Abstract Representations and Generalized Frequent Pattern Discovery

Eray Özkural(✉)

Gök Us Sibernetik Ar&Ge Ltd. Şti., Istanbul, Turkey
examachine@gmail.com

Abstract. We discuss the frequent pattern mining problem in a general setting. From an analysis of abstract representations, summarization and frequent pattern mining, we arrive at a generalization of the problem. Then, we show how the problem can be cast into the powerful language of algorithmic information theory. We formulate and prove a universal pruning theorem analogous to the well-known Downward Closure Lemma in data mining. This result allows us to formulate a simple algorithm to mine all frequent patterns given an appropriate compressor to recognize patterns.

1 Introduction

The field of data mining is changing faster than we can define it. In recent years, foundations of data mining have received considerable interest, helping remove some of the ad-hoc considerations in the theory of data mining and expanding the frontiers. The problem definitions of early data mining research have now been analyzed meticulously, considering especially the performance and scalability of methods, giving a performance-oriented character to most data mining research. Qualitative work has usually focused on slight variations of the original problems; staying within the framework of basic problems such as association rule mining and sequence mining. However, the ever expanding computational and storage capacity challenges us to devise new ways to look at the data mining tasks, to discover more interesting/useful patterns. The subject of this paper is a substantial revision of the frequency mining problem, this time mining for any kind of a pattern instead of frequent item sets. We arrive at our formulation from a philosophical analysis of the problem, conceiving what the problem might look like in the most general setting. After reviewing some of the recent literature on generalizing data mining problems, we examine the relation of abstraction to the summarization task and in particular frequent pattern discovery. We then present a novel formulation of the frequent pattern discovery problem using algorithmic information theory, derived from our philosophical analysis. We show that our formulation exhibits similar formal relations to the original frequent itemset mining problem, and is arguably a good generalization of it. Then, we present the

The present paper was originally written and circulated in 2006, and its findings inform our other AGI methods including Heuristic Algorithmic Memory [13].

T. Everitt et al. (Eds.): AGI 2017, LNAI 10414, pp. 67–76, 2017.
DOI: 10.1007/978-3-319-63703-7_7

Micro-Synthetic algorithm which has the capability to detect any kind of a pattern given our information theoretic definition of pattern occurrence. The algorithm is similar to the Apriori algorithm in its logic of managing the task in a small number of database scans. After discussing the pros and cons of our approach, we outline future research directions.

2 Background and Related Work

We will skip the definitions and methods of traditional frequent pattern discovery for considerations of space. For an introduction to the subject, see [3,9,18]. There has been some promising research in applying the generic methods of Kolmogorov complexity to data mining. The authors report favorable results for classification and deviation detection tasks in [7]. A mathematical theory of high frequency patterns which uses granular computing was presented in [11]. We will now take a closer look at algorithmic methods which have attracted a great deal of interest.

2.1 Algorithmic Information Theory

Algorithmic information theory (AIT) gives an absolute characterization of complexity for arbitrary bit strings [6]. A computer is a computable partial function $C(p, q)$ of self-delimiting program strings p and data q, where both input and output datum are bitstrings in $\{0, 1\}^*$. Empty string is denoted with Λ and the shortest program which computes s is denoted with $s*$. U is a universal computer that can simulate any other computer C with $U(p', q) = C(p, q)$ and $|p'| \leq |p| + sim(C)$ where $sim(C)$ is the length of simulation program for C. An admissible universal computer is LISP with its eval function.

The algorithmic information content $H(s)$ of a bit string s is the size of minimal program $s*$ which computes it. $H(s/t)$ is the algorithmic information content of s relative to t (conditional algorithmic entropy). Another definition from AIT is mutual algorithmic information $H(s : t)$ which is relevant to our work. $H(s : t)$ is the extent to which knowing s helps one to calculate t. The probability $P(s)$ of a bitstring s is the probability a program evaluates to s. Likewise, the conditional probability $P(s/t)$ is the probability a program evaluates to s given the minimal program $t*$ for calculating t.

AIT gives an analogous formalism to information theory, and is deemed more fundamental since Shannon information can be derived from algorithmic (Kolmogorov) information. It is not possible to include all theorems here, but some relevant consequences and results will be stated, mostly without proof.

