Hybridization of Evolutionary Mechanisms for Feature Subset Selection in Unsupervised Learning

Dolores Torres[1], Eunice Ponce-de-León[1], Aurora Torres[1],
Alberto Ochoa[2], and Elva Díaz[1]

[1] Universidad Autónoma de Aguascalientes, Centro de Ciencias Básicas. Avenida Universidad
# 940 Col Ciudad Universitaria, Aguascalientes, AGS. C.P. 20100. México
{mdtorres,eponce,atorres}@correo.uaa.mx,
elva.diaz@itesm.mx
[2] Universidad Autónoma de Ciudad Juárez, Instituto de Ingeniería y Tecnología, Departamento
de Ingeniería Eléctrica y Computación. Henry Dunant # 4016 Circuito Pronaf Apdo. Postal
1594-D, Juárez, Chihuahua. CP 32310. México
megamax8@hotmail.com

Abstract. Feature subset selection for unsupervised learning, is a very important topic in artificial intelligence because it is the base for saving computational resources. In this implementation we use a typical testor's methodology in order to incorporate an importance index for each variable. This paper presents the general framework and the way two hybridized meta-heuristics work in this NP-complete problem. The evolutionary mechanisms are based on the Univariate Marginal Distribution Algorithm (UMDA) and the Genetic Algorithm (GA). GA and UMDA – Estimation of Distribution Algorithm (EDA) use a very useful rapid operator implemented for finding typical testors on a very large dataset and also, both algorithms, have a local search mechanism for improving time and fitness. Experiments show that EDA is faster than GA because it has a better exploitation performance; nevertheless, GA' solutions are more consistent.

Keywords: Hybridized Evolutionary Mechanisms, Feature Subset Selection, Univariate Marginal Distribution Algorithm, Genetic Algorithm, Unsupervised Learning.

1 Introduction

Even though supervised learning has obtained very good results, this, is not the case of feature selection in unsupervised learning [22], The main goal of feature subset selection for unsupervised learning, consists on finding out the subset of features that better describes an original dataset; taking out noise or distortion for identifying the class data belong. When a good feature subset selection is done, a very important task is made because removing irrelevant data, increases accuracy of learning and its comprehension according with Blum and Langley [3]. Reducing data dimensionality has become very important in machine learning during the past decades. We can see large datasets with instances described by many features in problems as image processing,

A. Hernández Aguirre et al. (Eds.): MICAI 2009, LNAI 5845, pp. 610–621, 2009.

text mining and bioinformatics. To efficiently deal with those data, dimension reduction techniques emerged as a useful preprocessing step in the task of data analysis. A subset of these techniques makes reference to feature subset selection techniques. The difference between these techniques and other reduction techniques (like projection and compression techniques) is that the first ones do not transform the original input features, but select a subset of them [17]. The work presented here, is located in Automated Learning; that according with Tom Mitchell, refers to the study of computational algorithms that improve themselves with experience [19]. Moreno et al., say that it is the kind of learning a machine can achieve [20]. Nowadays, the efficient use of the resources is not a privilege, but also, it is a necessity. An universal problem, all intelligent agent has, consists on defining where to focus its attention [12], that is why features subset selection has so much importance in data mining and automated learning area.

In recent years, real world and large scale problems have been solved by using combinations of metaheuristics with other techniques and mechanisms; that is because the use of a pure metaheuristic has some restrictions. The idea of hybridization refers to take advantage of the potentialities of some of them and save their weaknesses using mechanisms or techniques from others sources that make them more robust; this way, they can provide a more efficient behavior and a higher flexibility. Some recommendations about how hybridization could be done through of the knowledge about the problem that is treated, can be seen on [8], specifically for GA's. This paper presents two evolutionary metaheuristics hybridized that have their base in the well known genetic algorithm and in the estimation of the distribution algorithm (univariate marginal distribution algorithm).

Genetic algorithm (GA) was presented in the sixties decade, and it has proved be an excellent evolutionary tool for solving different kind of problems; this tool is based on natural evolution [10]. The GA presented, is based on the simple genetic algorithm created by Goldberg [9]. Obviously, many novelty modifications have been incorporated to it. This algorithm has been hybridized for improving exploitation and exploration mechanisms and also testor's theory has been added.

