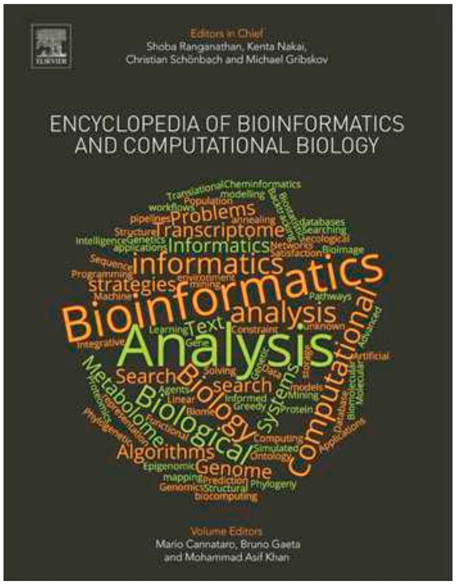

Editors in Chief
Shoba Ranganathan, Kenta Nakai,
Christian Schönbach and Michael Gribskov
ENCYCLOPEDIA OF BIOINFORMATICS
AND COMPUTATIONAL BIOLOGY
Volume Editors
Mario Cannataro, Bruno Gaeta
and Mohammad Asif Khan

Giuseppe Di Fatta (2019) Association Rules and Frequent Patterns. In: Ranganathan, S., Gribskov, M., Nakai, K. and Schönbach, C. (eds.), Encyclopedia of Bioinformatics and Computational Biology, vol. 1, pp. 367–373. Oxford: Elsevier.

Association Rules and Frequent Patterns

Giuseppe Di Fatta, University of Reading, Reading, United Kingdom

Introduction

Association Rule Mining (ARM) (Agrawal *et al.*, 1993; Agrawal and Srikant, 1994; Hipp *et al.*, 2000) is often referred to as frequent pattern mining (Goethals, 2003; Han *et al.*, 2007; Aggarwal and Han, 2014; Fournier-Viger *et al.*, 2017). Frequent pattern mining focuses on the extraction of the frequent patterns, ARM also offers a specific and more descriptive representation of the frequent patterns in the form of association rules: If the pattern X is present, then also the pattern Y is. Anyway, since ARM relies on the extraction of the frequent patterns, which is a particularly complex task, the two are often considered equivalent, at least, from the computational point of view. In the most common formulation of the ARM problem, patterns take the form of sets and, more specifically, of sets of items (itemsets). An item is a binary feature, such as the presence of a specific attribute or property. In general, ARM is applied to extract and explicitly represent events (attributes) that occur together in the data and more frequently than others. The assumption is that events that frequently occur together are more important of those not occurring together or not frequently enough. For example, in market basket analysis the itemset is the set of items purchased by a customer within a single transaction. The interesting relations to be exposed are the explicit or implicit associations made by customers in emerging shopping behaviours.

In the life sciences the adoption of ARM to address some data-intensive problems (Atluri *et al.*, 2009) did not receive the same attention as for other statistical and data mining techniques, such as correlation, regression, classification and clustering. However, since the widespread adoption of genomic sequencing and other high-throughput digital technologies, very large datasets have become more common and often publicly available. For example, mining association rules in genomic data (Hanash, 2003; Alves *et al.*, 2010; Oellrich *et al.*, 2014; Chen *et al.*, 2015) has become of more interest and adopted for two main tasks: identifying the significant patterns in subsets of genes and between gene regulation and phenotype.

Specific domain knowledge is useful to understand if the representation of the itemsets as binary attributes may force an undesired asymmetric semantics. The presence of an attribute in a sample is encoded by the presence of an item in the itemset: the item typically has an explicit positive meaning. The lack of an attribute often is not significant and is not explicitly encoded in the itemset. However, when this is not the case and the lack of some attribute is of interest, a specific 'negative' item can be used to encode this information explicitly in the itemsets. Alternatively Negative Association Rules (Antonie and Zaïane, 2004) can be applied to extract relations, for example, between the absence of an item and the presence of others.

Arguably some limiting factors have hindered a more widespread adoption of ARM. Firstly, the discovery of interesting patterns from data requires complex combinatorial algorithms and often these benefit from high performance computing. For real-world problems the search space may be prohibitively large. Secondly, it is not uncommon that even for a relatively small input dataset, the set of discovered patterns is very large, even larger than the input dataset. In this case ARM is only a first step in more complex data workflows that include other data mining techniques. For these reasons, a multidisciplinary approach that combines expertise from both computer science and the specific application domain is often critical.

The rest of the article is organised as follows. Section "Problem Definition" introduces the basic ARM problem and discuss the combinatorial nature of the computational task. Some important ARM algorithms, the search space of patterns and some interestingness criteria are presented in Section "ARM Algorithms". Some extended ARM problems are briefly reviewed in Section "Extended Association Rule Mining" and, finally, Section "Conclusions" provides some conclusive remarks.

