Fuzzy Associative Rule-based Approach for Pattern Mining and Identification and Pattern-based Classification

Ashish Mangalampalli
International Institute of Information Technology
Hyderabad, India
ashish_m@research.iiit.ac.in

Supervised by Vikram Pudi
International Institute of Information Technology
Hyderabad, India
vikram@iiit.ac.in

ABSTRACT

Associative Classification leverages Association Rule Mining (ARM) to train Rule-based classifiers. The classifiers are built on high quality Association Rules mined from the given dataset. Associative Classifiers are very accurate because Association Rules encapsulate all the dominant and statistically significant relationships between items in the dataset. They are also very robust as noise in the form of insignificant and low-frequency itemsets are eliminated during the mining and training stages. Moreover, the rules are easy-to-comprehend, thus making the classifier transparent.

Conventional Associative Classification and Association Rule Mining (ARM) algorithms are inherently designed to work only with binary attributes, and expect any quantitative attributes to be converted to binary ones using ranges, like "Age = [25, 60]". In order to mitigate this constraint, Fuzzy logic is used to convert quantitative attributes to fuzzy binary attributes, like "Age = Middle-aged", so as to eliminate any loss of information arising due to sharp partitioning, especially at partition boundaries, and then generate Fuzzy Association Rules using an appropriate Fuzzy ARM algorithm. These Fuzzy Association Rules can then be used to train a Fuzzy Associative Classifier. In this paper, we also show how Fuzzy Associative Classifiers so built can be used in a wide variety of domains and datasets, like transactional datasets and image datasets.

Categories and Subject Descriptors

H.2.8 [**Information Systems**]: Database Applications—*Data mining*

General Terms

Algorithms, Experimentation

1. INTRODUCTION

Classification is a very old problem which involves building a model or classifier to predict the unknown class labels of future data. This problem has been addressed in numerous ways, prominent of which is by building rule-based classifiers. Other previous studies such as decision trees [24], naive-bayes [10] and other statistical approaches have developed heuristic/greedy search techniques for building classifiers. Machine learning approaches like Support Vector Machines (SVMs) [5] do classification by learning boundaries between classes and checking on which side of the boundary, the query lies. Thus, building accurate and efficient classifiers for large databases is one of the essential tasks of data mining and machine learning research.

A new classification approach called associative classification has gained a lot of popularity of late, because of its accuracy, which can be attributed to its ability to mine huge amounts of data to build a classifier. It integrates association rule mining and classification by using association rule mining (ARM) algorithms, such as Apriori [1], to generate the complete set of association rules. These methods mine high quality association rules, and build classifiers based on them. Associative classifiers have several advantages:

- Frequent itemsets capture all the dominant relationships between items in a dataset.
- As these classifiers deal only with statistically significant associations, the classification framework is robust.
- Moreover, low-frequency patterns (noise) are eliminated during the ARM stage.
- The rules, used for classification, are very transparent and can be easily understood by the user, as opposed to the black-box-like approach popular classifiers, like SVMs and Artificial Neural Networks (ANNs) have.

However, the major problem with the associative classification is that crisp associative classification is not suitable, both theoretically and practically, for datasets and domains which make heavy use of numerical attributes, or have a mix of numerical and binary attributes. One method to circumvent this problem is to use binning or clustering to convert numerical attributes to categorical attributes. But using crisp binning or crisp clustering introduces *uncertainty* especially at the *boundaries* of bins or clusters, leading to *loss of information*. Small changes in the selection of number of bins or clusters may lead to *polysemy* (one bin or cluster containing features with different meanings) and *synonymy* (two features with same meaning mapped into different bins or clusters), thus generating misleading results. A more effective way to solve this problem is having features belong to clusters with some *membership value* in the interval [0, 1], instead of belonging entirely to a particular cluster. Thus, fuzzy features replace binary ones.

Copyright is held by the International World Wide Web Conference Committee (IW3C2). Distribution of these papers is limited to classroom use, and personal use by others.
WWW 2011, March 28–April 1, 2011, Hyderabad, India.
ACM 978-1-4503-0637-9/11/03.

