Agglomerating Local Patterns Hierarchically with ALPHA

Loïc Cerf
Université de Lyon, CNRS
INSA-Lyon, LIRIS, UMR5205
F-69621 France
lcerf@liris.cnrs.fr

Pierre-Nicolas Mougel
Université de Lyon, CNRS
INSA-Lyon, LIRIS, UMR5205
F-69621 France
pnmougel@insa-lyon.fr

Jean-François Boulicaut
Université de Lyon, CNRS
INSA-Lyon, LIRIS, UMR5205
F-69621 France
jboulica@liris.cnrs.fr

ABSTRACT

To increase the relevancy of local patterns discovered from noisy relations, it makes sense to formalize error-tolerance. Our starting point is to address the limitations of state-of-the-art methods for this purpose. Some extractors perform an exhaustive search w.r.t. a declarative specification of error-tolerance. Nevertheless, their computational complexity prevents the discovery of large relevant patterns. ALPHA is a 3-step method that (1) computes complete collections of closed patterns, possibly error-tolerant ones, from arbitrary n-ary relations, (2) enlarges them by hierarchical agglomeration, and (3) selects the relevant agglomerated patterns.

Categories and Subject Descriptors: H.2.8 [**Database Management**]: Database Applications—*Data mining*

General Terms: Algorithms

Keywords: Noise, n-ary relation, clustering, relevancy

1. INTRODUCTION

Many datasets of practical interest are n-ary relations and we are designing pattern discovery techniques from such 0/1 data where a 1 (resp. 0) value denotes that a given n-tuple does (resp. does not) belong to the relation. This paper deals with discovering relevant local patterns from noisy n-ary relations. The complete approaches extracting exact patterns (e.g., closed sets or formal concepts) are not suited. Indeed, even in the simpler case of binary relations, a faint noise makes the collections of computed patterns become far too large. Moreover their patterns are much smaller than what would be found in the same relations deprived of noise. To guarantee extraction completeness while improving the relevancy, several definitions of *error-tolerant patterns* have been proposed (see [5] for a survey). Nevertheless, the space and/or time complexities of the related algorithms make them fail in many practical settings. This situation gets even worse when n increases. Therefore, tedious or even impossible interpretation phases have to be performed by experts. However, they know that agglomerating much overlapping patterns often allows to discover some relevant larger ones (see, e.g., the concept of quasi-synexpression group discovery considered in [1]). The ALPHA method exploits this idea. It is structured in three steps:

1. Complete collections of (possibly error-tolerant) closed patterns holding in an n-ary relation are computed.
2. Using both the collection of patterns computed beforehand and the noisy relation they were extracted from, the local patterns are hierarchically agglomerated.
3. Relevant clusters are selected from the clustering structure. A coverage property leads this step.

The next section provides the needed definitions and it discusses the use of a hierarchical agglomeration. Section 3 describes the selection of the relevant local patterns. In Sec. 4, the added-value of ALPHA is empirically assessed. Section 5 discusses related work, and Sec. 6 briefly concludes.

2. GROUPING LOCAL PATTERNS

2.1 Input Local Pattern Collection

ALPHA supports the discovery of relevant local patterns from arbitrary n-ary relations. FENSTER [3] is an error-tolerant complete extractor that does not make any assumption on the arity of the mined relation. ALPHA uses it for its first step. FENSTER extracts every *error-tolerant closed n-set*. It essentially is a dense subspace of the dataset, i.e., representing an n-ary relation as a hyper-rectangle of 0/1 values, it is a sub-hyper-rectangle (modulo any permutation of the hyper-plans of any dimension) containing *few* 0 values.

Definition 1. Given n sets $(D^i)_{i=1\dots n}$, $\mathcal{R}$ is an *(n-ary) relation* on $(D^i)_{i=1\dots n}$ iff $\mathcal{R} \subseteq D^1 \times \cdots \times D^n$. Given $\mathcal{R}$, $(X^1, \dots, X^n)$ is an *n-set* iff $\forall i = 1 \dots n, X^i \subseteq D^i$.

