

# Sequential Association Rule Mining Revisited: A Study Directed at Relational Pattern Mining for Multi-morbidity

Alexandar Vincent-Paulraj, Girvan Burnside, Frans Coenen, Munir Pirmohamed, and Lauren Walker

University of Liverpool, Liverpool, UK

**Abstract.** Sequential rule mining is a well-established data mining technique for binary valued data. Many variations have been proposed, most approaches use the support-confidence-lift framework. Existing approaches make assumptions concerning the definition of what a sequence is. However, this definition is application dependent. In this paper we look at sequential rule mining with respect to multi-morbidity disease prediction which entails a rethink of the definition of what a sequence is, and a consequent rethink of the operation of the support-confidence-lift framework. A novel sequential rule mining algorithm is proposed designed to address the challenge of multi-morbidity disease prediction. The SEquential RELational N-Disease Pattern (SERENDIP) algorithm.

**Keywords:** Sequential Rule Mining, Multi-morbidity Disease Prediction

## 1 Introduction

Sequential Association Rule Mining (SARM) is an established extension of Association Rule Mining (ARM). The fundamental foundation of ARM, and by extension SARM, is the *support-confidence-lift* framework, used to distinguish good quality rules from poor quality rules and to limit computational complexity [2, 10]. The idea is to identify frequently occurring sequences within a binary valued input data set and to use these frequent item sets to formulate Association Rules (ARs). Frequency in this context is defined in terms of a *support threshold*  $\sigma$ . The value of  $\sigma$  also serves to limit the number of frequent item sets discovered using the “downward-closure property of item sets” principle. Hence many ARM and SARM algorithms operate in what is referred to as an Apriori manner, after the Apriori algorithm described in [1]. The issue that the downward closure property addresses is that there are  $2^n - 1$  candidate item sets given a binary valued data set comprised of  $n$  items (columns).

To limit the number of rules generated a confidence threshold  $\lambda$  is also used to prune the rule set so that only “high confidence” rules are retained. However, just because we have high confidence in a rule this does not mean that it is necessarily

a good rule. We would like to know the correlation between the antecedent and consequent of the rule. This is typically done using a the *lift* measure. If the lift for a rule is greater than one there exists a positive correlation between the antecedent and consequent, if the lift is less than one there is a negative correlation, and if the lift is equal to one there is no correlation. Typically we are interested in positive correlations.

Traditional approaches to ARM assumes the items are independent of one another, the input simply comprises a bag of items; the order in which items appear in a rule is not important. SARM, however, assumes that there exists an ordering over the items and that we are looking for frequent sequences of items to translate into Sequential ARs (SARs). Traditional approaches to SARM [6–8, 16] assume a sequence is any set of items that occur in order, which may be preceded and/or proceeded with other items or be interrupted by other items. An exemplar application domain where this assumption is valid, and that used for reference with respect to the work presented in this paper, is the retail domain. However, this assumption is not valid with respect all application domains where sequential patterns are of interest. One example of the latter is in the context of multi-morbidity disease prediction.

Multimorbidity, the presence of two or more Long-Term health Conditions (LTCs), arises from combinations of physical and mental health conditions that often require the use of daily preventative medicines. People with multimorbidity are major users of health care resources [9, 14]. Certain conditions predictably occur together as they share a common aetiology; consequently the behavioural, environmental and genetic risk factors that contribute to one condition are applicable across all [11]. A good example of this is obesity, high blood pressure, diabetes and heart disease. The organisation of care for individuals with these commonly co-occurring disease clusters is relatively well understood and the treatment goals are aligned (i.e. improving one condition will improve the others within the cluster). What is less well understood is how these conditions accumulate over time; an understanding of the order (sequence) of such co-occurring conditions would provide for their better management, and thus outcomes. SARM would seem to provide the answer. However, unlike traditional approaches to SARM, given a sequence of conditions, any preceding conditions are important while at same time any identified sequence should not be interrupted by additional conditions. In this paper we explore what the support-confidence-lift framework means in the context of SARM for multi-morbidity analysis and prediction, and propose a solution, the SEquential RELational N-Disease Pattern (SERENDIP) algorithm. A feature of this algorithm is the usage of an *occurrence count matrix* to provide efficiency gains. The algorithm has been published at <http://serendip.org.uk>, where it can be run as a client-server application using users' own data, or using the provided demonstration test data

The remainder of this paper is structured as follows. Section 2 presents some background to the technical challenge that this paper seeks to address. Section 3 presents a discussion of the support-confidence-lift framework in the context of multi-morbidity disease prediction. Section 4 then presents the proposed

SERENDIP algorithm. The evaluation of the proposed approach is presented in Section 5. The main findings and conclusions are the presented in Section 6, together with some suggestions for future work.