2. RELATED WORK

CBA [17] was one of the first associative classifiers and focuses on focusing on mining a special subset of association rules, called class association rules (CARs). CMAR [16] considers multiple rules at the time of classification using weighted χ^2. CPAR [28] takes a totally different approach to associative classification in that it does not directly use the rules generated by ARM, but only uses the frequent itemsets and their respective counts to build a classifier using a FOIL-like technique called PRM. Even before the advent of associative classifiers, many non-associative classifiers were proposed, prominent of which is C4.5, which is a widely known decision tree classifier [24].

Thabtah in [25] provides a very detailed survey of current associative classification algorithms. The author has compared most of the well-known algorithms, like CBA, CMAR, and CPAR among others, highlighting their pros and cons, and describing briefly how exactly they work. He lays more emphasis on pruning techniques, rule ranking, and predication methods used in various classifiers, and also provides valuable insights into the future directions which should be undertaken in associative classification.

Until now, the only algorithms being used for fuzzy ARM are various fuzzy adaptations of Apriori [1]. Over the years it has been shown [23], [13] that Apriori, per se, is a slow ARM algorithm for most large datasets, even crisp ones. Thus, fuzzy versions of Apriori do not perform fast against very large datasets and datasets with a large number of dimensions/attributes. More details of Fuzzy Apriori, the defacto algorithm used for fuzzy association rule mining, can be found in [27], [4], [26], [14].

[3], [27], [4] discuss in detail about fuzzy implicators and t-norms for discovering fuzzy association rules, especially negative association rules, from datasets. In [27] Yan *et. al.* provide a strong motivation for the need of fuzzy association rules. Though most of the papers on fuzzy ARM define the support and confidence measures used, De Cock *et. al.* in [4] clearly define in detail the actual semantics behind such support and confidence measures. [3] tries to discover the inherent two-sidedness of knowledge by mining association rules by using positive as well as negative examples, which need not necessarily be complementary. In order to do so, the authors introduce new measures of quality, especially for negative association rules. In fact, Dubois *et. al.* [8] make a very detailed analysis of t-norms and implicators with respect to fuzzy partitions, and provide a firm theoretical basis for their conclusions. They define various types of implications that can be used in the context of fuzzy association rules. Verlinde *et. al.* [26] describe in a fair amount of detail as to how fuzzy Apriori can be used to generate fuzzy association rules. In [14] Hüllermeier and Yi justify the relevance of fuzzy logic being applied to association rule mining in today's data-mining setup. [7] describes in great detail the general model and application of fuzzy association rules.

[11] uses k-Medoids (CLARANS) for the hard clustering, where as [15] uses CURE for the same. The hard clusterings so generated are then used to derive the fuzzy partitions. In such cases, where hard clustering is used, typically the middle point of each fuzzy partition is taken as reference (membership $\mu = 1$) with respect to which the memberships for other values belonging to that partitions are calculated.

3. FUZZY ASSOCIATION RULE MINING

Most real-life data are neither only binary nor only numerical but a combination of both. Quantitative attributes such as age, take values from a partially ordered, numerical scale which is often a subset of the real numbers. The general method adopted is to convert numerical attributes into binary attributes using ranges (for example, any numeric value for attribute Age would fit in ranges like "up to 25", "25-60", "60 and above"). This reduces the paradigm to traditional association rule mining with binary values.

A better way to solve this problem is to have attribute values represented in the interval [0, 1], instead of just 0 and 1, and to have transactions with a given attribute represented to a certain extent (in the range [0, 1]). Thus, we need to use fuzzy methods, by which quantitative values for numerical attributes are converted to fuzzy binary values. Doing so ensures that there is no loss of information whatever may be the value of any numerical attribute. Moreover, the inherent uncertainty that is present in numerical data (as far as ARM is concerned) is also appropriately taken care of. The corresponding mining process yields fuzzy association rules. For example, we have a crisp association rule like "People between the ages of [25, 60] earn an income [$80K, $150K]". With this rule a 24-year old person earning $100K is not accounted for. But a fuzzy association rule like, "Middle-aged people earn high incomes", is more flexible, and reflects this person's salary in a more appropriate manner.

