GENERATING ASSOCIATION RULES IN DATABASES USING DISCRETIZATION BASED ON CLUSTERING

Daniela JOITA, Senior lect., PhD

„TITU MAIORESCU" University, Bucharest- Faculty of Computer Science

daniela.joita@utm.ro

ABSTRACT

In this paper we analyse of the performance of two discretization methods based on clustering on some classification algorithms. We also use the same methods as the pre-processing step for generating association rules using the Apriori Algorithm and we test them against other discretization methods.

Keywords: discretization, clustering, k-means, Apriori Algorithm, HAC

1. INTRODUCTION

In [10] we presented an unsupervised static discretization method based on the k-means clustering method, different from the classic one by the fact that the values of the attribute need no sorting before the discretization. We proposed a technique of choosing the initial centers of the clusters designed specifically for the clustering of a one–dimensional vector of real valued data, to be used in the process of discretization of a single attribute, which avoids the *O(n log n)* time requirement for sorting the data points. In [11] we presented the discretization techniques based on the hierarchical clustering algorithms.

In the first part of this paper we compare four classification algorithms: Naive Bayes, C4.5, ID3 and C-RT for two well-known databases: weather (used for the first time by Quinlan (1986) [15]) and wine_quality. We have chosen the Naive Bayes algorithm because it requires all the input attributes to be discrete. Even that the others can have both continuous and discrete input attributes and therefore no discretization is needed, we wanted to see whether, for the databases that we analyse, the performance of these classification algorithms is higher than the performance of the Naive Bayes algorithm.

In the second part we compare the two discretization methods mentioned above with the classical ones (equal-width interval discretization and equal-frequency interval discretization) as the pre-processing step for the generation of association rules in databases. The algorithm used for the affinity analysis is the Apriori Algorithm.

The data mining software we used in all the applications described in this paper is Tanagra [16] created by Rakotomalala, for academic and research purposes. It proposes several data mining methods from exploratory data analysis, statistical learning, machine learning and databases area.

2. DISCRETIZATION BASED ON CLUSTERING VS. EQUAL-WIDTH AND EQUAL-FREQUENCY INTERVAL DISCRETIZATION

2.1 Comparison regarding the use of discretizations for Classification Algorithms

Many classification algorithms often require that the attributes of the data sets are discrete. If this is not the case, one has either to choose a different algorithm or to find a way to discretize the continuous data attributes prior to applying the desired algorithm. Such an algorithm is the Naive Bayes algorithm. The algorithm is based on the Bayes's Theorem which relates the conditional and marginal probabilities of two events. The Bayesian classifer assumes that all the input attributes are independent. For a given example ("the observed event"), it computes the probabilities that the target attribute takes each one of its possible value. The class to which the example belongs is the one with the higher probability. Other algorithms that allow both continuous and discrete input attributes are ID3 (the decision tree induction algorithm created by Quinlan that uses information gain criterion to find the best attribute for division of a tree node), C4.5 (an improved version of ID3) and C-RT (classification and regression trees system for learning decision trees (Breiman et al. (1984)[2]). For a detailed presentation of the classification algorithms see [17].

For the comparison of the discretization methods mentioned above we run the Naive Bayes algorithm for the datasets: weather and wine_quality, from the datasets downloaded from Tanagra's website [19]. The weather dataset has 5 attributes: Outlook, Temperature, Humidity, Windy and Class and 14 examples. It has to be decided whether the weather conditions are favourable for playing a baseball game. The attribute Class has two values: Play and Don't Play, Outlook and Windy are also discrete attributes and the others are continuous. The

wine_quality dataset contains 34 wines classified into 3 groups "good", "medium" and "bad". The attributes are the weather descriptors: sum of daily temperature(^{o}C), sun(h), heat (days) and rain (mm) and quality (the target attribute).

For both datasets we discretized first the attributes that are not discrete and then we applied the classifier on the training set. We discretized each attribute independent of the other attributes. (In the case of discretization based on clustering we take each attribute and distribute the data points in clusters based only on the values of the attribute that is discretized. We repeat the clustering for each input attribute.) The results are summarized in the following tables (Table1 and Table 2).

