Decision Support Systems 57 (2014) 139–149

Contents lists available at ScienceDirect

Decision Support Systems

journal homepage: www.elsevier.com/locate/dss

A hybrid heuristic approach for attribute-oriented mining

Maybin K. Muyeba [a,*], Keeley Crockett [a], Wenjia Wang [b], John A. Keane [c]

[a] SCMDT, Manchester Metropolitan University, UK
[b] School of Computing Sciences, University of East Anglia, UK
[c] School of Computer Science, University of Manchester, UK

ARTICLE INFO

Article history:
Received 16 February 2012
Received in revised form 18 July 2013
Accepted 22 August 2013
Available online 4 September 2013

Keywords:
Induction
Heuristic
Threshold
Interestingness
Cluster
Algorithm

ABSTRACT

We present a hybrid heuristic algorithm, *clusterAOI*, that generates a more interesting generalised table than obtained via attribute-oriented induction (AOI). AOI tends to overgeneralise as it uses a fixed global static threshold to cluster and generalise attributes irrespective of their features, and does not evaluate intermediate interestingness. In contrast, *clusterAOI* uses attribute features to dynamically recalculate new attribute thresholds and applies heuristics to evaluate cluster quality and intermediate interestingness. Experimental results show improved interestingness, better output pattern distribution and expressiveness, and improved runtime.

1. Introduction

Pattern interestingness is determined by an objective measure [18] or by subjective user interpretation [15]. Threshold-driven algorithms [18,12] generate many rules which need to be filtered to determine interestingness [15]. Attribute-oriented induction (AOI) [9] extracts high-level generalised rules by repeatedly replacing and clustering [14,16] attribute values using domain knowledge[1] [9,17]. AOI uses attribute and relation generalisation thresholds to limit the number of distinct attributes and rules generated.

1.1. Problem and approach

We aim to obtain generalised and hence more interesting rules than AOI. AOI overgeneralises to “ANY” values [3,13,14] as it uses a fixed global static threshold to generalise attributes irrespective of their features and does not dynamically evaluate interestingness. The aim here is to use attribute features to dynamically recalculate new thresholds, and to apply heuristics to evaluate cluster quality and interestingness. The most interesting rules consist of mostly interior concepts [6,7,14].

This paper presents *clusterAOI*, a hybrid heuristic algorithm based on [16], which produces a more interesting generalised table than AOI. A three-fold strategy is used: (1) generalise conservatively [14] selected clusters of attribute values that share common properties and satisfy a newly computed local attribute threshold; (2) evaluate intermediate interestingness result for each attribute and of the algorithm, using heuristic functions [16]; (3) apply Kullback–Leibler (KL) divergence and cluster quality (CQ) interestingness [16] to the output [10]. Experiments show improved interestingness (up to 4 times), better output pattern distribution and expressiveness (about 1.5 times), and improved runtime (about 2 times).

The process is as follows: (1) *pre-clusterAOI* analyses attribute features to dynamically generate local thresholds; (2) *intermediate-clusterAOI* uses probabilistic semantic similarity between clusters of attribute values and evaluates cluster interestingness, resulting in improved interestingness and runtime; (3) in *post-clusterAOI* the final output table's interestingness is determined using CQ (a global harmonic mean) and KL.

As an example we apply AOI and *clusterAOI* to the cancer Wisconsin dataset [19] (Table 1). We calculate KL for divergence and cluster quality (CQ). *clusterAOI* gave 0% overgeneralisation while AOI gave 50%; KL was 1.7 times higher and CQ 3 times higher. *clusterAOI* also produced twice as many informative rules (NOT-ANY), i.e. 100% compared to 50% for AOI. Similar weaknesses were highlighted in [21]. Overall, *clusterAOI* improves pattern understandability, intelligent interpretation and interestingness.

The rest of the paper is structured as follows: Section 2 discusses related work; Section 3 presents prerequisites and definitions; Section 4 introduces *pre-clusterAOI*; Section 5 presents *intermediate-clusterAOI* and *post-clusterAOI*; Section 6 describes experimentation; and Section 7 concludes. A running *Case Study* (Table 2) is extended as each aspect of the approach is introduced.