A pre-requisite for any fuzzy associative classifier building is to have efficient fuzzy ARM algorithms. There are many efficient crisp ARM algorithms like ARMOR, especially for large datasets which are omnipresent nowadays. But the same is not true for fuzzy ARM algorithms. The fuzzy ARM algorithms proposed (FAR-Miner and FAR-HD) try to solve this problem so that the actual fuzzy associative classifier building and classification can be done smoothly and quickly.

To obtain frequent patterns and fuzzy association rules, we have come up with an efficient and fast fuzzy association rule mining algorithm called FAR-Miner [20]. FAR-Miner works very fast on even very large datasets (in the order of millions of transactions). In fact, we are in the process of refining many parts of FAR-Miner in order to make it faster and more scalable to even larger datasets. From our experiments on a medium-sized real-life dataset and a large real-life dataset, we have observed that FAR-Miner is 8-19 times faster for large datasets, and 6-10 times for medium-sized datasets as compared to fuzzy Apriori. This efficiency in performance is obtained thanks to the properties like two-phased tidlist-style processing using multiple partitions, and also novel attributes, like the use of sub-partitions within partitions, byte-vector representation of tidlists, effective compression of tidlists, quicker generation of new tidlists, and a tauter and faster second phase of processing.

4. PRE-PROCESSING FOR FUZZY ASSOCIATION RULE MINING

An important aspect of fuzzy ARM is the pre-processing of the dataset to make it suitable for the fuzzy ARM process. Unlike crisp ARM, any dataset cannot be used directly for the fuzzy ARM process. A dataset needs substantial amount of pre-processing before it can be used as input to any fuzzy ARM algorithm. Any such pre-processing needs to neces-

sarily be based on only fuzzy methods, like fuzzy clustering. The pre-processing consists of three major steps:

- Creation of fuzzy partitions for each of the numerical attributes in the crisp data set given.
- Then, using these fuzzy partitions to create a fuzzy version of the dataset by converting crisp numerical attributes and associated numerical values to fuzzy attributes and associated values and membership degrees.

By standardizing the pre-processing and data representation (using our algorithm called FPrep - Fuzzy Clustering driven Efficient Automated Pre-processing for Fuzzy Association Rule Mining [22]), we simplify the fuzzy ARM process and bring it to a point from where any standard fuzzy ARM algorithm can be used, depending on various specifications, like domain and size of data-set. FPrep is much faster than other such comparable transformation techniques, which in turn depend on non-fuzzy techniques, like hard clustering (CLARANS and CURE). It is nearly 9 to 44 times faster than the CURE-based method, and 2672 to 13005 times faster than the CLARANS-based method. FPrep is not only much faster than other related methods, but also generates very high quality fuzzy partitions and fuzzy versions of original datasets, that too without much user-intervention.

5. FUZZY ASSOCIATIVE CLASSIFICATION USING FUZZY ARM

In this section we present two fuzzy associative classification algorithms which leverage fuzzy frequent itemsets and fuzzy association rules for use in use in two different domains and conditions - one for general databases/datasets with a mix of binary and numerical features (Section 5.1) and the other for specialized (image-related) datasets containing only numerical features (Section 5.2).

5.1 FACISME

We have come up with a new fuzzy associative classification algorithm called FACISME (Fuzzy Associative Classification using Iterative Scaling and Maximum Entropy) [21]. It uses maximum entropy and iterative scaling to build a theoretically sound classifier which is meant for accurate and efficient performance on any kind of datasets (irrespective of size and type of attributes - numerical or binary). The maximum entropy principle is well-accepted in the statistics community. It states that given a collection of known facts about a probability distribution, we choose a model for this distribution that is consistent with all the facts, but otherwise is as uniform as possible. Hence, the chosen model for FACISME does not assume any independence between its parameters that is not reflected in the given facts. In FACISME, we use the Generalized Iterative Scaling (GIS) algorithm [6] to compute the maximum entropy model. Maximum entropy models are interesting because of their ability to combine many different kinds of information.