Problem Definition

Consider a set of binary attributes $\mathcal{I}=\{A, B, C, \ldots\}$ ($|\mathcal{I}|=d$) called items. A proper subset $s \subset \mathcal{I}$ is referred to as *itemset*. A *k–itemset s* is an itemset of k items, i.e., $|s|=k$. A transaction over $\mathcal{I}$ is a pair $t=<id, s>$, where *id* is the transaction identifier and s is an itemset. $\mathcal{T}$ is the set of transactions ($|\mathcal{T}|=n$), a transactional database. A transaction $t=<id, s>$ is said to support an itemset x, if $s \supseteq x$. For example, the transaction $<0F36A, \{BCDH\}>$ supports the itemset $\{BD\}$ and it does not support the itemset $\{BDE\}$. The support σ of an itemset x is the fraction of transactions that support x, i.e., $\sigma(x) = \frac{|\{<id,s> \in \mathcal{T}, s \supseteq x\}|}{n}$.

An association rule (Agrawal and Srikant, 1994) is an implication expressed in the form $X \Rightarrow Y$ ("X implies Y", "X then Y"), where X and Y are itemsets ($X, Y \subset \mathcal{I}$) with $X \cap Y = \emptyset$. The left term is referred to as the antecedent, the right term as the consequent of the rule. The 'support' of a rule is the fraction of transactions that support $X \cup Y$, which corresponds to the empirical joint probability $P(X \cup Y)$.

$$support(X \Rightarrow Y) = \sigma(X \cup Y) \tag{1}$$

The 'confidence' of a rule is the fraction of transactions supporting X that also support Y. The confidence of a rule corresponds to the empirical conditional probability $P(Y|X)$.

doi:10.1016/B978-0-12-809633-8.20333-6

$$confidence(X \Rightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)} \tag{2}$$

A rule is said 'frequent', if it is supported by at least a minimum fraction of transactions in $\mathcal{I}$. A user-defined parameter for the minimum support (*minsup*) is required for this purpose. A rule is said 'strong', if it has a confidence greater or equal to a user-defined parameter *minconf*.

Given a transactional database $\mathcal{I}$ and two user-defined parameters *minsup* and *minconf*, the Association Rule Mining problem is to find all frequent and strong rules.

$$\begin{gathered} X \Rightarrow Y, \quad \text{where} \quad X, Y \subset \mathcal{I}, \quad X \cap Y = \varnothing, \\ \sigma(X \cup Y) \geq minsupp \quad \text{and} \quad \frac{\sigma(X \cup Y)}{\sigma(X)} \geq minconf \end{gathered} \tag{3}$$

The Search Space

The search space of the ARM problem is limited by all possible combinations for the antecedent and the consequent of a rule. The total number of itemsets is the cardinality of the power set, $|Power(I)| = 2^d$.

Although the disjoint constraint reduced the number of possible combinations, they are still expected to be exponential in the number d of items. The total number R of possible association rules is given by all possible itemset combinations for the antecedent times all possible itemset combinations for the consequent, as expressed by Eq. (4). There are $\binom{d}{k}$ possible k-itemsets as antecedent and, given a k-itemset as antecedent, there are $\binom{d-k}{j}$ possible j-itemsets as consequent.

The total number R of possible association rules is given by

$$R = \sum_{k=1}^{d-1} \left[\binom{d}{k} \cdot \sum_{j=1}^{d-k} \binom{d-k}{j} \right] = 3^d - 2^{d+1} + 1 \tag{4}$$

where the following equation derived from the binomial theorem has been applied for the reduction of the expression.

$$\sum_{k=0}^{m} \binom{m}{k} \cdot r^k = (1+r)^m \tag{5}$$

For example, a set of 10 items generates 57,002 rules; while 20 items generate almost 3.5 billion rules. Obviously a brute force approach that enumerates all possible rules and validates them against the minimum support and confidence constraints, is not a feasible solution to the ARM problem.

The computational complexity of the problem with binary attributes and its other variants (e.g., a quantitative approach) has been shown to be NP-complete (Wijsen and Meersman, 1998; Yang, 2004; Angiulli *et al.*, 2001). The challenging combinatorial nature of the problem has given rise to an intense research (Agrawal *et al.*, 1993; Agrawal and Srikant, 1994, 1986; Hipp *et al.*, 2000; Zaki *et al.*, 1997; Savasere *et al.*, 1995; Toivonen, 1996; Brin *et al.*, 1997; Han *et al.*, 2000) and a number of efficient algorithms, the more representative of which are briefly reviewed in the next section.

ARM Algorithms

The observation that the computation of the support of the rule $\sigma(X \cup Y)$ is necessary in both constraints (Eqs. (1) and (2)) suggests an efficient problem decomposition into two sub-problems, which was originally introduced in the *Apriori* algorithm (Agrawal and Srikant, 1994).