Finding all typical testors is a very expensive procedure; all described algorithms have exponential complexity, and also, they depend on the size of the matrix [24]. That is why, nowadays, this is a current problem.

The problem of features subset selection, has been treated with data mining [22], metaheuristics [11, 12, 23, 30, 31, 33] (actually, Inza et al., were the pioneers using EDAs in feature subset selection), multi-objective point of view [18], etc. Reader could review the state of the art for feature selection in [3, 7, 12]. Nevertheless, results at this time are not conclusive. That is why, we propose two evolutionary algorithms hybridized that use our novelty framework that is adaptable to supervised and unsupervised learning and also we reduced considerably the time used for identifying typical testors by means the hybridization.

In this paper, a comparison between two evolutionary algorithms is presented; both of them: Genetic Algorithm and Estimation Distribution Algorithm use an accelerating mechanism that let them have a pseudo-random behavior that address the search toward a promising area, while the improvement mechanism lead toward a local peak (just like a partial hill-climbing). Both algorithms are focused on finding the feature subset selection that better distinguishes the clusters found. The subsequent sections of this work are organized as follow. In section 2, the overall framework is presented; also some important topics related with the problem and the accelerating and improvements mechanisms are presented here. Section 3 is dedicated to describe the two algorithms to be compared: GA and UMDA. Experiments and results are presented on section 4 and finally, conclusions and future work are discussed on section 5.

2 Framework

In 2005, Peng Liu et al. [22] presented an interesting proposal for unsupervised learning using data mining. The main contribution of Peng Liu research consists on first develop clustering phase, and after the searching phase. This strategy was created for making easier and more understanding wrapper-based methodologies. We use that work as a base for our framework named: "Evolutionary Mechanisms Hybridized for Feature Subset Selection in both: supervised and unsupervised learning". Figure 1 shows the framework we created.

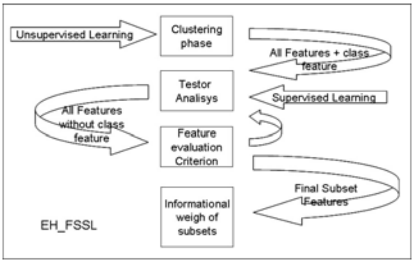
Unsupervised Learning
Clustering phase
All Features + class feature
Testor Analisys
Supervised Learning
All Features without class feature
Feature evaluation Criterion
Final Subset Features
Informational weigh of subsets
EH_FSSL

Fig. 1. Framework

An interesting model for intelligent learning environments that also supports supervised an unsupervised learning in other context could be review on [2].

As we can see, our framework is so flexible, that it works in supervised learning as well as in unsupervised learning. We used this framework with a GA and with an UMDA that were hybridized with tools like:

- A logical combinatory approach (typical testors).
- A local search mechanism (improvement mechanism).
- A global search operator (accelerating operator).

In this research, we used k-mean as our clustering algorithm because it is good for finding groups of cases given a specific k (number of classes).

As it can seen in the framework, when unsupervised learning is used:

1. All features are entered to the k-means algorithm.
2. The classification feature is obtained and then, all features plus the classification one are used for starting the testor analysis.
3. If we are working with supervised, then the process starts with the analysis. After that, (in a cyclical process), a subset of features for the clusters given is proposed (as a testor or typical testor).

4. Each subset proposed by the evolutionary algorithm is pondered and a fitness is assigned to each one.
5. Finally, all the typical testors found in the process, are used to calculate the informational weigh of each feature.

The framework presented uses some important concepts that will be presented as follow.

2.1 Typical Testor

The concept of Typical Testor appeared in the middle of the fifties in Russia [4]; it began to be used in failure detection in electrical circuits. Then it was extended to variables selection on geological problems [1]. The pioneer work in the use of typical testors for feature subset selection was from Dmitriev Zhuravlev et al. [6].