Thus, FACISME integrates maximum-entropy-based associative classification with fuzzy logic. Because of the use of maximum entropy, FACISME has a very strong theoretical foundation, and does not assume independence of parameters in the classification process, thus providing very good accuracy. And, this accuracy is easily extensible over any kind of datasets (irrespective of size and type of attributes - numerical or binary) and domains, through the use of fuzzy logic, by creating a fuzzy associative classifier.

We compare FACISME with state-of-the-art classifiers (associative and non-associative) on UCI-ML datasets. Fig. 1 shows the accuracy obtained by each classifier on the iris dataset. FACISME performs the best in terms of accuracy as compared to the other datasets. From Fig. 2 we get to know the accuracies for all the classifiers on the breast (breast cancer) dataset. And, FACISME performs nearly as well as the most accurate classifier for this dataset. Finally, Fig. 3 depicts the results of the experimental analysis done on pima dataset. Even in this case, FACISME performs the best as compared to the other classifiers. Thus, the basic inference from this experimental analysis is that FACISME consistently performs very well on the basis of accuracy, and is even the best in two cases.

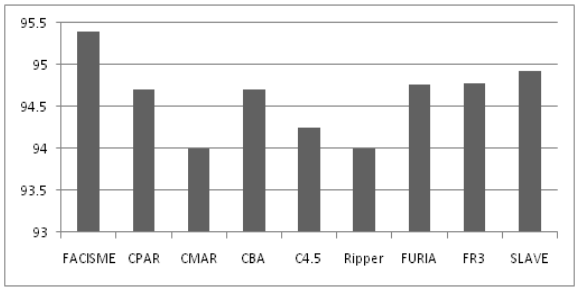


Figure 1: Experimental results on Iris dataset (Classifier vs. Accuracy)

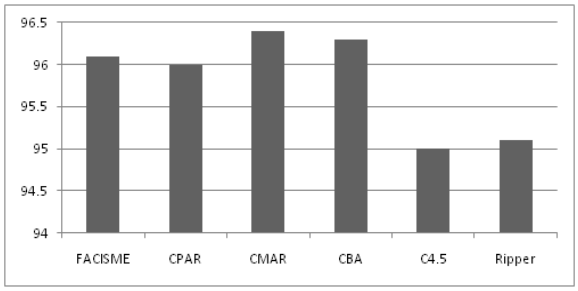


Figure 2: Experimental results on Breast dataset (Classifier vs. Accuracy)

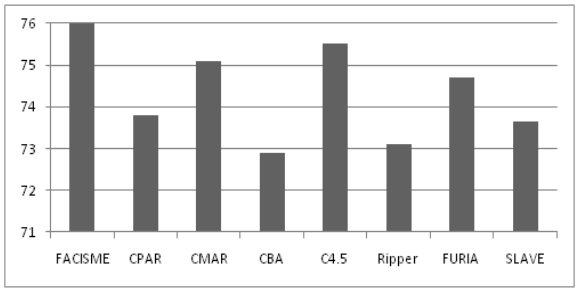


Figure 3: Experimental results on Pima dataset (Classifier vs. Accuracy)

