

QuantMiner: A Genetic Algorithm for Mining Quantitative Association Rules

Ansaf Salleb-Aouissi* Christel Vrain** Cyril Nortet**

*CCLS, Columbia University 475 Riverside Drive New York NY 10115 USA
ansaf@ccls.columbia.edu

**LIFO, Université d'Orléans Rue Léonard de Vinci BP 6759 45067 Orléans cedex 02 France
{Christel.Vrain, Cyril.Nortet}@univ-orleans.fr

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 data mining tool.

Keywords: Association rules, quantitative (numeric) attributes, unsupervised discretization, genetic algorithm

1 Introduction

Mining association rules that aims at finding strong relations between attributes has already long been studied (see for instance, [Agrawal *et al.*, 1993]). In that context, an association rule is an expression $C_1 \Rightarrow C_2$, where C_1 and C_2 express conditions on the attributes describing the objects.

The strength of such a rule is usually evaluated by means of statistical measures, as for instance the *support* and the *confidence*, defined as follows:

- $\text{Support}(C)$, where C expresses conditions on attributes, is the number of tuples in the database that satisfy C .
- $\text{Support}(C_1 \Rightarrow C_2) = \text{Support}(C_1 \wedge C_2)$
- $\text{Confidence}(C_1 \Rightarrow C_2) = \text{Support}(C_1 \wedge C_2) / \text{Support}(C_1)$

Given two thresholds MinSupp and MinConf , a rule is *strong*, when its support is greater than MinSupp and its confidence greater than MinConf . Discovering strong rules is usually a two-step process. The first step consists in mining the set of all frequent itemsets w.r.t. MinSupp . The second step relies on this set to discover strong rules w.r.t. MinConf .

Mining association rules on both categorical and numeric attributes, also called *quantitative association rules*, has been less studied. Basically, this task involves several problems:

First, numeric attributes are usually defined on a wide range of different values. It is useless to work on all possible numeric values, as done for categorical values, because in most cases, a given numeric value will not appear frequently. A classical way to deal with numeric attributes is to divide their domains into intervals. This is called *discretization*. But, as already pointed out in [Srikant and Agrawal,

1996], choosing intervals for numeric attributes is quite sensitive to the support and the confidence measures: small intervals lead to low supports, thus for a rule $A \Rightarrow B$ increasing B leads to a higher confidence whereas increasing A can either increase or decrease the confidence. A tradeoff has to be found to discretize “correctly” numeric attributes with respect to MinSupp and MinConf . For instance $\text{Weight} \in [50, 70]$ can be frequent, but the rule $\text{Weight} \in [55, 65] \rightarrow \text{Height} \in [1.60, 1.70]$ may have a higher confidence than the rule $\text{Weight} \in [50, 70] \rightarrow \text{Height} \in [1.60, 1.70]$. This shows that it is difficult to find a “good” discretization of numeric attributes, considering them individually. As a consequence, the classical two-step process for mining association rules, namely mining frequent itemsets and then extracting solid association rules is no longer appropriate, as well as the embedded traditional level-wise algorithm for mining frequent itemsets.

Therefore, mining quantitative association rules is not a simple extension of mining categorical association rules. Since the search space is infinite, our goal is to find a finite set of interesting solutions, close to the optimal solutions. This explains why we have chosen to solve this search problem with metaheuristics, mainly genetic algorithms [Holland, 1975].

In this paper, we consider that given a pattern for an association rule, finding “good” intervals for numeric attributes occurring in that pattern is an optimization problem and we propose QUANTMINER a system that has been designed in that framework. In that system, constraints on rules, namely the attributes appearing in the left-hand side and in the right-hand side, and possibly values for the categorical attributes, are specified by the user and for each possible rule template fulfilling these constraints, a genetic-based algorithm is then applied for learning “good” intervals optimizing a given quality criteria. Let us emphasize on the fact that intervals for numeric attributes are thus dynamically optimized during the mining process, and depend on all the numeric attributes occurring in the rule.