1. Find all Frequent Itemsets (FI):

$$FI = \{x | x \subset \mathcal{I} \quad and \quad \sigma(x) \geq minsupp\}$$

2. Generate all strong rules from FI.

In the second step only the confidence of the rules has to be tested, as any rule generated from the FI set is guaranteed to have sufficient support. The second problem can be solved by a straightforward polynomial time algorithm and most of the computational complexity lies in the first problem. All the possible itemsets correspond to the powerset $\mathcal{P}(\mathcal{I})$, where $|\mathcal{P}(\mathcal{I})| = 2^d$. This is still an exponential search space, though $3^d – 2^d + 1 > 2^d$ for $d > 2$, and it requires efficient combinatorial algorithms, optimisation techniques, high performance computing and, ultimately, additional domain-specific constraints to limit the search space.

The powerset $\mathcal{P}(\mathcal{I})$ can be represented as a lattice of sets. For example, **Fig. 1** shows the lattice of all possible itemsets for a set of four items, which are represented as natural numbers ($\mathcal{I} = \{1,2,3,4\}$). The binomial coefficients on the left indicates the number of subsets at each tier of the lattice and the edges indicate a subset-superset relation between itemsets in consecutive tiers. **Fig. 2**

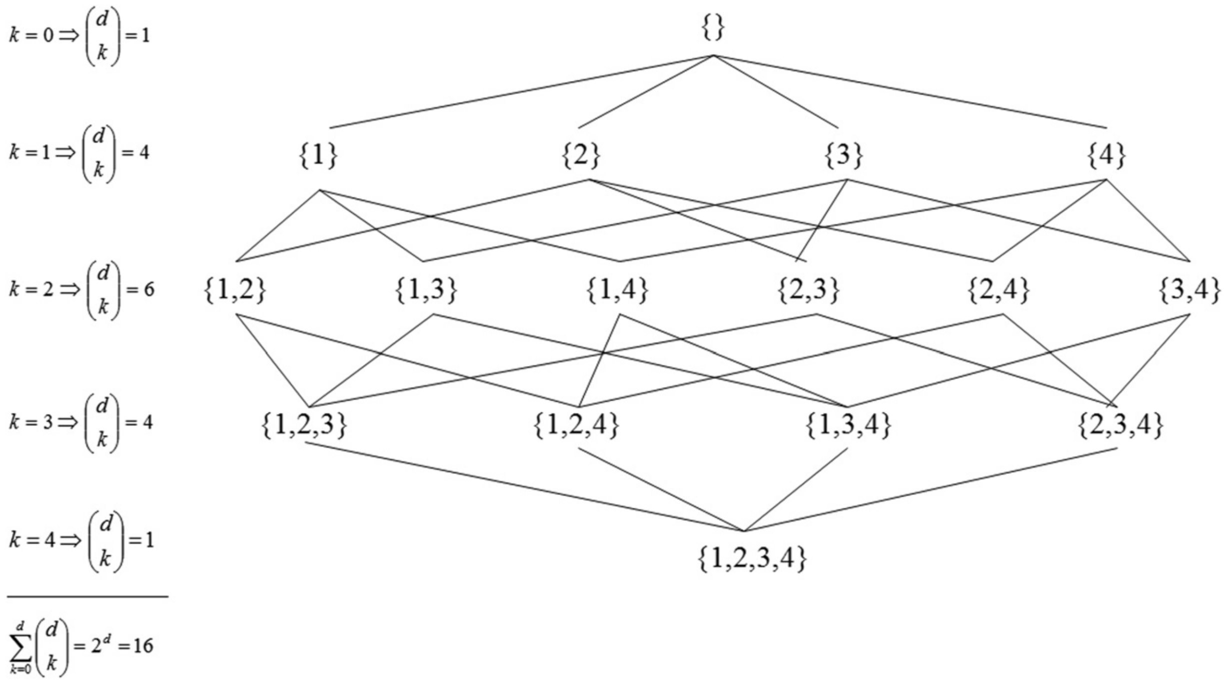
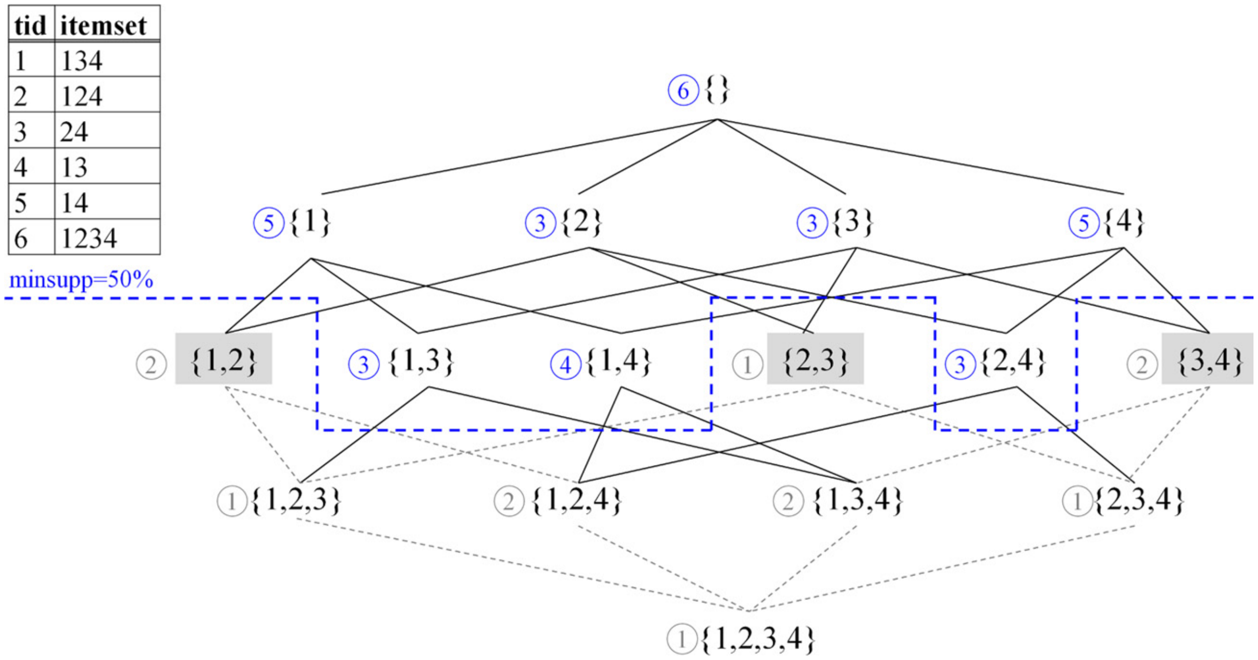