## 2 Background

Sequential pattern mining has been extensively used to mine patterns from sequence databases, and then to express those patterns as SARS. Typically the well-established support-confidence-lift framework is adopted. Example algorithms include: GSP (Generalised Sequential Patterns) [3], SPADE (Sequential Pattern Discovering using Equivalence classes) [16] and SPAM (Sequential Pattern Mining) [4]. These algorithms use different approaches for candidate generation, and their performance varies with volume of data and computation infrastructure. For example, GSP does not use all the frequent items in a database, it works on time bounds to mine patterns, and tends to be used for generalised pattern mining. The SPAM algorithm stores sequence patterns using a bitmap compression technique (SARM algorithms generate a lot of intermediate data), and significance support counting. SPADE is a depth-first search algorithm which employs vertical formatting to mine patterns. It has been shown to be efficient for the mining of large data sets and support counting. A more recent discussion can be found in [13] where the use of Hadoop-MapReduce is proposed with respect to a retail sequential pattern mining scenario.

What all of the above algorithms have in common is that they assume that it does not matter if a sequences, in the input data, is preceded and/or interrupted by other items, for it to be valid sequence. As noted above, this assumption does not hold with respect to all applications. One example, and that of interest with respect to this paper, is SARM for multi-morbidity disease prediction.





## 3 The traditional versus the Sequential Support-Confidence-Lift Framework

In the traditional approach to ARM the input is a data set  $D = \{r_1, r_2, \dots, r_n\}$  where each record comprises an item set  $\{a_1, a_2, \dots, a_m\}$  taken from a super-set of items  $A$ , thus  $r_i \subset A$ . In shopping basket analysis  $A$  is a set of products that might be purchased in a single transaction. In SARM the ordering becomes important. Thus instead of a set we have a list  $[a_1, a_2, \dots, a_m]$  such that (say) product  $a_i$  was bought before product  $a_{i+1}$ . In both cases the concept of a support count and the downward closure property of item sets still holds, but the way that they are calculated differs. In the traditional approach the support for an item set  $I$ , some subset of  $A$ , is simply the occurrence count of  $I$  in  $D$ . Usually it is more convenient to express this as the probability of  $A$  appearing in  $D$  (Equation 1).

$$support_{trad}(I) = \frac{\text{occurrence count of } I \text{ in } D}{n} \quad (1)$$

The confidence of a rule  $X \Rightarrow Y$  would then be calculated using Equation 2 where  $X, Y \in A$  and  $X \cap Y = \emptyset$ .

$$conf_{trad}(X \Rightarrow Y) = \frac{support_{trad}(\{X \cup Y\})}{support_{trad}(\{X\})} \quad (2)$$

Example Sequential Patterns	Retail Exemplar Scenario	Multi-morbidity Disease Prediction Scenario
	✓	✓
	✓	x
	✓	x
	✓	x

**Fig. 1:** Distinction between frequent sequential patterns relevant to a retail scenario and a multi-morbidity scenario (sequential patterns indicated by shaded boxes, additional items by filled boxes).