This paper is organized as follows: § 2 gives the state of the art for mining quantitative association rules. § 3 is devoted to QUANTMINER, the algorithm we propose for mining quantitative association rules. Experimental tests on real-life and synthetic datasets are described in § 4. We conclude in § 5 with a summary of our contribution and future directions.

2 Related Work

Discretization-based approach A classical method to deal with numeric attributes is to perform a preprocessing step, discretizing (also called binning) numeric attributes into intervals before the mining task (see for example [Lent *et al.*, 1997; Miller and Yang, 1997; Srikant and Agrawal, 1996; Wang *et al.*, 1998]). This can be achieved by either using domain knowledge expressed by concept hierarchies, or automatic discretization methods, such as equi-depth and equi-width binning. Nevertheless, this discretization is univariate (an attribute is discretized without taking into account the other attributes). To reduce the influence of pre-discretization, most work in this approach use either clustering or intervals merging. However, this work remain sensitive to outliers and does not reflect the distribution of data.

Distribution-based approach In a statistical approach proposed by Aumann and Lindell [Aumann and Lindell, 1999], the right-hand side of a rule can express the distribution of the values of numeric attributes such as the mean or variance, as for instance:

```
Sex=female  $\implies$  Height: mean = 168
 $\wedge$  Weight: mean = 68
```

Two kinds of rules are considered :

- A set of categorical attributes on the left-hand side and several distributions on the right-hand side,
- A *single* discretized numeric attribute on the left-hand side and a *single* distribution on the right-hand side.

Webb [Webb, 2001] has extended this framework to other measures such as standard deviation, minimum, count etc. Although quite interesting, the kinds of rules in these approaches have a very specific form and cannot handle several numeric attributes in the left-hand side of a rule.

Optimization-based approach In this approach, numeric attributes are *optimized* during the mining process. The term *optimization* was first used by Fukuda *et al.* in [Fukuda *et al.*, 1996b]. In their works, a very interesting optimization criterion, called the *Gain* is proposed as a trade-off between support and confidence. It is defined by :

$$\text{Gain}(A \Rightarrow B) = \text{Supp}(AB) - \text{MinConf} * \text{Supp}(A)$$

Nevertheless, the main limitation of this work is that rules contain at most two numeric attributes, since it is based on a technique inspired from image segmentation to find an image region that produces an *optimized* association rule. Rastogi and Shim in [Rastogi and Shim, 1999] have proposed a support-optimization method, generalizing the work proposed in [Fukuda *et al.*, 1996a; 1996b] to handle disjunctions on one numeric attribute. Their work has then been extended to a Gain-optimization approach by Brin *et al.* [Brin *et al.*, 2003]. Although these works allow disjunctions between intervals, the forms of the rules remain restricted to one or two numeric attributes.

A quite different algorithm, named GAR has been proposed by Mata *et al.* [Mata *et al.*, 2002]. The idea is to use a genetic algorithm in order to optimize the support of itemsets defined on uninstantiated intervals of numeric attributes. The fitness

function of each individual (a k-itemset) depends on its absolute support (coverage) and 3 factors: 2 penalization factors are used to avoid getting the whole domains of the attributes and to avoid overlapping between itemsets with respect to the support and one factor to favor specific itemsets with many attributes :

$$\text{Fitness} = \text{cov} - (\psi * \text{ampl}) - (\omega * \text{mark}) + (\mu * \text{nAtr})$$

The effect of these 3 factors can be adjusted with 3 parameters. The authors suggest then to use an apriori-like algorithm to mine association rules using these optimized itemsets. This approach has several drawbacks: it is limited to numeric attributes, only the support is optimized which does not guarantee to get rules with high confidences, and finally, intervals are static in the rules and thus do not care of the implication direction. This may miss a lot of interesting rules. More recently, an approach based on half-spaces has been suggested in [Rückert *et al.*, 2004]. In this work, the left-hand and the right-hand side of an association rule contain a linear inequation on attributes, such as: $x_1 > 20 \implies 0.5x_3 + 2.3x_6 \geq 100$. Although quite original and useful in many applications, the expressiveness of such rules is restricted, and this approach cannot handle categorical attributes.

