Rule Discovery in web-based educational systems using Grammar-Based Genetic Programming

C. Romero, S. Ventura, C. Hervás, P. González
Universidad de Córdoba, Campus Universitario de Rabanales, 14071, Córdoba, España. {cromero,sventura,chervas,pgonzalez}@uco.es

Abstract

This paper describes the use of data mining methods in e-learning system for providing feedback to courseware authors. The discovered information is presented in the form of prediction rules since these are highly comprehensible and they show important relationships among the presented data. The rules will be used to improve courseware, specially Adaptive Systems for Web-based Education (ASWE). We propose to use evolutionary algorithms as rule discovery methods, concretely Grammar-Based Genetic Programming (GBGP) with multi-objective optimization techniques. We have developed a specific tool named EPRules (Education Prediction Rules) to facilitate and simplify the knowledge discovery process for usage data in web-based education systems.
Keywords: web usage mining, rule discovery, web-based adaptive education, evolutionary algorithms, grammar-based genetic programming.

1 Introduction

The application of knowledge discovery techniques and data mining in web-based education systems is a very novel and promising research area (20). The same idea has been successfully used for a long time in e-commerce systems (15). But whereas the e-commerce objective is to guide clients in making purchase decisions, the e-learning objective is to guide students in learning. Currently there are a lot of tools, both commercial and freeware, to carry out data mining tasks, and mainly rule discovery. Among all those, DBMiner (21) and Weka (19) stand out because they are very popular public domain systems and they have an integrated graphic environment that lets them carry out almost all data mining tasks. The main inconvenience is that those tools are very difficult to use for a non-expert in data mining. (Teachers are typically not experts in data mining.) In addition they are of a general purpose nature, so they can't carry out a specific treatment of domain knowledge. In particular they do not contain features that are specific to the area of Adaptive Systems for Web-based Education (ASWEs) (2). To resolve these problems we have developed a specific tool that has been denominated EPRules (Education Prediction Rules), to simplify the process of discovering prediction rules (13). In the following, we describe the process of discovering information, implemented in EPRules. Then the tool is described and finally the performed tests are presented together with some instances of the discovered rules and the main conclusions obtained.

2 The Knowledge Discovery Process

The whole process of knowledge discovery with ASWEs can be carried out by the teacher or author of the course, using the EPRules tool (13). Figure 1 presents the process, beginning with the selection and pre-processing of course usage data, and proceeding with visualizing the discovered rules by applying data mining algorithms. Rules selected by the author can then be used to improve the course.

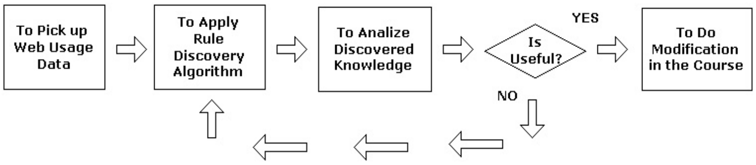

To Pick up Web Usage Data
To Apply Rule Discovery Algorithm
To Analize Discovered Knowledge
Is Useful?
YES
NO
To Do Modification in the Course

Figure 1: Knowledge discovery process.

The discovery process always begins with the selection of the database where the pre-processed usage data of the course to be used are stored. (These data are taken from log files and then stored in a database.) Then the knowledge discovery algorithms to be applied must be selected as well as their specific parameters and both the objective and subjective restrictions that we want the discovered rules to fulfil. After finishing the algorithm execution, the group of discovered prediction rules is displayed: the elements of the rule antecedent and consequent as well as the evaluation measures of each rule are shown, and it is determined if the group of discovered rules are interesting or not. This depends on the number of rules, on their quality with respect to the different measures, and on their semantic meaning. Then it is decided which of the discovered rules are sufficiently interesting to use them to take decisions on possible modifications to the course. If the rules are not considered sufficiently interesting the algorithm is applied again, with different parameters and restrictions, in order to discover a more interesting group of rules.