Database	weather						
Classifier	Naive Bayes classifier				C 4.5	C-RT	ID3
	Equal-frequency interval discretization	Equal-width interval discretization	Discretization with k-means	Discretization with HAC			
error rate	0.0714	0.1429	0.1429	0.2143	0.0714	0.0000	0.0000

Table 1. Performance of classification algorithms for weather database

Database	wine_quality						
Classifier	Naive Bayes classifier				C 4.5	C-RT	ID3
	Equal-frequency interval discretization	Equal-width interval discretization	Discretization with k-means	Discretization with HAC			
error rate	0.1765	0.2647	0.2941	0.2059	0.0588	0.0588	0.0588

Table 2. Performance of classification algorithms for wine_quality database

We observed that the classification algorithms C4.5, C-RT and ID3 have better performance on these two databases than Naive Bayes, no matter the discretization technique used. Also, regarding the four discretization methods, the equal- frequency interval discretization works better for these two databases.

2.2 Comparison regarding the use of discretizations for Apriori Algorithm

Finding association rules in databases is very important in the process of finding patterns in databases. The most commonly used algorithm for finding association rules is the Apriori Algorithm developed by Agrawal and Srikant in [1]. Even that it was designed to operate on databases involved in basket market analysis and therefore on datasets with transactions and items that are bought at transactions, it works on any dataset with discrete attributes. It finds rules of the type "*If X = x then Y= y*" where *X* and *Y* are attributes and *x* and *y* possible values of *X* respectively *Y*.

For the comparison of the four discretization methods as the pre-processing step for the generation of the association rules we run the Apriori Algorithm on three databases: weather, wine_quality mentioned above and winequality_white. The last one is from the UCI machine learning repository [20] and is related to white vinho verde wine samples, from the north of Portugal. The database is large (it has 4899 examples) and it was used by Cortez et al. to model wine quality based on physicochemical tests in [4]. The database has 11 input attributes : fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol and a target attribute variable (based on sensory data): quality (score between 0 and 10). All attributes are continuous.

As we did before, for all datasets we discretized first each continuous attribute independent of the other attributes. For the winequality_white database we discretized the target attribute using only the equal-width interval discretization into three intervals. Instead of having a score between 0 and 10 we have the quality of wine described by one of the possible values classified "good" (score $\geq$7), "medium" (5 $\leq$ score < 7) and "bad" (score < 5). After that we performed the Apriori Algorithm with different values for minsup (minimum support) and minconf (minimum confidence). The results are summarized in the following tables (Tables 3-5).

Database: weather minsup = 0.3 minconf = 0.75	Equal-frequency interval discretization	Equal-width interval discretization	Discretization with k-means	Discretization with HAC
number of rules	5	2	15	7
number of rules with confidence 1	1	1	5	3

Table 3. Number of association rules generated using the Apriori Algorithm for the database weather

We observed that for the database weather the rule found by all discretizations is the rule "*Outlook = overcast → Class = Play"*. Also, there are interesting rules found by discretization with k-means but not discovered by the others, like: "*Outlook = rain → Temp < 80"* or "*Humidity < 85 and Windy = no → Class = Play"*. The discretization with HAC generated a stronger rule than the last one: "*75 ≤ Humidity < 85 and Windy = no → Class = Play"*.

Database: wine_quality minsup = 0.3 minconf = 0.75	Equal-frequency interval discretization	Equal-width interval discretization	Discretization with k-means	Discretization with HAC
number of rules	2	10	49	12
number of rules with confidence 1	0	4	20	5

Table 4. Number of association rules generated using the Apriori Algorithm for the database wine_quality

We observed that for the database wine_quality the discretization with k-means generates much more rules with confidence 1 than the others. Some of the rules are: "*Quality = good → Temperature ≥ 3182*", "*Quality = good → Rain < 414*", "*Rain < 414 and Temperature < 3182 → Heat < 23*", "*Quality = bad →Sun < 1259*".

Database: winequality_white			
minsup = 0.3 minconf = 0.75			
	Equal-frequency interval discretization	Equal-width interval discretization	Discretization with k-means
number of rules	45	323	267
number of rules with confidence ≥ 0.9	4	0	54
minsup = 0.4 minconf = 0.75			
	Equal-frequency interval discretization	Equal-width interval discretization	Discretization with k-means
number of rules	11	323	47
number of rules with confidence ≥ 0.9	0	0	14
minsup = 0.5 minconf = 0.75			
	Equal-frequency interval discretization	Equal-width interval discretization	Discretization with k-means
number of rules	0	287	7
number of rules with confidence ≥ 0.9	0	0	3

Table 5. Number of association rules generated using the Apriori Algorithm for the database winequality_white

We observed that for the database winequality_white there were generated no rules with confidence 1 using any of the four discretization methods. We also observed that equal-width interval discretization generates a lot more rules than discretization with k-means, but no rule with confidence $\geq$ 0.9. It seems that the discretization with k-means generates more reasonable rules and with higher probability. A rule generated for minsup = 0.5 and confidence $\geq$ 0.9 is: "*density < 0.99434 → residual sugar < 7.6 and chlorides < 0.104*".

