

A New Approach for Multi-label Classification Based on Default Hierarchies and Organizational Learning

Rosane M.M. Vallim
Instituto de Ciências
Matemáticas e de
Computação
Universidade de São Paulo
São Carlos, Brazil 13560-970
rvallim@icmc.usp.br

David E. Goldberg
Illinois Genetic Algorithms
Laboratory
University of Illinois at
Urbana-Champaign
Urbana, Illinois 61801
deg@illgal.ge.uiuc.edu

Xavier Llorà
National Center for
Supercomputing Applications
University of Illinois at
Urbana-Champaign
1205 W. Clark Street, Urbana,
IL, 61801
xllora@illgal.ge.uiuc.edu

Thyago S.P.C. Duque
Illinois Genetic Algorithms
Laboratory
University of Illinois at
Urbana-Champaign
Urbana, Illinois 61801
thyago@illgal.ge.uiuc.edu

André C.P.L.F. Carvalho
Instituto de Ciências
Matemáticas e de
Computação
Universidade de São Paulo
São Carlos, Brazil 13560-970
andre@icmc.usp.br

ABSTRACT

Learning Classifier Systems (LCSs) are a class of expert systems that use a knowledge base of decision rules and a genetic algorithm (GA) [9] as a discovery mechanism. The set of decision rules allows the LCS to represent and learn control strategies, while the robust search ability of the GA allows it to search for new rules based on the performance of existing rules. LCS were first designed to solve machine learning problems, especially classification problems. Classification problems are problems where instances of a data set belong to a set of classes, and the system needs to infer, based on past experience, the correct class (or classes) of new, previously unseen, instances. However, the features of LCSs are also very useful for solving reinforcement learning problems, a class of problems where the system should learn to operate in the environment based only on performance feedback. This paper considers LCSs as an approach to classification problems, more specifically a more complex kind of classification called multi-label classification. This paper analyses the default hierarchy formation theory presented by [14] as a way of favoring the hierarchical arrangement of rules, and also the organizational learning theory [17] for adjusting the degree of individual and collective behaviors. We suggest a new method, combining both organizational learning and default hierarchy formation, for solving multi-label problems. The preliminary results with a simple multi-label problem show the potential of this method. Final discussion presents the conclusions and directions for further research.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO'08, July 12–16, 2008, Atlanta, Georgia, USA.
Copyright 2008 ACM 978-1-60558-131-6/08/07 ...\$5.00.

Categories and Subject Descriptors

I.2 [Artificial Intelligence]: Learning

General Terms

Theory

Keywords

LCS, default hierarchies, organizational sizing

1. INTRODUCTION

The multi-label classification is a more complex type of classification problem where an instance can belong to more than one class at the same time. The initial motivation for the research in this area arise with difficulties caused by ambiguities in text categorization problems [15]. Although text categorization is still the major area of application of multi-label techniques, other areas as pattern recognition and bioinformatics benefit from them. Since the objective is to correctly assign all the possible classes of an instance, a system made to achieve this goal needs to adapt itself during the learning process to include a missing class to the set of assigned classes of the instance or to exclude a wrong predicted class from this set.

LCSs are adaptive expert systems consisting of a production system and a discovery mechanism. Holland first proposed the LCS in 1971 [8]. The production system is a rule-based system that performs inference about the problem one wishes to solve. The discovery mechanism uses a GA to search for new promising rules based on the actual set of rules in the system. GAs are search and optimization procedures based on the mechanics of natural selection, proposed by Charles Darwin, and on basic genetics.

A lot of effort has been made on the field of LCSs research [18, 4, 2, 12], not only with applications in classification, but also in reinforcement learning tasks [13]. The LCS capacity of learning control strategies and searching for

new solutions is what makes this system very useful for reinforcement learning. LCSs represent their knowledge using decision rules. Based on the type of individual representation used by LCSs, two major approaches coexist. Michigan-style LCS represents an individual as a single classifier while Pittsburgh-style LCS represents an individual as an entire rule set. Both approaches have advantages and disadvantages [3].

