Journal of Machine Learning Research 14 (2013) 3153-3157 Submitted 3/11; Revised 4/13; Published 10/13

QuantMiner for Mining Quantitative Association Rules

Ansaf Salleb-Aouissi ANSAF@CCLS.COLUMBIA.EDU
Center for Computational Learning Systems (CCLS)
Columbia University
475 Riverside Drive, New York, NY 10115, USA

Christel Vrain CHRISTEL.VRAIN@UNIV-ORLEANS.FR
Cyril Nortet CYRIL.NORTET@UNIV-ORLEANS.FR
Laboratoire d'Informatique Fondamentale d'Orléans (LIFO)
Université d'Orléans
BP 6759, 45067 Orléans Cedex 2, France

Xiangrong Kong SHARONXKONG@GMAIL.COM
Vivek Rathod VIVEKMRATHOD@GMAIL.COM
Center for Computational Learning Systems (CCLS)
Columbia University
475 Riverside Drive, New York, NY 10115, USA

Daniel Cassard D.CASSARD@BRGM.FR
French Geological Survey (BRGM)
3, avenue Claude Guillemin
BP 6009, Orléans Cedex 2, France

Editor: Mikio Braun

Abstract

In this paper, we propose QUANTMINER, a mining quantitative association rules system. This system is based on a genetic algorithm that dynamically discovers "good" intervals in association rules by optimizing both the support and the confidence. The experiments on real and artificial databases have shown the usefulness of QUANTMINER as an interactive, exploratory data mining tool.

Keywords: association rules, numerical and categorical attributes, unsupervised discretization, genetic algorithm, simulated annealing

1. Introduction

In this paper, we propose a software for mining quantitative and categorical rules, that implements the work proposed in Salleb-Aouissi et al. (2007). Given a set of categorical and quantitative attributes and a data set, the aim is to find rules built on these attributes that optimize given criteria. Expressions occurring in the rules are either $A = v$ for a categorical attribute, or $A \in [l, u]$ for a quantitative one. The main difficulty is to find the best bounds l and u of the intervals. Mining quantitative association rules cannot be considered as a direct extension of mining categorical rules. While this task has received less attention than mining Boolean association rules (Agrawal et al., 1993), it remains a very important one from the point of view of applications. Several approaches have been designed for this task. For instance, a preprocessing step, discretizing (also called binning) numeric attributes into intervals is performed before the mining task in Srikant and Agrawal (1996). Some approaches (e.g., Aumann and Lindell, 1999) restrict learning to special kind of rules: the right-hand side of a rule expresses the distribution (e.g., mean, variance) of numeric attributes, the left-hand side is composed either of a set of categorical attributes or of a *single* discretized numeric attribute. Optimization-based approaches handle numeric attributes during the mining process. The first one proposed in Fukuda et al. (1996) introduces a new optimization criterion, called the *gain*, taking into account both the support and the confidence of a rule. Extensions have been proposed in Brin et al. (2003) but the forms of the rules remain restricted to one or two numeric attributes. A genetic algorithm is also proposed in Mata et al. (2002) to optimize the support of itemsets defined on uninstantiated intervals of numeric attributes. This approach is limited to numeric attributes and optimizes only the support before mining association rules. QuantMiner handles both categorical and numerical attributes and optimizes the intervals of numeric attributes during the process of mining association rules. It is based on a genetic algorithm, the fitness function aims at maximizing the gain of an association rule while penalizing the attributes with large intervals. Recent work (e.g., Alcalá-Fdez et al., 2010) show the interest of evolutionary algorithms for such a task.

2. QuantMiner

In the following, an item is either an expression $A = v$, where A is a categorical (also called qualitative) attribute and v is a value from its domain, or an expression $A \in [l, u]$ where A is a numerical (also called quantitative) attribute. QuantMiner optimizes a set of rule patterns produced from a user-specified rule template.