[*] Corresponding author.
E-mail addresses: m.muyeba@mmu.ac.uk (M.K. Muyeba), k.crockett@mmu.ac.uk (K. Crockett), Wenjia.Wang@uea.ac.uk (W. Wang), jak@cs.man.ac.uk (J.A. Keane).

[1] Use of domain knowledge has been limited [20] and is an acknowledged difficult problem [5].

0167-9236/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.dss.2013.08.012

Table 1
Comparing final output on NOT-ANY values, breast cancer dataset [19].

Algorithm	cellSize	bNuclei	nNuclei	Mitoses	Count	%ANY	%NOT-ANY
AOI	AboutAve	AboutAve	Any	Any	485	50	50
	AboutAve	AboveAve	Any	Any	93		
G.Thr = 2, KL = 0.63, CQ = 11.59							
clusterAOI	AboutAve	AboutAve	AboutAve	AboutAve	483	0	100
	AboutAve	AboveAve	AboutAve	AboutAve	99		
	AboveAve	AboveAve	AboveAve	AboutAve	71		
G.Thr = 2, KL = 1.08, CQ = 34.6							

2. Related work

Generally, AOI algorithms [3,4,8,9,13,14] are threshold-driven i.e. they stop generalisation when thresholds are reached (the only interestingness measure used), and do not consider attribute features and proprieties [3]. Further, most AOI algorithms evaluate interestingness before (pre-AOI) and after (post-AOI) generalisation, less so during (intermediate-AOI) generalisation. Differences still exist in these algorithms. For pre-AOI, [21] removes discriminating data that may affect interestingness. Others [1,7] analyse depths and weighted heights of concept hierarchies to determine interestingness, but only use a single fixed weight value for interior concepts which, naturally, may vary between hierarchies. For intermediate-AOI, [3] uses multiple-level support thresholds per attribute and order generalised tuples according to association strength. Others [1,11] select the next attribute generalisation path to follow but are computationally-intensive. In [13], repeating attribute values are preserved to minimize overgeneralisation and produce many output rules. In post-AOI [6,16], the number of interior concepts in the output is applied as an interestingness measure using only the original global thresholds. Further the algorithm in [21] is unsuitable for large datasets (its order complexity being $O(n^3)$).

Existing work has been applied in isolation and largely overgeneralises the rules. Still, there remain issues such as manual selection of thresholds that may be unsuitable to apply globally to all attributes (one size fits all problem). Secondly, no AOI algorithm evaluates interestingness in all three phases: pre-, intermediate- and post-. Our earlier work suggests that there can be improvements in interestingness of generalised patterns [16].

We propose a coordinated hybrid algorithm, *clusterAOI*, to address these limitations: *clusterAOI* has three phases: *pre-* (Section 4)—addresses attribute feature measure (or entropy); *intermediate-* (Section 5)—evaluates interestingness during generalisation (locally and globally) using attribute clustering functions [16,17]; and *post-* (Section 5)—evaluates interestingness of rules (from clusters) using cluster similarity, tightness and local interestingness functions; and overall via the KL function [10], often used in information theory, which gives differences in information divergence between data distributions in the rules.

Table 2
Ball data.