Although Michigan and Pittsburgh approaches are the most used ones, there is still another one that has been proposed by Wilcox [17]. In his thesis, Wilcox proposes a concept for a classifier system capable of exhibiting both individual and collective behavior, the so-called Organizational Classifier System (OCS). This system simultaneously evolves groups of classifiers, where these groups can vary in size and can also interact with each other. The idea behind the OCS is based on an economic analogy of transaction costs.

Another important feature of the LCS paradigm is the possible formation of adaptive default hierarchies. Default hierarchies are layered sets of rules where imperfect default rules cover some system responses, and exception rules handle responses when the default are incorrect. Smith [14] proposes a method for favoring default hierarchy formation using a necessity auction and different priority factors.

This paper proposes a new classifier system for solving multi-label problems. We propose a representation of multi-label problems as layered sets of default and exception rules, as an attempt to correctly learn the possible correlations between classes inside the structure of the hierarchy. Also, we search for a solution capable of separating partially correct rules, those that advocate just a fraction of all the correct classes of an instance, from the correct rules, giving the system the ability of searching for solutions containing just optimal classifiers. For achieving these objectives we make use of the OCS capacity of putting optimal classifiers together and Smith’s method for favoring default hierarchy formation. The preliminary results using a simple multi-label problem show the potential of the method proposed.

The next sections are organized as follows. Section 2 gives a short overview on multi-label classification. Section 3 presents the default hierarchy concept and a brief revision on Smith’s work on adaptive default hierarchy formation. Next, Section 4 explains the main idea behind the OCS and how it manages organizations. Section 5 proposes a new classification scheme that joins default hierarchy theory and OCS theory into a single classifier system. The preliminary results of the proposed method are presented on Section 6. Finally, Section 7 brings the main conclusions and the directions for future research.

2. MULTI-LABEL CLASSIFICATION

Unlike single-label classification, where one instance belongs to just one of the possible classes of the problem, in multi-label classification the classes are not disjoint. This means that one instance can have one or more classes associated to it. As mentioned before, the initial motivation for multi-label classification arise as an attempt to deal with ambiguities found in text categorization problems. Although text categorization is still the major area of application of multi-label techniques, nowadays these techniques are increasingly required by modern applications, such as semantic scene classification [5] and protein classification [19].

The difficulty of multi-label classification resides in learning all the possible classes that can be assigned to a given instance. A learning system designed to solve this type of classification problems needs to adapt its knowledge to include missing classes of an instance or to remove a wrong class that has been included in its set of classes.

Some techniques have been proposed to solve multi-label classification. One of the approaches divides the problem into a set of single-label problems [16]. This division of the original problem is done based on the labels of the classes or in the instances themselves. Doing this transformation, in principle any learning algorithm could be used to solve the problem. This strategy has two drawbacks: it is not scalable and it does not consider possible correlations between the classes.

Other techniques propose modifications in existing learning algorithms to make them capable of solving multi-label problems [11, 1, 19].

Finally, new algorithms designed specifically to solve multi-label classification have been proposed [6].

3. DEFAULT HIERARCHY FORMATION

Default Hierarchies are layered rule sets where default rules cover a broad part of the system responses, and exception rules are responsible for handling responses where the default is incorrect. Although exception rules are used where the default rules are imperfect, they may also be imperfect. As an example of a default hierarchy consider the following rule set, adapted from [14] :

```
1##/000
111/111
```

where / delimits actions.

If an exception rule is matched, other less specific rules in the hierarchy will also be matched. To make sure that the correct exception fires, we have to guarantee that it has priority over the other rules in the conflict-resolution process. Traditionally, this prioritization is based on a specificity factor introduced into each classifier’s bid, which is a portion of a classifier’s strength used in the conflict-resolution scheme. Smith proposes another method, where each classifier posts a portion of its strength as a potential bid. The classifier that wins the competition does not pay out the amount of its potential bid. Instead, it only pays out an actual bid that is equal to the bid of its nearest competitor.