Definition 2. Given a relation $\mathcal{R}$ on $(D^i)_{i=1\dots n}$, n error-tolerance parameters $(\epsilon^1, \dots, \epsilon^n) \in \mathbb{N}^n$ and an n-set $X = (X^1, \dots, X^n)$, X is an error-tolerant closed n-set iff it satisfies the two following constraints:

$$\mathcal{C}_{\text{dense}}(X) \equiv \quad \forall i = 1 \dots n, \forall x \in X^i, \\ |(X^1 \times \cdots \times \{x\} \times \cdots \times X^n) \setminus \mathcal{R}| \leq \epsilon^i$$

$$\mathcal{C}_{\text{closed}}(X) \equiv \quad \forall Y \in \{(Y^1, \dots, Y^n) | \forall i = 1 \dots n, X^i \subseteq Y^i\}, \\ \mathcal{C}_{\text{dense}}(Y) \Rightarrow Y = X$$

CIKM'09, November 2–6, 2009, Hong Kong, China.

An error-tolerant closed n-set generalizes the closed n-set pattern type (when $(\epsilon^1, \ldots, \epsilon^n) = (0, \ldots, 0)$), which, itself, encompasses the popular closed itemset (associated with its support) that was often used to mine transactional datasets.

Further details on the considered pattern types are not necessary to understand ALPHA. Indeed ALPHA neither imposes any restriction on the number of n-tuples absent from the relation a pattern contain nor on its maximality. Actually, other pattern types capturing some relevant regularities may feed ALPHA. This will be briefly discussed in Sec. 5.

2.2 On the Impact of Noise

Forcing the extracted local patterns to be *closed* discards many smaller patterns that are directly derivable from a closed one ($\mathcal{C}_{\text{dense}}$ being anti-monotonic). Nevertheless, in practical settings, the complete collection of closed patterns remains huge and many of them are very similar to each other. In the particular case of binary relations, [7] theoretically and empirically shows that the number of frequent itemsets exponentially grows with the level of noise while their sizes exponentially decrease with it. With a relation of a higher arity n, the situation turns out to be dramatic. Indeed, the number of n-tuples a hidden pattern covers exponentially increases with n. As a consequence the noise has an exponentially greater probability to damage some of these n-tuples. Moreover, at a fixed number of false-negative n-tuples, which should be present in the relation but are missing, the number of closed n-sets linearly increases with n ($\mathcal{C}_{\text{closed}}$ is a maximality constraint w.r.t. *every* attribute).

Mining complete collections of error-tolerant patterns in n-ary relations is computationally hard. Error-tolerance widens the traversed search space because it delays pruning strategies. As a result, using large ϵ^i values (see Def. 2) for complete error-tolerant closed n-set extractions can be infeasible on large datasets. In practice, even a faint noise brings any complete extractor to its limits, i.e., the noise altering the "real" patterns cannot be entirely compensated while keeping the extraction tractable. Instead of damning the analyst to the interpretation of long lists of insufficiently error-tolerant (hence too small and much-overlapping) closed n-sets, ALPHA supports an *automatic* intermediary task between the complete extraction of patterns and the actual interpretation. This approach looks trustier than fully heuristic ones because the lossy heuristics are delayed as far as possible in the knowledge discovery process.

2.3 A Pattern Clustering Scheme

The intermediary task, introduced in the previous section, can be compared to an n-dimensional jigsaw puzzle: every piece is a pattern returned during the complete extraction phase and the image to construct is a perfect version of the one given on the box (the dataset), which is, contrary to classical jigsaw puzzle, altered by some noise. The perfect image must be composed of large (possibly) overlapping hyper-rectangles (modulo any permutation of the hyper-plans of any dimension) of 1 values "embedded" in a 0 valued hyper-space. A pattern clustering approach will help us solve this tough game.

