Book Title	Encyclopedia of Machine Learning	
Chapter Number	00403	
Book CopyRight - Year	2010	
Title	Frequent Pattern	
Author	Particle	
	Given Name	**Hannu**
	Family Name	**Toivonen**
	Suffix	
	Email	hannu.toivonen@cs.helsinki.fi
Affiliation	Division	Department of Computer Science
	Organization	University of Helsinki
	Street	P.O. Box 68 (Gustaf Hallstromin katu 2b)
	Postcode	FI-00014
	City	Helsinki
	State	
	Country	Finland

F

Frequent Pattern

Hannu Toivonen
University of Helsinki, Finland

Definition

Given a set $\mathcal{D}$ of examples, a language $\mathcal{L}$ of possible patterns, and a minimum frequency *min_fr*, every pattern $\theta \in \mathcal{L}$ that occurs at least in the minimum number of examples, i.e., $|\{e \in \mathcal{D} \mid \theta \text{ occurs in } e\}| \geq min\_fr$, is a frequent pattern. Discovery of all frequent patterns is a common data mining task. In its most typical form, the patterns are ▸frequent itemsets. A more general formulation of the problem is ▸constraint-based mining.

Motivation and Background

Frequent patterns can be used to characterize a given set of examples: they are the most typical feature combinations in the data.

Frequent patterns are often used as components in larger data mining or machine learning tasks. In particular, discovery of ▸frequent itemsets was actually first introduced as an intermediate step in ▸association rule mining (Agrawal, Imieliński & Swami, 1993) ("frequent itemsets" were then called "large"). The frequency and confidence of every valid association rule $X \rightarrow Y$ are obtained simply as the frequency of $X \cup Y$ and the ratio of frequencies of $X \cup Y$ and X, respectively.

Frequent patterns can be useful as ▸features for further learning tasks. They may capture shared properties of examples better than individual original features, while the frequency threshold gives some guarantee that the constructed features are not so likely just noise. However, other criteria besides frequency are often used to choose a good set of candidate patterns.

Structure of Problem

A frequent pattern often is essentially a set of binary ▸features. Given a set $\mathcal{I}$ of all available features, the pattern language $\mathcal{L}$ then is the power set of $\mathcal{I}$. An example in data $\mathcal{D}$ covers a pattern $\theta \in \mathcal{L}$ if it has all the features of θ. In such cases, the frequent pattern discovery task reduces to the task of discovering ▸frequent itemsets. Therefore, the structure of the frequent pattern discovery problem is best described using the elementary case of frequent itemsets.

Let $\mathcal{I}$ be the set of all items (or binary features); subsets of $\mathcal{I}$ are called itemsets (or examples or patterns, depending on the context). The input to the frequent itemset mining problem is a multiset $\mathcal{D}$ of itemsets (examples described by their features), and a frequency threshold. The task is to output *all* frequent itemsets (patterns) and their frequencies, i.e., all subsets of $\mathcal{I}$ that exceed the given frequency threshold in the given data $\mathcal{D}$.

Example 1 *Assume the following problem specification:*

- *Set of all items* $\mathcal{I} = \{A, B, C, D\}$.
- *Data* $\mathcal{D} = \{\{A, B, C\}, \{A, D\}, \{B, C, D\}, \{A, B, C\}, \{C, D\}, \{B, C\}\}$.
- *Frequency threshold is 2.*

All possible itemsets and their frequencies:

C. Sammut, G. Webb (eds.), *Encyclopedia of Machine Learning*, DOI 10.1007/978-0-387-30768-8, © Springer-Verlag Berlin Heidelberg, 2010

Itemset	*Frequency*
$\{A\}$	**3**
$\{B\}$	**4**
$\{C\}$	**5**
$\{D\}$	**3**
$\{A,B\}$	**2**
$\{A,C\}$	**2**
$\{A,D\}$	*1*
$\{B,C\}$	**4**

Itemset	*Frequency*
$\{B,D\}$	*1*
$\{C,D\}$	**2**
$\{A,B,C\}$	**2**
$\{A,B,D\}$	*0*
$\{A,C,D\}$	*0*
$\{B,C,D\}$	*1*
$\{A,B,C,D\}$	*0*

The frequent itemsets are $\{A\}$, $\{B\}$, $\{C\}$, $\{D\}$, $\{A,B\}$, $\{A,C\}$, $\{B,C\}$, $\{C,D\}$, $\{A,B,C\}$.