To determine what we mean by support in a sequential setting we first need to establish the nature of the sequential ARs we wish to generate. If we consider the sequential frequent pattern  $[a, b, c, d]$ , where  $\{a, b, c, d\} \in A$ , we can identify the rule  $[a] \Rightarrow [b]$ , if  $a$  happens then  $b$  will happen next. But is  $[a] \Rightarrow [c]$  also a valid rule that can be extracted from this record? What about  $[b] \Rightarrow [c]$ ? Whether these two rules are valid or not depends on whether we are interested purely in the order in which things happen and that, in the case  $[a] \Rightarrow [c]$  the fact that  $b$  happens between  $a$  and  $c$  is not important and in the case of  $[b] \Rightarrow [c]$  the fact that  $a$  precedes  $b$  is not important; or that we are interested in the actual sequences and preceding items and intervening items are important. In the case of multi-morbidity disease sequences, the application focus of this paper, any preceding and intervening diseases are important as they may very well have an influence on what happens next. The point is illustrated in Figure 1 which presents a comparison between the sequential patterns that are relevant in the context of a retail scenario and those that are relevant in the context of a multi-morbidity disease prediction scenario. Thus, in the case of the multi-morbidity application scenario, and given the example frequent sequential pattern  $[a, b, c, d]$ , we should identify the following valid SARs:

$$\begin{aligned} [a] \Rightarrow [b] \quad [a] \Rightarrow [b, c, d] \quad [a, b] \Rightarrow [c, d] \\ [a] \Rightarrow [b, c] \quad [a, b] \Rightarrow [c] \quad [a, b, c] \Rightarrow [d] \end{aligned}$$

Thus, in the case of sequential ARM, and given our multi-morbidity application domain where all preceding and intermediate diseases are important, support should be calculated as shown in Equation 3, where  $n$  is the number of records, and confidence is calculated as shown Equation 4. The notation  $X + Y$  indicates the concatenation of sub-sequence  $Y$  to the end of sub-sequence  $X$ .

$$support_{seq}(X) = \frac{\text{occurrence count where } X \text{ is a leading subsequence}}{n(I)} \quad (3)$$

$$conf_{seq}(X \Rightarrow Y) = \frac{support_{seq}(X + Y)}{support_{seq}(X)} \quad (4)$$

If we assume the following trivial data set of sequences  $D$ :

$[a, b, c, d]$   
 $[c, b, d]$   
 $[d, c, a]$   
 $[a, b, d]$   
 $[a, d, c, b]$

and consider the rule  $[a] \Rightarrow [b]$ . The sequential support for  $[a]$  is  $3/5 = 0.6$ , and the sequential support for  $[a, b]$  is  $2/5 = 0.4$ . The confidence is then  $0.4/0.6 = 0.67$ .

Traditionally the lift of a rule is calculated as shown in Equation 5. In the case of sequential ARM, and given the constraints imposed by our multi-morbidity application domain, lift is calculated as indicated by Equation 6 where the support of the consequent is calculated as shown in Equation 7. The variable  $i$  is the index of the start of the consequent, calculated as  $|X| + 1$  where  $X$  is the number of items in  $X$  (the antecedent of the rule). Thus, the lift for our sequential example rule  $[a] \Rightarrow [b]$  (see above) is  $0.67/0.60 = 1.11$ .

$$lift_{trad}(X \Rightarrow Y) = \frac{conf_{trad}(X \Rightarrow Y)}{support_{trad}(Y)} \quad (5)$$

$$lift_{seq}(X \Rightarrow Y) = \frac{conf_{seq}(X \Rightarrow Y)}{support_{consequent}(Y, i)} \quad (6)$$

$$support_{consequent}(Y, i) = \frac{\text{occurrence count where } Y \text{ is at position } i}{n} \quad (7)$$

The sequential support-confidence-lift framework as described above, in the context of multi-morbidity disease prediction, requires a rethink of the traditional Apriori approach to ARM and established sequential ARM algorithms such as SPADE [16] where a sequence can be preceded by items not included in the sequence, and that there may be intervening items in the data set that are not included in the sequence. This is considered in the following section, Section 4.