The quality of the constructed image should not be reduced to how well every piece interlocks with its neighbors. This image should also match the noisy one on the box. To take into account this objective, the global (at every iteration, a global clustering refines the previous one) and divisive (the complete collection of local patterns, considered as a whole, is successively divided into smaller clusters) clustering approaches are, by nature, not much suitable. On the contrary, a bottom-up agglomeration successively merges previously established clusters (the previously assembled pieces) into larger ones, hence allowing to test the partial results against the original dataset (the noisy image on the box). The fact it does not require to fix, a priori, a number of clusters is also extremely useful. Indeed, the number of relevant local patterns to discover usually is unknown. Therefore a hierarchical agglomeration is chosen and ALPHA requires a merging operator and a metric.

1	**1**	**1**	0	1	1
1	**1**	**1**	1	0	1
1	**1**	**1**	1	1	0
0	1	1	**1**	**1**	**1**
1	0	1	**1**	**1**	**1**
1	1	0	**1**	**1**	**1**

(a)

1	**1**	**1**	0	0	0
1	**1**	**1**	0	0	0
1	**1**	**1**	0	0	0
0	0	0	**1**	**1**	**1**
0	0	0	**1**	**1**	**1**
0	0	0	**1**	**1**	**1**

(b)

Figure 1: Two binary relations.

Our merging operator essentially is a union denoted $\cup$. Since the hidden local patterns are hyper-rectangles, the agglomeration of two n-sets is defined as the n-set with the minimal envelope enclosing both of them.

Definition 3. Given two n-sets $X = (X^1, \ldots, X^n)$ and $Y = (Y^1, \ldots, Y^n)$, $X \cup Y$ denotes $(X^1 \cup Y^1, \ldots, X^n \cup Y^n)$.

The importance of using the initial relation to define the metric on n-sets is illustrated by the two relations in Fig. 1. Restricted to the closed 2-sets having at least 3 elements per attribute (bold 1 in the figures), the two settings are identical. Nevertheless the closed 2-sets of Fig. 1(a) obviously are better candidates for an agglomeration than those of Fig. 1(b). In fact, the reasoning behind the definition of an error-tolerant local pattern can be applied to the n-set whose outline is the envelope of the two patterns. Thus, our distance between two n-sets is based on the proportion of n-tuples absent from the relation in the worst hyper-plan of their union. When the noise distribution is approximately known (and not uniform), weights can be used:

Definition 4. Given a weight function $w : \cup_{i=1}^{n} D^i \to \mathbb{R}^+$ (greater weights mean less tolerance to noise) and an n-set $X = (X^1, \ldots, X^n)$, $d(X)$ denotes its intrinsic distance:

$$d(X) = \max_{i=1}^{n} \left(\max_{x \in X^i} \left(w(x) \frac{|K \setminus \mathcal{R}|}{|K|} \right) \right)$$

where $K = X^1 \times \cdots \times X^{i-1} \times \{x\} \times X^{i+1} \times \cdots \times X^n$.

Definition 5. The distance between two n-sets X and Y is $d(X \cup Y)$.

The agglomerated n-sets can be organized in a dendrogram. It contains more local patterns than the collection initially extracted[1] but some of them now are relevant because they tolerate enough noise. For a visual interpretation, the height of an agglomerated pattern (or *cluster*) in the dendrogram is advantageously set to its intrinsic distance. Nevertheless, ALPHA further delays the manual interpretation by automatically ranking the clusters by relevancy and dropping the least relevant ones thanks to a coverage test.

[1] $2N - 1$ clusters for N closed error-tolerant n-sets.

3. SELECTING RELEVANT N-SETS

3.1 Quantifying Cluster Relevancy

According to [6], "A local pattern is a data vector serving to describe an anomalously high local density of data points". In the context of an n-ary relation $\mathcal{R}$, a relevant cluster X describes an "anomalously high local density" of n-tuples present in $\mathcal{R}$ when, *simultaneously*, (a) it is apart from the rest of the data ("anomalously"), i.e., it maximizes its distance with the other clusters (but its ancestors and descendants), and (b) it minimizes the proportion of n-tuples absent from $\mathcal{R}$ on its worst hyper-plan ("high local density"). Both are easily quantified:

(a) The minimal distance between a parent cluster and its two children X and Y, $d(X \cup Y) - \max(d(X), d(Y))$, is how distant X and Y are from each other and, even more, from the other clusters (since these two clusters were the closest when they were agglomerated).