$H(s, t)$ is the joint algorithmic information of s and t where "," denotes concatenation of bitstrings (at any rate it is straightforward to convert between any two pair encodings). Algorithmic information is asymptotically symmetric, e.g. $H(s, t) = H(t, s) + O(1)$ since in high level languages it is not problematic to accomplish this sort of feat with a short constant program. The conditional entropy of a string with itself is constant, similarly.

Theorem I8 of [6] states that conditional entropy measures how easier it is to compute two strings together than separately.

$$H_C(t/s) = H(s,t) - H(s) + c \tag{1}$$

Theorem I9 of [6] exposes the relationships between joint, mutual and conditional information, as well as probability and joint probability. In particular, algorithmic information is subadditive and conditional and mutual information can be calculated from probabilities.

$$H(s,t) = H(s) + H(t/s) + O(1) \tag{2a}$$
$$H(s:t) = H(s) + H(t) - H(s,t) + O(1) \tag{2b}$$
$$H(s:t) = H(t:s) + O(1) \tag{2c}$$

There are several other interesting theorems in AIT, however they fall beyond the scope of the present work.

2.2 Algorithmic Distance Metrics for Classification

$H(s)$ is uncomputable. However, it can be approximated with a reasonable compression program from the above. The standard UNIX compression programs gzip and bzip2 have been used exactly for this purpose by Cilibrasi et al. [16] for clustering music files. In the predecessors to this paper, Vitanyi et al. [5,10,14] have introduced a distance function based on algorithmic information theory which can be used for domain unspecific classification and clustering algorithms.

In another work, Kraskov et al. propose using mutual information both in Shannon's version and Kolmogorov's version based on the same proof [1]. These studies are relevant to our problem in that they show the versatility of Kolmogorov complexity. We shall now try to answer if we can achieve similar feats in data mining.

3 Abstract Representations

Before proceeding with our formulation of frequent pattern discovery from an information theoretic perspective, it is worthwhile giving a philosophical overview of the task. The main objective of frequency mining is to summarize a large data set. With a suitable threshold, we obtain a smaller data set that is representative of the most significant patterns in the data. By means of such an abstract representation, one then achieves more specific tasks such as discovering association rules or clustering the data.

Recent formulations of association rule mining have characterized the task as generalization of the data. This is a necessary condition for any successful abstraction, else what use can we imagine of an abstract representation? According to Marvin Minsky, another way of putting this would be the removal of unnecessary details from the representation [12]. Statistically, "detail" could be understood as infrequent patterns in data, which is precisely what frequent item set mining eliminates. Thus, a comparison of the common sense notion of "abstraction" and the familiar data mining task of summarization is in order.

Let us conceive of an abstract sketch A. If this drawing is an abstraction of a lively picture B, we expect to find the most "important" features of B in A, perhaps only some of them. We would also expect to see the details, for instance the texture, shading and colors of B to be removed in A (assuming that it is quite abstract). In addition, we would not like to see anything in A that does not correspond to a significant feature in B. Some caricatures, like those of politicians drawn in a clean generic style, may set a good example of this kind of sensory abstraction (Note however that some caricatures are highly stylized and will set a bad example for abstraction). The facial features in a caricature are highly informative; they convey much information about the facial identity of the person at a small cost of representation. On the other hand, like any other image, the abstract representation must be built from low-level components, which are apparently not part of the original image. If these components, such as the basic drawing patterns of the caricaturist, are kept simple enough, the resulting work will look abstract.

If we are to relate the above characterization of abstraction to data mining, the most problematic part might be the "important" term. After all, an important feature for one task might be unimportant for another. Consider the notes of a symphony. The pitch and duration information is considered significant because it helps us to quickly discern one piece of music from the other. This is true for any given application domain. For recognition of music, it is the pitch or the interval that matters. But for speech, it is the phoneme that matters. The truly generic summarization algorithm might be able to discover the concept of note or phoneme merely by looking at the data. If we take B to be only one datum in a data set, we will find it more productive to think of the importance of a feature determined by the frequency of its occurrence. This approach suggested also in the beginning of the section does not completely solve the problem, however. We also need universal and objective criteria for determining if a feature approximately occurs in a given datum.