The ▶hypothesis space for itemsets obviously is the power set of $\mathcal{I}$, and it has an exponential size ($2^{|\mathcal{I}|}$) in the number of items. Since all frequent itemsets are output, this is also the size of the output in the worst case (e.g., if the frequency threshold is zero, or if all examples in $\mathcal{D}$ equal $\mathcal{I}$), as well as the worst case time complexity.

In practical applications of frequent itemset mining, the size of the output as well as the running times are much smaller, but they strongly depend on the properties of the data and the frequency threshold. The useful range of thresholds varies enormously among different datasets. In many applications – such as ▶basket analysis – the number $|\mathcal{I}|$ of different items can be in thousands, even millions, while the typical sizes of examples are at most in dozens. In such sparse datasets a relatively small number of frequent itemsets can reveal the most outstanding co-occurrences; e.g., there are not likely to be very large sets of books typically bought by the same customers. In dense datasets, in turn, the number of frequent patterns can be overwhelming and also relatively uninformative. E.g., consider the dense dataset of books that have *not* been purchased by a customer: there are a huge number of sets of books that have not been bought by the same customers.

Theory/solutions

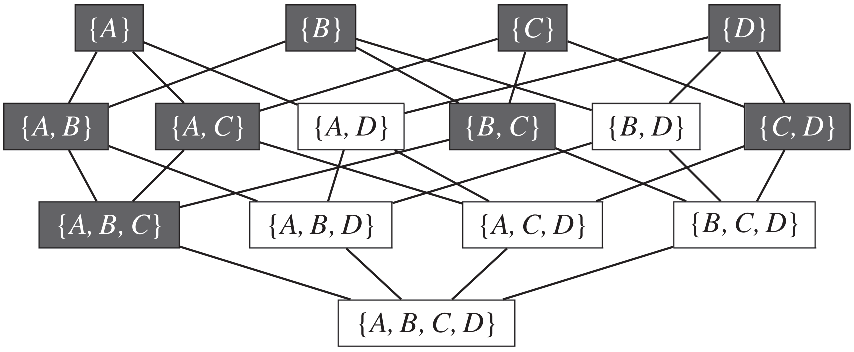
The most widely known solution for finding all frequent itemsets is the ▶Apriori algorithm (Agrawal, Mannila, Srikant, Toivonen, & Verkamo, 1996). It is based on the monotonicity of itemset frequencies (a ▶generalization relation): the frequency of a set is at most as high as the frequency of any of its subsets. Conversely, if a set is known to be infrequent, then none of its supersets can be frequent.

Apriori views the ▶hypothesis space of itemsets as a (refinement) lattice defined by set containment, and performs a ▶general-to-specific search using ▶breadth-first search. In other words, it starts with singleton itemsets, the most general and frequent sets, and proceeds to larger and less frequent sets. The search is pruned whenever a set does not reach the frequency threshold: all supersets of such sets are excluded from further search. Apriori deviates from standard breadth-first search by evaluating all sets of equal size in a single batch, i.e., it proceeds in a levelwise manner. This has no effect on the search structure or results, but can reduce disk access considerably for large databases. See the entry ▶Apriori Algorithm for an outline of the method.

Example 2 *Figure 1 illustrates the search space for the data $\mathcal{D}$ of Example 1. Dark nodes represent frequent itemsets, i.e., the answer to the frequent itemset mining problem. Apriori traverses the space a level at a time. For instance, on the second level, it finds out that $\{A,D\}$ and $\{B,D\}$ are not frequent. It therefore prunes all their supersets, i.e., does not evaluate sets $\{A,B,D\}$, $\{A,C,D\}$, and $\{B,C,D\}$ on the third level.*

Other search strategies have also been applied. A ▶depth-first search without the subset check allows faster identification of candidates, at the expense of having more candidates to evaluate and doing that without natural batches (e.g., Zaki, 2000). FP-growth (Han, Pei, Yin, & Mao, 2004) uses a tree structure to store the information in the dataset, and uses it to recursively search for frequent itemsets.