2.2 Prediction Rule Discovery

Prediction rules are IF-THEN rules, one of the most popular forms of knowledge representation, due to its simplicity, compressibility and expressive power (21). Depending on the knowledge they store, there are different types of rules. They are referred to as: decision rules, association rules, classification rules, prediction rules, causal rules, optimization rules, etc. The types of rules we will use are the prediction rules and the concrete format of the rules, in EBNF (Extended Backus Naur Form), is shown in Figure 2.

```
<rule>::="IF" <antecedent> "THEN" <consequent>
<antecedent>::=<antecedent> "AND" <condition> | <condition>
<consequent>::=<condition>
<condition>::=<level-attribute> "=" <level-value> |
                              <time-attribute> "=" <time-value> |
                              <success-attribute> "=" <success-value>
<level-attribute>::="LEVEL" Name of a valid level attribute
<success-attribute>::="SUCCESS" Name of a valid success attribute
<time-attribute>::="TIME" Name of a valid time attribute
<level-value>::="BEGINNER" | "NORMAL" | "EXPERT"
<success-value>::="YES" | "NO"
<time-value>::="HIGH" | "MEDIUM" | "LOW"
```

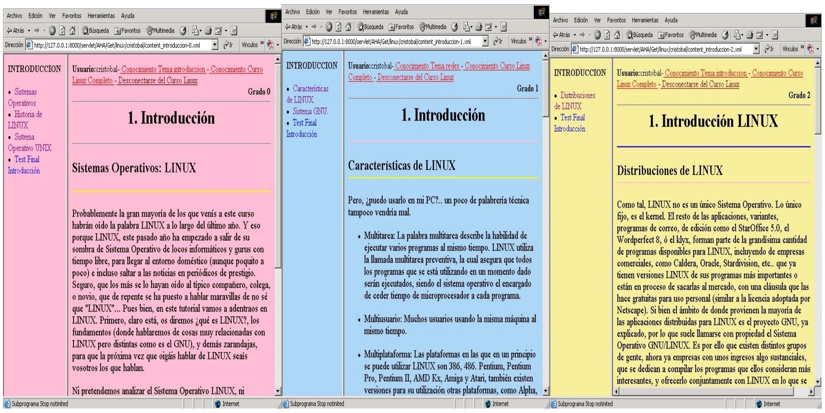
Figure 2: Example of IF-THEN rule format.

The objective of prediction rules (10) is to predict an objective attribute depending on the values of another group of attributes. Its syntax is similar to classification rules that have only one condition in the consequent, but now it is similar to any other condition (about the value of the attribute to predict). And any of the attributes can appear in the consequent of the rule as it occurs in association rules. We are going to discover prediction rules, doing for that a dependence modeling task. This data mining task consists of the prediction of relationships among attributes specified (or not) by the user (7). Prediction rules are very popular in data mining because they usually represent discovered knowledge at a high level of abstraction and it can be used directly in the decision making process. Dependence modeling can be considered like a generalization of discovering classification rules (11) or a specialization of discovering association rules (1).

The rule discovery is carried out by Grammar Based Genetic Programming (GBGP) (17) using multi-objective optimization techniques (4) which let us use several criteria for the quality of the rules. Grammar-based Genetic Programming is a genetic programming paradigm in which individuals are represented by trees derived from a grammar defined by the user to specify the solutions of the problem, in our case all the possible prediction rules. This paradigm has been chosen because the expressivity it represents, which is going to simplify the interaction with the user, restricting the grammar so the requested rules will be the only ones generated. There are plenty of metrics to evaluate the quality of the rules (8), (16), each one centred on some (different) aspects of the quality. However, there isn't any metric which clearly surpasses the others in all application domains. For this reason this problem has been set out like a multi-objective optimization problem (4). In this case there would not be a single attitude function associated to a metric, but several functions to perform the optimization at the same time. There are several ways to deal with the multi-objective optimization problem using evolutionary algorithms: the first one uses the "aggregation function", while the second one, which is the one used, uses the Pareto Front concept in which there is a vector of objectives to optimize within each individual, and the purpose of the algorithms is to make the solution for the individual converge towards the group of the best solutions (in terms of all objectives together and not in any specific objective) (6). The fitness function used is going to be made up of a 3-valued vector in which each value measures one of the three main aspects of the discovered knowledge using a data mining algorithm (7), that is, compressibility, interestingness and accuracy. The metrics selected as partial objectives are the ones named certainty factor measure (14), interestingness measure (16) and simplicity measure (9).