Let us now make our explanations more precise. We say that A is an abstract representation of B if and only if:

1. A is substantially less complex than B.
2. Every important feature of A is similar to an important feature of B.

Note that condition 2 can also be stated as: "There is no important feature in A that is not similar to an important feature in B".

This definition is more relevant to abstraction than lossy compression. Especially, in lossy compression the only purpose is to reproduce the data set with a low error rate (e.g. defined in terms of how well the reproduction is), it does not necessarily take into account simplification of condition 1. Neither does it address the "similar" predicate of the last condition. One might decide to exclude color from the abstract representation of a house, but in traditional image compression such choices would not be considered. Furthermore, lossy compression does not take into account the generalization power of the representation over an ensemble of objects. However, in frequency mining, we can give a rigid meaning to importance, e.g., statistically significant patterns.

If we now consider a frequent pattern discovery algorithm, we may say that the set of frequent patterns satisfy conditions 1 and 2 to be an abstract representation of the entire data set. A useful frequent pattern set is smaller than the transaction set and each frequent pattern (all of them above the given support can be said to be important) occurs in B as an important feature. In this sense, the pattern set does not only model the current data set, but presumably also future extensions of the data source. (We can note here that the non-traditional statistics provided by the frequent itemset-like computation may have use for predictive modelling in general).

3.1 Analysis of Common Objections

An objection may be raised at this point with respect to the traditional duality of syntax vs. semantics. It may be suggested that abstraction crucially depends on semantics which does not seem to be mentioned in our definition. It need not be, since semantic relations, too, may be accounted for in the "similar" predicate. On the other hand, it must be reminded that cryptic references should not in general be considered as abstract in themselves. By abstraction, we refer to manifestly useful, generalized, compact representations. Any cryptic representation may be conceived of as an encrypted form of such an underlying "successful" abstract representation.

3.2 Other Approaches for Pattern Interestingness

Equating frequency with importance may not be the only or satisfactory way of defining interestingness of a pattern objectively. If we go back to the caricature example, an approach which takes the locality and statistics of the image might be able to produce abstract features which are closer to the common sense description of interestingness. In particular, using wavelets may capture the locality of many data types [4]. Compare also the approach of non-linear PCA to image analysis (for the later task of classification, etc.) [15].

4 Algorithmic Information and Patterns

As noted by [11], a pattern may be conceived of the shortest program that generates a string. Otherwise, the concept of a pattern is something else entirely in every machine learning and data mining paper. By using bit strings and programs, we can give an objective, and universal definition of a pattern. Algorithmic information theory can then be used to define pattern operators in a way that is surprisingly close to cognitive processes. However, at this stage of our research, we do not yet concern ourselves with the programs, our patterns are simply bit strings for now.

In particular, information distance and normalized information distance which were briefly covered in Sect. 2 are universal measures of similarity that are completely independent of the application domain, and some amazingly simple implementations have achieved success in diverse domains and learning tasks. Our use of information theory is directly related to the concept of information distance. We also use conditional entropy to quantify structural difference.

5 A General Model

We are now going to generalize the set-theoretic definition of the classical frequent item set mining to cover a wider range of scientific measurement. Assume that we have samples of sensor data from a "fixed" instrumentation device, for instance image data from a radioastronomy telescope examining a certain region of space. Another example could be seismograph data which transmits measurements irregularly and for any number of samples.

Let transaction multi-set (set with repetition) $T = \{y \mid y \in \{0,1\}^*\}$ be the unordered list of observations drawn from the same domain. Let also bitstrings $x, y \in \{0,1\}^*$. We will say that an *abstract* pattern x occurs approximately in datum $y \in T$ iff:

1. $H(x) \leq c_1.H(y)$ (entropy reduction)
2. $H(x/y) \leq c_2.H(y)$ (noise exclusion)

where $0 < c_2 < c_1 < 1$. We denote "x occurs approximately in y" by $x \prec y$. Second condition is equivalent to stating that pattern x and datum y has mutual information as expected, i.e., $H(x : y) > 0$. Since $H(x : y) = H(x) - H(x/y) + O(1)$, $H(x/y) < H(x) - O(1)$, which is satisfied as $c_2.H(y) < H(x)$. Note that many equations introduce a small additive constant in AIT, which must be correctly handled by the algorithms, or non-patterns may be detected.