The search strategy of Apriori is optimal in a certain sense. Consider the number of sets evaluated, and assume that for any already evaluated set we know whether it was frequent or not but do not consider its frequency. Apriori evaluates the frequencies of all frequent itemsets plus a number of candidates that turn out to be infrequent. It turns out that every infrequent candidate must actually be evaluated under the given assumptions: knowing which other sets are frequent



Frequent Pattern. Figure 1. The search space of frequent itemsets for data *D* of the running example. Dark nodes: frequent itemsets; white nodes: infrequent itemsets

and which are not does not help, regardless of the search order. This observation leads to the concept of *border*: the border consists of all those itemsets whose all proper subsets are frequent and whose all proper supersets are infrequent (Gunopulos et al., 2003; Mannila & Toivonen, 1997). The border can further be divided into two: the positive border contains those itemsets in the border that are frequent, the negative border contains those that are not. The positive border thus consists of the most specific patterns that are frequent, and corresponds to the "S" set of ▶version spaces.

Example 3 *Continuing our running example, Figure 2 illustrates the border between the frequent and infrequent sets. Either the positive or the negative border can alone be used to specify the collection of frequent itemsets: every frequent itemset is a subset of a set in the positive border ($\{A, B, C\}$, $\{C, D\}$), while every infrequent itemset is a superset of a set in the negative border ($\{A, D\}$, $\{B, D\}$).*

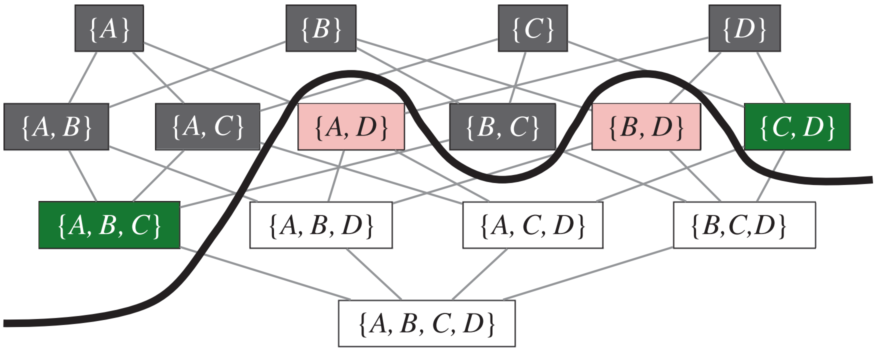
One variant of frequent itemset mining is to output the positive border only, i.e., to find the *maximal frequent itemsets* (Bayardo, 1998). This can be implemented with search strategies that do not need to evaluate the whole space of frequent patterns. This can be useful especially if the number of frequent itemsets is very large, or if the maximal frequent itemsets are large (in which case the number of frequent itemsets is large, too, since the number of subsets is exponential in the length of the maximal set). As a trade-off, the result does not directly indicate frequencies of itemsets.

Condensed Representations: Closed Sets and Nonderivable Sets Closed sets and nonderivable sets are a powerful concept for working with frequent itemsets, especially if the data is relatively dense or there are strong dependencies. Unlike the aforementioned simple model for borders, here also the known frequencies of sets are used to make inferences about frequencies of other sets.

As a motivation for closed sets (Pasquier, Bastide, Taouil, & Lakhal, 1999), consider a situation where the frequency of itemset $\{i, j\}$ equals the frequency of item *j*. This implies that whenever *j* occurs, so does *i*. Thus, any set $A \cup \{j\}$ that contains item *j* also contains item *i*, and the frequencies of sets $A \cup \{j\}$ and $A \cup \{i, j\}$ must be equal. As a result, it sufficies to evaluate sets $A \cup \{j\}$ to obtain the frequencies of sets $A \cup \{i, j\}$, too.

More formally, the *closure* of set *A* is its largest superset with identical frequency. *A* is *closed* iff it is its own closure, i.e., if every proper superset of *A* has a smaller frequency than *A*. The utility of closed sets comes from the fact that frequent closed sets and their frequencies are a sufficient representation of all frequent sets. Namely, if *B* is a frequent set then its closure is a frequent closed set in $\mathcal{C}\ell$, where $\mathcal{C}\ell$ denotes the collection of all frequent closed itemsets. *B*'s frequency is obtained as $\text{fr}(B) = \max\{\text{fr}(A) \mid A \in \mathcal{C}\ell \text{ and } B \subseteq A\}$. If *B* is not a frequent set, then it has no superset in $\mathcal{C}\ell$. ▶Formal concept analysis studies and uses closed sets and other related concepts.

