A Meta-Heuristic Method to Feature Selection By Means of Harmony Search

Chikram Sridhar[1], Chenagoni Nagaraju[2], Ramesh Polisetti[3]
[1, 2, 3] Assistant Professor, Dept of CSE
[1, 2, 3] Sree Dattha Group of Institutions, Telangana

Abstract- *Most of the real world applications deal with large amounts of data that may be in Gigabytes or Terabytes. When we come to analyze this type of data it is not so easy, some practical problems arise (i.e. curse of dimensionality) , so here we use a concept called Feature Selection. The main aim of the Feature Selection is to discover a minimal feature subset from a problem domain, while retaining a suitably high accuracy in representing the original data. Many search strategies have been exploited for the task of Feature Selection, in an effort to identify more compact and better quality subsets. In this work, a novel FS approach based on harmony search (HS) is presented. Harmony Search is a recently developed meta-heuristic algorithm that mimics the improvisation process of a music player. Each musician plays a note while finding best notes of Harmony altogether. The simplicity of the Harmony Search is exploited to reduce overall complexity of search process. This work has described a flexible Feature Selection method based on general Harmony Search (HS). The thesis shows that the Harmony Search is capable of identifying better-quality feature subsets for most data sets than correlation feature selection (cfs) subset evaluator and consistency based subset evaluator.*

Keywords- Feature Selection, Harmony Search, Cfs, Subset evaluator

I. INTRODUCTION

A. Problem Specification:

This project presents a feature selection [2] method based on Harmony Search is to discover a minimal feature subset from a problem domain while retaining a suitably high accuracy in representing the original data [1]. Practical problems that arise when analyzing data in real-world applications are often related to the number of features (so-called “curse of dimensionality” [6] [10] [11]), and the inability to identify and extract patterns or rules[14] easily due to high interdependence among individual features, or the behavior of combined features. Human evaluation and sub sequent pattern identification are limited when considering data sets which have very large numbers of Features [8][9]. Techniques such as text Processing and classification can benefit greatly from FS once the noisy, irrelevant, redundant, or misleading features are removed.

B. Methodology:

In this work Optimization method which is realistic in the real world is used. Harmony search (HS) is a recently developed meta-heuristic algorithm that mimics the improvisation process of music players. The HS algorithm has been very successful in a wide variety of engineering optimization problem and machine learning [3] tasks. It has demonstrated several advantages over traditional optimization techniques.

HS imposes only limited mathematical requirements and is not sensitive to the initial value settings. Although it is a population-based approach, HS works by generating a new vector that encodes a candidate solution, after considering a selection of existing quality vectors.

C. Solutions:

Generally Feature Selection is done with 2 types of approaches. Those are

a. Filter Based Approach: which are usually used as a preprocessing step and are independent of any learning algorithm that may be subsequently employed.

b. **Wrapper Based Approach:** in contrast to filter approaches, these are often used in conjunction with a learning or data mining algorithm, where the learning algorithm forms part of the validation process. The generalized wrapper algorithm is similar to the filter approach apart from the fact that a learning algorithm is employed in place of an evaluation metric as used in the filter approach. Then Hybrid Algorithms came into exist to combine the benefits provided by both types of approach. To locate the “optimal” feature subset, an exhaustive method may be used; however, it is often impractical for most datasets. In this work Harmony Search is used for finding the minimal feature subset with high accuracy.

D. Contributions:

In this work Harmony Search is applied for Feature Selection, which gives minimal feature subsets with high accuracy for the most of UCIML bench mark data sets. This is worth and compared with the existing algorithms gives best accuracy for most of the data sets.

II. DESIGN AND IMPLEMENTATION

A. Harmony Search:

Harmony search (HS) is a recently developed meta-heuristic algorithm that mimics the improvisation Process of music players. HS mimics the improvisation process of musicians during which each musician plays a note for finding a best harmony all together. In such a search process, each decision variable (musician) generates a value (note) for finding a global optimum (best harmony).

a. Key Concepts:

i. Musical Inst. → Decision Var.
ii. Note → Decision Var. value
iii. Pitch Range → Value Range
iv. Harmony → Solution Vector
v. Aesthetics → Objective Function
vi. Practice → Iteration
vii. Experience → Memory Matrix

a) Original HS uses five parameters:

i. HMS (Harmony Memory Size)
ii. HMCR (Harmony Memory Considering Rate)
iii. K (Maximum number of iterations)
iv. PAR (Pitch Adjustment Rate)
v. FW (Fret Width)

(a). Harmony*:* It is the same as gene in the Genetic algorithm. It is the set of the values of all the variables [1] of the objective function.