3 QuantMiner

QuantMiner is a genetic-based algorithm for mining quantitative association rules. 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 quantitative attribute. QuantMiner works directly on a set of *rule templates*. A rule template is a preset format of a quantitative association rule, either chosen by the user or computed by the system. It is used as a starting point for the mining process¹. For each rule template, the algorithm looks for the best intervals for the numeric attributes occurring in that template, relying on a Genetic Algorithm.

More precisely, a rule template is defined by the set of attributes occurring in the left hand side and the right hand side of the rule. For each categorical attribute, the value is either defined by the user in the template, or computed by QuantMiner. In this case, let A_1, \dots, A_n be the non instantiated categorical attributes occurring in the template. QuantMiner computes the frequent itemsets built on A_1, \dots, A_n , *i.e.*, the set of instantiations v_1, \dots, v_n such that $A_1 = v_1 \wedge \dots \wedge A_n = v_n$ is frequent and generate a rule template for each such instantiation. This leads to as many templates as the number of instantiations. Note that this step is similar to *apriori* [Agrawal *et al.*, 1993].

• **Individual representation** 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.

• **Initial population** The initial population of individuals is generated as follows: in the first individuals, the intervals

¹Note that QUANTMINER can consider all possible association rules with no restrictions.

$[l_i, u_i]$ represent the whole domain of the i^{th} numeric attribute, and the following individuals encode intervals with decreasing amplitudes (length of intervals) until they reach a minimum support in the dataset. Once the amplitudes are fixed for an individual, the bounds l_i and u_i are chosen at random. This ensures to start with enough diversity in the initial population that model general and specific rules.

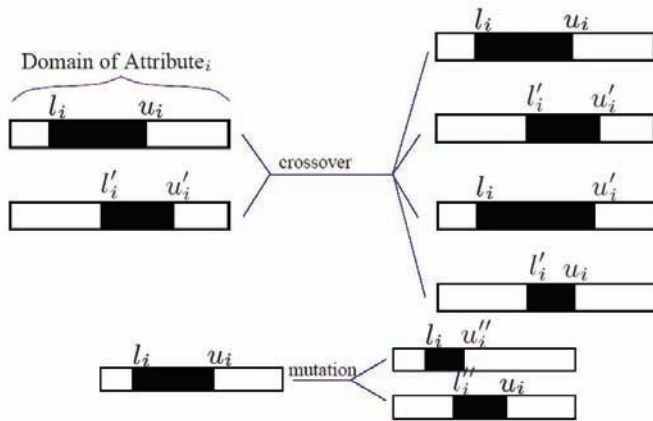


Figure 1: Crossover and mutation operators.

- **Genetic operators** *Mutation* and *crossover* (Figure 1) are both used in order to transform a generation of individuals into another one, improving its quality. The crossover operator consists in taking two individuals, called parents, at random and generating new individuals: for each attribute the interval is either inherited from one of the parents or formed by mixing the bounds of the two parents. Mutation works on a single individual and increases or decreases the lower or upper bound of its intervals respectively. Moving interval bounds is done so as to discard/involve no more than 10% of tuples already covered by the interval.

- **Fitness function** The fitness function used in QUANTMINER is based on the *Gain* measure proposed in [Fukuda *et al.*, 1996b]. If the gain is positive (the confidence of the rule exceeds the minimum confidence threshold), we take into account the proportions of the intervals (defined as the ratios between the amplitudes and the domains), so as to favor those with small amplitudes. Moreover, rules with low supports are penalized by decreasing drastically their fitness values by the number of tuples in the database.