5.2 I-FAC

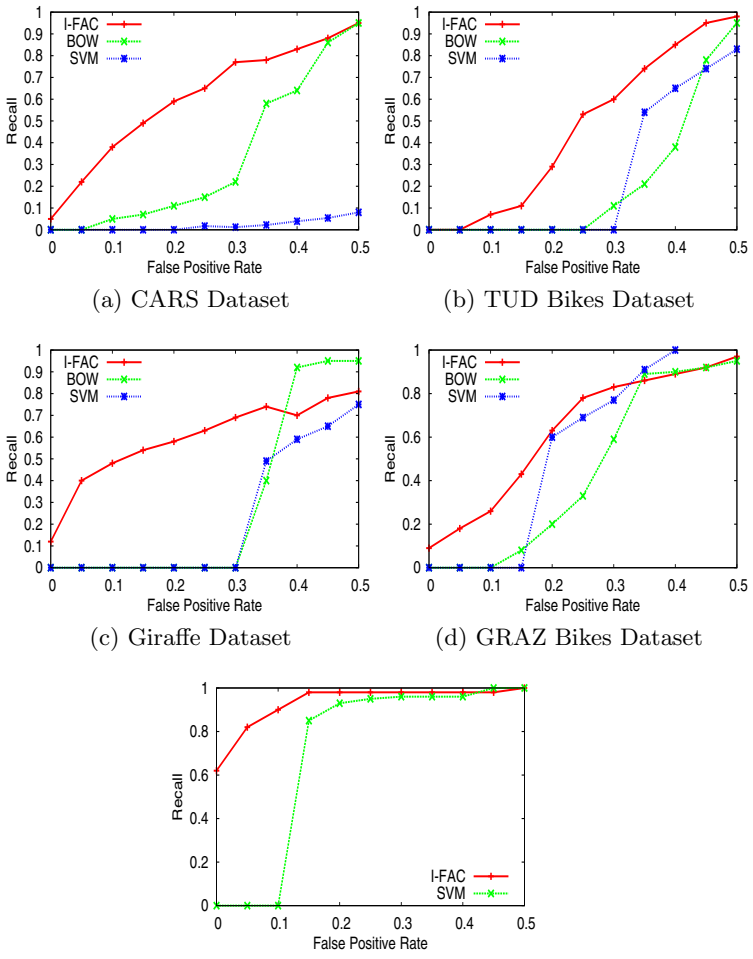
I-FAC (Efficient Fuzzy Associative Classifier for Object Classes in Images) [19] adapts fuzzy associative classification to fit the image classification perspective, by leveraging Speeded-Up Robust Features (SURF) that can be extracted

from images [2]. SURF is a fast scale and rotation-invariant interest point detector and descriptor for images. These interest points which can vary in number from image to image, can be used for further processing, like clustering and classification. Generally, obtaining the negative class set C_N is an issue in image classification due to its ill-defined nature as compared to the positive class C_P. Effectively, the negative class set $C_N = U - C_P$, where U is the universal set of all images. But, conventional classifiers need both positive and negative classes for training. Because C_N is not well-defined, classifiers so trained (on a subset of the negative class) may not perform well on disparate test images. The advantage of I-FAC is that *only positive class* samples are required to train the classifier, with *no reliance* on *negative class* samples for training and without the need for unlabelled examples or outliers. In the literature, one-class classifiers which rely on unlabelled examples [12] or treat outliers and noise as negative examples for training [18], have been proposed.

In the bag-of-words (BOW) approach, each SURF point belongs only to one of the clusters in the code-book, which is created by applying crisp clustering (like k-means) on a sizeable set of images. Using fewer number of clusters would avoid synonymy, but would at the same time give rise to polysemy. Thus, in BOW deciding upon the number of clusters that should be used is an important but difficult task, because of which ≈1000-3000 are generally used. But, I-FAC relies on fuzzy c-means (FCM) clustering [9] and creates far less number of clusters (≈100) using only the positive class training images, as compared to the number of clusters used for the code-book in BOW, thus avoiding synonymy. Due to the fuzzy nature of clusters, it is able to address polysemy as well. The salient features of I-FAC are:

- We use of fuzzy sets and logic in object class classification in images. By doing so we can deal with polysemy and synonymy better as compared to crisp sets.
- Next, I-FAC is an associative classifier, which unlike traditional classifiers like SVM, relies on frequent itemsets which encapsulate all dominant relationships between items in a dataset. This helps a lot in achieving better results and making the algorithm resilient to noise.