(b). Harmony Memory (HM): The places where the harmonies are stored.

(c). HM Size*:* The number of places that HM has. The best harmony is stored in the first place and the rest harmonies are classified according to their performance.

(d). Max.Iteration: It defines the termination criterion.

(e). Pitch Adjustment Rate (PAR): It is the rate of choosing neighboring value.

b) Presentation of Harmony Search Algorithm:

In the following, the concept of Harmony-based Algorithm is analyzed and further explained. Then, the structure of the new algorithm is presented. The abilities and the perspectives of HSA are discussed and it is proven that the specific algorithm it is not only an innovative idea, but also a strong tool in the hands of engineers and other scientists.

c) Analysis of Harmony Search Algorithm:

(a). Seeking Harmony in Music.

The new algorithm was inspired by the improvisation process that a skilled musician follows when he is playing in a music band. During his performance the musician has one of the following choices:

i. To play the famous tune, the melody that characterizes the music piece. This specific melody is called “theme” in music. Obviously, every member of the band knows the theme and can play it by heart. In other words all musicians have this melody in their minds, stored in their memory.
ii. A common choice a musician has is to play something similar to the theme. Very often, musicians try to enrich a music piece slightly changing or adjusting pitches of the memorized theme. In this way, musicians are free to explore the theme and listeners hear its new versions. Tasteless iterations of the same tune are avoided.
iii. Finally, another choice is to start an improvisation. This choice, which is so common in Jazz music, gives the freedom to the musician to play random tunes, sometimes notes with very small (or no) relation to the performed piece. The performer uses his talent and imagination; he explores new music worlds and refreshes the music material with new themes.

B. Basic Elements of HSA:

a. Harmony*:* Harmony is similar to the gene in GA. It is the set of the values of all the variables of the objective function.

b. Harmony Memory (HM)*:* The places where harmonies are stored.

c. Harmony Memory Size (HMsize): The number of places that HM has.

The best harmony is stored in the 1st place and the rest harmonies are classified according to their performance. Definition of HMsize is an important part of the calibration of the model.

d. Maximum number of Iterations (MaxIter): Defines the termination criterion. It is similar to the maximum number of generations in GA.

C. HSA's Process:

HSA Algorithm is mimicking the choices mentioned in section2 and uses them in order to optimize a specific problem. First of all, the Algorithm fills the Harmony Memory with random values. HSA applies the three following procedures in every iteration. Procedure „b" is used (in a percentage) only if procedure „a" is activated. Option „c" is applied every time procedure „a" is not selected:

a. HS is choosing any value from HS Memory. This process is defined as Memory Consideration and it is very important because it ensures that good harmonies (values that give good results) will be considered through the solution. Moreover, these "good" harmonies will be the material (similar with parents in GA) for the creation of new, even better harmonies. In order to use this process effectively, Harmony Memory Considering Rate (HMCR) was defined. This index will specify the probability that New Harmony will include a value from the historic values that are stored in the Harmony Memory. If this rate is too low, only few elite harmonies will be selected. As a result HSA will converge slowly. Of course an HMCR value of 1.0 is not recommended because the exploration of the entire feasible range will be obstructed and optimization will fail. Typical values of HMCR are always greater than 70%.

b. Every component of the new harmony chosen from HM, is likely to be pitch-adjusted. For example a Pitch Adjusting Rate (PAR) of 10%, indicates that algorithm will choose neighboring values for the 10% of the harmonies chosen from HM. The New Harmony will include the value

xinew which will be:
xinew = xi ± Random·bw,
where, xi is the existing pitch stored in HM, Random is a random number between 0 and 1, and bw is the bandwidth of the adjustment Pitch Adjustment is similar to Mutation procedure in GA. Although PAR usually takes small values (≈5%), recent literature regards PAR as a very important factor, responsible for the convergence.