(b) The intrinsic distance measure of X, $d(X)$, is the proportion of absent n-tuples on its worst hyper-plan.

Both quantities being proportions of n-tuples absent from $\mathcal{R}$, the relevancy of X can now be computed by difference.

Definition 6. Given an n-set X and its parent $X \cup Y$ after hierarchical agglomeration, $r(X)$ denotes the relevancy of X:

$$r(X) = d(X \cup Y) - \max(d(X), d(Y)) - d(X)$$

3.2 Selecting the Relevant Clusters

Though sorted by relevancy, the list of patterns to interpret remains very long and its tail contains poorly relevant clusters. In particular, it contains the initial collection of patterns (leaves of dendrogram). ALPHA automatically decides where to cut off this tail. It simply reads, by decreasing relevancy order, the list of clusters and considers the leaves they cover. Once every leaf is covered by at least one previously read cluster, ALPHA removes the clusters with lower relevancy values. The completeness of the initial extraction is somehow preserved since every pattern of the related collection is part of at least one output cluster. Thus, ALPHA assumes that every initially extracted closed error-tolerant n-set is a fragment of some relevant local pattern. As a consequence, a cluster with a lower relevancy than at least one of its ancestors is not to be kept. Indeed, it must be a fragment of such a larger ancestor. However both a large cluster and one of its sub-clusters are considered relevant if the latter has a greater relevancy: it describes an "anomalously high local density" of present n-tuples inside another anomalously high, but lower, local density of present n-tuples. Notice also that the actual satisfaction of the assumption made by ALPHA does not matter a lot. If a closed error-tolerant n-set partly covers regions of the dataset out of any relevant local pattern, it receives a high relevancy (being far away from the other clusters) but, being small, it can easily be filtered out by size constraints in a final post-processing step.

4. EXPERIMENTAL VALIDATION

A real-life 4-ary relation was successfully mined with ALPHA. Because of space constraints, the results published here are only those obtained on synthetic relations that allow to quantitatively benchmark the quality. Indeed, the hidden patterns to be discovered and the exact level of introduced noise are known. Ten synthetic 3-ary relations, with 32 elements per attribute domain, are generated. Four hidden 3-sets, with eight elements per attribute, are randomly placed and may overlap. A uniform random noise is then applied on every relation. Its *level* is the proportion of 3-tuples switched from "present in the relation deprived of noise" to "absent" and vice versa. The figures in this section plot the mean of the measures obtained with the ten relations. In practical settings, the analyst usually cannot know he/she is looking for $8 \times 8 \times 8$ patterns. Hence, the output n-sets are only forced to gather at least five elements per attribute.

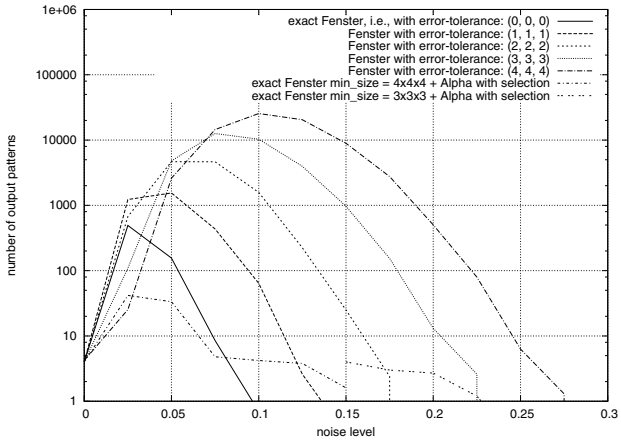

Figure 2: Size of the output collections.

Given the collection of hidden local patterns (those in the relation deprived of noise) and the collection of extracted local patterns, two pieces of information rate how useful the latter is for discovering the former: the size of the returned collection (cost for interpreting it) and the average similarity between every hidden pattern and its most similar counterpart among the extracted patterns (quality of the knowledge discovery). The latter is formally expressed as follows.