Smith’s work also proposes two separate factors for each classifier. Factor Π , called reward estimate, is an estimative of the classifier’s expected reward. Factor Φ , called priority factor, indicates priority in the auction. The potential bid of a classifier is then the product of its reward estimate and priority factor. The reward estimate for classifier x is updated using a linear update procedure, while the priority factor for this classifier is updated using the necessity auction. The update equations are as follows:

$$\Pi_x^{i+1} = \Pi_x^i(1 - C) + CR$$

$$\Phi_x^{i+1} = \Phi_x^i - C\Pi_y^i + CR$$

Where Π_y^i is the classifier x ’s nearest competitor reward estimate at time i . If no competitors exist, the priority factor

is not updated. This method achieves adequate separation between default and exception classifiers, because maintaining separate priority factors each classifier can use its nearest competitor's reward estimate to update its own priority. However, this scheme may result in an excessive growth of a classifier's priority, not permitting that newly introduced classifiers be sufficiently evaluated. Difficulties associated with unbounded priorities can be corrected by limiting the effects of the necessity auction. This can be achieved by introducing a margin M that limits separation. Thus, the update of the priority factor is changed to:

$$\Phi_x^{i+1} = \Phi_x^i + CR - \begin{cases} C\Pi_y & \text{if } \Phi_x - \Phi_y < M \\ C\Pi_x & \text{if } \Phi_x - \Phi_y \geq M \end{cases}$$

4. ORGANIZATIONAL CLASSIFIER SYSTEM

The main idea behind the OCS is to separate rules that lead to optimal decisions from those that lead to sub-optimal decisions [17]. It tries to achieve this goal by forming appropriately sized organizations of classifiers using a mechanism inspired on economics. More specifically, it uses some techniques to reduce transaction costs [7]. The basic hypothesis is that forming organizations one can reduce the transaction costs associated with exchanging goods and services. The OCS consists of three components:

- Production system
- Conflict-resolution and credit-allocation schemes
- Organizational Growth component

The production system contains a population of organizations containing classifiers. These organizations can be of variable size and can interact with each other. Like in Michigan-style LCSs, the production system matches inputs with the actual set of rules, chooses an action and send that action to the environment and may receive a reward signal. The conflict-resolution scheme uses reputation to determine interactions of classifiers and organizations, while the credit-allocation process assigns reputation to classifiers and organizations.

Each classifier and organization carries two values, a short-term reputation value (ST), and a long-term reputation value (LT). A classifier's reputation is determined by its success at obtaining reward. The ST reputation of a classifier measures its most recent performance. The ST reputation for the i th classifier to fire is calculated as:

$$V_{STC}^i = R + b \sum \left\{ V_{LTC}^{i+1} \right\}$$

Where R is the reward received from the environment and V_{LTC}^{i+1} is a fraction of the sum of LT reputation values for classifiers that fire after reading the internal posted message. The second term is related to the bucket brigade algorithm [10]. The LT reputation of a classifier is calculated as follows:

$$V_{LTC}^i = (1 - b)V_{LTC}^i + b \sum \left\{ V_{LTC}^{i+1} \right\} + R$$

This is similar to the strength values in Michigan-style classifier systems.

An organization's reputation values are indirectly determined by the success of its classifiers. The ST reputation of an organization is updated as follows:

$$V_{STO}^{t+1} = (1 - T_O)V_{STO}^t + \sum_{c \in O} T_c V_{LTC}^c - F$$

Where T_O is a system tax, T_c is a tax on each classifier c that belongs to the organization O , and F is a fee paid when the organization is selected to affect the environment. The LT reputation of an organization measures the performance of an organization over its lifetime, and it is calculated as follows:

$$V_{LTO}^i = \frac{S_{success}}{S_{attempts}}$$

Where $S_{success}$ is the number of times the organization caused a reward signal when it was selected to affect the environment and $S_{attempts}$ is the number of times that the organization was selected to affect the environment.

The OCS conflict-resolution scheme uses LT reputation to select an organization and ST reputation to select a classifier.