Moreover, recent studies suggest dynamic change of PAR and bw during the performance of the Algorithm. Pitch Adjusting is the local search mechanism which controls the ability for fine-tuning. Because of the importance of PAR, some scientists think of increasing its percentage even up to 50%.

c. The third choice is to select a totally random value from the possible value range. Randomization occurs with probability (100-HMCR) % and increases the diversity of the solutions. Although pitch adjustment has a similar role, it is limited in a local area. Randomization can drive the algorithm to explore the whole range and attain the global optimality. A flow diagram [5] of the search process is also shown in Figure 1.

D. Steps of Harmony Search Algorithm:

a. Initialization of an Optimization Problem
Definition of Problem and Algorithm parameters
b. Initialization of Harmony Memory.
Typically HM is filled with random values as many as the Harmony Memory Size.
c. Improvisation of a new harmony.
d. Update of Harmony Memory.

If a new harmony is better than any existing harmony, it replaces it.
e. Repetition of Steps 3 and 4 until the termination criterion is satisfied.

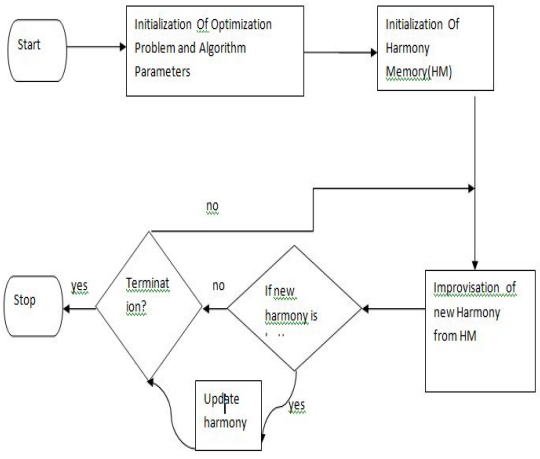

Figure 1 : Flow Chart of Harmony Search

a) Example for Harmony Search:

A simple example problem is used for a better Illustration Minimize

$(a - 2)\ 2 + (b - 3)4 + (c - 1)2 + 3$

Where a, $b, c \in \{1, 2, 3, 4, 5\}$.

(a). Initialize problem domain: In the beginning, the parameters used in the search need to be established. According to the problem at hand, the number of musicians is initialized to be equal to the number of variables (3), each corresponding to the decision attributes *a*, *b*, and *c*. Harmony memory is filled with randomly generated solution vectors. In the example problem, three randomly generated solution vectors may be {2, 2, 1}, {1, 3, 4}, and {5, 3, 3}.

(b). Improvise New Harmony: A new value is chosen randomly by each musician out of their note domain and together forms a new harmony. In the example, musician *a* may randomly choose 1 out of {2, 1, 5}, *b* chooses 2 out of {2, 3, 3}, and *c* chooses 3 out of {5, 3, 3}, forming a new harmony {1, 2, 3}

(c). Update harmony memory: If the new harmony is better than the worst harmony in the harmony memory, judged by the objective function, the new harmony is then included in the resulting harmony memory, and the existing worst harmony is removed. The new harmony {1, 2, 3} has the evaluation score of 9, making it better than the worst harmony in the memory {1, 3, 4} which has a score of 16; therefore, the harmony {1, 3, 4} is removed from memory, replaced with {1, 2, 3}. If {1, 2, 3} had a larger score than 16, it would be the one being discarded. The algorithm continues to iterate until the maximum number of iterations *K* is reached. In the example, if the musicians later choose {2, 3, 1}, which is likely as those numbers are already in the note domains, the problem will be solved with a minimal fitness score of 3.

b) Parameter control in HS:

Traditional HS uses fixed predefined parameters throughout the entire search process, making it hard to determine a "good" setting without extensive trial runs.Figure2 illustrates the details. The parameters are also non-independent from each other; therefore, Finding a good setting often becomes an optimization problem itself. The search results usually provide no hint on how parameters should be adjusted in order to gain a performance increase. To eliminate the drawbacks lying with the use of fixed parameter values, a dynamic parameter adjustment scheme is proposed to modify parameter values at run time shown in Figure 2. By using tailored sets of parameter values for the initialization, intermediate, and termination stages, the search process can benefit greatly from this dynamic parameter environment. At the beginning of a search, as the musicians are just, starting to explore the solution space, the note domains contain only randomly initialized low-quality notes. Therefore, having a large harmony memory is not essential. In fact, having to keep a large pool of suboptimal harmonies may only confuse the musicians, preventing them from choosing good values during improvisation. Lower **HMCR** at this stage may also encourage the musicians to seek values outside of the current harmony memory.