Having generalized the pattern occurrence operator in the set theoretic definition from the subset operation to the information-theoretic conditions, the problem definition is straightforward. Let the frequency function $f(T, x) = |\{x \prec y \mid y \in T\}|$. Our objective is the discovery of frequent patterns in a transaction set with a frequency of ϵ and more. The set of all frequent patterns is $\mathcal{F}(T, \epsilon) = \{x \in \{0,1\}^* \mid f(T, x) \geq \epsilon\}$, which is finite due to the entropy reduction condition. (Note that we consider the classical definition of Kolmogorov complexity as mentioned in Sect. 2). However, the size of $\mathcal{F}$ can be quite large, as in the frequent item set mining problem.

The downward closure lemma which states that the subsets of a frequent pattern are also frequent makes the Apriori algorithm possible in the context of frequent item set mining [2]. There is an analogue of the contrapositive of this lemma for our general formulation. Note that to simplify matters we assume a self-delimiting program encoding such as LISP. The analysis without the self-delimiting condition would introduce an additive logarithmic term which we would address separately.

Theorem 1. *If $x \notin \mathcal{F}(T, \epsilon)$ then $xy \notin \mathcal{F}(T, \epsilon)$. Less formally, any extension of an infrequent pattern is also infrequent.*

Proof. If $x \notin \mathcal{F}(T, \epsilon)$ then, $f(T, x) < \epsilon$. Let z be any datum in T for which it is not the case that $x \prec z$. Then, at least one of the pattern occurrence conditions does not hold. We can now analyze whether an extended $xy \prec z$.

– Suppose that the entropy reduction condition does not hold: $H(x) > c_1.H(z)$. Then, $H(x, y) > c_1.H(z)$ since $H(x, y) > H(x)$.
– Alternatively, suppose that the noise exclusion condition does not hold: $H(x/z) > c_2.H(z)$. Then, it doesn't hold for x, y either. $H((x, y)/z) = H(y/(x, z)) + H(x/z) + O(1)$ by subadditivity of algorithmic information. Since $H(y/(x, z)) > 0$ (since it has to be at least $O(1)$), then we find that $H((x, y)/z) > c_2.H(z)$.

Therefore, it is not the case that $xy \prec z$. Then, $f(T, xy) \leq f(T, x) < \epsilon$ which entails that $xy \notin \mathcal{F}(T, \epsilon)$.

6 Abstract Pattern Synthesis

By Theorem 1, we are inspired to write an algorithm which starts with a number of primitive candidate patterns and searches the pattern space in breadth first fashion like the Apriori algorithm. First, let us look at the calculation of pattern occurrence conditions.

6.1 Approximate Calculations

Algorithmic information content is uncomputable using a universal computer. Neither of the conditions we give are recursively enumerable. Fortunately, that should not trouble us too much, for we can use the methods mentioned in Sect. 2.2 to approximate these uncomputable values. However, it is arguable whether using a dictionary-based simple compressor is sufficient for the range of data mining applications we are interested in. At the present, the only obvious advantage of using a traditional compressor would seem to be efficiency.

We again approximate the conditional entropy using subadditivity of information $H(t/s) \approx H(s, t) - H(s)$. With a compressor $C(\cdot)$ such as gzip, the conditions become:

1. $C(x) \leq c_1.C(y)$ (entropy reduction)
2. $C(x, y) \leq (1 + c_2).C(y)$ (noise exclusion)

6.2 BFS in Pattern Space

We will adapt a generate and test strategy similar to Apriori for our first algorithm, applying the theory introduced in the paper. The pruning logic is quite similar, we do not extend infrequent patterns by Theorem 1. We will keep the algorithm as close as possible to APRIORI to show the relation, although there could be many efficiency improvements following various frequent itemset mining algorithms. MICRO-SYNTHETIC extends the pattern length by n bits at each iteration of the algorithm. Initially, a fast algorithm finds all frequent patterns up to n bits (akin to discovery of large items). The GENERATE procedure extends the frequent patterns of the previous level up to n bits. Then, a database pass is performed and the pattern occurrence conditions are checked for each candidate pattern and transaction element. Then, the algorithm iterates, generating candidates from the last level of frequent patterns discovered, until we reach a level where there are no frequent patterns, exactly as in APRIORI.