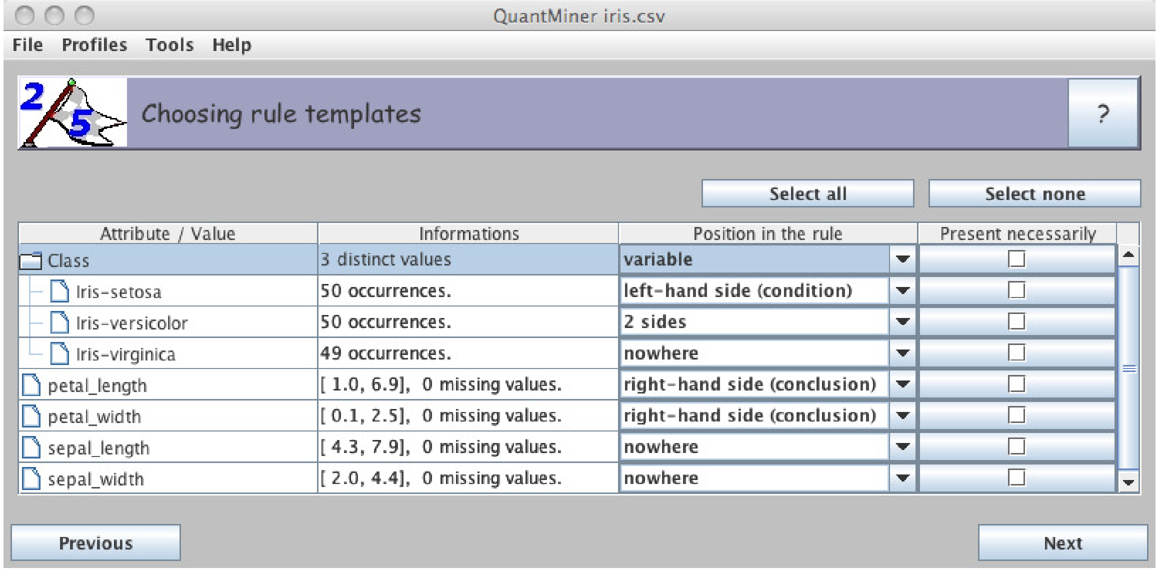
Rule templates: A *rule template* is a preset format of a quantitative association rule used as a starting point for the quantitative mining process. It is defined by the set of attributes occurring in the left hand side and the right hand side or both sides of the rule. Categorical attributes may or may not have specific values. Furthermore, an attribute may be mandatory or optional, thus allowing to generate rules of different lengths from the same rule template. Given a rule template, first for each unspecified categorical attribute in the template, the frequent values are computed. Then, a set of *rule patterns* verifying the specifications (position of the attributes, mandatory/optional presence of the attributes, values for categorical attributes either provided, or computed as frequent) are built. For each rule pattern, the algorithm looks for the "best" intervals for the numeric attributes occurring in that template, relying on a genetic algorithm.

Example 1 *Consider the Iris data set from the UCI machine learning repository.*[1] *An example of rule template and a specific example of rule are given in Figure 1 and in Figure 2 respectively.*

Population: An individual is a set of items of the form $attribute_i \in [l_i, u_i]$, where $attribute_i$ is the i^{th} numeric attribute in the rule template from the left to the right. The process for generating the population is described in Salleb-Aouissi et al. (2007).

Genetic operators: Mutation and *crossover* are both used in QuantMiner. For the crossover operator, for each attribute the interval is either inherited from one of the parents or formed by mixing the bounds of the two parents. For an individual, mutation increases or decreases the lower or upper bound of its intervals. Moving interval bounds is done so as to discard/involve no more than 10% of tuples already covered by the interval.

[1] The data set is available here: `http://archive.ics.uci.edu/ml/datasets/Iris`.