- **Algorithm**

QUANTMINER (Algorithm 2) follows a prototypical genetic algorithm scheme. The inputs are the minimum support (MinSupp), the minimum confidence (MinConf), the population size (PopSize), the number of generations (GenNb), the fraction of population to be replaced by crossover (CR) and the mutation rate (MR).

We developed QUANTMINER² in JAVA as a 5-step GUI wizard allowing an interactive mining process. After opening a

²In collaboration with the French Geological Survey (BGRM).

```

Function Fitness ( $A \Rightarrow B$ )
TempFitness = Gain( $A \Rightarrow B$ )
if TempFitness  $\geq 0$  then
  foreach interval  $I$  in  $A \Rightarrow B$  do
    //favor small intervals
    TempFitness * = (1-Prop(I))2
    if Support( $A \Rightarrow B$ ) < MinSupp then
      //penalize low support rules
      TempFitness - = Nbtuples
return TempFitness

```

Algorithm 2: QUANTMINER

```

Input: A dataset composed of NbTuples, PopSize,
        GenNb, CR, MR, MinSupp, MinConf
Output: Quantitative association rules  $\mathcal{R}$ 
Select a set of attributes
Let  $\mathcal{R}_t$  a set of rule templates defined on these attributes
Compute the set of frequent itemsets on categorical
attributes in  $\mathcal{R}_t$ 
 $\mathcal{R} = \emptyset$ 
foreach  $r \in \mathcal{R}_t$  do
  Generate a random population POP of PopSize
  instantiated rules following the template  $r$ 
  i=1
  while  $i \leq GenNb$  do
    Form the next generation of population by
    mutation and crossover w.r.t. MR and CR.
    Keep PopSize rules in POP with the best Fitness
    values
    i++
   $\mathcal{R} = \mathcal{R} \cup Argmax_{R \in POP} Fitness(R)$ 
return  $\mathcal{R}$ 

```

dataset, the user can choose attributes, choose some rule templates specifying the position of items in the left-hand and/or right-hand side of the rule³, choose the optimization technique and set its parameters, launch the process, and finally display the rules with various sorting: support, confidence, rule-length. The user can save the mining-context, go back to previous steps, change the method, parameters, templates and restart the learning.

We have set the default parameters of the genetic algorithm to 250 individuals, 100 generations, 40% of mutations and 50% of crossovers. They have been chosen experimentally so as to produce approximately stable intervals from a run of the system to another one. To give an idea about execution time, with the default parameters, on a dataset of 2 500 tuples, QUANTMINER can process on average one rule template/second. Note that simulated annealing is implemented in QuantMiner as an alternative optimization method. Our experiments have shown that simulated annealing is 10 to 20 times faster than the genetic algorithm, but generally the optimized intervals are much less precise and also sometimes

³Items are $A = v$ or $A \in [l, u]$ where l, u are not fixed, v can be instantiated or not.

Species	Attr	Min	Max	Mean	StdDev	Rule
Setosa	PW	1	6	2.46	1.05	Species= Setosa $\Rightarrow \left\{ \begin{array}{l} \text{PW} \in [1, 6] \text{ SW} \in [31, 39] \\ \text{PL} \in [10, 19] \text{ SL} \in [46, 54] \end{array} \right\}$ 23% 70%
	SW	23	44	34.28	3.79	
	PL	10	19	14.62	1.74	
	SL	43	58	50.6	3.52	
Versicolor	PW	10	18	13.26	1.98	Species= Versicolor $\Rightarrow \left\{ \begin{array}{l} \text{PW} \in [10, 15] \text{ SW} \in [22, 30] \\ \text{PL} \in [35, 47] \text{ SL} \in [55, 66] \end{array} \right\}$ 21% 64%
	SW	20	34	27.7	3.14	
	PL	30	51	42.6	4.70	
	SL	49	70	59.39	5.16	
Virginica	PW	14	25	20.26	2.75	Species= Virginica $\Rightarrow \left\{ \begin{array}{l} \text{PW} \in [18, 25] \text{ SW} \in [27, 33] \\ \text{PL} \in [48, 60] \text{ SL} \in [58, 72] \end{array} \right\}$ 20% 60%
	SW	22	38	29.74	3.22	
	PL	45	69	55.52	5.52	
	SL	49	79	65.88	6.36	