$k=0 \Rightarrow \binom{d}{k}=1$
$k=1 \Rightarrow \binom{d}{k}=4$
$k=2 \Rightarrow \binom{d}{k}=6$
$k=3 \Rightarrow \binom{d}{k}=4$
$k=4 \Rightarrow \binom{d}{k}=1$
$\sum_{k=0}^{d}\binom{d}{k}=2^d=16$
{}
{1}
{2}
{3}
{4}
{1,2}
{1,3}
{1,4}
{2,3}
{2,4}
{3,4}
{1,2,3}
{1,2,4}
{1,3,4}
{2,3,4}
{1,2,3,4}

Fig. 1 Lattice of itemsets: Powerset for $\mathscr{I}=\{1,2,3,4\}$ ($d=4$).



tid	itemset
1	134
2	124
3	24
4	13
5	14
6	1234

Fig. 2 Border between frequent and infrequent itemsets: The number in the circle indicates the support of the itemset.

provides an example of the border of the frequent itemsets for the transactions listed in the table and for a minimum support of 3 transactions (50%).

The main algorithmic strategy is to generate candidate frequent itemsets and test their support efficiently and without any redundancy, considering the computing system limitations. In most real-world applications the search space is too large to fit in main memory. Moreover, in spite of the exponential complexity optimisation techniques can provide a sufficient run time efficiency to tackle problems large enough to be of interest.

To this aim a number of optimisation techniques have been proposed, which refer to the strategy of visiting the lattice (breath vs depth first), to the generation of the candidate itemsets, and to the data structure and the algorithm for testing the constraint of the minimum support of candidate itemsets.

In particular, one fundamental optimisation technique is based on the anti-monotonicity (or downward-closure) property of the support. Any superset of an itemset cannot have a support greater than its subset. In Fig. 2, the infrequent itemsets are highlighted in grey: the edges from them moving forward to the next tier are represented as dashed lines to indicate that their supersets are implicitly, a priori, known to fail the support constraint. The algorithm *Apriori* (Agrawal and Srikant, 1994) was the first to introduce and exploit these concepts with the so-called *Apriori* 'pruning' optimisation.

Among the many algorithms proposed in the last two decades or so, three of them have a particular importance as each introduced a specific novel approach that inspired many others. The algorithm *Apriori* (Agrawal and Srikant, 1994) adopts a breadth-first strategy by generating the itemsets of a single tier at a time. The algorithm *Eclat* (Zaki, 2000) adopts the opposite depth-first strategy reducing the memory requirements. The algorithm *FP-growth* (Han *et al*., 2000) also adopts a depth-first search and introduced a clever tree-based data structure to improve the running time significantly.

Efficient implementations of algorithms for mining frequent itemsets are available as executable or source code (Borgelt, 2003, 2012, 2017), or by means of extensions of data science development environments, such as *arules* (Hornik *et al*., 2005), which is an R package for generating, managing and analysing frequent itemsets and association rules.

Interestingness Measures and Criteria

Interesting correlations among items and itemsets in patterns and association rules may not always be represented only by frequency (support and confidence). For example, the support does not consider normalisation of the frequencies of the items and may be skewed towards highly frequent items. Moreover, when patterns are considered interesting only on their frequency, the number of discovered patterns is often very large with a significant redundancy.

For these reasons other interestingness criteria and measures have also been considered. In some cases additional domain-specific measures can be used for ranking the patterns and prioritise their analysis. In others, pruning unnecessary and redundant patterns is a possible approach.