The organizational growth component controls how organizations vary their size and membership through the use of two operators; Grow and Shrink. For each organization in the population, OCS randomly selects one of the two operators. The Grow operator is responsible for increasing the size of an organization. It selects one classifier from the entire population not contained in the selected organization. Then, the salary of each organization is calculate as the organization's ST reputation divided by its number of classifiers. Between the two selected organizations the one with the highest salary keeps the selected classifier. In the other hand, the Shrink operator manages to reduce the size of an organization, removing classifiers that are performing poorly. The Shrink operator compares each classifier in the selected organization with a threshold set near zero. The classifiers whose strength is below this threshold are removed from the organization and join a separate empty organization.

5. PROPOSED APPROACH

Rule-based solutions for some classification problems require some sort of hierarchical differentiation among its rules. To efficiently solve these problems, rule-based classification systems need to encourage the formation of hierarchical rule sets. In this sense, the LCS paradigm presents an important feature, which is the possible adaptive formation of default hierarchies. As presented in Section 3, in a default hierarchy an exception rule needs to have priority over default rules. Favoring default hierarchy formation helps to construct an elegant solution for hierarchical problems, allowing a graceful refinement of the rule set [14].

Although default hierarchies seem like a good solution for problems where we need to construct a hierarchy between some rules, they are still not good enough to solve problems where solutions can be partially correct. One example of such a problem is the multi-label classification problem, where we have instances that can simultaneously belong to

more than one class and, as a result, we can have rules that assign just a fraction of all the correct classes to an instance. This is an example of a partially correct rule, because although it correctly assigns some of one instance's classes, it still misses some classes. Inside the production system of an LCS a variable set of rules can be found, including the correct rules at some point of the learning process, but also a sub-set of sub-optimal rules. In a multi-label problem, these sub-optimal rules can not only be rules that advocate wrong actions, but also rules that advocate partially wrong, or partially correct, actions. So, it seems reasonable that the final solution contains only the optimal classifiers and not the only partially correct ones.

Hereby, this paper proposes a hierarchical representation of multi-label problems. We propose the use of Smith's default hierarchy formation theory to correctly separate the optimal rule set in a hierarchy and the OCS theory to adequately separate optimal from sub-optimal classifiers.

The proposed system contains four main components: production system, conflict-resolution and credit-allocation mechanism, organizational growth component and discovery mechanism.

The production system contains a population of organizations. Each organization can have at least one and at most N classifiers, where N is the total number of classifiers in the population. Also, each organization carries two values: the LT reputation and the ST reputation, calculated as in the OCS system. The classifiers carry two values, but here we do not use reputation for classifiers. Instead, we use a reward estimate value and a priority factor as proposed by Smith. When an input signal is received from the environment the system searches for all the organizations that contain classifiers that match this input. Then conflict-resolution scheme selects the most promising organization based on its LT reputation value. With the selected organization, conflict-resolution again takes places to select which classifier to fire, using the bids of the matched classifiers that are calculated as the product of a classifier's reward estimate and priority factor. The classifier with the highest bid wins the competition and sends its action to the environment. The environment then sends a reward signal that is used by the credit-allocation to update organization's and classifier's parameters.

The update procedure updates the classifier's reward estimate and priority factor, as well as organizations' LT and ST reputation, exactly as proposed in Sections 3 and 4 respectively.

At each k iterations, the organizational growth component is executed. It is responsible for sizing organizations, favoring classifiers with similar performance to be kept together. This is very similar to the OCS organizational growth component explained in section 4, using the ST reputation of organizations for selecting which organization to grow, but with two small additions. The first is associated to the Grow operator and it is simply an extra test that guarantees that if a classifier is about to leave its organization to enter another one, its performance has to be at least greater than a percentage of the performance of the organization it is about to enter. This modification was included to guarantee that classifiers known to be poor do not enter good organizations. The performance of a classifier is considered to be its reward estimate; the performance of the organization is the mean of all its classifier's reward estimate.