Table 1: Statistical Distribution and some rules of the Iris dataset

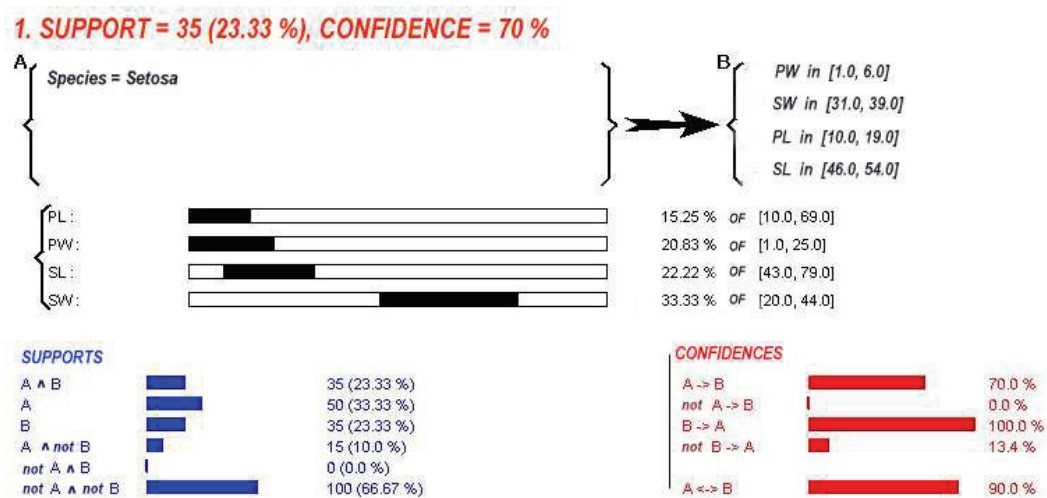


Figure 2: Example of rule as produced and displayed in QUANTMINER. The top part shows the rule $A \rightarrow B$ followed by the proportion of each interval appearing in the rule in its corresponding domain. Note that further support and confidence measures are given to assess the quality of the rule as for example $confidence(\neg A \rightarrow B)$.

cover the whole attribute domains.

4 Experiments

We begin this section with an experiment on a simple dataset to illustrate our algorithm, followed by an empirical study on a real-life and synthetic datasets to assess the usefulness and the scalability of QUANTMINER. Our experiments have been performed on a PC AMD Athlon XP2000+ with 256 MB main memory, running Windows XP.

4.1 Iris Dataset

The IRIS data set [Murphy and Aha, 1995] is composed of 150 samples of flowers from the iris species *setosa*, *versicolor*, and *virginica*. For each species, there are 50 observations described by the attributes Sepal Length (SL), Sepal Width (SW), Petal Length (PL), and Petal Width (PW) in mm.

For instance QUANTMINER has extracted, the *descriptive* association rules presented in Table 1. The intervals produced are coherent with the normal distribution of the Iris dataset. The rule 1, as displayed in QuantMiner, is presented in Figure 2.