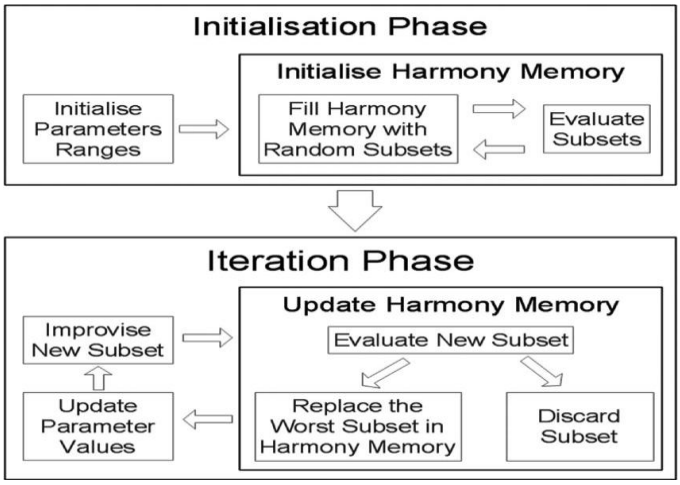

Figure 2. HS Illustration

c) Harmony Search with Feature Selection:

In this section, a description of HSFS [12] is given, based on the initial work. It explains how FS problems can be translated into optimization problems, further solved by HS. This section includes illustrative examples of the encoding scheme used to convert feature subsets into harmony representation in table1.

Table 1: Concept mapping from Harmony Search to Feature Selection

Harmony Search	**Optimisation**	**Feature Selection**
Musician	Variable	Feature Selector
Note	Variable Value	Feature
Harmony	Solution Vector	Subset
Harmony Memory	Solution Storage	Subset Storage
Harmony Evaluation	Fitness Function	Subset Evaluation
Optimal Harmony	Optimal Solution	Optimal Subset

For conventional optimization problems, the number of variables is predetermined by the function to be optimized. However, for FS [13], there is no fixed number of features in a subset. The size of the emerging subset itself should be reduced in parallel to the optimization of the subset evaluation score. Therefore, when converting concepts, as shown in Table 1, a musician is best described as an independent expert or "feature selector," [7] where the available features for the feature selectors translate to notes for musicians. Each musician may vote for one feature to be included in the feature subset when such an emerging subset is being improvised. The harmony is then the combined vote from all musicians, indicating which features are being nominated. The entire pool of the original features forms the range of notes available to each of the musicians. Multiple musicians are allowed to choose the same attribute, and they may opt to choose no attribute at all. The fitness function used will become a feature subset evaluation method, which analyzes and merits each of the new subsets found during the search process.

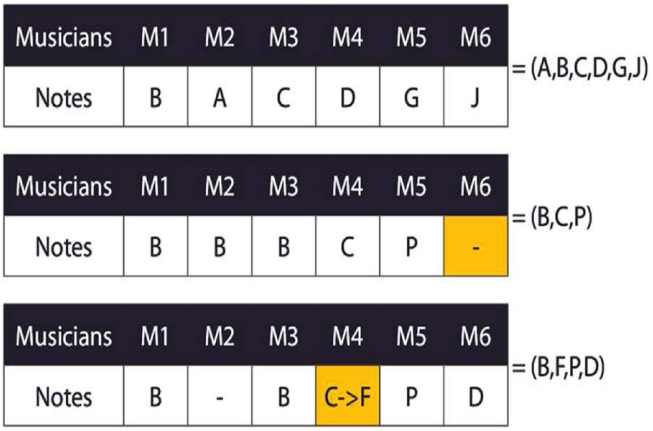

Figure. 3 Harmony encoded feature subsets.

For example,as shown in Figure. 3, the harmony *{B,A,C,D,G, J}* represents a subset of size 6, where all musicians decided to choose distinctive notes. The second harmony *{B, B, B,C,P, −}* demonstrates a duplication of choices from the first three musicians, and a discarded note (represented by -) from the last, resulting in a much reduced subset *{B,C,P}* of size 3. The last harmony *{B,−,B, C → F, P,D}* will translate into feature subset *{B,F,P,D}*, where *C → F* indicates that the original vote from musician 4 was for *C*, but it was forced to change into *F* by **HMCR** activation.