2.1 Students Usage data

The usage information we have used to carry out the prediction rule discovery is the usage information captured from a Linux operating system course. We have developed the Linux course using AHA! (5) because apart from being a generic adaptive hypermedia system, it captures all the user's usage information, and its source code is available so we can (and are allowed to) modify it. We have modified AHA! in order to increase its adaptation power in education (12). More precisely, we wanted to adapt or personalize the system to each particular student depending on his knowledge level. In figure 3 the same theme of the course is shown at three different difficulty levels (expert, normal and beginner).


Archivo Edición Ver Favoritos Herramientas Ayuda
Atrás Búsqueda Favoritos Multimedia
Dirección http://127.0.0.1:8000/servlet/AHA/Get/linux/cristobal/content_introduccion-0.xml
Ir Vínculos
Subprograma Stop notinited
Internet

INTRODUCCION

- Sistemas Operativos
- Historia de LINUX
- Sistema Operativo UNIX
- Test Final Introducción

Usuario:cristobal- Conocimiento Tema introduccion - Conocimiento Curso Linux Completo - Desconectarse del Curso Linux

Grado 0

1. Introducción

Sistemas Operativos: LINUX

Probablemente la gran mayoría de los que venís a este curso habrán oído la palabra LINUX a lo largo del último año. Y eso porque LINUX, este pasado año ha empezado a salir de su sombra de Sistema Operativo de locos informáticos y gurus con tiempo libre, para llegar al entorno doméstico (aunque poquito a poco) e incluso saltar a las noticias en periódicos de prestigio. Seguro, que los más se lo hayan oído al típico compañero, colega, o novio, que de repente se ha puesto a hablar maravillas de no sé que "LINUX"... Pues bien, en este tutorial vamos a adentraos en LINUX. Primero, claro está, os diremos ¿qué es LINUX?, los fundamentos (donde hablaremos de cosas muy relacionadas con LINUX pero distintas como es el GNU), y demás zarandajas, para que la próxima vez que oigáis hablar de LINUX seaís vosotros los que hablan.

Ni pretendemos analizar el Sistema Operativo LINUX, ni

Archivo Edición Ver Favoritos Herramientas Ayuda
Atrás Búsqueda Favoritos Multimedia
Dirección http://127.0.0.1:8000/servlet/AHA/Get/linux/cristobal/content_introduccion-1.xml
Ir Vínculos
Subprograma Stop notinited
Internet

INTRODUCCION

- Características de LINUX.
- Sistema GNU.
- Test Final Introducción

Usuario:cristobal- Conocimiento Tema redes - Conocimiento Curso Linux Completo - Desconectarse del Curso Linux

Grado 1

1. Introducción

Características de LINUX

Pero, ¿puedo usarlo en mi PC?.. un poco de palabrería técnica tampoco vendría mal.

- Multitarea: La palabra multitarea describe la habilidad de ejecutar varios programas al mismo tiempo. LINUX utiliza la llamada multitarea preventiva, la cual asegura que todos los programas que se está utilizando en un momento dado serán ejecutados, siendo el sistema operativo el encargado de ceder tiempo de microprocesador a cada programa.
- Multiusuario: Muchos usuarios usando la misma máquina al mismo tiempo.
- Multiplataforma: Las plataformas en las que en un principio se puede utilizar LINUX son 386, 486. Pentium, Pentium Pro, Pentium II, AMD Kx, Amiga y Atari, también existen versiones para su utilización otras plataformas, como Alpha,

Archivo Edición Ver Favoritos Herramientas Ayuda
Atrás Búsqueda Favoritos Multimedia
Dirección http://127.0.0.1:8000/servlet/AHA/Get/linux/cristobal/content_introduccion-2.xml
Ir Vínculos
Subprograma Stop notinited
Internet

INTRODUCCION

- Distribuciones de LINUX
- Test Final Introducción

Usuario:cristobal- Conocimiento Tema introduccion - Conocimiento Curso Linux Completo - Desconectarse del Curso Linux