4.2 Real-life Dataset

We have also experimented QUANTMINER on a real-life medical database donated by the Stulong⁴ project to discover

⁴The study STULONG was realized at the 2nd Department of Medicine, 1st Faculty of Medicine of Charles University and Charles University Hospital, U nemocnice 2, Prague 2 (head. Prof. M. Aschermann, MD, SDr, FESC), under the supervision of Prof. F. Boudik, MD, ScD, with collaboration of M. Tomeckova, MD, PhD and Ass. Prof. J. Bultas, MD, PhD. The data were transferred to the electronic form by the European Center of Medical Informatics,

descriptive rules of the atherosclerosis disease. Stulong is a dataset concerning a 20 years lasting study of the risk factors of the atherosclerosis in a population of 1 419 middle aged men. The patients have been classified into three groups : normal, risk and pathological. The database we have used has about 27 categorical attributes and 17 numerical attributes. Each patient is described by his physical activities, drinking of alcohol, smoking, physical and biochemical examination, etc. In our experiments, we have focused on rule patterns such that the antecedent of the rule describes a population of patients, and the consequent gives a description of this population and *vice-versa*. Here are 3 examples of discovered rules where ALCO_CONS is patient's alcohol daily consumption expressed in liter, TOBA_CONSO gives the number of packs of cigarettes consumed daily, TOBA_DURA is the smoking duration in years, BMI stands for Body Mass Index, the weight in kilograms divided by the square of the height in meters (kg/m²). According to the World Health Organization the BMI falls into one of these categories: below 18.5 Underweight, 20-25 Normal, 25-30 Overweight, 30 and above Obese:

$$\text{DEATH?} = \begin{cases} \text{YES} \\ \text{NO} \end{cases} \Rightarrow \left\{ \begin{array}{l} \text{ALCO_CONS} \in [1.0, 1.28] \\ \& \text{TOBA_CONSO} \in [0.5, 1.25] \\ \& \text{TOBA_DURA} \in [15, 20] \end{array} \right\} \begin{matrix} 18\% \\ 68\% \end{matrix}$$

$$\left\{ \begin{array}{l} \text{ALCO_CONS} \in [1.1, 1.2] \\ \& \text{BMI} \in [23.18, 26.15] \\ \& \text{TOBA_CONSO} \in [0.0, 0.5] \end{array} \right\} \Rightarrow \begin{matrix} \text{DEATH?} = \\ \text{NO} \end{matrix} \begin{matrix} 9.5\% \\ 90\% \end{matrix}$$

$$\left\{ \begin{array}{l} \text{ALCO_CONS} \in [1.07, 1.18] \\ \& \text{BMI} \in [24.73, 29.03] \\ \& \text{TOBA_CONSO} \in [0.85, 1.25] \\ \& \text{TOBA_DURA} \in [20, 20] \end{array} \right\} \Rightarrow \begin{matrix} \text{GROUP} = \text{R} \\ \end{matrix} \begin{matrix} 8\% \\ 86\% \end{matrix}$$

Our results have shown that the atherosclerosis disease is strongly related to the overweight and the obesity commonly assessed by BMI. The intervals discovered by QUANTMINER (see for example rule 2 and 3) are quite coherent with those given by the World Health Organization. As shown in these 3 rules, our experiments pointed out the big influence of tobacco consumption on the death of patients. Typically, the patients who died have smoked a lot for a long time. Some other risk factors emerged, such as a high cholesterol rate and blood pressure and the presence of skin fold above some muscles. The intervals discovered by QUANTMINER were considered as relevant and interesting when presented to a physician of STULONG. Figure 3 shows the convergence of the process for a given rule template of 15 numeric attributes by considering the maximum fitness according to the number of generations. It shows also the execution time w.r.t. the number of numeric attributes.

Statistics and Epidemiology of Charles University and Academy of Sciences (head. Prof. RNDr. J. Zvarova, DrSc). The data resource is on the web pages <http://euromise.vse.cz/STULONG>. At present time the data analysis is supported by the grant of the Ministry of Education CR Nr LN 00B 107.'