Choosing rule templates

Attribute / Value	Informations	Position in the rule	Present necessarily
Class	3 distinct values	variable	☐
Iris-setosa	50 occurrences.	left-hand side (condition)	☐
Iris-versicolor	50 occurrences.	2 sides	☐
Iris-virginica	49 occurrences.	nowhere	☐
petal_length	[1.0, 6.9], 0 missing values.	right-hand side (conclusion)	☐
petal_width	[0.1, 2.5], 0 missing values.	right-hand side (conclusion)	☐
sepal_length	[4.3, 7.9], 0 missing values.	nowhere	☐
sepal_width	[2.0, 4.4], 0 missing values.	nowhere	☐

Figure 1: An example of rule template exploring petal attributes for 2 categories of iris. *petal_length* and *petal_width* are chosen to be on the right side of the rules template.

Fitness function: It is based on the *Gain* proposed in Fukuda et al. (1996), defined in Equation 1:

$$Gain(A \Rightarrow B) = Supp(A \wedge B) - MinConf * Supp(A). \tag{1}$$

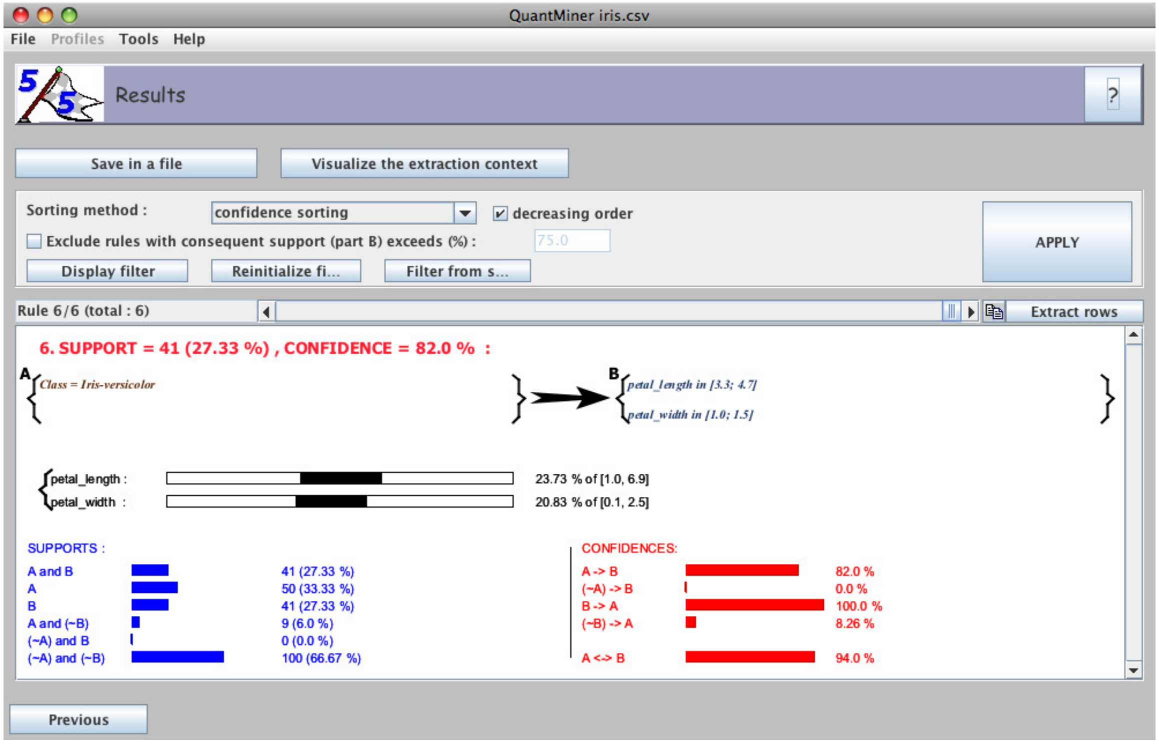
Let $\mathcal{A}_{num}$ the set of numerical attributes present in the rule $A \Rightarrow B$. Let I_a denote the interval of the attribute $a \in \mathcal{A}_{num}$. In the following, $size(a)$ denotes the length of the smallest interval which contains all the data for the attribute a and $size(I_a)$ denotes the length of the interval I_a. If $Gain(A \Rightarrow B)$ is negative, then the fitness of the rule is set to the gain. If it is positive (the confidence of the rule exceeds the minimum confidence threshold), the proportions of the intervals (defined as the ratios between the sizes and the domains) is taken into account, so as to favor those with small sizes as shown in Equation 2. Moreover, rules with low supports are penalized by decreasing drastically their fitness values.