This research presents two mechanisms that were applied to both: supervised and unsupervised learning. A testor is a subset of features that distinguishes objects from different classes. Satiesteban and Pons say that a typical testor is the testor which we cannot eliminate any of its features without changing its condition of testor [26]. We could say it in other words: a typical testor is a testor whose redundancy has been eliminated. Let's suppose that U is a collection of objects, and these objects are described by a set of n features; also let's suppose that the objects are grouped into k classes. By comparing each pair of features that belong to different classes, using any criterion, we can obtain the difference matrix DM that is made capturing all the differences between objects from different classes; this difference is coded with 1 if difference exists and 0 if difference does not exist. This DM can be very large when we identify many differences between members of a class with regard to others. Let T to be a subset of the entire set of labels in the columns of DM. We call Basic Matrix BM to a special data set obtained from eliminating all rows belonging to DM that are not basic rows.

Let a and b be two rows from DM, a is sub-row from b if condition presented in ec.1 is satisfied, and also exists at less one that satisfies ec. 2 [15].

$$(\forall_i \mid b_i = 0 , a_i = 0) . \qquad (1)$$

In other words, a is a basic row from DM, if there is not any other row less than a in DM.

$$(\exists_i \mid b_i = 1 \wedge a_i = 0) . \qquad (2)$$

Given a DM, we create the BM, that is the matrix composed exclusively by the basic rows in DM [15]. BM is composed by 1's and 0's too, because it has the basic differences between classes. We have to pay attention to the three declarations that follow for finding typical testors:

1. T is a testor from a Learning Matrix (LM) if no zero's rows exist in M after eliminating all columns that do not belong to the set T.
2. The set T is typical if by eliminating any feature j | j∈T, T loses its condition of testor.
3. The set of all typical testors of DM is equal to the set of all typical testors of MB [15].

Some algorithms for determining the complete set of typical testors from a basic matrix have been created. It can be mentioned: BT [28], TB [28], CT-EXT [24], LEX [26] and REC [27] among others as [25]. The framework presented here, use a genetic algorithm on one hand, and on the other, a univariate marginal distribution algorithm to find the entire set of typical testors, to determine how representative is each variable. This way, the informational weigh can be calculated. Interested reader can review the main ideas using testors in [16].

2.2 Feature Subset Selection

Regularly, feature subset selection is used to reduce dimensionality. This task can be done taking out irrelevant or redundant features. This is an important task because reducing the number of features may help to decrease the cost of acquiring data and also make the classification models easier to understand [21]. Also, the number of features could affect the accuracy of classification [32]. Some authors have also studied the bias feature subset selection for classification learning [29, 32]. This justifies the importance of a good feature subset selection for learning.

We can see two groups of authors: who consider that reduced subsets have to conserve features that are relevant and no redundant [5] and who consider the importance of the difference between optimal and relevant features that reduced subset have kept [12]. The two most frequently used feature selection methods, are filter and wrapper [12, 21, 22], Filter methodology selects features based on the general characteristics of the training data, while wrapper methodology uses a learning algorithm to evaluate the accuracy of the potential subsets in predicting a target.

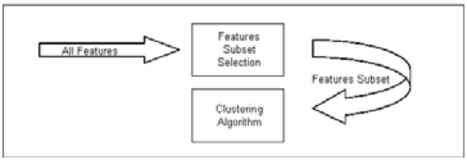

Fig. 2. Filter

Filter methods are independent of the classifier and select features based on properties that a good feature sets are presumed to have, such as class separability or high correlation with the target while wrapper treats the induction algorithm as a black box that is used by the search algorithm to evaluate each candidate [21]. While wrappers methodologies give good results in terms of accuracy of the final classifier are computationally expensive and may be impractical for large datasets [21]. That is why we tried to rescue the better of both methods creating a new framework that is based on the Pen Liu et al. [22]; but ours is more flexible, because it fits to both: supervised and unsupervised learning.

On figure 2, we can see the main components of filter methodology while figure 3 presents wrapper's.

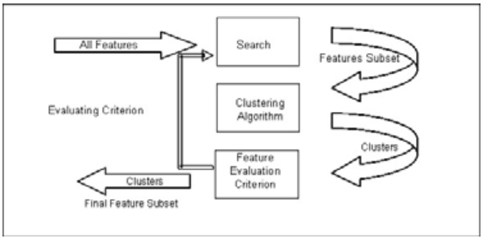

Fig. 3. Wrapper

Even though wrapper methodology has higher computational costs than filter [22], it has received considerable attention [3, 12]. Nevertheless, filter methodologies are faster and also adequate for large dataset [21, 22]. Filter methods can be further categorized into two groups: attribute evaluation algorithms and subset evaluation algorithms, which refer to whether they rate the relevance of individual features or feature subsets. Attribute evaluation algorithms study the features individually, and ponder each one according to each feature's degree of relevance to the target feature. In this research, we focus on feature subset selection (FSS) for both supervised and unsupervised learning.