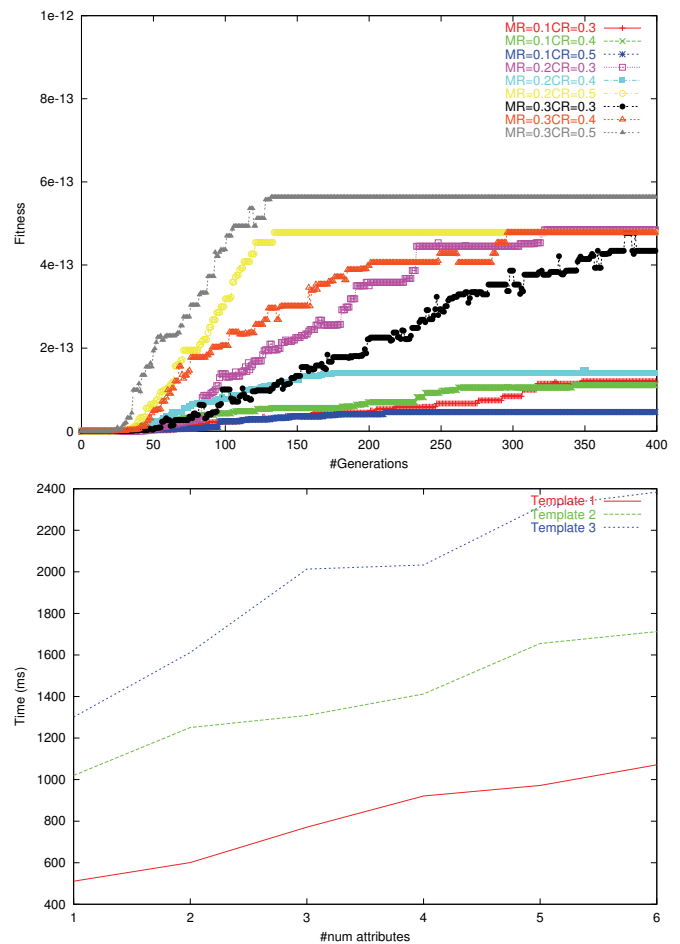


Figure 3: Maximum fitness w.r.t to the number of generations for different values of CR and MR (left). Execution time w.r.t to the number of numeric attributes (right)

4.3 Synthetic Datasets

We have tested the scalability and the sensitivity of QUANTMINER to the Minimum confidence threshold. For this purpose we generated several synthetic datasets⁵ for a person database in which each person is described by nine attributes: salary, commission, age, education, car, zipcode, housevalue, yearsowned and loan. Attributes education and zipcode are categorical and the rest are numeric. We have projected our experiments in Figure 4 where the optimization time of a template is given according to the number of tuples in the dataset and to MinConf respectively. Here, k indicates the number of numeric attributes in the rule template. For a given k, the execution time reported here is an average time of a number of rules having k numeric attributes. This number varies between 10 and 150 rules according to the templates. We have set the parameters of QUANTMINER to the following: GenNb=100, PopSize=250, CR=0.5, MR=0.4, MinSupp=0.1 and MinConf=0.6.