In applications of frequent itemset mining, the user typically applies an exploratory and iterative approach starting from large values of the support threshold (*minsupp*). Such large values of *minsupp* are chosen so that the number of frequent itemsets is small. Unfortunately, this often leads to frequent itemsets of small size, which may not be particularly interesting. Hence, in order to find more interesting frequent itemsets, the support threshold needs to be set to smaller values, for which the number of the frequent itemsets can be quite large, possibly too large for user inspection and analysis.

It turns out that some of the frequent itemsets may not be of particular interest as they have identical support as their supersets. For example, consider the two frequent itemsets 3 and 13 in Fig. 2, they are supported by the same three transactions ($id=1$, $id=4$ and $id=6$). In this case, the frequent itemset 13 is 'maximal' within the three supporting transactions, while 3 is not.

Following this consideration, a method of reducing the large amount of frequent itemsets is to use one of the so-called compact or condensed representations, such as the Maximal Frequent Itemsets (MFI) and the Closed Frequent Itemsets (CFI) (Uno *et al*., 2004).

MFI are frequent itemsets for which none of their supersets is also frequent. Clearly, any frequent itemset is a subset of a maximal frequent itemset. In other words, a frequent itemset is maximal if it is not a proper subset of any other frequent itemset.

A frequent itemset is closed if it contains all items that occur in all transactions in which it is bought, i.e., it is the intersection of its supporting transactions. In other words, a closed frequent itemset is a frequent itemset whose support is higher than the supports of all its proper supersets. In particular, all maximal frequent itemsets are also closed.

The underlying idea of both definitions is that the set of all maximal or closed frequent itemsets can be used as a compact representation of all frequent itemsets: It can be seen as a form of compression.

The set MFI allows to reconstruct all frequent itemsets by simply generating all their subsets. However, the support of these subsets cannot be reconstructed: the support of a frequent itemset can be different from the support of its maximal itemset. An additional database scan is needed to reconstruct this information, if necessary. MFI can be seen as a form of compression with loss of information.

This issue is overcome with the set CFI. The support of a non-closed itemset is the support of the smallest superset that is closed. CFI can be seen as a form of compression without loss of information.

Many other sampling criteria and definitions for subsets of frequent itemsets have also been proposed and include free frequent itemsets (Boulicaut *et al*., 2003), non-derivable itemsets (Calders and Goethals, 2007) and margin-closed frequent itemsets (Moerchen *et al*., 2011).

If the number of itemsets in a condensed subset is still too large for a manual analysis, then other data mining techniques can be applied to the complete set of frequent itemsets, or one of its subsets, in order to generate a model or a view at a higher level of abstraction (e.g., (Di Fatta *et al*., 2006)).

Extended Association Rule Mining

The basic concept of mining association rules and frequent patterns can be extended in a number of ways in order to gain more generality or to account for other attribute types (e.g., numerical or temporal attributes) in the data.

Generalised Association Rules (Sarawagi and Thomas, 1998) adopt a hierarchical taxonomy (concept hierarchy) of the items. Considering that coke and pepsi are soft drinks and assuming they are frequently bought with chips, nuts or crackers, a generalised rule may be express that "60% of transactions that contain soft drinks also contain snacks".

Quantitative Association Rules (Srikant and Agrawal, 1996) considers both quantitative and categorical attributes in the data. For example, the rule "10% of married people between age 50 and 60 have at least 2 cars" is able to associate a qualitative attribute with a quantitative one. In Interval Data Association Rules (Miller and Yang, 1997) the range of quantitative attributes is partitioned: For example, age can be partitioned into 5-year-increment intervals.

Mining Maximal Association Rules and Closed Association Rules (Uno *et al.*, 2004) directly discovers the condensed subsets of the frequent itemsets, as discussed in Section "Interestingness Measures and Criteria".

Sequential Association Rules (Sarawagi and Thomas, 1998) consider temporal data and discover subsequences: For example, users frequently buy first a PC, then a printer and, finally, a digital camera. Sequential pattern mining (Mabroukeh and Ezeife, 2010) can be applied to a variety of domains where the order of the items is relevant, e.g., Web log data and DNA sequences (Abouelhoda and Ghanem, 2010).

In Frequent Subgraph Mining (Kuramochi and Karypis, 2001) the data in the transactions are in the form of graphs. These may include many types of networks, such as social networks, protein-protein interaction networks (Shen *et al.*, 2012) and molecular compounds (Deshpande *et al.*, 2002). Finding frequent subgraphs in a set of graphs involves graph and subgraph isomorphism testing, which is more complex than subset testing required in frequent itemsets mining, and efficient and parallel computing approaches may be required for large datasets (Di Fatta and Berthold, 2005, 2006).