We compare I-FAC with two baseline approaches, namely BOW and SVM, both based on SURF points. This comparison is done on five standard image datasets which are, CALTECH Cars Rear, TUD Motorbikes, ETHZ Giraffes, GRAZ Bikes, and CALTECH Faces. I-FAC consistently performs well on the basis of FPR-versus-recall when compared to either BOW (by *high margins* on all five datasets) or SVM (by *high margins* on three datasets - Cars, Faces, and Giraffes, and by reasonable margins on the remaining two datasets - Fig. 4). It especially performs *very well* at low FPRs (≤ 0.3), which is highly desirable for an image classifier. The performance of I-FAC can be attributed to two broad reasons. First, its fuzzy nature helps avoid polysemy and synonymy, which are common problems with BOW. Second, SVM has to deal with a lot of noise in the training images, which hampers the creation of a clear hyper-plane, affecting the assignment of probability with which the positive class occurs in a given image. This problem does not occur in I-FAC which uses only the positive class for training.



(a) CARS Dataset (b) TUD Bikes Dataset

(c) Giraffe Dataset (d) GRAZ Bikes Dataset

(e) Faces Dataset

Figure 4: Results on Standard Datasets

6. FUTURE WORK

In this section, we describe briefly the work that we are currently doing, or will be doing in the near future.

6.1 FAR-HD

Fuzzy association rules can also be derived from high-dimensional numerical datasets, like image datasets, in order to train fuzzy associative classifiers. But, because of the peculiarity of such datasets, traditional fuzzy ARM algorithms are not able to mine rules from them efficiently, since such algorithms are meant to deal with datasets with relatively much less number of attributes/dimensions. We are currently working on FAR-HD (A Fast and Efficient Algorithm for Mining Fuzzy Association Rules in Very Large High-Dimensional Datasets) which is a fuzzy ARM algorithm designed specifically for very large high-dimensional datasets. FAR-HD processes fuzzy frequent itemsets in a tree-like (depth first search – DFS) manner using a two-phased multiple-partition tidlist-based strategy. It also uses a byte-vector representation of tidlists, with the tidlists stored in the main memory in a compressed form (using a fast generic compression method). FAR-HD uses Fuzzy Clustering to convert each numerical vector of the original input dataset to a fuzzy-cluster-based representation, which is ultimately used for the actual fuzzy ARM process.

6.2 SEAC and F-SEAC

We are also currently in the process of implementing a new associative classification algorithm called SEAC (Simple and Effective Associative Classifier) and its fuzzy version called F-SEAC (Fuzzy-SEAC), both of which use association rules directly for classification, after appropriate processing and pruning. Until now it has been purported that associative classification has a disadvantage in that it generates a very large number of rules during ARM, and it takes a lot of efforts to select high quality rules from among them. Hence, the newer associative classification algorithms use alternate techniques, like FOIL, PRM, and GIS, to generate high quality classification rules from the itemsets generated by algorithms like Apriori. But, with SEAC and F-SEAC we take a diametrically opposite approach to associative classification and show that association rules can be directly used for classification if suitable pruning is done and appropriate information metric (information gain in the case of SEAC and F-SEAC) is used for processing such rules.

SEAC is a holistic, straightforward, and fast classification algorithm, which uses association rules directly for classification, after two phases of processing and pruning. The first phase performs association rule mining (ARM) and subsequent classifier building globally by taking the whole dataset into account. The second phase does the same albeit locally, *i.e.* only for those classes which are either under-represented or not represented at all in the first phase. Doing so bolsters the accuracy, and drastically reduces the chances of an unclassified tuple, with a rare consequent class label, being misclassified or not classified at all. SEAC uses very few relative parameters, which can be very easily and effortlessly configured. It relies on information gain and entropy to build a set of optimal rules, which is a first as far as associative classifiers are concerned.

7. REFERENCES

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *VLDB*, pages 487–499, 1994.

[2] H. Bay, A. Ess, T. Tuytelaars, and L. J. V. Gool. Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[3] M. D. Cock, C. Cornelis, and E. E. Kerre. Fuzzy association rules: A two-sided approach. In *FIP*, pages 385–390, 2003.