Grado 2

1. Introducción LINUX

Distribuciones de LINUX

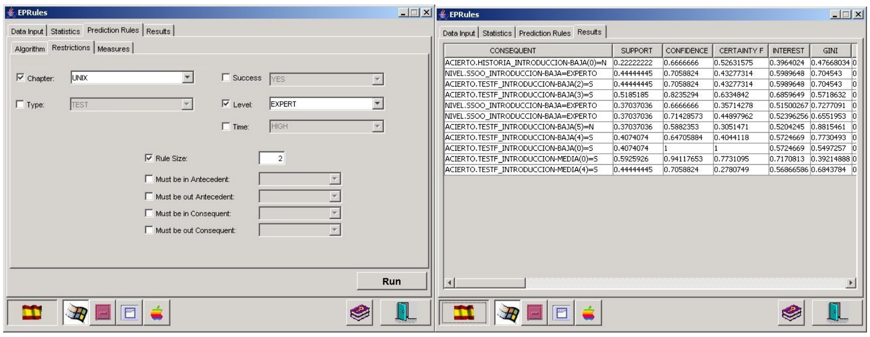
Como tal, LINUX no es un único Sistema Operativo. Lo único fijo, es el kernel. El resto de las aplicaciones, variantes, programas de correo, de edición como el StarOffice 5.0, el Wordperfect 8, ó el klyx, forman parte de la grandísima cantidad de programas disponibles para LINUX, incluyendo de empresas comerciales, como Caldera, Oracle, Stardivision, etc.. que ya tienen versiones LINUX de sus programas más importantes o están en proceso de sacarlas al mercado, con una cláusula que las hace gratuitas para uso personal (similar a la licencia adoptada por Netscape). Si bien el ámbito de donde provienen la mayoría de las aplicaciones distribuidas para LINUX es el proyecto GNU, ya explicado, por lo que suele llamarse con propiedad el Sistema Operativo GNU/LINUX. Es por ello que existen distintos grupos de gente, ahora ya empresas con unos ingresos algo sustanciales, que se dedican a compilar los programas que ellos consideran más interesantes, y ofrecerlo conjuntamente con LINUX en lo que se

Figure 3: Three different levels of the Introduction Chapter of our Linux course.

This course has been taken by 50 students of computer science engineering of the University of Cordoba. The specific usage information use for data mining is the logged data for each student's interaction with the Linux course: the access time of each page, correct answers to tests and knowledge levels. Before applying data mining algorithms we have transformed and moved all the log information to a relational database. This made (repeated) data extraction easier and increased the speed of the algorithms. During this process we have carried out the following pre-processing task (7): attribute selection, data cleaning, discretization of continuous attributes and data integration.

3 Using the EPRules tool

The EPRules tool (13) is a visual tool to discover prediction rules and it is oriented to be used by the teacher or author of the course. It has been implemented in the Java programming language and its main characteristic is its specialization in education through attributes, filters and specific restrictions for the ASWE domain. Furthermore, it is a dynamic tool, because it lets the (advanced) user add new rule discovering algorithms and new rule evaluation measures, by modifying only some configuration files (Java properties files) and by selecting the new types in the algorithm directory or the new evaluation measures. The graphic interface of the EPRules application consists of four main windows (Figure 4):



☑ Chapter: UNIX

□ Type: TEST

□ Success YES

☑ Level: EXPERT

□ Time: HIGH

☑ Rule Size: 2

□ Must be in Antecedent:

□ Must be out Antecedent:

□ Must be in Consequent:

□ Must be out Consequent:

CONSEQUENT	SUPPORT	CONFIDENCE	CERTAINTY F	INTEREST	GINI	
ACIERTO.HISTORIA_INTRODUCCION-BAJA(0)=N	0.22222222	0.6666666	0.52631575	0.3964024	0.47668034	0
NIVEL.SSOO_INTRODUCCION-BAJA=EXPERTO	0.44444445	0.7058824	0.43277314	0.5989648	0.704543	0
ACIERTO.TESTF_INTRODUCCION-BAJA(2)=S	0.44444445	0.7058824	0.43277314	0.5989648	0.704543	0
ACIERTO.TESTF_INTRODUCCION-BAJA(3)=S	0.5185185	0.8235294	0.6334842	0.6859649	0.5718632	0
NIVEL.SSOO_INTRODUCCION-BAJA=EXPERTO	0.37037036	0.6666666	0.35714278	0.51500267	0.7277091	0
NIVEL.SSOO_INTRODUCCION-BAJA=EXPERTO	0.37037036	0.71428573	0.44897962	0.52396256	0.6551953	0
ACIERTO.TESTF_INTRODUCCION-BAJA(5)=N	0.37037036	0.5882353	0.3051471	0.5204245	0.8815461	0
ACIERTO.TESTF_INTRODUCCION-BAJA(4)=S	0.4074074	0.64705884	0.4044118	0.5724669	0.7730493	0
ACIERTO.TESTF_INTRODUCCION-BAJA(0)=S	0.4074074	1	1	0.5724669	0.5497257	0
ACIERTO.TESTF_INTRODUCCION-MEDIA(0)=S	0.5925926	0.94117653	0.7731095	0.7170813	0.39214888	0
ACIERTO.TESTF_INTRODUCCION-MEDIA(4)=S	0.44444445	0.7058824	0.2780749	0.56866586	0.6843784	0

Figure 4: EPRules tool windows.

- **Data input:** In this window you can either open an existing database with the usage data of a course or create a new one and add new students to it. To create a new database or to add data, the course usage files must be selected (students' log files) that will be pre-processed and integrated into a relational database. The algorithm parameters of the time-variable discretization can be also selected and configured. (This consists of transforming continuous attributes into discrete attributes). The following unsupervised global methods have been implemented: the equal-width method, the equalfrequency method and the manual method (in which you have to specify the cut-off points.). We have only discretized the time attribute, assigning three values: HIGH, MEDIUM and LOW.

- **Data view:** If a course database has been opened from this window all students' pre-processed usage data can be visualized. These data are about the access times, correct answers and knowledge levels obtained by students for the different web pages (activities and contents) that make up the course. You can select either visualizing all students' data or a specific student's data, or just about a particular theme of the course or about a specific concept of the theme, or the visibility and difficulty level of a particular theme (high, normal, low), or a type of particular information (time, level or correct answers).
- **Prediction rules:** This is the most important part of the tool because this is where the different algorithms for rule discovery are applied. The implemented algorithms are algorithms for decision tree building like the ID3 (11), an algorithm of association rules like Apriori (1), an algorithm for induction rules like Prism (3) and different versions of evolutionary algorithms, specifically the grammar based genetic programming algorithm with or without multiobjective optimization (Pareto). You can select the algorithm to use and its particular parameters of execution and also the subjective restrictions that the rules have to fulfil (figure 3, left), so that the rules finally shown to the user are really interesting for him.
- **Results:** this window (figure 4, right) appears automatically after finishing the algorithm execution and lets us visualize all the discovered prediction rules (figure 3). Concretely, for each discovered prediction rule the conditions of the rule antecedent and consequent are shown and then all the values for each rules evaluation measure (16) (8) (confidence, support, interest, gini, laplace, etc., currently 40 available measures). In a predetermined way, they appear ordered from the first discovered one to the last one, but they can be rearranged taking into account a condition or the value of any measure by simply clicking the desired column.

4 Experimental Results

To carry out the tests we have used the log information of 50 students of computer science engineering at the Cordoba university taking a course about the LINUX operating system. The course was developed with (and served through) the AHA! system (5). Different tests have been carried out in order to compare the results that each implemented algorithm produces in the task of knowledge discovery. The objective is to compare the number of discovered rules in each case and the quality of them based on the previously set out measures about accuracy, interestingness and comprehensibility.

Because evolutionary algorithms are not deterministic the evolutionary algorithms have been executed 10 times, and we have used the average values of all the executions. Furthermore, three different tests have been carried out: first using all data, then only the frequent data (those with a support higher than 0.5) and finally, the range data (those with a support higher than 0.2 and lower than 0.9).

Algorithm	Number of discovered rules		
	All	Range	Frequent
ID3	474	131	89
Prism	657	172	62
Apriori	7960	491	70
AE-GBGP	198	162	51

Table 1: Comparison of the number of discovered rules.

Algorithm	Percentage of exact rules		
	All	Range	Frequent
ID3	46,0	51,9	60,3
Prism	71,9	53,7	91,9
Apriori	84,3	90,0	93,0
AE-GBGP	76,5	86,1	96,3

Table 2: Comparison of the percentage of exact rules discovered.