2.3 Accelerating Operator

The accelerating operator is a very important contribution to our framework because it applies to any metaheuristic we could use. This operator, improves the exploration mechanisms of both: GA and UMDA because it let them to look for in promissory areas. This operator was created for taking the better of exterior and interior scale algorithms for finding typical testors. Exterior scale algorithms, are those that determine typical testors by generating the power set elements of the complete set of columns from BM in a specific order, this way, the algorithm can jump some combinations [27]. While interior scale algorithms, find typical testors studying the internal structure of the matrix, finding conditions which guarantee the condition of testor or typical testor.

Both GA and UMDA have the possibility of exploring different combinations of chains formed by 1's and 0's, for proving if each combination is or not, a typical testor. Nevertheless, we hybridized the two algorithms incorporating an operator that contains the knowledge we obtained from the shallow study of the BM. Such analysis, let us identify atypical cases; which reveal the ownership of certain feature to a typical testor. Once, the entire atypical cases presented in BM have been identified; a special string is built. This string has been called "belonging string", because it is an important little piece of any typical testor. In the core of our metaheuristics, this operator was applied for guiding searching mechanisms. This operator, reduces times in a very considerable way in both metaheuristics.

2.4 Improvement Mechanism

A special mechanism called "improvement mechanism" was used in both metaheuristics. This mechanism acts like a partial hill climbing for improving a particular solution. Obviously, this tool improves the exploitation mechanism. The main idea of using this mechanism, is to increase the probability of obtaining a typical testor from an already obtained testor. This process consists on changing 1's by 0's in the actual chromosome. It was designed based on the knowledge that all testor has inside at least one typical testor. This mechanism, was inspired on the improvement component that scatter search uses [13].

2.5 Informational Weight

The use of the informational weight for feature subset selection in unsupervised learning, is an excellent tool that gives us tangible results [32]. This mechanism consists on determining the most of the typical testor we could; and then, compute a factor of the appearance frequency.

3 Hybridized Evolutionary Algorithms

The framework was used with two algorithms, a GA, and an UMDA; the GA was called “Hybridized Evolutionary Genetic Algorithm for Feature Subset Selection in Learning” (EHGAFSSL) while the UMDA was called “Hybridized Evolutionary Univariate Marginal Distribution Algorithm for Feature Subset Selection in Learning” (EHUMDAFSSL). The two subsections that follow, show each algorithm used.

3.1 HEGAFSSL

The Hybridized Evolutionary Genetic Algorithm for Feature Subset Selection in Learning is shown as follow:

Algorithm 1. - HEGAFSSL

```
begin /* Hybridized Evolutionary GA for FSS in
Learning*/
  Initial pre-processing for taking out some
    features
  If supervised learning then goto ⌂
  else Training clustering mechanism.
  Cluster phase.
⌂ Generating DM phase.
  Generating BM phase.
  Generate initial population (randomly).
  Apply accelerating operator.
  Apply improvement mechanism.
  Compute fitness
  population <- new initial population
  Repeat
     Begin /* New Generation */
        Repeat
         Begin /* reproductive cycle for pairs of
              individuals */
           Apply selection operator.
           Apply crossover operator.
           Apply accelerating operator.
           Apply improvement mechanism.
           Compute fitness.
           Count= Count+1;
         End.
        Until Count = (generation size/ 2).
        Order of population by fitness.
        Apply elitism.
        Population <- new population
     End.
  Until (stopping criterion is reached)
  Final set of typical testors analysis.
  Compute informational weight for each feature
  Get final features subset selection.
End /* Hybridized Evolutionary AG for FSS in Learning*
```

Here, we can see that the metaheuristic hybridized with a local mechanism that improves a particular solution looking for a typical testor from a single testor. Besides that, elitism was used for accelerating the typical testors search and also, a global search operator was incorporated. This global operator, was obtained from a simple analysis of the BM. This way, we rescued the best of the interior and exterior scale algorithms, used for finding typical testors. Finally the logical combinatory focus was integrated to our methodology.

