Information Fusion 40 (2018) 112–125

Contents lists available at ScienceDirect

Information Fusion

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

# Preference rules for label ranking: Mining patterns in multi-target relations

Cláudio Rebelo de Sá[a,b,*], Paulo Azevedo[d], Carlos Soares[c], Alípio Mário Jorge[e], Arno Knobbe[b]

[a] INESC TEC, Porto, Portugal
[b] LIACS, Leiden, Netherlands
[c] INESC TEC, Faculdade de Engenharia, Universidade do Porto, Porto, Portugal
[d] HasLab, INESC TEC, Departamento de Informática, Universidade do Minho, Braga, Portugal
[e] INESC TEC, Faculdade de Ciências, Universidade do Porto, Porto, Portugal

ARTICLE INFO

*Article history:*
Received 18 August 2016
Revised 22 June 2017
Accepted 15 July 2017
Available online 17 July 2017

*Keywords:*
Label ranking
Association rules
Pairwise comparisons

ABSTRACT

In this paper, we investigate two variants of association rules for preference data, Label Ranking Association Rules and Pairwise Association Rules. Label Ranking Association Rules (LRAR) are the equivalent of Class Association Rules (CAR) for the Label Ranking task. In CAR, the consequent is a single class, to which the example is expected to belong to. In LRAR, the consequent is a ranking of the labels. The generation of LRAR requires special support and confidence measures to assess the similarity of rankings. In this work, we carry out a sensitivity analysis of these similarity-based measures. We want to understand which datasets benefit more from such measures and which parameters have more influence in the accuracy of the model. Furthermore, we propose an alternative type of rules, the Pairwise Association Rules (PAR), which are defined as association rules with a set of pairwise preferences in the consequent. While PAR can be used both as descriptive and predictive models, they are essentially descriptive models. Experimental results show the potential of both approaches.



## 1. Introduction

Label ranking is a topic in the machine learning literature [1–3] that studies the problem of learning a mapping from instances to rankings over a finite number of predefined labels. One characteristic that clearly distinguishes Label Ranking problems from classification problems is the order relation between the labels. While a classifier aims at finding the true class on a given unclassified example, the label ranker will focus on the relative preferences between a set of labels/classes. These relations represent relevant information from a decision support perspective, with possible applications in various fields such as elections, dominance of certain species over the others, user preferences, etc.

Due to its intuitive representation, Association Rules [4] have become very popular in data mining and machine learning tasks (e.g. mining rankings [5], classification [6] or even Label Ranking [7,8]). In [7], association rules were adapted for the prediction of rankings, which are referred to as Label Ranking Association Rules (LRAR). A different approach, Rule-Based Label Ranking (RBLR) [8], adapts the Dominance-based Rough Set Approach (DRSA) [9] for predicting rankings in the Label Ranking task. Both LRAR and RBLR can be used for predictive or descriptive purposes.

LRAR are relations, like typical association rules, between an antecedent and a consequent ($A \rightarrow C$), defined by interest measures. The distinction lies in the fact that the consequent is a complete ranking. Because the degree of similarity between rankings can vary, it leads to several interesting challenges. For instance, how to treat rankings that are very similar but not exactly equal. To tackle this problem, similarity-based interest measures were defined to evaluate LRAR. Such measures can be applied to existing rule generation methods [7] (e.g. APRIORI [4]).

One important issue for the use of LRAR is the threshold that determines what should and should not be considered sufficiently similar. Here we present the results of sensitivity analysis study to show how LRAR behave in different scenarios, to understand the effect of this threshold better. Whether there is a rule of thumb or this threshold is data-specific is the type of questions we investi-

* Corresponding author at: INESC TEC, Porto, Portugal.
*E-mail addresses:* claudio.r.sa@inesctec.pt, claudio84@gmail.com, c.f.pinho.rebelo.de.sa@liacs.leidenuniv.nl (C.R. de Sá), pja@di.uminho.pt (P. Azevedo), csoares@fe.up.pt (C. Soares), amjorge@fc.up.pt (A.M. Jorge), a.j.knobbe@liacs.leidenuniv.nl (A. Knobbe).

http://dx.doi.org/10.1016/j.inffus.2017.07.001
1566-2535/

gate here. Additionally we also want to understand which parameters have more influence in the predictive accuracy of the method.

Another important issue is related to the large number of distinct rankings. Despite the existence of many competitive approaches in Label Ranking, Decision trees [2,10], *k*-Nearest Neighbor [2,11] or LRAR [7], problems with a large number of distinct rankings can be hard to model. One real-world example with a relatively large number of rankings, is the sushi dataset [12]. This dataset compares demographics of 5000 Japanese citizens with their preferred sushi types. With only 10 labels, it has more than 4900 distinct rankings. Even though it has been known in the preference learning community for a while, no results with high predictive accuracy have been published, to the best of our knowledge. This might be due to noise in the data or simply because of inconsistency in the ratings provided by the people interviewed [13]. Cases like this have motivated the appearance of new approaches, e.g. to mine ranking data [5], where association rules are used to find patterns within rankings.

We propose a method which combines the two approaches mentioned above [5,7], to extract interesting information from datasets even when the number of different rankings is very high. We define Pairwise Association Rules (PAR) as association rules with one or more pairwise comparisons in the consequent. In this work, we present an approach to identify PAR and analyze the findings in two real world datasets.