---

**Algorithm 1** Sequential ARM for Multi-morbidity ( $|A|, D, max, \sigma$ )

---

```
1:  $M = occurrenceCountMatrixGeneration(max, |A|)$  (Algorithm 2)
2:  $I_1 = \emptyset$ , set to hold frequent one-item patterns
3: for  $j = 1$  to  $j = |A|$  do
4:   if  $M_{1,j} \geq \sigma$  then
5:      $I_1 = I_1 \cup \{a_j\}$ , add  $a_j \in A$  to the set of frequent one-item sets  $I_1$  so far
6:   end if
7: end for
8: Add content of  $I_1$  to the set enumeration tree.
9:  $C_2 = generateCandidateItemset(k, I_1, M)$  (Algorithm 3)
10:  $k = 2$ 
11: while  $C_k \neq \emptyset$  do
12:    $I_k = \emptyset$ , set to hold frequent  $k$  item sets if any
13:   for  $\forall S_i \in C_k$  do
14:     if  $support(S_i) \geq \sigma$  then
15:        $I_k = I_k \cup S_i$ 
16:     end if
17:   end for
18:   Add  $I_k$  to the set enumeration tree.
19:    $k++$ 
20:    $C_k = generateCandidateItemset(k, I_{k-1}, M)$  (Algorithm 3)
21: end while
22: Step through the set enumeration tree and generate a set of SARs.
```

---

## 4 The SEquential RELational N-Disease Pattern (SERENDIP) Algorithm

This section presents the proposed SEquential RELational N-Disease Pattern (SERENDIP) algorithm for multi-morbidity disease prediction where the preceding and intervening items within a given sequence are important; in other words, the proposed algorithm operates using a different definition of a item sequence than that used by established SARM algorithms such as those presented in [3, 4, 16]. Using the proposed algorithm, the identified frequent item sets are held in a “set enumeration tree” structure where each node holds: (i) a single item set label, (ii) a support value and (iii) a set of links to child nodes (or “null” if there are no child nodes). This offers the advantage of fast look up and efficient storage as oppose to the alternative nested set of arrays approach. The use of set enumeration trees in ARM is well established with respect to existing algorithms (see for example [5]).

The pseudo code for SERENDIP is given in Algorithm 1. The inputs are: (i) the input data set  $D$ , (ii) the set of attributes  $A$  that feature in  $D$ , (iii) the maximum length  $max$  of a record in  $D$ , and (iv) the support threshold  $\sigma$ . The variables  $D$  and  $A$ , input data and the associated set of attributes respectively, are assumed to be global variables. We commence, line 1, by generating a  $max \times |A|$  “occurrence count matrix”  $M$  to hold single item occurrence counts according to the position of the items in each records in  $D$  (an item may, of course, not

---

**Algorithm 2** *occurrenceCountMatrixGeneration(max, |A|)*

---

```
1:  $M = max \times |A|$  matrix with 0 values
2: for  $k = 1$  to  $|D|$  do
3:   for  $i = 1$  to  $i = |r_k|$  ( $r_k \in D$ ) do
4:      $j =$  index of attribute  $a_i \in r_k$  w.r.t.  $A$ 
5:      $M_{i,j} = M_{i,j} + 1$ 
6:   end for
7: end for
8: return  $M$ 
```

---

exist in a particular record). A value at  $m_{i,j} \in M$  is the occurrence count of attribute  $a_j \in A$  when at index  $i$  in  $D$ . The usage of  $M$  provides for efficiency gains in that the individual support values need only be calculated once. The pseudo code for generating  $M$  is given in Algorithm 2. The inputs are: (i) the maximum length  $max$  of a record in  $D$ , and (ii) the number of attributes  $|A|$  in the set  $A$  from which the items in records can be drawn. Note that by definition  $max \leq |A|$ . The  $max \times |A|$  matrix is defined in line 1. We then, line 2, step through the data set  $D$  record by record ( $k$  is the record index). For each record  $r_k \in D$  we then, line 3, step through the record attribute by attribute ( $i$  is the location index in a record). For each attribute  $a_i \in r_k$  we obtain its index  $j$  with respect to the set  $A$  (line 4). The index  $j$  is the column index in  $M$  and  $i$  the row index in  $M$ . We then, line 5, increment the value in  $M$  at row index  $i$  and column index  $j$ ; thus  $m_{i,j}$ . The occurrence count matrix thus holds information on the frequency of occurrence of individual items according to their position in the individual sequences (records) in  $D$ .

Returning to Algorithm 1, the next stage is to identify the frequently occurring one item sets and place these in a set  $I_1$  (lines 2 to 7). The set  $I_1$  is declared on line 2. We then step through the first row in the occurrence count matrix  $M$  (this  $i = 1$ ). Each value  $m_{1,j}$  is compared against the support threshold  $\sigma$ , and if  $m_{1,j} \geq \sigma$  the associate attribute,  $a_j \in A$  is added to  $I_1$ . Once we have the complete set of frequent one item sets,  $I_1$  these are added as child nodes of the root node in the set enumeration tree (line 8).