Definition 7. Given $\mathcal{H}$ a set of hidden patterns, $\mathcal{P}$ a set of extracted patterns and $s : \mathcal{H} \times \mathcal{P} \to [0, 1]$ a similarity measure, the quality of $\mathcal{P}$, ranging in $[0, 1]$, is:

$$\frac{1}{|\mathcal{H}|} \sum_{H \in \mathcal{H}} (\max_{P \in \mathcal{P}}(s(H, P))$$

In our experiments, the similarity measure s between two n-sets is a classical distance between their sets of elements.

Definition 8. Given two n-sets $X = (X^1, \ldots, X^n)$ and $Y = (Y^1, \ldots, Y^n)$, the similarity between X and Y is:

$$s(X, Y) = \frac{1}{n} \sum_{i=1}^{n} \frac{|X^i \cap Y^i|}{|X^i \cup Y^i|}$$

Figures 2 and 3 compare the results of FENSTER to those of FENSTER + ALPHA. When run alone, FENSTER is tested with five different tolerances to noise (the ϵ^i parameters of Def. 2). When used as a pattern collection provider to ALPHA, FENSTER extract *exact* closed 3-sets, i.e., $(\epsilon^1, \epsilon^2, \epsilon^3) = (0, 0, 0)$. The "completeness as far as tractable" (see Sec. 2.2) is not respected here so that a clear assessment of ALPHA's

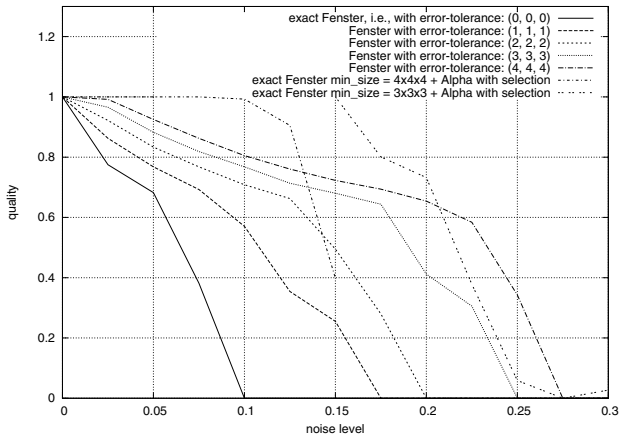

Figure 3: Quality of the output collections.

added-value can be achieved (it deals with *all* the noise). Despite this setting and even with the greatest tested tolerance to noise (implying very long extraction times) for FENSTER alone, FENSTER + ALPHA significantly outperforms it. It returns far less patterns (see Fig. 2) and almost perfectly recovers the hidden ones up to a noise level of 0.15 (see Fig. 3).

The size constraints imposed by FENSTER to the closed 3-sets feeding ALPHA are meant to provide enough *fragments* to construct relevant agglomerated patterns. Thus, they are set by merely looking at the size of the returned collection. Hence, in our experiments, when the noise level reaches 0.15, these constraints are loosened to "at least three elements per attribute". Indeed, with a noise level of 0.15, FENSTER extracts, in average, only 16.1 closed 3-sets gathering four elements per attribute or more. Clearly, this is insufficient.

5. RELATED WORK

ALPHA takes, as input, any collection of n-sets (see Def. 1) but a complete collection is trustier. Many such extractors work with binary relations. Some tolerate noise (see [5] for a survey). DATA-PEELER [3] and its generalization [2] respectively extract *exact* and *error-tolerant* closed n-sets (see Def. 2). CACTUS [4] and CLICKS [10] extract *subspace clusters* from any n-ary relations. These *local patterns* (rather than *clusters*) are of the form $(X^1, \ldots, X^m)$ where every X^i is a subset of one attribute domain and $m \leq n$. Those with $m = n$ could feed ALPHA. They tolerate noise in a far more constrained way than the closed error-tolerant n-sets.