3.2 HEUMDAFSSUL

The Hybridized Evolutionary Univariate Marginal Distribution Algorithm for Feature Subset Selection in Learning, was inspired on the EDA called UMDA [14]. And it is shown as follow:

```
Algorithm 2 - HEUMDAFSSL
begin /* Hybridized Evolutionary UMDA for FSS in
Learning*/
  Initial pre-processing for taking out some
    features
  If supervised learning then goto ⌂
  else Training clustering mechanism.
  Cluster phase.
⌂ Generating DM phase.
  Generating BM phase.
  Generate initial population (randomly).
  Apply accelerating operator.
  Apply improvement mechanism.
  Compute fitness
  population [P(0)] <- new initial population
  Repeat
     Begin /* New Generation */
        Select population of promising solutions
          s(t).
        Build probabilistic model P(t) for s(t).
```

```
          Apply elitism.
          Sample P(t) to complete O(t).
          Apply accelerating operator.
          Apply improvement mechanism.
          Compute fitness.
          Order of P(t) by fitness.
          P(t) <- O(t).
          t=t+1.
       End.
    Until (stopping criterion is reached)
    Final set of typical testors analysis.
    Compute informational weight for each feature
    Get final features subset selection.
  End /* Hybridized Evolutionary AG for FSS in Learning*
```

As can be seen, the algorithm includes elitism, an added global search mechanism, and an additional local search mechanism; just like the first algorithm based on GA “HEGAFSSL”.

4 Experiments and Results

The main problem, is related with health risk factors in Mexican pregnant women. The cost of taking care of these critical risk factors is insignificant and the benefits are invaluable; so, many researches in health care area, consider it a very important topic. We found that the most important variables, women have to take care of, are 25; but 24 are indispensable while 1 is less vital than the others. The original dataset had 46 features, but we kept 29 after preprocessing phase, after the whole process we kept 25. The experimental results show that even though GA is known as a rapid convergence algorithm, thanks our mechanisms, we had a very stable algorithm that conserved the diversity required for looking in the whole solutions space.

4.1 Accelerating Operator and Improvement Mechanism

We will use AO to refer to the accelerating operator, and IM for the improvement mechanism. Experimental results related with the AO and IM, showed the improvement presented in table 1, for a very large Basic Matrix of 65549 rows and 29 features. An exhaustive algorithm has to do 536 870 911 searches on a basic matrix and this increases exponentially with the size of the matrix. The number of possible combination is 2^n-1, and it depends on the number of cluster we need to look.

Both metaheuristics reported improved in the proportion of table 1. The fitness function (ff) had only three possible values: 5 for no testors, 10 for single testors, and

Table 1. Proportion of testors

Series	Proportion
Without AO-IM	1 of each 100 000
With AO-IM	99 980 of each 100 000

20 for typical testors. We worked with a real-word and benchmark datasets, and here we are reporting here the real-problem results because it was the largest problem we used to probe the algorithms.

4.2 Experimental Results

The results presented in this section, were obtained from 10 executions using both algorithms. Experiments indicate that EDA is faster than GA.

Statistical tests, were done using SPSS software. First of all, a Shapiro Wilk test was selected, because we used the times observed in 10 executions; so, we needed a test for few data. The Shapiro Wilk test threw the following result: p-value is 0.0007434 (less than 0.05), that's why we could not accept that the errors of times had a normal distribution. Since errors have not a normal behavior, a Wilcoxon-Man Whitney test was done, to prove if times of GA and EDA were significantly different. P-value of Wilcoxon-Man Whitney test was 0.002089 (less than 0.05); therefore we can say that times are different. Invariably, EDA was faster; nevertheless, GA was more consistent conserving more diversity of results. EDA found typical testors earlier than GA.

Table 2. Fitness function (ff)

Algorithm	Generation's mean to find any typical Testor	Percentage found of the whole set of typical testors
EDA	0	70
GA	1	100

EDA tended to a single solution while GA showed more diversity even in the last generation. These results are presented in table 2.