$$Fitness(A \Rightarrow B) = Gain(A \Rightarrow B) \times \prod_{a \in \mathcal{A}_{num}} \left(1 - \frac{size(I_a)}{size(a)}\right)^2. \tag{2}$$

3. Implementation

We developed QUANTMINER in JAVA as a 5-step GUI wizard allowing an interactive mining process.[2] After opening a data set, the user can choose attributes, a rule template, the optimization technique and set its parameters, launch the process, and finally display the rules with various sorting options: support, confidence or rule length. The user can save the mining-context, go back to

[2] The software is available at `http://quantminer.github.com/QuantMiner/`.



Sorting method : confidence sorting ☑ decreasing order

□ Exclude rules with consequent support (part B) exceeds (%) : 75.0

Rule 6/6 (total : 6)

6. SUPPORT = 41 (27.33 %) , CONFIDENCE = 82.0 % :

A { *Class = Iris-versicolor* } → B { *petal_length in [3.3; 4.7]* *petal_width in [1.0; 1.5]* }

petal_length : 23.73 % of [1.0, 6.9]
petal_width : 20.83 % of [0.1, 2.5]

SUPPORTS :	
A and B	41 (27.33 %)
A	50 (33.33 %)
B	41 (27.33 %)
A and (~B)	9 (6.0 %)
(~A) and B	0 (0.0 %)
(~A) and (~B)	100 (66.67 %)

CONFIDENCES:	
A -> B	82.0 %
(~A) -> B	0.0 %
B -> A	100.0 %
(~B) -> A	8.26 %
A <-> B	94.0 %

Figure 2: Example of rule visualization in QuantMiner. The top part shows filtering criteria that facilitate exploring the rules. The bottom part shows a specific rule followed by the proportion of each interval in its corresponding domain. More measures are given to assess the quality of the rule, for example, $confidence(\neg A \Rightarrow B)$.

previous steps, change the method, parameters, templates and restart the learning. Note that simulated annealing is implemented in QuantMiner as an alternative optimization method. A tentative for mining rules with disjunctive intervals is also implemented. We hope this functionality will be further investigated.

Acknowledgments

This project has been supported by the BRGM-French Geological Survey, LIFO-Université d'Orléans and CCLS-Columbia University. Many thanks to everyone who contributed to QuantMiner with support, ideas, data and feedback.

References

R. Agrawal, T. Imielinski, and A. N. Swami. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD*, pages 207–216, 1993.

J. Alcalá-Fdez, N. Flugy Papè, A. Bonarini, and F. Herrera. Analysis of the effectiveness of the genetic algorithms based on extraction of association rules. *Fundam. Inform.*, 98(1):1–14, 2010.

Y. Aumann and Y. Lindell. A statistical theory for quantitative association rules. In *Knowledge Discovery and Data Mining*, pages 261–270, 1999.

S. Brin, R. Rastogi, and K. Shim. Mining optimized gain rules for numeric attributes. *IEEE Trans. Knowl. Data Eng.*, 15(2):324–338, 2003.

T. Fukuda, Y. Morimoto, S. Morishita, and T. Tokuyama. Mining optimized association rules for numeric attributes. In *Proc. of the fteenth ACM SIGACTSIGMOD -SIGART PODS'96*, pages 182–191. ACM Press, 1996.

J. Mata, J. L. Alvarez, and J. C. Riquelme. An evolutionary algorithm to discover numeric association rules. In *Proceedings of the ACM SAC'2002*, pages 590–594, 2002.

A. Salleb-Aouissi, C. Vrain, and C. Nortet. Quantminer: A genetic algorithm for mining quantitative association rules. In *IJCAI*, pages 1035–1040, 2007.

R. Srikant and R. Agrawal. Mining quantitative association rules in large relational tables. In *Proc. of the ACM SIGMOD*, pages 1–12, 1996.