Algorithm	Percentage of interesting rules		
	All	Range	Frequent
ID3	1,5	7,6	15,6
Prism	2,5	11,6	49,3
Apriori	3,6	7,9	53,1
AE-GBGP	21,9	60,4	76,6

Table 3: Comparison of the percentage of interesting rules discovered.

The obtained results show (Table 1, 2, 3) that, in general, evolutionary algorithms generate a lower number of rules but with higher interest than classic algorithms, making them more suitable to be used on-line, like for the extraction of knowledge in an adaptative system of education. Classic algorithms, and above all Apriori, produce very exact rules, but fail when generating rules with a higher interest and, furthermore, the length of the produced rules makes these rules difficult to comprehend. In addition, when we use all data (that could happen when the user wants to extract global information about the system without applying any type of restriction over the group) it generates a group of rules so large that it becomes impossible to exploit the rules later. The obtained results using the proposed evolutionary algorithms show that, in general, the algorithms produce a lower number of rules than classic algorithms. Moreover, the use of algorithms based on Pareto Front (MOGA and NSGA) let us optimize the three set out objectives at the same time, producing in all the executions the higher proportion of exact, comprehensible and interesting rules.

4.1 Description of the discovered information

As we have commented previously our objective is to show a group of useful rules to the teacher so that he can take decisions about how to improve the course. The discovered rules show different types of relationships depending on which attributes are in the rule consequent:

- **Time.** It shows which attributes (in the rule antecedent) have an influence on the time (attribute of the rule consequent).
- **Level.** It shows which attributes (in the rule antecedent) have an influence on the level (attribute of the rule consequent).
- **Success.** It shows which attributes (in the rule antecedent) have an influence on the success (attribute of the rule consequent).

These relationships can make reference to chapters (levels obtained in initial and final tests) or to concepts (times, successes and levels obtained in exposition content pages and evaluation activities) of web-based adaptive courses. Using these discovered relations a teacher can make decisions about which modifications in the course are the most appropriate in order to increase the relationship (if he considers it to be desirable) or on the contrary to eliminate the relationship (if he considers it not to be desirable), changing or modifying the contents, the structure or the adaptation of the course. Next we are going to describe the meaning and the possible use of one discovered rules.

IF *LEVEL.interface-network-high = EXPERT*
THEN ***LEVEL.tcpip-telnet-medium = EXPERT***
(Interest = 0.57, Factor Certainty = 0.75, Simplicity = 1)

This rule shows that the knowledge level obtained in the evaluation activities of the two mentioned concepts have been simultaneously very high (EXPERT). This indicates that the concepts (NETWORK, with HIGH difficulty level in the INTERFACE chapter, and TELNET, with MEDIUM difficulty level in the TCPIP chapter) may be related to each other. In this case, the teacher should check the presentation content of both concepts and try to find what the reason of the relationship is. And he should then decide if joining both concepts into a single concept, putting both concepts in the same chapter, setting them to the same level of difficulty, correcting the rules that assign levels, or any other modification is most appropriate. In this concrete rule example we have considered that both concepts should have the same level of difficulty. But if the levels refer to initial or final test instead of activities, it can be concluded that the chapters are related. Then the teacher can join the chapters, or put them one after the other, or on the contrary, create more distance between them.

5. Conclusions and Future Work

In this paper we have introduced a visual tool to discover knowledge in the form of prediction rules in order to help teachers to improve adaptive Web-based courses. Particularly, we have proposed the use of Grammar-based genetic programming with multiobjective approaches. The quality of the results, depending on the number of rules obtained and the interestingness, accuracy and compressibility factor of the rules, is higher than for a number of classic algorithms that use only one measure or a composition of some of them to evaluate the rules. Regarding the usability of the discovered rules for taking decisions about the modifications to be carried out in ASWEs, the different types of rules, and the utilities that they can have to improve the course have been described and illustrated using specific instances of the discovered rules in a Linux course. To simplify the whole process of knowledge discovery a specific tool, named EPRules, has been developed, that lets us carry out the pre-processing of usage data in web courses, the selection of restrictions on the type of information to be discovered as well as the application of data mining algorithms to extract the rules and to visualization them.