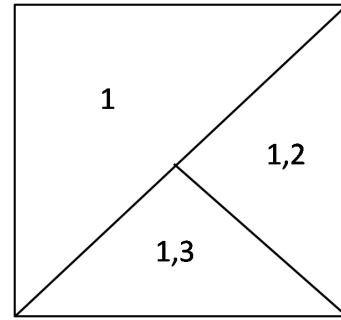


Figure 1: A simple multi-label problem

The second modification regards the Shrink operator and has almost the same logic. Originally, in the OCS the Shrink operator compares the strength of the classifiers inside of an organization with a threshold and removes those which are under this threshold. Here, instead of comparing a classifier strength (determined by its reward estimate) with a fixed threshold we again calculate the mean performance of the organization and compare the classifier's strength with a percentage of it. If the classifier's strength is smaller than this percentage of the mean performance, it is removed from the organization and joins a separate empty organization.

The discovery mechanism uses the robust search ability of GAs to create new rules, based on the experience obtained by the actual rules. In this paper the GA is set off, but some criteria presented here were included to enable a better use of its capabilities, as the reward scheme presented in section 6.

By putting together mechanisms from the OCS and from Smith's default hierarchy formation theory we expect the final system to be capable of solving multi-label classification problems, using a hierarchical rule set and separating optimal from sub-optimal classifiers. Besides, we expect the proposed system to be able to construct solutions formed by easily comprehensible rule sets, thanks to the organization capacity of OCS. The next section provides the preliminary results obtained with the proposed system in a simple multi-label problem.

6. RESULTS

In order to test the potential usefulness of the proposed classifier system, it was tested in a very simple multi-label problem, defined by the Figure 1.

In this problem, some instances belong only to class 1, while other instances belong to either classes 1 and 2 or classes 1 and 3 simultaneously. The following three rules, which form a default hierarchy, are sufficient to represent this problem:

```
Antecedent/Consequent
00/1,2
01/1,3
##/1
```

We represented these rules with a ternary codification as follows:

```
Antecedent/Consequent
00/110
01/101
##/100
```

Input	Selected Classifier	Expected output	Reward
00	00/1,2	1,2	1
00	00/1	1,2	0.666
00	00/1,3	1,2	0.333
00	00/3	1,2	0

Table 1: Example of how the reward scheme works

Where the # is the "don't care" symbol, and admits both 0 and 1 values. At the consequent of the rules we used one bit for each class of the problem. A value of 1 in one of these bits indicates the presence of the associated class in the solution, and a value of 0 indicates the absence of this class. We provided the system an initial population of rules containing all the three correct rules that represent the problem and four other rules. Among these four, two are completely wrong rules, i.e, advocate just wrong classes for the instances they match, and the remaining two are partially wrong rules, advocating some correct classes and omitting others, or advocating correct and wrong classes. The four extra rules presented to the system are as follows:

Antecedent/Consequent
00/3
01/2
00/1,2,3
00/1

Since we have correct, partially correct and wrong rules, we need a criterion to reward these rules based on their quality. This is important because providing different reward levels induces a different reward estimate for each type of rule, and provides the GA a fitness guidance. Without this guidance, the GA search would be reduced to a blind search among bad classifiers until it finds the optimal solution, like in a *needle in a haystack* problem. To calculate the reward one rule should receive, we first calculate the symmetric difference between the set of classes advocated by the rule and the set of correct classes for the current input instance. The symmetric difference between two sets A and B is the set of all x such that either $x \in A$ or $x \in B$, but not both. Afterwards, we calculate the size of this set and divide it by the number of classes in the problem. The reward is defined by subtracting 1 from this result. Now, partially correct classifiers receive a reward, but not as much reward as the optimal classifiers. Table 1 presents an example.

This reward scheme induces the need of a modification in the LT reputation value of an organization. Now, the LT reputation is calculated as the total amount of rewards receive by the organization during its lifetime divided by the number of times it was selected to affect the environment. Since the GA is turned off, there are no new classifiers added to the system during its learning process and therefore the goal of the system is to correctly learn and organize the initial rules.

Initially, each classifier was assigned to one organization. The LT reputation of the organizations was set to 0.95 and the ST reputation was set to 1. The reward estimate of the classifiers was set to 0.5 and the priority factor to 0. Every time a new organization is created by the Shrink operator its LT and ST reputations are initialized with the same initial values, i.e, 0.95 and 1, respectively. This is done to give the newly created organizations a chance to be evaluated.