GA is more consistent then EDA because the algorithm is able to find more results. EDA find a pattern and it guides strongly its behavior. This research considered a complex problem if the number of cases of the BM is 1 000 000 or more and the number of features is 20 or more; other case is considered as a simple problem.

5 Conclusions and Future Work

It can be concluded that hybridization improves in a very important proportion the results for the problem we attacked. To include the knowledge we have about a particular problem in the algorithm, will help to obtain important improves in our results. Finding the typical testors of a data set is an exponential problem, the accelerator operator had a powerful effect in reducing the time of this process. As long as the string is, is the time for finding the testors. Informational variable weigh is also a very important concept, since it let us to qualify the importance of each variable in a clustering process. In short time, experiments for contrasting the whole set of typical testors in a pareto's front will be done, because we consider that this problem is an inherently multi-objective one. The improvement mechanism will be perfected starting in a random point of our string doing a complete hill-climbing; We hope, applying those strategies, our mechanism will be able to tend to all results and not to find one of them (the EDA). This behavior could be achieved because the accelerating operator guarantees that the common part of any typical testors is considered.

References

1. Alba, C.E., Santana, R., Ochoa, R.A., Lazo, C.M.: Finding Typical Testors By Using an Evolutionary Strategy. In: Proceedings of V Iberoamerican Workshop on Pattern Recognition Lisbon, Portugal, pp. 267–278 (2000)
2. Amershi, S., Conati, C.: Unsupervised and supervised machine learning in user modeling for intelligent learning environments. In: IUI 2007: Proceedings of the 12th international conference on Intelligent user interfaces, pp. 72–81 (2007)
3. Blum, A., Langley, P.: Selection of relevant features and examples in machine learning. In: Artificial Intelligence 1997, pp. 245–271 (1997)
4. Cheguis, I.A., Yablonskii, S.V.: About testors for electrical outlines. Uspieji Matematicheskij Nauk 4(66), 182–184 (1955) (in Russian)
5. Dash, M., Liu, H.: Feature selection for classification. Intelligent Data Analysis 1, 131–156 (1997)
6. Dmitriev, A.N., Zhuravlev, Y. I., Krendeleiev, F.P.: On the mathematical principles of patterns and phenomena classification. Diskretnyi Analiz 7, 3–15 (1966)
7. Dy, J.G., Brodley, C.E.: Feature Selection for Unsupervised Learning. The Journal of Machine Learning Research 5, 845–889 (2004)
8. Glover, F.W., Kochenberger, G.A., Fred, W. (eds.): Handbook of Metaheuristics, pp. 55–82. Springer, Heidelberg (2005)
9. Goldberg, D.E.: Genetic Algorithms in Search. In: Optimization, and Machine Learning, Addison-Wesley, New York (1989)
10. Holland, J.H.: Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. In: Ann Arbor, The University of Michigan Press (1975); 2nd edn. The MIT Press, Cambridge (1992)
11. Inza, I., Larrañaga, P., Etxeberria, R., Sierra, B.: Feature Subset Selection by Bayesian networks based optimization. Artificial Intelligence 123(1-2), 157–184 (1999)
12. Kohavi, R.y., Jhon, G.H.: Wrappers for Feature Subset Selection. In: Artificial Intelligence 1997, pp. 273–324 (1997)
13. Laguna, M., Martí, R.: Scatter Search Methodology and Implementations in C. Kluwer Academic Press, Dordrecht (2003)
14. Larrañaga, P., Lozano, J.A.: Estimation of distribution algorithms: a new tool for evolutionary computation. Springer, Heidelberg (2002)
15. Lazo-Cortés, M., Ruiz-Shulcloper, J.: Determining the feature relevance for non classically described objects and a new algorithm to compute typical fuzzy testors. Pattern Recognition Letters 16, 1259–1265 (1995)
16. Lazo-Cortes, M., Ruiz-Shulcloper, J., Alba Cabrera, E.: An Overview of the Evolution of the Concept of Testor. Pattern Recognition 34, 753–762 (2001)
17. Lozano, J.A., Larrañaga, P., Inza, I., Bengoetxea, I.: Towards a new evolutionary computation: advances in the estimation of distribution algorithms, pp. 55–82. Springer, Heidelberg (2006)
18. Mierswa, I., Wurst, M.: Information Preserving Multi-Objective Feature Selection for Unsupervised Learning. In: Proc. of the Genetic and Evolutionary Computation Conference (GECCO 2006), pp. 1545–1552. ACM Press, New York (2006)
19. Tom, M.M.: Machine Learning. Mc.Graw Hill, New York (1997)