By decomposing rankings into the unitary preference relation i.e. *pairwise comparisons*, we can look for sub-ranking patterns, which are expected to be more frequent.

LRAR and PAR can be regarded as a specialization of general association rules that are obtained from data containing preferences, which we refer to as *Preference Rules*. These two approaches are complementary in the sense that they can give different insights from multi-target relations that can be found in preference data [14]. We use LRAR and PAR in this work as predictive and descriptive models, respectively.

The paper is organized as follows: Sections 2 and 3 introduce the task of association rule mining and the Label Ranking problem, respectively; Section 4 describes the Label Ranking Association Rules and Section 5 the Pairwise Association Rules proposed here; Section 6 presents the experimental setup and discusses the results; finally, Section 7 concludes this paper.

## 2. Association rule mining

An association rule (AR) is an implication: $A \rightarrow C$ where $A \bigcap C = \varnothing$ and $A, C \subseteq desc(\mathbb{X})$, where $desc(\mathbb{X})$ is the set of descriptors of instances in the instance space $\mathbb{X}$, typically pairs $\langle attribute, value \rangle$. The training data is represented as $D = \{\langle x_i \rangle\}$, $i = 1, \ldots, n$, where $x_i$ is a vector containing the values $x_i^j$, $j = 1, \ldots, m$ of $m$ independent variables, $\mathcal{A}$, describing instance $i$. We also denote $desc(x_i)$ as the set of descriptors of instance $x_i$.

### 2.1. Interest measures

There are many interest measures to evaluate association rules [15], but typically they are characterized by *support* and *confidence*. Here, we summarize some of the most common, assuming a rule $A \rightarrow C$ in $D$.

*Support.* Percentage of the instances in $D$ that contain $A$ and $C$:

$$sup(A \rightarrow C) = \frac{\#\{x_i | A \cup C \subseteq desc(x_i), x_i \in D\}}{n}$$

*Confidence.* Percentage of instances that contain $C$ from the set of instances that contain $A$:

$$conf(A \rightarrow C) = \frac{sup(A \rightarrow C)}{sup(A)}$$

*Coverage.* Proportion of examples in $D$ that contain the antecedent of a rule: *coverage* [16]:

$$coverage(A \rightarrow C) = sup(A)$$

We say that a rule $A \rightarrow C$ covers an instance $x$, if $A \subseteq desc\ (x)$.

*Lift.* Measures the independence of the consequent, $C$, relative to the antecedent, $A$:

$$lift(A \rightarrow C) = \frac{sup(A \rightarrow C)}{sup(A) \cdot sup(C)}$$

*Lift* values vary from 0 to $+\infty$. If $A$ is independent from $C$ then $lift(A \rightarrow C) \sim 1$.

### 2.2. Methods

The original method for induction of AR is the APRIORI algorithm, proposed in 1994 [4]. APRIORI identifies all AR that have support and confidence higher than a given minimal support threshold (*minsup*) and a minimal confidence threshold (*minconf*), respectively. Thus, the model generated is a set of AR, $\mathcal{R}$, of the form $A \rightarrow C$, where $A, C \subseteq desc(\mathbb{X})$, and $sup(A \rightarrow C) \geq minsup$ and $conf(A \rightarrow C) \geq minconf$. For a more detailed description see [4].

Despite the usefulness and simplicity of APRIORI, it runs a time consuming candidate generation process and needs substantial time and memory space, proportional to the number of possible combinations of the descriptors. Additionally it needs multiple scans of the data and typically generates a very large number of rules. Because of this, many alternative methods were previously proposed, such as hashing [17], dynamic itemset counting [18], parallel and distributed mining [19] and mining integrated into relational database systems [20].

A major breakthrough in itemset mining has been brought by the algorithm FP-Growth (Frequent pattern growth method) [21], which starts by efficiently projecting the original data base into a compact tree data structure (FP-tree). From the FP-tree, itemset support can be calculated without revisiting the original dataset, which also avoids the generation of candidate itemsets. With respect to APRIORI there is an enormous reduction both on computational time and space necessary. FP-growth approach is also able to efficiently find long itemsets.

### 2.3. Pruning

AR algorithms typically generate a large number of rules (possibly tens of thousands), some of which represent only small variations from others. This is known as the rule explosion problem [22] which should be dealt with by pruning mechanisms. Many rules must be discarded for computational and simplicity reasons.

Pruning methods are usually employed to reduce the amount of rules without reducing the quality of the model. For example, an AR algorithm might find rules for which the confidence is only marginally improved by adding further conditions to their antecedent. Another example is when the consequent $C$ of a rule $A \rightarrow C$ has the same distribution independently of the antecedent $A$. In these cases, we should not consider these rules as meaningful.

*Improvement.* A common pruning method is based on the improvement that a refined rule yields in comparison to the original one [22]. The *improvement* of a rule is defined as the smallest difference between the confidence of a rule and the confidence of all sub-rules sharing the same consequent:

$$imp(A \rightarrow C) = min(\forall A' \subset A, conf(A \rightarrow C) - conf(A' \rightarrow C))$$