---

**Algorithm 3** *generateCandidateItemset(k, I, M)*

---

```
1:  $C_k = \emptyset$ 
2: for  $\forall X_i \in I$  do
3:   for  $\forall m_{k,j} \in M$  do
4:     if  $m_{k,j} \geq \sigma$  and  $a_j \notin X_i$  then
5:        $C_k = C_k \cup (X_i + [a_j])$  where  $a_j \in A$ 
6:     end if
7:   end for
8: end for
9: return  $C_k$ 
```

---

We are now in a position to generate the two-item candidate set  $C_2$  by stepping through  $I_1$  and, for each items set  $X_i \in I_1$ , appending the associated attributes from the second row in  $M$  ( $j = 2$ ) to  $X_i$  if the associated support value at  $m_{2,j} \geq \sigma$ , and provided that the associated attribute is not already in  $X_i$ . The pseudo code for candidate item set generation is given in Algorithm 3. The inputs to the algorithm are: (i) the item set size  $k$  for the candidate items we wish to generate ( $k = 2$  for two item sets); (ii) the  $k - 1$  sequential frequent item sets already discovered, the set  $I_{k-1}$ ; and (iii) the occurrence count matrix  $M$ . Note that  $k$  will also be the relevant first index for  $M$ , The algorithm commences, line 1, by declaring the empty set  $C_K$  in which to hold the candidate sets. We then step through  $I_{k-1}$  and for each item set  $X_i$  in  $I_{k-1}$  we add attributes  $a_j$  from  $A$  to  $X_i$  to form candidate item sets by stepping through row  $k$  in  $M$ . In each case we create a candidate item set if: (i)  $a_j$  is not already in  $X_i$  and (ii) the support for  $a_j$  in  $M$ ,  $m_{k,j}$ , is greater or equal to  $\sigma$ . On completion the algorithm returns the set of candidate  $k$  item sets,  $C_k$ .

Returning to Algorithm 1, the algorithm next enters into a Apriori style “generate and test” loop (lines 11 to 21), testing the identified candidate item sets against the threshold  $\sigma$  and generating new candidate item sets. The loop continues until no more candidate item sets can be generated. As the loop progresses the set enumeration tree is further populated (line 18).

In the case of the multi-morbidity application domain, we wish to use the sequential ARs for prediction purposes. Thus we are interested in sequential Classification Association Rules (sequential CARs) [12] where the consequent is a single item, the class we wish to predict. In the case of the multi-morbidity application, the class we are interested in is the next disease that a patient can be expected to get. To calculate the lift for a rule  $X \Rightarrow Y$  we need the consequent support for  $X$ . This can be obtained directly from the occurrence count matrix  $M$  generated earlier, and would be the value at  $M_{i,j}$  where  $i$  is the index of interest ( $i = |X| + 1$ ) and  $j$  is the index of  $Y \in A$ . Note that this assumes the consequent is comprised of a single item as required in the context of sequential CARs, but could easily be adjusted to calculate consequent supports for consequents comprised of more than one item.

These rules can now be used for prediction purposes. Given a query antecedent  $Q$  we start at the top of the list looking for matches with the listed rule antecedents. It is possible that we have rules with the same antecedent, but different consequents. The ordering of the rules in the list are therefore important. There are a range of schemes that can be used for rule ranking [15], typically founded on confidence, lift and rule antecedent size. In the case of our multi-morbidity application an exact match is required for a rule to be fired, because of the interplay of diseases, so the size of the rule antecedents is not important. Some applications use subset “matching”, where  $Q$  is only required to be a subset of a rule antecedent for the rule to be fired, in which case more specific rules, rules whose antecedents have a large number of items, should be listed first. The example rules listed in the following section have been ranked using confidence as the primary ranking and lift as the secondary ranking. We



can also simply fire the first rule we get to, or the top  $k$  and use some voting mechanisms should conflicting classes be predicted. Only the first is applicable given that we require exact matching.