Diameter	Colour
2	Red
7	Blue
34	Yellow
25	Green
28	Orange
8	Violet
16	Red

3. Prerequisites and definitions

clusterAOI addresses interestingness as follows: let relation R be defined on dataset $D \subseteq R$ with n tuples; attribute A_i and attribute hierarchy H_i pairs exist for m attributes i.e. $\{(A_1,H_1),(A_2,H_2),\ldots,(A_m,H_m)\}$, A_{m+1} is an attribute storing the count of tuples in R and t is a global threshold. Then $\sum |A_{m+1}, \varnothing| = n$, with domain values $Dom(A_{m+1}, \varnothing) \in Z^+$, where Z^+ are positive integers, with $H_{m+1} = \varnothing$. Given a generalisation space $B_i = A_i \cup H_i$ for each attribute, we use entropy function $\nabla(A_i)$ to generate new local thresholds $\{L.thr_i : L.Thr_i \geq G.Thr,\ i = 1,\ldots,m\}$, for each A_i, where $G.Thr$ is the global threshold. $L.Thr_i \geq G.Thr$ means at most $|L.Thr_i|$ distinct values per attribute, thus minimising over generalisation. With a description language $\mathrm{L} = (\mathrm{B_i}, f)$, there is a level-by-level "nearest parent" generalisation function $f : B_i \rightarrow Dom(H_i)$ and a partial order $(\prec, B_i)$ for finding parents (or descriptions) $\{\varphi_1,\varphi_2,\ldots,\varphi_k\}$ in B_i i.e. a cluster $\{\varphi_1,\ldots,\varphi_j\}$ has parent $\varphi' = min\,\{f(\varphi_1),\ldots,f(\varphi_j)\}$ [17]. This leads to Definitions 1 and 2.

Definition 1. *Cluster.* Given attribute $A_i = \{a_1^i,..,a_k^i\}$, an attribute cluster of A_i is defined as $C_j = \{c_1,\ldots,c_n\}$, $C_j \subseteq A_i$, $n \leq k$.

Definition 2. *Generalisable cluster.* Given a cluster $C_j = \{c_1,\ldots,c_n\}$ of A_i and local threshold $\alpha = L.Thr_i$, cluster C_j is *generalisable* if $|C_j| \geq \alpha$ and $f(c_k) = f(c_l), \forall\, k, l \leq n, k \neq l$.

Generalisation of each attribute stops when its optimal value (*a local interestingness value*) is reached (Definition 3, Section 5.1), and in the global case, when a global optimal value is encountered (Theorem 1, Section 5.1). We derive these values by applying heuristic functions to attribute clusters. Without loss of generality, interestingness [16] can be described by both distance and cluster tightness [17] depending on tuple distribution in a summary table [10] (a cluster of attribute values). The agglomerative hierarchical clustering distance δ_n and tightness τ_n functions [17] are used for overall interestingness: $G_n : (\delta_n, \tau_n) \rightarrow \Re^+$. These functions exhibit both monotone and anti-monotone properties during generalisation. Therefore, the problem of mining generalised patterns is a 4-tuple $(\nabla, I_L^i, f, I_g^T)$ (see Appendix E) defined as follows:

(1) Find attribute significance (entropy) $\nabla(A_i)$ and generate new local threshold $L.Thr_i$; discussed in Sections 4.1 and 4.2;
(2) Find local attribute interestingness in iteration k and aggregate values using a cluster tightness function $I_L^i(A_i)$ discussed in Section 5.1;
(3) Generalise values using distance function $f(B_i)$ subject to $L.Thr_i$;
(4) Find global cluster interestingness $I_g^T(\ldots)$ of table T by aggregating local values from (2) using Eq. (7) discussed in Sections 5.2–5.4.

After rule generation, we then measure divergence (using KL) and cluster quality (CQ) in the rules. KL is an information divergence measure between two probability distributions (uniform and actual): higher values show good distribution and variety of output values, indicating improved interestingness [10]. Given m tuples in a table $T = \{t_1,\ldots,t_m\}$ and actual probabilities $\{p_1,\ldots,p_m\}$, the divergence is $KL(T) = \log_2 m - \sum_{i=1}^{m} p_i \log_2 p_i$, where $KL(T) \geq 0$, bounded by $\log_2 m$. CQ is an interestingness heuristic function [16] applied to the top k rules of the output (Eq. (7), Section 5.2), similar to heuristics in [1,7].[2]

4. Pre-clusterAOI

Pre-clusterAOI aims to find each attribute's local threshold *L.Thr* (from *G.Thr*) using attribute features such as concept hierarchy and distinct values.

[2] Notation is collected in Appendix E.