequencesis huge, this approach becomes less relevant and time expensive. Frequent drug sequences can reveal which drugs or combinations of drugs are more likely to be recommended when we know the previous prescriptions. We will predict the most likely drugs to be prescribed and compare the result with the real valuesto determine the accuracy of the predictions.

Rules construction To describe the links between the drugs from frequent sequences, we will generate rules of form (antecedent, consequent, support) withthe following meanings:

[2] https://www.nhs.uk/conditions/heart-failure/treatment/,

https://www.heart.org/en/health-topics/heart-failure/treatment-options-for-heart-failure/medications-used-to-treat-heart-failure

- For a sequence $s = \langle e_1, e_2, \dots, e_n \rangle$, the antecedent will contain the first (n−1) events $\langle e_1, e_2, \dots, e_{n-1} \rangle$, and the consequent will be the last event e_n. Only sequences containing at least two elements are considered.
- The support of the rule will correspond to the sequence support.

Some examples of rules generated from frequent sequences for *Need for prophylactic vaccination and inoculation against viral hepatitis* diagnosis are given in Table 5**.**

Table 5
Rules generated for the Need for prophylactic vaccination and inoculation against viral hepatitis diagnosis

Antecedent	Consequent	Support
{'erythromycin ophthalmic', 'phytonadione (vitamin k1)'}	{'erythromycin ophthalmic' 'hepatitis b vaccine' 'phytonadione (vitamin k1)'}	11
{'phytonadione (vitamin k1)'}	{'gentamicin'}	1324
{'phytonadione (vitamin k1)'}	{'acetaminophen'}	2356
{'lidocaine', 'acetaminophen'}	{'hepatitis b vaccine'}	189
{'triple dye'}	{'hepatitis b vaccine'}	2525
{'erythromycin ophthalmic', 'phytonadione (vitamin k1)'}	{'phytonadione (vitamin k1)'}	7428

Predictions using rules Before making any predictions, the list of rules is sorted using a multi-level approach: first, descending by the number of events from the antecedent and then descending by support. To narrow the search space, we also created athreshold dictionary as follows: for each length of the antecedent that exists in the previously sorted list, store the index of the first corresponding rule. For example, the following threshold dictionary, denoted $thresholds = \{8: 0, 7: 97, 6: 1178, 5: 6174, 4: 17020, 3: 29901, 2: 39094, 1: 43006, 0: 43908\}$ reveals that there are eight distinct lengths of the rules' antecedent. The rules that have an antecedent containing *x* events, $1 \leq x \leq 8$, will be found in the list starting with position $thresholds[x]$ and up to position $thresholds[x-1] - 1$.

Having a patient's prescribed medication sequence during a hospitalization $s = \langle e_1, e_2, \dots, e_n \rangle$ and a sorted list of rules, the predictions will be made as follows:

1. If $n \geq 1$, iterate through the rules with the number of events from the antecedent equal to the number of events in *s*. The threshold list will be used.
2. For each rule, check if there is a match between the antecedent and the sequence *s*. If a match is found, the event from the consequent is added to a list.
3. If five matches are found, the search ends. Otherwise, the first event from the sequence *s* is removed and the previous steps are repeated. Deleting the first item from *s* means, in fact, that we are trying to test on the patient's more recent history.

At the end, the list of maximum five events represents the predictions of drugs for the patient with the sequence *s* as history.

To test the accuracy of the predictions, we used the hospitalizations for whichthe frequent sequences were found and which, implicitly, were used to generatethe rules and the dictionary of thresholds. Denote by p_{max} the maximum value of a key in the threshold dictionary, or the maximum length of an antecedent for the current rules. The sequences of each hospitalization are divided into segments of length p_{max}. If they are not divided exactly, the last segment willbe considered if its length is at least two. The last event is removedfrom each segment, as it will be used to verify the correctness of the predictions. Predictions are made based on these segments, and if at least one of the drugs

contained in the predicted events is found in the event set aside, then we will consider the prediction is correct. Accuracy is computed as the percentage of correct predictions out of the total predictions made.

Predictions results The prediction results are given in Table 6.