Generators are a complementary concept, and also constitute a sufficient representation of frequent itemsets. (To be more exact, in addition to frequent generators, generators in the border are also needed). Set *A* is a *generator* (also known as a key pattern or a free set) if



Frequent Pattern. Figure 2. The positive border ($\{A, B, C\}$, $\{C, D\}$) and negative border ($\{A, D\}$, $\{B, D\}$) of frequent itemsets

all its proper subsets have a larger frequency than A has. Thus, in an equivalence class of itemsets, defined by the set of examples in which they occur, the maximal element is unique and is the closed set, and the minimal elements are generators. The property of being a generator is monotone in the same way that being frequent is, and generators can be found with simple modifications to the Apriori algorithm.

Example 4 *Figure 3 illustrates the equivalence classes of itemsets by circles. For instance, the closure of itemset $\{A, B\}$ is $\{A, B, C\}$, i.e., whenever $\{A, B\}$ occurs in the data, C also occurs, but no other items. Given just the frequent closed sets and their frequencies, the frequency of, say, $\{B\}$ is obtained by finding its smallest frequent closed superset. It is $\{B, C\}$, with frequency 4, which is also B's frequency. Alternatively, using generators as the condensed representation, the frequency of itemset $\{B, C\}$ can be obtained by finding its maximal generator subset, i.e., $\{B\}$, with which it shares the same frequency.*

Nonderivability of an itemset (Calders & Goethals, 2002) is a more complex but often also a more powerful concept than closed sets. Given the frequencies of (some) subsets of itemset A, the frequency of A may actually be uniquely determined, i.e., there is only one possible consistent value. A practical method of trying to determine the frequency is based on deriving upper and lower bounds with inclusion–exclusion formula from the known frequencies of some subsets, and checking if these coincide. An itemset is derivable if this is indeed the case, otherwise it is *nonderivable*. Obviously, the collection of nonderivable frequent sets is a sufficient representation for all frequent sets.

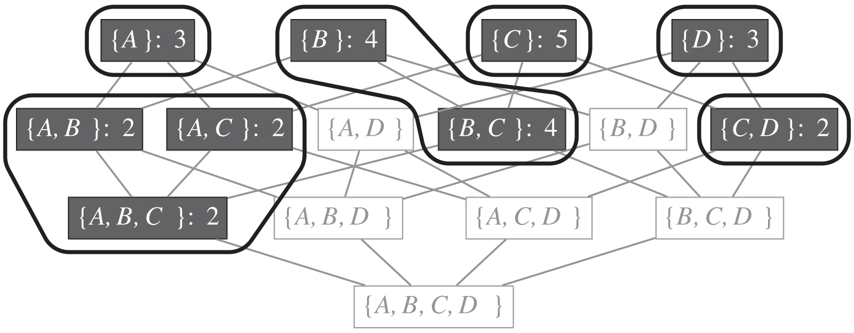
Bounds for the absolute frequency of set I are obtained from its subsets as follows, for any $X \subseteq I$:

$$\mathrm{fr}(I) \leq \sum_{J:X\subseteq J\subset I} (-1)^{|I\setminus J|+1}\mathrm{fr}(J) \text{ if } |I \setminus X| \text{ is odd,} \quad (1)$$

$$\mathrm{fr}(I) \geq \sum_{J:X\subseteq J\subset I} (-1)^{|I\setminus J|+1}\mathrm{fr}(J) \text{ if } |I \setminus X| \text{ is even.} \quad (2)$$

Using all subsets X of I, one can obtain a number of upper and lower bounds. If the least upper bound equals the greatest lower bound, then set I is derivable. The conceptual elegance of this solution lies in the fact that derivable sets follow logically from the nonderivable ones – the aforementioned formula is one way of finding (some) such situations – whereas with closed sets the user must know the closure properties.

Generalizations of Frequent Patterns The concept of frequent patterns has been extended in two largely orthogonal directions. One is to more complex patterns and data, such as frequent sequences, trees (see ►tree mining), graphs (see ►graph mining), and first-order logic (Dehaspe & Toivonen, 1999). The other direction to generalize the concept is to ►constraint-based mining, where other and more complex conditions are considered beyond frequency. We encourage the interested reader to continue at the entry for ►constraint-based mining, which also gives further insight into many of the more theoretical aspects of frequent pattern mining.