Currently we are working on the complete automation of the knowledge discovery process in ASWEs, so that the discovered rules can be applied directly on the course, without manual intervention by the teacher or author of the course, except for accepting or rejecting the changes proposed by the rules. But these automatic modifications can be only about improvements of the course structure and not about the course content in which the teacher have to do manually the changes. Another future line of investigation will research the metrics related to the subjective interest of professionals in the discovered rules. From this point of view there are references in AEs (18) in which there isn't an aptitude function but individuals are assessed by an expert in each cycle of the algorithm.

Acknowledgments

The authors gratefully acknowledge the financial support provided by the Spanish Department of Research of the Ministry of Science and Technology under TIC2002-04036-C05-02 (Department of Computer Science, University of Cordoba) Projects. FEDER also provided additional funding.

References

[1] Agrawal, R., Imielinski, T., Swami, A., Mining association rules between sets of items in large databases, *ACM SIGMOD International Conference on Management of Data,* 1993.

[2] Brusilovsky, P., Adaptative Educational Systems on the World-Wide-Web: A Review of Available Technologies, *Int. Conf. on Intelligent Tutoring System,* 1998.

[3] Cendrowska, J., PRISM: an algorithm for inducing modular rules. *Journal of Man-Machine Studies,* 27, pp. 349-370, 1987.

[4] Coello, C.A, Veldhuizen, D.A., Lamount, G.B., *Evolutionary Algorithms for Solving Multi-Objective Problems,* Kluwer, 2002.

[5] De Bra, P., Ruiter, J., AHA! Adaptive Hipermedia for All. *Proc. of the WebNet Conference,* pp. 262-268, 2001.

[6] Fonseca, C.M., Fleming, P.J., Genetic Algorithms for Multiobjective Optimization: Formulation, Discusión and Generalization, *Conf. on Genetic Algorithms,* San Mateo, CA, 1993.

[7] Freitas, A.A., *Data Mining and Knowledge Discovery with Evolutionary Algorithms,* Springer-Verlag, 2002.

[8] Lavrac, N., Flach, P., Zupan, B., Rule Evaluation Measures: A Unifying View, *ILP-99*, Berlin Heidelberg, 1999.

[9] Liu, J.L., Kwok, J.T., An Extended Genetic Rule Induction, *Conf. On Evolutionary Computation.* 2000.

[10] Noda, E., Freitas, A., Lopes, H.S., Discovering interesting prediction rules with a genetic algorithm, *Proc. on Evolutionary Computation*, 1999.

[11] Quilan, J.R., Generating Production rules from decision trees, *Proceeding of IJCAI-87,* 1987.

[12] Romero, C., De Bra, P., Ventura, S., De Castro, C., Using Knoledge Level with AHA! For Discovering Interesting Relationship, *World Congress ELEARN,* Montreal, 2002.

[13] Romero, C., Ventura, S., de Castro, C., De Bra, P., Discovering Prediction Rules in AHA! Courses, *LNAI 2702 User Modeling'03*, pp. 25-34, 2003.

[14] Shortliffe, E., Buchanan, B., A model of inexact reasoning in medicine. *Mathematical Biosciences*, 23, pp. 351-379, 1975.

[15] Spiliopoulou, M., Web Usage Mining for Web Site Evaluation, *Communicacion of the ACM,* 2000.

[16] Tan, P., Kumar, V., Interesting Measures for Association Patterns. *Technical Report TR00-036,* Department of Computer Science, University of Minnnesota, 2000.

[17] Whigham, P.A., Gramatically-based Genetic Programing, *Proceeding of the Workshop on Genetic Programming*, pp. 33-41, 1995.

[18] Williams, G.J., Evolutionary Hot Spots Data Mining, An Architecture for Exploring for Interesting Discoveries, *Conf. on Knowledfe Discovery and Data Mining,* 1999.

[19] Witten, I.H., Frank, E., Data Mining. *Practical Machine Learning Tools and Techniques with Java Implementations*, Morgan Kaufmann. 1999.

[20] Zaïene, O.R., Web Usage Mining for a Better Web-Based Learning Environment, *Technical Report*, 2001.

[21] Zytkow, J., Klosgen, W., Handbook of Data Mining and Knowledge Discovery, *Oxford University Press*, 2001.