Clustering complete collections of patterns to make them larger and tolerate some noise is not a new idea. Back in 1995, [8] applies it to association rules. However, to the best of our knowledge, the first metrics taking into account regions of the data outside the two patterns to agglomerate, were published last year. Indeed, [9] considers the (binary) relation they were extracted from too. Nevertheless, it favors one attribute, whereas ALPHA does not. The extension of pattern clustering methods beyond 2D-datasets is recent too. TRICLUSTER [11] discovers complete collections of local patterns in real-valued tensors (in particular it can extract exact closed 3-sets) and a post-process handles noise by deleting or merging some patterns. The involved distances only are counts of covered 3-tuples, i.e., their values (0 or 1 in the case of a relation) are not taken into consideration.

6. CONCLUSION

Mining relevant local patterns from noisy arbitrary n-ary relations is challenging. In practical settings, the computational complexity of complete extractors does not allow to fix as much error-tolerance as required. The proposed method post-processes a collection of such (insufficiently) error-tolerant patterns. First, the patterns are hierarchically agglomerated. The involved metric takes into account not only the clustered patterns but also the relation they were extracted from. Then, the agglomerated patterns are ranked by relevancy. This measure is not only based on the distance to the other clusters but also on the inherent density of the large patterns they represent. Finally, a simple coverage test drops the least relevant patterns.

Acknowledgement: This work was partly funded by the ANR project BINGO2 (MDCO 2007).

7. REFERENCES

[1] S. Blachon, R. Pensa, J. Besson, C. Robardet, J.-F. Boulicaut, and O. Gandrillon. Clustering formal concepts to discover biologically relevant knowledge from gene expression data. *In Silico Biology*, 7(0033):1–15, July 2007.

[2] L. Cerf, J. Besson, T. K. N. Nguyen, and J.-F. Boulicaut. An exhaustive search for error-tolerant patterns in arbitrary n-ary relations. Technical report, LIRIS, June 2009. Under evaluation.

[3] L. Cerf, J. Besson, C. Robardet, and J.-F. Boulicaut. Closed patterns meet n-ary relations. *ACM Trans. on Knowledge Discovery from Data*, 3(1):1–36, March 2009.

[4] V. Ganti, J. Gehrke, and R. Ramakrishnan. CACTUS—Clustering categorical data using summaries. In *KDD '99: Proc. of the fifth SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pages 73–83. ACM Press, 1999.

[5] R. Gupta, G. Fang, B. Field, M. Steinbach, and V. Kumar. Quantitative evaluation of approximate frequent pattern mining algorithms. In *KDD '08: Proc. of the 14th SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pages 301–309. ACM Press, 2008.

[6] D. J. Hand. Pattern detection and discovery. In *Proc. of the ESF Exploratory Workshop on Pattern Detection and Discovery*, volume 2447 of *LNCS*, pages 1–12. Springer, Heidelberg, 2002.

[7] J. Liu, S. Paulsen, X. Sun, W. Wang, A. B. Nobel, and J. Prins. Mining approximate frequent itemsets in the presence of noise: Algorithm and analysis. In *SDM '06: Proc. of the 6th SIAM Int. Conf. on Data Mining*, pages 405–416. SIAM, 2006.

[8] H. Toivonen, M. Klemettinen, P. Ronkainen, K. Hätönen, and H. Mannila. Pruning and grouping discovered association rules. In *Proc. of the ECML '95 Workshop on Statistics, Machine Learning and Knowledge Discovery in Databases*, pages 47–52, 1995.

[9] A. K. C. Wong and G. C. L. Li. Simultaneous pattern and data clustering for pattern cluster analysis. *IEEE Trans. on Knowledge and Data Engineering*, 20(7):911–923, July 2008.

[10] M. J. Zaki, M. Peters, I. Assent, and T. Seidl. CLICKS: An effective algorithm for mining subspace clusters in categorical datasets. *Data & Knowledge Engineering*, 60(1):51–70, January 2007.

[11] L. Zhao and M. J. Zaki. TRICLUSTER: An effective algorithm for mining coherent clusters in 3D microarray data. In *SIGMOD '05: Proc. of the 24th SIGMOD Int. Conf. on Management of Data*, pages 694–705. ACM Press, 2005.