20. Moreno, A., Armengol, E., Béjar, J., Belanche, L., Cortés, U., Gavalda, R., Gimeno, J., López, B., Martín, M., Sánchez, M.: Aprendizaje Automático. Ediciones de la Universidad Politécnica de Catalunya. SL (1994)
21. Pelikan, M., Sastry, K., Cantu-Paz, E.: Scalable Optimization via Probabilistic Modeling: From Algorithms to Applications. Springer, Secaucus (2006)
22. Peng L, Jiaxian Z, Lanjuan L, Yanhong L, Xuefeng Z. Data mining application in prosecution committee for unsupervised learning. In: International Conference on Services Systems and Services Management, Proceedings of ICSSSM 2005, 13-15 June 05, Vol. 2, pp 1061 - 1064 (2005)
23. Saeys, Y., Degroeve, S., Van de Peer, Y.: Digging into Acceptor Splice Site Prediction: An Iterative Feature Selection Approach. In: Boulicaut, J.-F., Esposito, F., Giannotti, F., Pedreschi, D. (eds.) PKDD 2004. LNCS (LNAI), vol. 3202, pp. 386–397. Springer, Heidelberg (2004)
24. Sanchez, D.G., Lazo, C.M.: CT-EXT: An Algorithm for Computing Typical Testor Set. In: Rueda, L., Mery, D., Kittler, J. (eds.) CIARP 2007. LNCS, vol. 4756, pp. 506–514. Springer, Heidelberg (2008)
25. Sánchez, D.G., Lazo, C.M.: Modificaciones al Algoritmo BT para Mejorar sus Tiempos de Ejecucion, cuba. Revista Ciencias Matemáticas XX(2), 129–136 (2002)
26. Santiesteban, A.Y., Pons, P.A.: LEX: Un Nuevo Algoritmo para el Cálculo de los Testores Típicos. Revista Ciencias Matemáticas 21(1), 85–95 (2003)
27. Shulcloper, J.R., Alba, C., Lazo, C.: Introducción al reconocimiento de Patrones: Enfoque Lógico Combinatorio, México. Serie Verde, CINVESTAV-IPN 51, 188 (1995)
28. Shulcloper, J.R., Bravo, M.A., Lazo, C.: Algoritmos BT y TB para el cálculo de todos los test típicos, cuba. Revista Ciencias Matemáticas VI(2), 11–18 (1985)
29. Singhi, S.K., Liu, H.: Feature Subset Selection Bias for Classification Learning. In: Proceedings of 23rd International Conference on Machine Learning, Pittsburgh, pp. 849–856 (2006)
30. Torres, M.D., Torres, A., de León, P., Ochoa, A.: Búsqueda Dispersa y Testor Típico. In: XI Simposio de Informática y 6° Mostra de Software Académico SIMS 2006, Uruguaiana, Brasil, Hífen, Noviembre del 2006, vol. 30(58) (2006)
31. Torres, M.D., Ponce, E.E., Torres, A., Luna, F.J.: Selección de Subconjuntos de Características en Aprendizaje no Supervisado utilizando una Estrategia Genética en Combinación con Testores y Testores Típicos. In: Tercer Congreso Internacional de Computación Evolutiva (COMCEV 2007), pp. 57–63 (2007)
32. Torres, M.D., Ponce, E.E., Torres, A., Torres, A., Díaz, E.: Selección de Características Basada en el Peso Informacional de las Variables en Aprendizaje no Supervisado mediante Algoritmos Genéticos. In: Avances en computación Evolutiva. Centro de Investigaciones en Matemáticas, pp. 100–106 (2008)
33. Kim, Y., Street, W.N., Menczer, F.: Feature selection in unsupervised learning via evolutionary search. In: KDD 2000, pp. 365–369 (2000)