III. RESULTS

The results of a number of experimentations carried out are reported to demonstrate the capabilities of the proposed approach. In this work, the quality of the discovered subsets, or the performance of the search approaches, is judged by the subset evaluation score, in conjunction with the size of subset.

Feature Selection with Harmony Search is tested over four datasets having continuous variables. J48 classifier has been used to compare the results. In order to ensure convergence for the more complex data sets, a large number of iterations are uniformly chosen. Stratified tenfold cross-validation (10-FCV) is employed for data validation.

In 10-FCV, a given data set is partitioned into ten subsets. Of these ten subsets, nine subsets are used to perform a training fold, where FS algorithms are used to select the feature subsets. A single subset is retained as the testing data so that a classifier learner can be tested using the selected feature subsets. This cross-validation process is then repeated ten times (the number of folds). In the experiment, 10-FCV is performed ten times in order to lessen the impact of random factors within the heuristic algorithms; these 10×10 sets of evaluations are then aggregated to produce the final experimental outcomes. The advantage of 10-FCV over random sub sampling is that all objects are used for both training and testing, and each object is used for testing only once per fold. The stratification of the data prior to its division into different folds ensures that each class label[4] (as far as possible) has equal representation in all folds, thereby helping to alleviate bias/variance problems.

Table 2 :UCIML Repository Benchmark Datasets

Dataset	Attributes	Instances	#Classes
Soybean	35	683	19
Ionosphere	35	354	2
Glass	9	214	7
Iris	4	150	3

Table 5.1 describes this project work input data sets taken from UCIML repository. The results of the experiments are presented in tables 3,4,5 and Figures 1,2,3,4,5 that follow.

Table 3: Selected Important Features Using Cfs Subset Evaluator

Dataset	Original Features	Selected Features	Classification Accuracy(%)
Soybean	35	1,3,4,5,7,8,9,10,11,12,13,15,17,18,19,22,23,24,26,28,30,35 (22 Features)	90.3367 %
Ionosphere	35	1,3,4,5,6,7,8,14,18,21,27,28,29,34 (14 Features)	90.5983 %
Glass	9	1,2,3,4,6,7,8,9 (8 Features)	68.6916 %
Iris	4	3,4 (2 Features)	96.00%

Table 4: Selected Important Features Using Consistency Subset Evaluator

Dataset	Original Features	Selected Features	Classification Accuracy (%)
Soybean	35	1,3,4,5,6,7,9,10,15,16,18,22,29 (13 Features)	82.1376 %
Ionosphere	35	5,6,8,13,22,27,34 (7 Features)	87.4644 %
Glass	9	1,2,3,4,6,7,9 (7 Features)	69.0935 %
Iris	4	3,4 (2 Features)	96.00%

Table 5 :Selected Important Features Using HS

Dataset	OriginalFeatures	Selected Features	Classification Accuracy (%)
Soybean	35	1,2,3,9,10,13,15,16,17,18,20,21,23,24,25,29,31,35 (18Features)	91.0688 %
Ionosphere	35	1,3,9,14,16,18,22,23,24,25,32 (11 Features)	91.168%
Glass	9	1,2,3,4,5,6,8 (7 Features)	69.626%
Iris	4	3,4 (2 Features)	96.00%

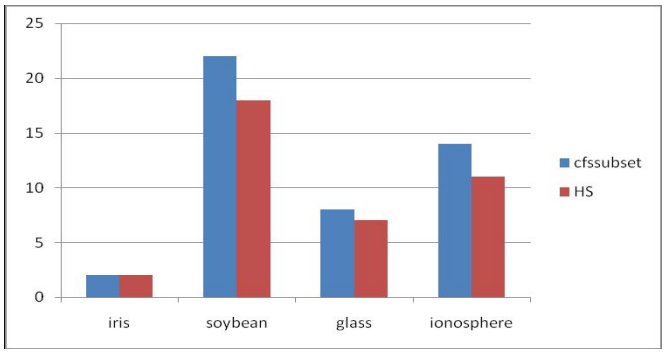

Figure 4:Comparision of Datasets with Selected Number of Features

Figure 4 describes the comparison of datasets with selected number of features. It clearly shows that for different datasets Harmony Search gives better features than cfs subset evaluator.