**Table 1:** Statistics analysis of the study population

Gender	Population
Male	45281
Female	54009
Total	99290

Age Band	Population
50-54	6093
55-59	12366
60-64	12687
65-69	13248
70-74	14553
75-79	12877
80+	27466
Total	99290

## 5 Evaluation

The proposed SERENDIP algorithm was evaluated using a set of multi-morbidity patient records obtained from the Clinical Practice Research Datalink (CPRD)<sup>1</sup>. CPRD is a large electronic health record database that contains anonymised health records of primary health care patients in the United Kingdom. It includes over twenty million patient records of which some five million patients were active at the time of writing. Approval was obtained from the Independent Scientific Advisory Committee (ISAC) in order to use the data for this study (Protocol No. 19159R1). Primary-care patient-level data from a random sample of hundred thousand patients was extracted for the period (1920 – 2020) for patients aged 50 and over (the age group most likely to be affected by multi-morbidity). Patients were considered eligible for inclusion if they had been registered in a general practice for a minimum of two years and their record indicating diagnosis of two or more recorded Life Threatening Conditions (LTCs). Diagnoses are recorded in CPRD using a coding system. Long-term conditions and associated case definitions were determined by reference to a clinical group with broad generalist and prescribing expertise, including two of the authors. Diagnostic code lists were developed and adapted from previous studies. The code lists are available online<sup>2</sup>. Some statistics concerning the evaluation data are given in Table 1. From the table it can be seen that female patients outnumber male patients, however this was to be expected given the over 50 age group under consideration.

For the evaluation results presented here,  $\sigma = 0.00005$  (0.005%) and  $\lambda = 0.1$  (10%) were used. A low value for  $\sigma$  was deliberately selected to ensure no relevant sequences were missed. The generated rules were presented to a clinical group for inspection. In total 1261 rules were identified using SERENDIP. Example rules are presented in Tables 2, 3, 4 and 5; the top ten two, three and four-item rules,

<sup>1</sup> <https://www.cprd.com>

<sup>2</sup> <http://hammerai.co.uk>

and the top five five-item rules, according to confidence respectively. Column one gives the rule and column two the support, confidence and lift respectively (calculated as shown in Equations 3, 4 and 6).

**Table 2:** Two Disease Sequential ARs generated using SERENDIP

Rule	Support	Confidence	Lift
Chronic Constipation $\Rightarrow$ Chronic Pain	0.00329	0.46714	5.03502
Allergic and Chronic Rhinitis and ... $\Rightarrow$ Asthma	0.00006	0.42857	17.24184
Abdominal Aortic Aneurysm $\Rightarrow$ Lipid Disorder	0.00005	0.38462	5.70573
Stroke CVA and Hypertension $\Rightarrow$ Lipid Disorder	0.00005	0.38462	5.70573
Polycythaemia Vera $\Rightarrow$ Hypertension	0.00005	0.33333	4.33259
Diabetic Eye Disease $\Rightarrow$ Diabetes	0.00028	0.32184	20.41879
Diabetes and Hypertension $\Rightarrow$ Lipid Disorder	0.00007	0.31818	4.72020
Ankylosing Spondylitis $\Rightarrow$ Chronic Pain	0.00041	0.31298	3.37337
Stable Angina $\Rightarrow$ Coronary Heart Disease	0.00020	0.29412	21.11565
Coronary Heart Disease $\Rightarrow$ Lipid Disorder	0.00355	0.29163	4.32633

**Table 3:** Three Disease Sequential ARs generated using SERENDIP

Rule	Support	Confidence	Lift
Chronic Constipation, Abdominal Hernia $\Rightarrow$ Chronic Pain	0.00005	0.80000	12.18171
Cancer Solid organ, Chronic Constipation $\Rightarrow$ Chronic Pain	0.00008	0.70000	10.65900
Diverticular Disease, Chronic Constipation $\Rightarrow$ Chronic Pain	0.00014	0.66667	10.15143
Anxiety, Chronic Constipation $\Rightarrow$ Chronic Pain	0.00005	0.66667	10.15143
Cataract, Chronic Constipation $\Rightarrow$ Chronic Pain	0.00005	0.66667	10.15143
Chronic Constipation, Osteoarthritis excluding spine $\Rightarrow$ Chronic Pain	0.00005	0.66667	10.15143
Chronic Constipation, Spondylosis $\Rightarrow$ Chronic Pain	0.00005	0.66667	10.15143
Diabetic eye disease, Hypertension $\Rightarrow$ Diabetes	0.00006	0.62500	30.16532
Gastritis and Duodenitis, Gout $\Rightarrow$ Chronic Pain	0.00006	0.62500	9.51696
Spondylosis, Chronic Constipation $\Rightarrow$ Chronic Pain	0.00006	0.62500	9.51696