ARM Applications

Association rule and, in general, frequent pattern mining was originally inspired and motivated by market basket and customer behaviour analysis (Agrawal *et al.*, 1993; Agrawal and Srikant, 1994) and over the last two decades it has been successfully applied to many other applications domains. Various applications have been developed initially in other computer science fields, such as software bug discovery, failure and event detection in telecommunication and computer networks, WWW user behaviour analysis, and later in multidisciplinary domains including bioinformatics, chemoinformatics and data-driven medical applications.

Arguably the most popular example of association rules was extracted from supermarket transactions to reveal buying behaviour of customers: $IF\{DIAPERS\} \Rightarrow \{BEER\}$. This alleged association rule is often used to underline the nature of the correlations emerging from data: there is no implication of causality, only co-occurrence. The goal of ARM is to systematically extract all frequent and strong rules with an empirical (data-driven) approach. Domain experts may be able to gain useful insights from the generated rules and use them to support decision making, for example, in shelf management, marketing and sale promotions. The true story behind the example of diapers and beer is that in 1992, a retail consulting group at Teradata analysed 1.2 million customer transactions from Osco Drug stores. SQL self joins queries, not ARM, were used to identify correlations between (expensive) baby's products and any other product. The analysis discovered that between 5 and 7 p.m. consumers often bought diapers and beer. Apparently no attempt was made to exploit this correlation: Beers were not relocated near diapers. Nevertheless, this unexpected correlation raised much attention in the data analytics and mining community, and inspired a stream of research on association rules.

In telecommunication and computer networks, sequential pattern mining can be applied to the detection of events and anomalies (Cui *et al.*, 2014), to the analysis of user behaviour from Web logs (Pei *et al.*, 2000) and of online social media content (Adedoyin-Olowe *et al.*, 2013).

Detection of software bugs is another interesting example of ARM applications. Errors and faults leading to unexpected results, i.e., non-crashing bugs, is a difficult case of software bug detection (Liu *et al.*, 2005). For example in (Fatta *et al.*, 2016), function calls are recorded during software test executions and are analysed with a frequent pattern mining algorithm. Frequent subgraphs in failing test executions that are not frequent in successful test executions are used to rank the functions according to their likelihood of containing a fault.

Frequent pattern mining has been applied to biological data (Rigoutsos and Floratos, 1998) as these may come in the form of sequences (e.g., Microarray data (Cong *et al.*, 2004, 2005) and RNA (Chevalet and Michot, 1992)) and graphs (e.g., protein-to-protein interaction networks (Cho and Zhang, 2010), phylogenetic trees (Shasha *et al.*, 2004), molecular compounds (Di Fatta and Berthold, 2005, 2006; Deshpande *et al.*, 2005)).

In real-world applications frequent pattern and association rule mining is often one step in a more complex data analytics workflow, which may include other data mining techniques, such as classification and clustering, and data visualisation. The set of discovered patterns or rules can even be larger than the input dataset and a direct user inspection is not feasible, nor desirable. In some cases, frequent pattern mining can be seen as a feature generation step in the knowledge discovery process. For example, in (Di Fatta *et al.*, 2006) the frequent patterns are used to identify candidate drugs for a target activity and are considered as attributes in a very high dimensional space. A self-organising map is used to generate a 2-dimensional map of drugs, where proximity in the map indicates that activity against a target disease is due to a similar molecular substructure.

Conclusions

This article provides a brief survey of association rule and frequent pattern mining, which is one of the most important method in data mining for data-driven scientific discovery. A large body of literature has been produced in the past two decades or so. From the early introduction of the problem definition and the algorithms for market basket analysis, more efficient methods, extended problem definitions, advanced interestingness criteria and a plethora of applications in many diverse domains are nowadays available. In particular, the application domains have gradually expanded to cover a wide range, including software engineering, computer networks, bioinformatics, chemoinformatics and many other life sciences and scientific domains.

A general problem definition is presented and the combinatorial complexity of the computational task discussed. In particular, the exact exponential number of possible rules that are implicitly defined by a set of items is derived. Some of the most relevant algorithms, the fundamental and advanced interestingness criteria are presented. The most relevant extended problems and some application domains are also discussed with examples.

ARM is arguably one of the most elegant data mining problems that has fascinated the data mining community for many years and has a great potential to contribute to many scientific domains. It can provide a formidable tool for exploratory investigations to reveal intriguing insights from data, and ultimately lead to unexpected data-driven scientific discoveries.

See also: Data Mining in Bioinformatics. Identification of Homologs. Identification of Proteins from Proteomic Analysis. Next Generation Sequencing Data Analysis. Population Analysis of Pharmacogenetic Polymorphisms. The Challenge of Privacy in the Cloud

References

Abouelhoda, M., Ghanem, M., 2010. String Mining in Bioinformatics. Springer. pp. 207–247.

Adedoyin-Olowe, M., Gaber, M.M., Stahl, F., 2013. TRCM: A methodology for temporal analysis of evolving concepts in twitter. In: Proceedings of 12th International Conference on Artificial Intelligence and Soft Computing, ICAISC 2013, Part II, pp. 135–145. Zakopane, Poland: Springer.