Frequent closed set $\mathcal{C}\ell = \{\{A\},\{C\},\{D\},\{B,C\},\{C,D\},\{A,B,C\}\}$.

Frequent generators: $\{\{A\},\{B\},\{C\},\{D\},\{A,B\},\{A,C\},\{C,D\}\}$.

Frequent Pattern. Figure 3. Frequencies and equivalence classes of frequent itemsets in data $\mathcal{D}$ of the running example, and the corresponding closed sets and generators

Programs and Data

Frequent itemset mining implementations repository:
http://fimi.cs.helsinki.fi/
Weka: http://www.cs.waikato.ac.nz/ml/weka/
Christian Borgelt's implementations:
http://www.borgelt.net/software.html
Data mining template library:
http://dmtl.sourceforge.net/

Applications

Frequent patterns are a general purpose tool for data exploration, with applications virtually everywhere. Market ▶basket analysis was the first application, telecom alarm correlation and gene mapping are examples of quite different application fields.

Future Directions

Work on frequent pattern mining is being expanded in several directions. New types of pattern languages are being developed, either to meet some specific needs or to increase the expressive power. Many of these developments are motivated by different types of data and applications. Within machine learning, frequent patterns are increasingly being used as a tool for feature construction in complex domains. For an end-user application, methods for choosing and ranking the most interesting patterns among thousands or millions of them is a crucial problem, for which there are no perfect solutions (cf. Geng & Hamilton, 2006). At the same time, theoretical understanding of the problem and solutions of frequent pattern discovery still has room for improvement.

Cross References

- ▶Apriori Algorithm
- ▶Association Rule
- ▶Basket Analysis
- ▶Constraint-Based Mining
- ▶Data Mining
- ▶Frequent Itemset
- ▶Graph Mining
- ▶Knowledge Discovery in Databases
- ▶Tree Mining

Recommended Reading

Agrawal, R., Imielinski, 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, Washington, DC* (pp. 207–216). New York: ACM.

Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., & Verkamo, A. I. (1996). Fast discovery of association rules. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, & R. Uthurusamy (Eds.), *Advances in knowledge discovery and data mining* (pp. 307–328). Menlo Park, CA: AVAAI Press.

Bayardo, R. J. Jr. (1998). Efficiently mining long patterns from databases. In *Proceedings of the 1998 ACM SIGMOD international conference on management of data, Seatle, Washington, DC* (pp. 85–93). New York: ACM.

Calders, T., & Goethals, B. (2002). Mining all non-derivable frequent itemsets. In *Proceedings of the 6th European Conference on principles of data mining and knowledge discovery, Helsinki, Finland.* Lecture Notes in Computer Science (vol. 2431, pp. 74–85). London: Springer.

Ceglar, A., & Roddick, J. F. (2006). Association mining. *ACM Computing Surveys 38*(2): Article No. 5.

Dehaspe, L., & Toivonen, H. (1999). Discovery of frequent datalog patterns. *Data mining and knowledge discovery 3*(1): 7–36.

Geng, L., & Hamilton, H. J. (2006). Interestingness measures for data mining: A survey. *ACM Computing Surveys 38*(3): Article No. 9.

Gunopulos, D., Khardon, R., Mannila, H., Saluja, S., Toivonen, H., & Sharma, R. S. (2003). Discovering all most specific sentences. *ACM transactions on database systems 28*(2): 140–174.

Han, J., Pei, J., Yin, Y., & Mao, R. (2004). Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data Mining and Knowledge Discovery 8*(1): 53–87.

Mannila, H., & Toivonen, H. (1997). Levelwise search and borders of theories in knowledge discovery. *Data Mining and Knowledge Discovery 1*(3): 241–258.

Pasquier, N., Bastide, Y., Taouil, R., & Lakhal, L. (1999). Discovering frequent closed itemsets for association rules. In *Proceedings of 7th international conference on database theory, Jerusalem, Israel*. Lecture Notes in Computer Science (vol. 1540, pp. 398–416). London: Springer.

Zaki, M. J. (2000). Scalable algorithms for association mining. In *IEEE transactions on knowledge and data engineering* 12(3): 372–390.