From the tables it can be seen that many of the rules feature low support, hence the selection of a low value for  $\sigma$  to ensure no significant rules were missed was appropriate. Some rules feature very high confidence, for example the rule:

Diabetes, Hypertension, Lipid Disorder, Diabetic eye disease  $\Rightarrow$  Chronic  
Kidney Disease

in Table 5 featured a confidence of 1.00000 (100%). The high lift values that feature in the results are also interesting. Recall that a lift greater than one indicates a positive correlation. For example if we consider the rule:

Stable Angina  $\Rightarrow$  Coronary Heart Disease

in Table 2, which had a lift of 21.11565 this indicates that as the incidence of “Stable Angina” (chest pain due to poor blood flow through the heart) increases we can expect a significant increase in the incidence of Coronary Heart Disease. Some of the rules had relatively low confidence but even a confidence of 0.25000 (25%), correlated with a lift greater than one, provides a good indicator of a likely follow on condition. Consultation with domain experts indicated that the rules that had been discovered “made sense”.

**Table 4:** Four Disease Sequential ARs generated using SERENDIP

Rule	Support	Confidence	Lift
Abdominal Hernia, Chronic Sinusitis, Hypertension ⇒ Lipid Disorder	0.00006	0.80000	10.42112
Hypertension, Chronic Kidney Disease, Diabetic eye disease ⇒ Diabetes	0.00006	0.80000	31.54886
Hypertension, Lipid Disorder, Diabetic eye disease ⇒ Diabetes	0.00012	0.72727	28.68078
Diabetes, Diabetic eye disease, Erectile Dysfunction ⇒ Hypertension	0.00006	0.66667	8.95264
Psoriasis, Chronic Pain, Thyroid Problem ⇒ Hypertension	0.00006	0.66667	8.95264
Chronic Pain, Abdominal Hernia, Chronic Kidney Disease ⇒ Hypertension	0.00006	0.57143	7.67369
Lipid Disorder, Chronic Pain, Erectile Dysfunction ⇒ Osteoarthritis excluding spine	0.00006	0.57143	14.26342
Chronic Pain, Chronic Constipation, Osteoporosis ⇒ Thyroid Problem	0.00007	0.55556	30.39148
Lipid Disorder, Hypertension, Diabetic eye disease ⇒ Diabetes	0.00007	0.50000	19.71804
Osteoarthritis excluding spine, Hypertension, Coronary Heart Disease ⇒ Lipid Disorder	0.00007	0.50000	6.51320

**Table 5:** Five Disease Sequential ARs generated using SERENDIP

Rule	Support	Confidence	Lift
Diabetes, Hypertension, Lipid Disorder, Diabetic eye disease ⇒ Chronic Kidney Disease	0.00005	1.00000	19.33938
Chronic Pain, Obesity, Hypertension, Osteoarthritis excluding spine ⇒ Lipid Disorder	0.00005	0.60000	8.40091
Hypertension, Lipid Disorder, Chronic Pain, Diabetes ⇒ Diabetic eye disease	0.00005	0.42857	33.14166
Chronic Pain, Hypertension, Lipid Disorder, Chronic Kidney Disease ⇒ Diabetes	0.00005	0.33333	11.32475
Chronic Pain, Lipid Disorder, Hypertension, Chronic Kidney Disease ⇒ Osteoarthritis excluding spine	0.00005	0.30000	7.17283