Aggarwal, C.C., Han, J., 2014. Frequent Pattern Mining. Springer.

Agrawal, R., Imieliński, T., Swami, A., 1993. 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, pp. 207–216. New York, NY: ACM.

Agrawal, R., Srikant, R., 1994. Fast algorithms for mining association rules in large databases. In: Proceedings of the 20th International Conference on Very Large Data Bases, VLDB '94, pp. 487–499. Morgan: Kaufmann.

Alves, R., Rodriguez-Baena, D.S., Aguilar-Ruiz, J.S., 2010. Gene association analysis: A survey of frequent pattern mining from gene expression data. Briefings in Bioinformatics 11 (2). 210–224.

Angiulli, F., Ianni, G., Palopoli, L., 2001. On the complexity of mining association rules. In: Proceedings of the Nono Convegno Nazionale su Sistemi Evoluti di Basi di Dati (SEBD), SEBD, pp. 177–184.

Antonie, M.-L., Zaïane, O.R., 2004. Mining positive and negative association rules: An approach for confined rules. In: Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases, vol. 3202 of PKDD, pp. 27–38. Springer.

Atluri, G., Gupta, R., Fang, G., *et al*., 2009. Association analysis techniques for bioinformatics problems. In: Proceedings of the First International Conference on Bioinformatics and Computational Biology, pp. 1–13. Berlin, Heidelberg: Springer.

Borgelt, C., 2003. Efficient implementations of apriori and eclat. In: Proceedings of Workshop of Frequent Item Set Mining Implementations, FIMI. Melbourne, FL.

Borgelt, C., 2012. Frequent item set mining, Wiley Interdisciplinary Reviews. Data Mining and Knowledge Discovery 2 (6). 437–456.

Borgelt, C., 2017. Implementations of various data mining algorithms. Available at: http://www.borgelt.net/software.html.

Boulicaut, J.-F., Bykowski, A., Rigotti, C., 2003. Free-sets: A condensed representation of boolean data for the approximation of frequency queries. Data Mining and Knowledge Discovery 7 (1). 5–22.

Brin, S., Motwani, R., Ullman, J., Tsur, S., 1997. Dynamic itemset counting and implication rules for market basket data. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 255–264. ACM Press.

Calders, T., Goethals, B., 2007. Non-derivable itemset mining. Data Mining and Knowledge Discovery 14 (1). 171–206.

Chevalet, C., Michot, B., 1992. An algorithm for comparing rna secondary structures and searching for similar substructures. Computer Applications in the Biosciences 8 (3). 215–225.

Cho, Y.R., Zhang, A., 2010. Predicting protein function by frequent functional association pattern mining in protein interaction networks. IEEE Transactions on Information Technology in Biomedicine 14 (1). 30–36.

Cong, G., Tan, K.-L., Tung, A.K.H., Xu, X., 2005. Mining top-k covering rule groups for gene expression data. In: Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data, SIGMOD '05, pp. 670–681. New York, NY: ACM Press.

Cong, G., Tung, A.K.H., Xu, X., Pan, F., Yang, J., 2004. Farmer: Finding interesting rule groups in microarray datasets. In: Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data, SIGMOD '04, pp. 143–154. ACM.

Cui, H., Yang, J., Liu, Y., Zheng, Z., Wu, K., 2014. Data mining-based dns log analysis, Annals of Data. Science 1 (3). 311–323.

Deshpande, M., Kuramochi, M., Karypis, G., 2002. Automated approaches for classifying structures. In: Proceedings of the 2nd International Conference on Data Mining in Bioinformatics, BIOKDD'02, pp. 11–18. Springer.

Deshpande, M., Kuramochi, M., Wale, N., Karypis, G., 2005. Frequent substructure-based approaches for classifying chemical compounds. IEEE Transactions on Knowledge and Data Engineering 17 (8). 1036–1050.

Di Fatta, G., Berthold, M.R., 2005. High performance subgraph mining in molecular compounds. In: Proceedings of the International Conference on High Performance Computing and Communications (HPCC), LNCS, pp. 866–877. Springer.

Di Fatta, G., Berthold, M.R., 2006. Dynamic load balancing in distributed mining of molecular compounds. In: Proceedings of the IEEE Transactions on Parallel and Distributed Systems, Special Issue on High Performance Computational Biology, pp.773–785.

Di Fatta, G., Fiannaca, A., Rizzo, R., *et al*., 2006. Context-aware visual exploration of molecular databases. In: Proceedings of the Sixth IEEE International Conference on Data Mining – Workshops (ICDMW'06), pp. 136–141.

Di Fatta, G., Leue, S., Stegantova, E., 2016. Discriminative pattern mining in software fault detection. In: Proceedings of the 3rd International Workshop on Software Quality Assurance (SOQUA), 14th ACM Symposium on Foundations of Software Engineering (ACM SIGSOFT), pp. 62–69. ACM.

Fournier-Viger, P., Lin, J.C.-W., Vo, B., *et al*., 2017. A survey of itemset mining, wiley interdisciplinary reviews. Data Mining and Knowledge Discovery 7 (4).