Classifier	Organization	Reward est.	Priority factor
00/001	1	0.000066	-0.131043
01/010	2	0.000083	-0.200000
00/110	3	0.999998	2.951449
01/101	3	0.999998	2.940476
##/100	3	0.999998	1.408944
00/111	4	0.665129	-0.066666
00/100	4	0.666665	-0.133332

Table 2: Configuration of the rule set at the end of the run

Organization	Long-Term reputation
1	0.802632
2	0.500000
3	0.999987
4	0.833595

Table 3: Long-term reputation of all organizations at the end of the run

The percentage of the mean performance used by the Grow and Shrink operators were both defined as 85%. Constant C is set to 0.2, T_O and T_c to 0.01, F to 0.1, M to 1.5 and the time between organizational growth component application as 30 cycles, where a cycle is defined by presenting all the input instances to the system. We have run the system 30 times during 20000 cycles, and on 99% of these runs the system finished with a configuration similar to the one presented in Table 2. The table shows the classifiers, the organizations to which they were allocated and the classifiers' reward estimate and priority factor.

The system separates the classifiers in four organizations. The three optimal classifiers were placed on one organization, the two partially correct classifiers were placed on another organization and the two wrong classifiers were left alone in their own organizations. Table 3 presents the LT reputation of the four organizations formed. We can observe that the system correctly learned to select the best organization when more than one matches. Also, looking inside the organization that contains the three optimal classifiers and analyzing the classifiers' priority factor we see that the system build the correct default hierarchy between them.

These preliminary results show that the proposed system managed to learn the optimal rule set that solves a simple multi-label classification problem and also learned the hierarchical structure of this rule set. Moreover, it constructed an easily comprehensible solution, separating correct, partially correct and wrong rules in different organizations.

7. CONCLUSIONS

LCSs are a well known technique capable of adapting rule sets to improve their performance. Because of its capacity of learning control strategies, the LCSs theory has been widely studied and applied to a large set of problems, especially in the areas of Classification and Reinforcement Learning. The OCS considers the possible existence of variable sized organizations of classifiers in the population and, therefore, tackles problems with a mixture of individual and collective behavior. While the goal of the OCS was to better separate optimal from sub-optimal classifiers, Smith's theory on adaptive default hierarchy formation proposed a new

method to encourage the formation of hierarchical rule sets in an LCS, making LCSs more capable to adequately separate default from exception rules in a hierarchy.

The main focus of this paper was to bring these ideas back in a new formulation that uses both of them in the same classifier system, trying to solve multi-label classification. Building a system capable of good separation between the partially correct and correct rules brings the opportunity of solutions containing only optimal classifiers. Also, the ability to construct a hierarchical structure inside these groups of classifiers could prove useful to learn the possible correlations between the classes.

The primary conclusions of this study are that using reward estimate and separate priority factors at the classifier level, the system is capable of building default hierarchies among the classifiers inside an organization. Changing the ST reputation values for bids and using the reward estimate as LT reputation for classifiers did not change the system ability of adequately sizing its organizations. In fact, we believe that this is true because the classifier's LT reputation and the reward estimate are calculated in a very similar way.

The preliminary results obtained applying the system to a simple multi-label classification problem are encouraging. It was able to overcome the difficulties associated to the problem and managed to build a solution that adequately separates more general rules, which advocate less classes, from more specific rules advocating a larger set of classes.

However, a more detailed analysis needs to be carried out considering scalability and convergence of the system. Also, the inclusion of noise in the organizational growth component could make the system more robust to parameter setting. The influence of the time between organizational growth component application in the performance of the system also needs a more detailed analysis. Another important aspect that needs study is the inclusion of the GA and how it affects the system performance. The authors also plan to evaluate the proposed approach using real multi-label classification data sets and compare its performance on these data sets with other multi-label classification approaches proposed in the literature

Finally, although the system was build trying to resolve a specific group of classification problem, the preliminary results showed that it may be useful for any simple classification and reinforcement learning problem where solutions can be represented with a hierarchical rule set.