## 6 Conclusions

This paper has presented the SERENDIP algorithm for SAR extraction, a feature of the algorithm is the usage of an occurrence count matrix  $M$ . The motivation for the work was the observation that existing SARM algorithms make certain assumptions about what a sequence is, typically permitting a sequence to be preceded by additional items and/or be interrupted by additional items. This assumption holds with respect to many applications, such as retail analysis and prediction, but those not hold for all applications. One such application, and the focus for the work presented in this paper, is multi-morbidity disease prediction where preceding and intervening conditions are important. In this case existing SARM algorithms are inappropriate because of the assumptions made, and the consequent way in which metrics such as support, confidence and lift are calculated. The proposed SERENDIP algorithm addresses these issues. For future work the authors intend to investigate mechanisms where by the variable interval between multi-morbidity conditions can be included in the SAR extraction process, currently a unit interval is adopted. The SERENDIP algorithm has been published at <http://serendip.org.uk> as a www service, where it can be run as a client-server application using a user's own data.

## References

1. Rakesh Agarwal and Ramakrishnan Srikant. Fast algorithms for mining association rules. In *Proc. of 20th Intl. Conf. on VLDB*, pages 487–499, 1994.
2. Rakesh Agrawal, T. Imieliński, and A. A. Swami. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD international conference on Management of data - SIGMOD '93*, page 207, 2019.
3. Rakesh Agrawal and Ramakrishnan Srikant. Mining sequential patterns. In *Proceedings of the Eleventh International Conference on Data Engineering*, pages 3–14, 1995.
4. Jay Ayres, Johannes Gehrke, Tomi Yiu, and Jason Flannick. Sequential pattern mining using a bitmap representation. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD'02)*, page 429–435, 2002.
5. Frans Coenen, Paul Leng, and Shakil Ahmed. Data structure for association rule mining: T-trees and p-trees. *IEEE Transactions on Knowledge and Data Engineering*, 16:774–778, 2004.
6. Philippe Fournier-Viger, Usef Faghihi, Roger Nkambou, and Engelbert Mephu Nguifod. Cmrules: Mining sequential rules common to several sequences. *Knowledge-Based Systems*, 22:63–76, 2012.
7. Philippe Fournier-Viger and Ted Gueniche nd Souleymane Zida nd Vincent S. Tseng. Erminer: Sequential rule mining using equivalence classes. In *Proceedings International Symposium on Intelligent Data Analysis (IDA 2014)*, pages 108–119, 2014.
8. Sherri K. Harms and Jitender S. Deogun. Sequential association rule mining with time lags. *Journal of Intelligent Information*, 22:7–225, 2004.

9. Anna Head, Kate Fleming, Christodoulos Kypridemos, Pieta Schofield, and Martin O’Flaherty. Dynamics of multimorbidity in england between 2004 and 2019: a descriptive epidemiology study. *Eurpoean Journal of Public Health*, 2020.
10. Manpreet Kaur and Shivani Kang. Market basket analysis: Identify the changing trends of market data using association rule mining. *Procedia Computer Science*, 85:78–85, 2016.
11. Rokas Navickas, Vesna-Kerstin Petric, Andrea B Feigl, and Martin Seychell. Multimorbidity: What do we know? what should we do? *Journal of Comorbidity*, 6(1):4–11, 2016.
12. Cynthia Rudin, Benjamin Letham, Ansaf Salieb-Aouissi, Eugene Kogan, and David Madigan. Sequential event prediction with association rules. *Proceedings of Machine Learning Research*, 19:615–634, 2011.
13. Neha Verma and Jatinder Singh. A comprehensive review from sequential association computing to hadoop-mapreduce parallel computing in a retail scenario. *Journal of Management Analytics*, 5(4):359–392, 2017.
14. Christine Vogeli, Alexandra E Shields, Todd A Lee, Teresa B Gibson, William D Marder, Kevin B Weiss, and David Blumenthal. Multiple chronic conditions: Prevalence, health consequences, and implications for quality, care management, and costs. *Journal of General Internal Medicine*, 22(3):391–395, 2007.
15. Yanbo J. Wang, Qin Xin, and Frans Coenen. Hybrid rule ordering in classification association rule mining. *Transactions on Machine Learning and Data Mining in Pattern Recognition*, pages 1–16, 2008.
16. Mohammed J. Zaki. Spade: An efficient algorithm for mining frequent sequences. *Interantional Journal of Machine Learning*, 42(1–2):31–60, 2001.