3. CONCLUSION

We found that, on the databases we tested, the discretization methods based on clustering are more efficient in finding the association rules with higher confidence than the classical ones (equal-width interval discretization and equal-frequency interval discretization). Future work may include testing the technique on more databases and finding characteristics of databases for which the technique works best.

BIBLIOGRAPHY

1. *R. Agrawal, R Srikant - Fast algorithms for mining association rules, Proc. 20th Int. Conf. Very Large Data Bases, VLDB, 1994*
2. *Breiman, L., J. H. Friedman, R. A. Olshen, and C. J. Stone. Classification and regression trees, Monterey, CA: Wadsworth. 1984*
3. *Cios, K., Pedrycz, W., Swiniarski, R., Kurgan, L., Data Mining A Knowledge Discovery Approach, Springer , 2007*
4. *P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis, Modelling wine preferences by data mining from physicochemical properties, Decision Support Systems, Elsevier, 47(4):547-553.*
5. *Dougherty, J., Kohavi, R., Sahami, M., Supervised and Unsupervised Discrimination of Continuous Features. In: Proceedings of the12th International Conference, Morgan Kaufman, (1995) p. 194-202*
6. *Fayyad, U. M., & Irani, K. B. On the Handling of Continuous-valued Attributes in Decision Tree Generation, Machine Learning 8, (1992). p. 87-102.*
7. *Forgy E.W., Cluster analysis of multivariate data: efficiency vs. interpretability of classifications, Biometrics 21 (1965), p.768–769.*
8. *Hartigan, J., Wong, M., A k-means Clustering Algorithm, Applied Statistics 28(1979) p.100–108.*
9. *Ho, K., Scott P., Zeta: A Global Method for Discretization of Continuous Variables, In: Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining (1997), p. 191-194*
10. *Joiţa D., Unsupervised static discretization methods in data mining, Conferinţa internaţională "Educaţie şi creativitate pentru o societate bazată pe cunoaştere", Universitatea Titu Maiorescu, secţiunea Ştiinţa şi tehnologia informaţiei, pag. 29-32, Ediţia a II-a, Bucureşti, 20-22 noiembrie, 2008*
11. *Joiţa D., Discretization Based on Clustering Methods, Conferinţa internaţională "Educaţie şi creativitate pentru o societate bazată pe cunoaştere", Universitatea Titu Maiorescu, secţiunea Ştiinţa şi tehnologia informaţiei, Ediţia a III-a, Bucureşti, noiembrie, 2009*
12. *Kurgan, L., Cios, K., CAIM discretization algorithm, IEEE Transactions on Knowledge and Data Engineering, (2004) 16, no.2, p.145-153*
13. *Larose F. T., Discovering Knowledge in Data: An Introduction to Data Mining, John Wiley & Sons, 2005*
14. *MacQueen J, Some methods for classification and analysis of multivariate observations. Proceedings of the 5th Berkeley Symposium on Mathematics, Statistics and Probability 3 (1967), p. 281–297.*
15. *Quinlan, J. R.. Induction of decision trees. Machine Learning 1(1) (1986), p.81–106.*
16. *Rakotomalala R., "TANAGRA: a free software for research and academic purposes", in Proceedings of EGC'2005, RNTI-E-3, vol. 2, pp.697-702, 2005. (in French)*
17. *Witten, I., Eibe, F., Data Mining. Practical Machine Learning Tools and Techniques, Second edition, Morgan Kaufman, 2005*
18. *Wu, X. et al, Top 10 Algorithms in Data Mining, Knowledge Information Systems (2008) 14, p.1–37*
19. *http://eric.univ-lyon2.fr/~ricco/tanagra/en/tanagra.html*
20. *http://archive.ics.uci.edu/ml/index.html*