Table 6
Predictions results

Diagnosis	Algorithm	Support	p_{max}	Sequences	Segments	Accuracy	Runtime (sec)
Heart failure	SPADE	0.025	12	25%	22139	25.83%	2914.35
Born in hospital	SPADE	0.008	8	100%	18165	65.15%	858.18
Need for prophylactic vaccination and inoculation against viral hepatitis	PrefixSpan	0.0001	13	100%	9962	80.14%	106.96
Circumcision	PrefixSpan	0.0001	13	100%	10854	89.32%	37.07
Encounter for immunization	PrefixSpan	0.0011	7	50%	3965	55.88%	266.22

For the heart failure diagnosis, for example, for 5718 sequences at least one correct prediction was obtained, meaning an accuracy of 25.83%, and for 4704 we could not find any prediction. If we take into account only the sequences on which predictions were found, then the accuracy would be 32.79%.

For certain diagnostics, like *Need for prophylactic vaccination and inoculation against viral hepatitis* and *Circumcision* the accuracy is high, while for other diagnoses like *Heart failure* it is small. Statistically, the number of prescriptions increases with age [17]. Intuitively, a diagnosis that contains the term 'born in hospital' refers to newborns, in which case certain standard medicines are required. The number of allowed drugs is lower (many drugs have age restrictions). In this case, it is easier to identify which drugs are more likely to be prescribed. The diagnoses *Need for prophylactic vaccination and inoculation against viral hepatitis* and *Encounter for immunization* indicate that a person needs administration of a vaccine. The person is not necessarily ill, so the number of drugs is not expected to be high. Instead, diagnoses that contain 'heart failure' indicate a serious, complex condition that is often found in the population over the age of 65.

To better clarify the possible reasons that affect the accuracy of predictions, we analysed other measures detailed in Table 7. The second column contains the total number of different drugs encountered in the sequences. The next column contains the average number of drugs per sequence. The last column contains the average difference between the date of the last prescription and the date of the first prescription.

Table 7
Measures that can influence the accuracy of predictions

Diagnosis	Total drugs	Drugs per sequence	Time diff (days)
Heart failure	2155	45.82	6
Born in hospital	305	4.92	3
Need for prophylactic vaccination and inoculation against viral hepatitis	200	2.96	1
Circumcision	112	3.76	1
Encounter for immunization	864	3.84	1

According to Table 7, when there is a wider range of drugs to choose from, theaccuracy tends to decrease. The average number of drugs per sequence influences the sequential pattern mining algorithms: it is necessary to usually choose a larger support, so as not to use too much memory, fact which also influences theaccuracy. Another parameter that could influence the results is the length of the period in which prescriptions were made. This may indicate complex diagnoses or, conversely, less severe cases.

The choice of the minimum support can influence the accuracy of the predictions, and indirectly the runtime and the memory. Table 8 exemplifies theway the support influences the accuracy. The last column is the time needed to compute the predictions.

Table 8
The influence of support on the accuracy of predictions

Diagnosis	Support	Frequent sequences	Accuracy	Time (sec)
Circumcision	0.01	246	85.54%	0.39
	0.001	2505	88.50%	1.38
	0.0001	268516	89.32%	37.07
Need for prophylactic vaccination and inoculation against viral hepatitis	0.01	132	71.36%	0.12
	0.001	1413	77.13%	0.45
	0.0001	99975	80.14%	6.97

As the minimum support decreases, the accuracy increases slightly, and the runtime also increases. Lowering the support is useful up to a certain limit, for which a reasonable execution time is obtained

5. Conclusions

Sequential Pattern Mining represents an effective technique to make predictions of medications based on the patient's past prescription history. This paper studies in particular the application of two algorithms, SPADE and PrefixSpan, as a means to find frequent sequences that reveal temporal relationships between medications. The resulting frequent sequences are general or specific to one or more diseases and are used to construct rules. Predictions are made by finding matches of a patient's

medication history in the list of rules. According to the experimental results, there are situations in which the predictions made can reach a satisfactory accuracy. Such a solution is especially useful for routine cases, for instance, immunizations, or for the treatment of newborns. Instead, for more complex diagnoses, additional study is needed to optimize the results.

Some improvements that can be made are the addition and the usage of supplementary patient information, such as laboratory results, age and supplementary medication details, like the dose, the method of administration.