[4] M. D. Cock, C. Cornelis, and E. E. Kerre. Elicitation of fuzzy association rules from positive and negative examples. *Fuzzy Sets and Systems*, 149(1):73–85, 2005.

[5] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 1 edition, 2000.

[6] J. N. Darroch and D. Ratcliff. Generalized iterative scaling for log-linear models. *The Annals of Mathematical Statistics*, 43(5):1470–1480, 1972.

[7] M. Delgado, N. Mar?n, D. Sánchez, and M. A. V. Miranda. Fuzzy association rules: General model and applications. *IEEE Transactions on Fuzzy Systems*, 11:214–225, 2003.

[8] D. Dubois, E. Hüllermeier, and H. Prade. A systematic approach to the assessment of fuzzy association rules. *Data Min. Knowl. Discov.*, 13(2):167–192, 2006.

[9] J. C. Dunn. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3:32–57, 1973.

[10] N. Friedman, D. Geiger, M. Goldszmidt, G. Provan, P. Langley, and P. Smyth. Bayesian network classifiers. In *Machine Learning*, pages 131–163, 1997.

[11] A. W.-C. Fu, M. H. Wong, S. C. Sze, W. C. Wong, W. L. Wong, and W. K. Yu. Finding fuzzy sets for the mining of fuzzy association rules for numerical attributes. In *IDEAL*, pages 263–268, 1998.

[12] B. C. M. Fung, K. Wang, and M. Ester. Hierarchical document clustering using frequent itemsets. In *SDM*, 2003.

[13] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In *SIGMOD Conference*, pages 1–12, 2000.

[14] E. Hüllermeier and Y. Yi. In defense of fuzzy association analysis. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 37(4):1039–1043, 2007.

[15] M. Kaya, R. Alhajj, F. Polat, and A. Arslan. Efficient automated mining of fuzzy association rules. In *DEXA*, pages 133–142, 2002.

[16] W. Li, J. Han, and J. Pei. Cmar: Accurate and efficient classification based on multiple class-association rules. In *ICDM*, pages 369–376, 2001.

[17] B. Liu, W. Hsu, and Y. Ma. Integrating classification and association rule mining. In *KDD*, pages 80–86, 1998.

[18] L. M. Manevitz and M. Yousef. One-class svms for document classification. *Journal of Machine Learning Research*, 2:139–154, 2001.

[19] A. Mangalampalli, V. Chaoji, and S. Sanyal. I-FAC: Efficient fuzzy associative classifier for object classes in images. In *ICPR*, pages 4388–4391, 2010.

[20] A. Mangalampalli and V. Pudi. Fuzzy association rule mining algorithm for fast and efficient performance on very large datasets. In *FUZZ-IEEE*, pages 1163–1168, 2009.

[21] A. Mangalampalli and V. Pudi. FACISME: Fuzzy associative classification using iterative scaling and maximum entropy. In *FUZZ-IEEE*, pages 1–8, 2010.

[22] A. Mangalampalli and V. Pudi. FPrep: Fuzzy clustering driven efficient automated pre-processing for fuzzy association rule mining. In *FUZZ-IEEE*, pages 1–8, 2010.

[23] V. Pudi and J. R. Haritsa. ARMOR: Association rule mining based on ORacle. In *FIMI*, 2003.

[24] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[25] F. A. Thabtah. A review of associative classification mining. *Knowledge Eng. Review*, 22(1):37–65, 2007.

[26] H. Verlinde, M. D. Cock, and R. Boute. Fuzzy versus quantitative association rules: A fair data-driven comparison. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, 36(3):679–683, 2005.

[27] P. Yan, G. Chen, C. Cornelis, M. D. Cock, and E. E. Kerre. Mining positive and negative fuzzy association rules. In *KES*, pages 270–276, 2004.

[28] X. Yin and J. Han. CPAR: Classification based on predictive association rules. In *SDM*, 2003.