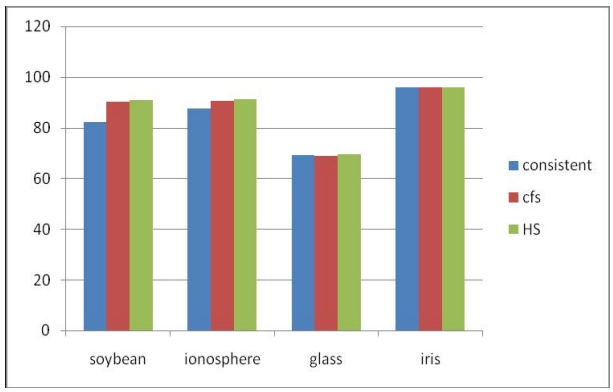

Figure 8:Comparision of Datasets with Selected Number of Features From the above figure it is clearly visible that the feature subsets obtained by the Harmony Search algorithm

have performed consistently better for all the four different data sets.

IV. CONCLUSION

This work has described a flexible Feature Selection method based on general Harmony Search. Initially Harmony Memory is filled with randomly generated Feature subsets. Here J48 classifier is used as a fitness function to evaluate the accuracy. This classifier is applied to each feature subset entry in the Harmony Memory (HM). Next another randomly generated feature subset is selected other than those in the Harmony Memory and its fitness score is calculated. If the fitness score of the newly generated subset is found to be more than the least fitness score present in the Harmony Memory, then the subset score with the least score in the HM is replaced by the new subset. This procedure should be done until maximum number of iterations or termination criterion is satisfied. This work offers a number of advantages over conventional approaches: fast Convergence, simplicity and efficiency in finding minimal subsets. Next, comparative study is done between Harmony Search and two existing algorithms. Experimental comparative studies show that the HS is capable of identifying good-quality feature subsets for most test data sets. Currently the total number of iterations is predefined, but a good subset may be found early during the search process. It would be useful to develop a better stopping criteria based on the harmony memory. And also a better iterative refinement algorithm may also be developed.

REFERENCES

[1] B.S.Everitt. “An Introduction to Latent Variable Models, Monographs on Statistics and Applied Probability”, Chapman & Hall, London. 1984.

[2] E.P.Xing.“Feature Selection in Microarray Analysis. A Practical Approach to Microarray Data Analysis”, Kluwer Academic Publishers. 2003.

[3] http://download.oracle.com/docs/cd/B28359_01/datamine .1 11/b28129/classify.html

[4] I KougiasandN.Theodosiou,”A New Music-Inspired Harmony Based Optimization Algorithm”. Theory and Applications Department of Civil Engineering Aristotle University of Thessaloniki 541 24 Thessaloniki, Greece.

[5] M.A. Carreira-Perpinan. “Continuous latent variable models for dimensionality reduction and sequential data reconstruction”. PhDthesis, University of Sheffield, UK. 2001.

[6] M.Dash and H. Liu, “Feature selection for classification,” Intell. Data Anal.,vol1, no.3,pp. 131–156, 1997.

[7] M. Dash, K. Choi, P. Scheuermann and H. Liu. “Feature Selection for Clustering – A Filter Solution”. In Proceedings of IEEE International Conference on Data Mining (ICDM), pp. 115–122. 2002.

[8] M. Shah, M. Marchand, and J. Corbeil, “Feature selection with conjunction of decision stumps and learning from microarray data,” IEEE Trans.Pattern Anal. Mach. Intell., vol. 34, no.1, pp. 174–186, Jan. 2012.

[9] Q. Shen and A. Chouchoulas. “A fuzzy-rough approach for generating classification Rules”. Pattern Recognition, Vol. 35, No. 11, pp. 341–354. 2002.

[10] R. E. Bellman, “Dynamic Programming”. Princeton, NJ: Princeton Univ. Press, 1957.

[11] Ren Diao and Qiang Sheen,” Feature Selection With Harmony Search” IEEE Trans. Systems, Man and cybernetics,vol.42,pages 1509-1523,Dec 2012.

[12] R.Jensen and Q.Shen. “Fuzzy-Rough Sets for Descriptive Dimensionality Reduction”.In Proceedings of the 11th International Conference on Fuzzy Systems, pp. 29–32002.

[13] W.H. Au and K.C.C. Chan. “An Effective Algorithm for Discovering Fuzzy Rules in Relational Databases”.In Proceedings of the 7th IEEE International Conference on Fuzzy Systems, pp. 1314–1319. 1998