6. References

[1] Schmiedl, S., Rottenkolber, M., Hasford, J., Rottenkolber, D., Farker, K., Drewelow, B., and Thürmann, P. Self-medication with over-the-counter and prescribed drugs causing adverse-drug-reaction-related hospital admissions: results of a prospective, long-term multi-centre study. Drug safety (2014): 37(4):225-235. doi: 10.1007/s40264-014-0141-3.

[2] Prather, J. C., Lobach, D. F., Goodwin, L. K., Hales, J. W., Hage, M. L., and Hammond, W. E. Medical data mining: knowledge discovery in a clinical data warehouse Proceedings: a conference of the American Medical Informatics Association. AMIA Fall Symposium (1997): 101-105.

[3] Agrawal, R., and Srikant, R. Mining sequential patterns. Proceedings of the Eleventh International Conference on Data Engineering (1995): 3-14

[4] Fournier-Viger, P., Lin, J. C. W., Kiran, R. U., Koh, Y. S., and Thomas, R. A survey of sequential pattern mining. Data Science and Pattern Recognition, 1(1), 54-77(2017)

[5] Norén, G. N., Bate, A., Hopstadius, J., Star, K., and Edwards, I. R. Temporal pattern discovery for trends and transient effects: its application to patient records. In Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining (2008): 963-971. doi: 10.1145/1401890.1402005.

[6] Reps, J., Garibaldi, J. M., Aickelin, U., Soria, D., Gibson, J. E., and Hubbard, R. B. Discovering sequential patterns in a UK general practice database. In Proceedings of 2012 IEEE-EMBS International Conference on Biomedical and Health Informatics (2012): 960-963. doi: 10.1109/BHI.2012.6211748.

[7] Wright, A., Wright, A., McCoy, A., Sittig, D.: The use of sequential pattern mining to predict next prescribed medications. Journal of biomedical informatics 53 (2015): 73-80. doi: 10.1016/j.jbi.2014.09.003.

[8] Helgason, Í.S. Predicting prescription patterns (Doctoral dissertation, Massachusetts Institute of Technology) (2008)

[9] Thibault, M., Lebel, D.: An application of machine learning to assist medication order review by pharmacists in a health care center. (2019). https://doi.org/10.1101/19013029.

[10] Jin, B., Yang, H., Sun, L., Liu, C., Qu, Y., Tong, J. A treatment engine by predicting next-period prescriptions. In Proceedings of the 24th ACM SIGKDD Inter-national Conference on Knowledge Discovery & Data Mining, (2018): 1608-1616. doi: 10.1145/3219819.3220095.

[11] Chen, J. H., Goldstein, M. K., Asch, S. M., Mackey, L., & Altman, R. B. Predicting inpatient clinical order patterns with probabilistic topic models vs conventional order sets. Journal of the American Medical Informatics Association, 24(3), 472-480(2017). doi: 10.1093/jamia/ocw136

[12] Johnson, A., Bulgarelli, L., Pollard, T., Horng, S., Celi, L. A., Mark, R. MIMIC-IV (version 1.0). PhysioNet (2021)

[13] Mooney, C.H., Roddick, J.F. Sequential pattern mining–approaches and algorithms. ACM Computing Surveys (CSUR), 45(2), 1-39 (2013). doi: 10.1145/2431211.2431218.

[14] Zaki, M. J. SPADE: An efficient algorithm for mining frequent sequences. Machine learning, 42(1), 31-60 (2001). doi: 10.1023/A:1007652502315.

[15] Pei, J., Han, J., Mortazavi-Asl, B., Wang, J., Pinto, H., Chen, Q., & Hsu, M.C. Mining sequential patterns by pattern-growth: The prefixspan approach. IEEE Transactions on knowledge and data engineering, 16(11), 1424-1440 (2004). doi: 10.1109/TKDE.2004.77
[16] Fournier-Viger, P., Lin, J. C. W., Gomariz, A., Gueniche, T., Soltani, A., Deng,Z., & Lam, H. T. The SPMF open-source data mining library version 2. In Joint European conference on machine learning and knowledge discovery in databases (pp. 36-40) (2016).
[17] Martin, C. B., Hales, C. M., Gu, Q., & Ogden, C. L. Prescription drug use in the United States, 2015–2016, (2019): 1-8.