Goethals, B., 2003. Survey on frequent pattern mining. Technical report. Helsinki Institute for Information Technology.

Hanash, C.C.S., 2003. Mining gene expression databases for association rules. Bioinformatics 19 (1). 79–86.

Han, J., Cheng, H., Xin, D., Yan, X., 2007. Frequent pattern mining: Current status and future directions. Data Mining and Knowledge Discovery 15 (1). 55–86.

Han, J., Pei, J., Yin, Y., 2000. Mining frequent patterns without candidate generation. In: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, SIGMOD '00, pp. 1–12. ACM.

Hipp, J., Güntzer, U., Nakhaeizadeh, G., 2000. Algorithms for association rule mining – A general survey and comparison. SIGKDD Explorations Newsletter 2 (1). 58–64.

Hornik, K., Grün, B., Hahsler, M., 2005. Arules – A computational environment for mining association rules and frequent item sets, Wiley Interdisciplinary Reviews. Data Mining and Knowledge Discovery 14 (15). 1–25.

Kuramochi, M., Karypis, G., 2001. Frequent subgraph discovery. In: Proceedings of the 2001 IEEE International Conference on Data Mining, ICDM '01, IEEE Computer Society, pp. 313–320.

Liu, C., Yan, X., Yu, H., Han, J., Yu, P.S., 2005. Mining behavior graphs for "backtrace" of noncrashing bugs. In: Proceedings of the 2005 SIAM International Conference on Data Mining, pp. 286–297.

Mabroukeh, N.R., Ezeife, C.I., 2010. A taxonomy of sequential pattern mining algorithms. ACM Computing Surveys 43 (1). 1–3.

Miller R.J., Yang Y., 1997. Association rules over interval data. In: Proceedings of the ACM 1997 SIGMOD International Conference on Management of Data, SIGMOD '97, pp. 452–461. ACM.

Moerchen, F., Thies, M., Ultsch, A., 2011. Efficient mining of all margin-closed itemsets with applications in temporal knowledge discovery and classification by compression. Knowledge and Information Systems 29 (1). 55–80.

Oellrich, A., Jacobsen, J., Papatheodorou, I., Smedley, D., 2014. Using association rule mining to determine promising secondary phenotyping hypotheses. Bioinformatics 30 (12).

Pei, J., Han, J., Mortazavi-Asl, B., Zhu H., 2000. Mining access patterns efficiently from web logs. In: Proceedings of the Knowledge Discovery and Data Mining, Current Issues and New Applications: 4th Pacific-Asia Conference, PAKDD 2000, pp. 396–407. Kyoto, Japan: Springer.

Rigoutsos, I., Floratos, A., 1998. Combinatorial pattern discovery in biological sequences: The teiresias algorithm. Bioinformatics 14 (1). 55–67.

Sarawagi, S., Thomas, S., 1998. Mining generalized association rules and sequential patterns using sql queries. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '98, ACM.

Savasere, A., Omiecinski, E., Navathe, S., 1995. An efficient algorithm for mining association rules in large databases. In: Proceedings of the 21th International Conference on Very Large Data Bases (VLDB), pp. 432–444.

Shasha, D., Wang, J.T.L., Zhang, S., 2004. Unordered tree mining with applications to phylogeny. In: Proceedings of 20th International Conference on Data Engineering, pp. 708–719.

Shen, R., Goonesekere, N., Guda, N., 2012. Mining functional subgraphs from cancer protein-protein interaction networks. BMC Systems Biology 6 (3).

Chen, C.-H., Tsai, T.-H., Li, W.-H., 2015. Dynamic association rules for gene expression data analysis. BMC Genomics 16 (786).

Srikant, R., Agrawal, R., 1996. Mining quantitative association rules in large relational tables. In: Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data, SIGMOD '96, pp. 1–12. ACM.

Toivonen, H., 1996. Sampling large databases for association rules. In: Proceedings of the 22nd International Conference on Very Large Data Bases (VLDB), pp. 134–145. Morgan: Kaufmann.

Uno, T., Kiyomi, M., Arimura, H., 2004. Efficient mining algorithms for frequent/closed/maximal item sets. In: Workshop of Frequent Item Set Mining Implementations, FIMI. Brighton, United Kingdom.

Wijsen, J., Meersman, R., 1998. On the complexity of mining quantitative association rules. Data Mining and Knowledge Discovery 2 (3). 263–281.

Yang, G., 2004. The complexity of mining maximal frequent itemsets and maximal frequent patterns. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '04, pp. 344–353. ACM.

Zaki, M., Parthasarathy, S., Ogihara, M., Li, W., 1997. New algorithms for fast discovery of association rules. In: Proceedings of the Third International Conference on Knowledge Discovery and Data Mining, pp. 283–286. AAAI Press.

Zaki, M.J., 2000. Scalable algorithms for association mining. IEEE Transactions on Knowledge and Data Engineering 12 (3). 372–390.