Algorithm 1. MICRO-SYNTHETIC(T, ϵ, c_1, c_2)

$F_0 \leftarrow \{|x| \leq n | x \in \{0,1\}^* \wedge f(T,x) >= \epsilon\}$
$k \leftarrow 1$
while $F_{k-1} \neq \emptyset$ **do**
  $C_k \leftarrow$ GENERATE(F_{k-1})
  for all $y \in T$ **do**
    for all $x \in C$ **do**
      if $C(x) \leq c_1.C(y) \wedge C(x,y) \leq (1+c_2).C(y)$ **then**
        $count[x] \leftarrow count[x] + 1$
      end if
    end for
  end for
  $F_k \leftarrow \{x \in C_k | \; count[x] \geq \epsilon\}$
  $k \leftarrow k + 1$
end while
return $\bigcup_k F_k$

7 Discussion

The algorithm is called MICRO-SYNTHETIC, because direct search in pattern space has obvious limitations. On the other hand, that is also what all frequent pattern discovery algorithms do, therefore it may not be at a greater disadvantage. Like in the basic frequent itemset mining algorithms, the support threshold must be given. However, we also require two extra parameters to delimit the pattern occurence. Unfortunately, our formulation falls short of the "parameter-free" ideal [7]. At the moment, we can give no guidelines for setting c_1 and c_2 except that they must be small enough. Especially c_2, which controls vagueness in our model. The basic frequent item set mining problem has no place for vagueness, the pattern relation is strict. On the other hand, our formulation places no bounds on the kind of data/pattern representation, and allows for vague representations, which are useful for a system that can abstract.

An implementation effort is ongoing. MICRO-SYNTHETIC has been implemented and tested on small datasets. We have tried a variety of compressors like gzip, bzip2 and PAQ8f for the information distance approximation.

While we have managed to find some interesting character patterns this way (such as finding an abstract pattern of 00000001111111 from example strings of different length which contain a sequence of 0's and 1's in them, with errors), we have observed that the suboptimality of the compressors (relative to the particular decompressor) causes too many random patterns to be found, which cannot be attenuated by the c_2 parameter. We have been thus working on a simple but optimal compressor that will fit out implementation better. After we get some results using the Micro-Synthetic on toy problems, we are planning to devise an algorithm with many optimizations to deal with more realistic data sets. We think that an implementation could demonstrate results on both traditional tabular datasets, and novel kinds of data due to the generality of data schema, depending on the availability of a suitable compressor.

An interesting merit of the Theorem 1 is that it might offer a partial but fundamental theoretical explanation of the success of hierarchical models typically used in deep learning, the compositionality of frequent patterns we exposed likely applies to any pattern recognition system.

Our approach has been criticized as having been superseded by the theory of Algorithmic Statistics [8], however the present paper only offers a generalized version of frequent pattern mining based on AIT, which was not addressed in that work, but perhaps may be reformulated in that framework. The abstract pattern definition was completely new at the time of the writing. The main theorem was also not seen elsewhere before we proposed it in 2006. A more directly relevant formulation of data analysis is Solomonoff's set induction model [17].

8 Conclusions and Future Work

We have made a high-level analysis of the frequent pattern discovery problem, by observing relations between the common sense notion of "abstraction", and the summarization task. We have determined objective criteria for a pattern to be an abstract representation. These criteria were interpreted as information theoretic conditions of reduced entropy and noise exclusion for a problem definition where patterns and data are any bitstring. We have replaced the pattern occurence operation in frequency mining with the conditions we have proposed. Thus, we have achieved a generalized version of the frequent pattern discovery problem. Thereafter, we have demonstrated that our conditions allow for pruning which is essential for the search in the vast but bounded pattern space. We have then used commonly employed methods to apply Kolmogorov complexity in real-world to design an algorithm suitable for the discovery task. Finally, we have introduced an Apriori like algorithm which enumerates all frequent patterns in our formulation.