8. ACKNOWLEDGMENTS

The authors would like to thank the Brazilian research councils FAPESP and CNPq for their financial support. This work was sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant FA9550-06-1-0370, the National Science Foundation under ITR grant DMR-03-25939 at Materials Computation Center and under grant ISS-02-09199 at the National Center for Supercomputing Applications, UIUC. The U.S. Government is authorized to reproduce and distribute reprints for government purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Office of Scientific Research, the National Science Foundation, or the U.S. Government.

9. REFERENCES

- [1] F. Aioli, F. Portera, and A. Sperduti. Speeding up the solution of multilabel problems with Support Vector Machines. In *Supplementary Proc. of the Joint 13th International Conference on Artificial Neural Networks and 10th International Conference on Neural Information Processing*, pages 118–121, 2003.
- [2] J. Bacardit and J. M. Garrell. Bloat control and generalization pressure using the minimum description length principle for a pittsburgh approach learning classifier system. In *Proceedings of the 6th International Workshop on Learning Classifier Systems*, 2003.
- [3] E. Bernadó-Mansilla, X. Llorà, and I. Traus. Multiobjective Learning Classifier Systems: An Overview. Technical report, University of Illinois at Urbana Champaign, Urbana, IL, 2005.
- [4] P. Bonelli, A. Parodi, S. Sen, and S. W. Wilson. NEWBOOLE: A fast GBML System. In *Seventh International Conference on Machine Learning*, pages 153–159. Morgan Kaufmann, 1990.
- [5] M. Boutell, J. Luo, X. Shen, and C. Brown. Learning multi-label scene classification. *Pattern Recognition*, 37(9):17571771, 2004.
- [6] A. Chan and A. A. Freitas. A New Ant Colony Algorithm for Multi-Label Classification with Applications in Bioinformatics. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 27–34, 2006.
- [7] R. H. Coase. *The firm, the market, and the law*. University of Chicago Press, Chicago, 1988.
- [8] J. H. Holland. Processing and Processors for schemata, contributed chapter in *Associative Information Processing*, EL jacks editor, 127–146, 1971.
- [9] J. H. Holland. *Adaptation in Natural and Artificial Systems*. The MIT Press, 1975.
- [10] J. H. Holland and J. S. Reitman. Cognitive systems based on adaptive algorithms. *Pattern directed inference systems*, pages 313–329, 1978.
- [11] A. Karalic and V. Pirnat. Significance level based multiple tree classification. *Informatica*, 15(5), 1991.
- [12] X. Llorà, K. Sastry, and D. E. Goldberg. The Compact Classifier System: Scalability Analysis and First Results. Technical report, University of Illinois at Urbana-Champaign, Urbana, IL, 2005.
- [13] J. Mendel and R. McLaren. Reinforcement learning control and pattern recognition systems, *Adaptive, Learning and Pattern Recognition Systems: Theory and Applications*, 1970.
- [14] R. Smith and D. Goldberg. Adaptive Default Hierarchy Formation. *Applied Artificial Intelligence*, 6(1):79–102, 1992.
- [15] G. Tsoumakas and I. Katakis. Multi-Label Classification: An Overview. *International Journal of Data Warehousing and Mining*, 3:1–13, 2007.
- [16] G. Tsoumakas and I. Katakis. Multi-Label Classification: An Overview. *International Journal of Data Warehousing and Mining*, 3:1–13, 2007.
- [17] J. R. Wilcox. Organizational learning within a learning classifier system. Master's thesis, University of Illinois at Urbana-Champaign, 1995. Illigal Report No.95003.
- [18] S. W. Wilson. Classifier Fitness Based on Accuracy. *Evolutionary Computation*, 3(2):149–175, 1995.
- [19] M.-L. Zhang and Z.-H. Zhou. A k-nearest neighbor based algorithm for multi-label classification. In *Proceedings of the 1st IEEE International Conference on Granular Computing (GrC'05)*, pages 718–721, 2005.