⁵<http://www.almaden.ibm.com/software/quest/Resources/>

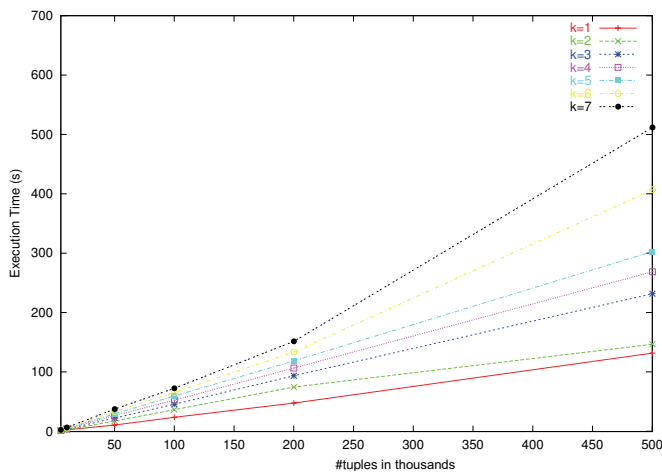


Figure 4: Execution time according to the number of tuples

5 Conclusion

The problem of mining association rules in large databases involving both numerical and categorical attributes is not straightforward. This is due to the very great number of possibilities for discretizing numeric attributes into intervals whereas in categorical association rules, all possible rules are considered. Actually, mining quantitative association rules is an *optimization* problem rather than being a simple *discretization* one. In this paper, we have proposed QUANTMINER, a genetic-based algorithm for mining such rules in databases. Unlike GAR [Mata *et al.*, 2002], QUANTMINER gets high-confidence rules and do take care of the position of the intervals w.r.t the implication direction. Our algorithm starts with a set of rule templates and then looks dynamically for the “best” intervals for the numeric attributes present in these templates. An optimization criterion based on both support and confidence is used to keep only high quality and interesting rules. In QUANTMINER, the user is highly solicited in order to guide the mining process thus avoiding the discovery of hundreds of rules, as it is usually the case in association rule mining. Our experiments have shown the usefulness of QuantMiner as an interactive data mining system and the pertinence of the intervals discovered by QUANTMINER in real datasets. In future works, we plan to extend QUANTMINER to handle disjunction between intervals.

Acknowledgments

We thank the French Geology Survey (BRGM) for supporting this work and Teddy Turmeaux for his help in the experiments. This work has been partly supported by a research contract from Consolidated Edison.

References

[Agrawal *et al.*, 1993] 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.

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

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

[Fukuda *et al.*, 1996a] T. Fukuda, Y. Morimoto, S. Morishita, and T. Tokuyama. Data mining using two-dimensional optimized association rules: Scheme, algorithms and visualization. In *Proc. of the Int’l Conf. ACM SIGMOD*, pages 12–23, 1996.

[Fukuda *et al.*, 1996b] T. Fukuda, Y. Morimoto, S. Morishita, and T. Tokuyama. Mining optimized association rules for numeric attributes. In *Proc. of the fteenth ACM SIGACTS SIGMOD -SIGART PODS’96*, pages 182–191. ACM Press, 1996.

[Holland, 1975] J. H. Holland. *Adaptation in Natural and Artificial Systems*. MIT Press, 1975.

[Lent *et al.*, 1997] B. Lent, A. N. Swami, and J. Widom. Clustering Association Rules. In *Proceedings of the 13th ICDE*, pages 220–231, 1997.

[Mata *et al.*, 2002] J. Mata, J. L. Alvarez, and J. C. Riquelme. An evolutionary algorithm to discover numeric association rules. In *Proceedings of the ACM symposium on Applied computing SAC’2002*, pages 590–594, 2002.

[Miller and Yang, 1997] R. J. Miller and Y. Yang. Association rules over interval data. *SIGMOD Rec.*, 26(2):452–461, 1997.

[Murphy and Aha, 1995] P. M. Murphy and D. W. Aha. *UCI Repository of Machine Learning Databases*. Technical report, University of California, Irvine, 1995. [Available by anonymous ftp from `ics.uci.edu`].

[Rastogi and Shim, 1999] R. Rastogi and K. Shim. Mining optimized support rules for numeric attributes. In *Proc. of the 15th ICDE*, pages 206–215, 1999.

[Rückert *et al.*, 2004] U. Rückert, L. Richter, and S. Kramer. Quantitative association rules based on half-spaces: An optimization approach. In *Proceedings of the Fourth IEEE International Conference on Data Mining (ICDM’04)*, pages 507–510, 2004.

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

[Wang *et al.*, 1998] K. Wang, S. H. W. Tay, and B. Liu. Interestingness-based interval merger for numeric association rules. In *Proc. 4th Int. Conf. Knowledge Discovery and Data Mining, KDD*, pages 121–128. AAAI Press, 27–31 1998.

[Webb, 2001] G. I. Webb. Discovering associations with numeric variables. In *Proceedings of the seventh ACM SIGKDD*, pages 383–388. ACM Press, 2001.