Our research requires yet a lot of work to be done, both in the theory and experimental studies. First, there are more theoretical properties to be clarified, and alternative search methods should be analyzed. Especially, pattern space clustering methods and efficient representations may be sought. We have given an algorithm only for all frequent pattern discovery, the analogues of closed/maximal mining may be investigated. Second, a synthetic data set generator should be written, which highlights the virtues of our model and if possible real-world data should be tried out. Third, the effects of different kinds of compressors must be analyzed.

The present algorithm is mostly a theoretical proof-of-concept, we expect a universal data mining solution to achieve a lot more and proceed search in program space instead of pattern space, although practical pattern space search may also be desirable. We shall investigate both approaches further.

References

1. Kraskov, A., Stögbauer, H., Andrzejak, R.G., Grassberger, P.: Hierarchical clustering based on mutual information. Bioinformatics (2003)
2. Agrawal, R., Imielinski, T., Swami, A.N.: Mining association rules between sets of items in large databases. In: Buneman, P., Jajodia, S. (eds.) Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, Washington, D.C., pp. 207–216 (1993)
3. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: Bocca, J.B., Jarke, M., Zaniolo, C. (eds.) Proceedings of 20th International Conference on Very Large Data Bases, VLDB, pp. 487–499. Morgan Kaufmann (1994)
4. Siebes, A., Struzik, Z.: Complex data: mining using patterns. In: Hand, D.J., Adams, N.M., Bolton, R.J. (eds.) Pattern Detection and Discovery. LNCS, vol. 2447, pp. 24–35. Springer, Heidelberg (2002). doi:10.1007/3-540-45728-3_3
5. Bennett, C., Gacs, P., Li, M., Vitányi, P., Zurek, W.: Information distance. IEEE Trans. Inf. Theor. **44**(4), 1407–1423 (1998)
6. Chaitin, G.J.: Algorithmic Information Theory. Cambridge University Press, Cambridge (1987)
7. Keogh, E., Lonardi, S., Ratanamahatana, C.A.: Towards parameter-free data mining. In: Proceedings of the 2004 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2004)
8. Gács, P., Tromp, J., Vitányi, P.M.B.: Algorithmic statistics. IEEE Trans. Inf. Theor. **47**(6), 2443–2463 (2001). https://doi.org/10.1109/18.945257
9. Hipp, J., Güntzer, U., Nakhaeizadeh, G.: Algorithms for association rule mining - a general survey and comparison. SIGKDD Explor. **2**(1), 58–64 (2000)
10. Li, M., Chen, X., Li, X., Ma, B., Vitányi, P.M.B.: The similarity metric. IEEE Trans. Inf. Theor. **50**(12), 3250–3264 (2004). https://doi.org/10.1109/TIT.2004.838101
11. Lin, T.: Mathematical theory of high frequency patterns. In: Proceedings of the IEEE ICDM 2004 Foundations of Data Mining Workshop (2004)
12. Minsky, M.: The Emotion Machine: Commonsense Thinking, Artificial Intelligence, and the Future of the Human Mind, chap. 6. Simon and Schuster (2006)
13. Özkural, E.: Teraflop-scale incremental machine learning. CoRR arXiv:1103.1003 (2011)
14. Cilibrasi, R., Vitányi, P.: Clustering by compression. Technical report CWI (2003)
15. Rosipal, R., Girolami, M., Trejo, L.: Kernel PCA for feature extraction and denoising in non-linear regression. Neural Comput. Appl. **10**, 231–243 (2000)
16. Cilibrasi, R., Vitanyi, P., de Wolf, R.: Algorithmic clustering of music
17. Solomonoff, R.J.: Two kinds of probabilistic induction. Comput. J. **42**(4), 256–259 (1999). https://doi.org/10.1093/comjnl/42.4.256
18. Zaki, M.J.: Parallel and distributed association mining: a survey. IEEE Concurr. **7**(4), 14–25 (1999)