Encyclopedia of Data Warehousing and Mining

Second Edition

John Wang
Montclair State University, USA

Volume III
K–Pri

Information Science REFERENCE

INFORMATION SCIENCE REFERENCE
Hershey • New York

Director of Editorial Content:	Kristin Klinger
Director of Production:	Jennifer Neidig
Managing Editor:	Jamie Snavely
Assistant Managing Editor:	Carole Coulson
Typesetter:	Amanda Appicello, Jeff Ash, Mike Brehem, Carole Coulson, Elizabeth Duke, Jen Henderson, Chris Hrobak, Jennifer Neidig, Jamie Snavely, Sean Woznicki
Cover Design:	Lisa Tosheff
Printed at:	Yurchak Printing Inc.

Published in the United States of America by
Information Science Reference (an imprint of IGI Global)
701 E. Chocolate Avenue, Suite 200
Hershey PA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@igi-global.com
Web site: http://www.igi-global.com/reference

and in the United Kingdom by
Information Science Reference (an imprint of IGI Global)
3 Henrietta Street
Covent Garden
London WC2E 8LU
Tel: 44 20 7240 0856
Fax: 44 20 7379 0609
Web site: http://www.eurospanbookstore.com

Library of Congress Cataloging-in-Publication Data

Encyclopedia of data warehousing and mining / John Wang, editor. -- 2nd ed.
p. cm.
Includes bibliographical references and index.
Summary: "This set offers thorough examination of the issues of importance in the rapidly changing field of data warehousing and mining"--Provided by publisher.
ISBN 978-1-60566-010-3 (hardcover) -- ISBN 978-1-60566-011-0 (ebook)
1. Data mining. 2. Data warehousing. I. Wang, John,
QA76.9.D37E52 2008
005.74--dc22
2008030801

British Cataloguing in Publication Data
A Cataloguing in Publication record for this book is available from the British Library.

All work contributed to this encyclopedia set is new, previously-unpublished material. The views expressed in this encyclopedia set are those of the authors, but not necessarily of the publisher.

Multi–Instance Learning with MultiObjective Genetic Programming

Amelia Zafra
University of Cordoba, Spain

Sebastián Ventura
University of Cordoba, Spain

INTRODUCTION

The multiple-instance problem is a difficult machine learning problem that appears in cases where knowledge about training examples is incomplete. In this problem, the teacher labels examples that are sets (also called bags) of instances. The teacher does not label whether an individual instance in a bag is positive or negative. The learning algorithm needs to generate a classifier that will correctly classify unseen examples (i.e., bags of instances).

This learning framework is receiving growing attention in the machine learning community and since it was introduced by Dietterich, Lathrop, Lozano-Perez (1997), a wide range of tasks have been formulated as multi-instance problems. Among these tasks, we can cite content-based image retrieval (Chen, Bi, & Wang, 2006) and annotation (Qi and Han, 2007), text categorization (Andrews, Tsochantaridis, & Hofmann, 2002), web index page recommendation (Zhou, Jiang, & Li, 2005; Xue, Han, Jiang, & Zhou, 2007) and drug activity prediction (Dietterich et al., 1997; Zhou & Zhang, 2007).

In this chapter we introduce MOG3P-MI, a multiobjective grammar guided genetic programming algorithm to handle multi-instance problems. In this algorithm, based on SPEA2, individuals represent classification rules which make it possible to determine if a bag is positive or negative. The quality of each individual is evaluated according to two quality indexes: sensitivity and specificity. Both these measures have been adapted to MIL circumstances. Computational experiments show that the MOG3P-MI is a robust algorithm for classification in different domains where achieves competitive results and obtain classifiers which contain simple rules which add comprehensibility and simplicity in the knowledge discovery process, being suitable method for solving MIL problems (Zafra & Ventura, 2007).

BACKGROUND

In the middle of the 1990's, Dietterich et al. (1997) described three Axis-Parallel Rectangle (abbreviated as APR) algorithms to solve the problem of classifying aromatic molecules according to whether or not they are "musky". These methods attempted to search the appropriate axis-parallel rectangles constructed by their conjunction of features. Their best performing algorithm (iterated-discrim) started with a point in the feature space and grew a box with the goal of finding the smallest box covered at least one instance from each positive bag and no instances from any negative bag. The resulting box was then expanded (via a statistical technique) to get better results.

Following Dietterich et al.'s study, a wide variety of new methods of multi-instance learning has appeared. Auer (1997) tried to avoid some potentially hard computational problems that were required by the heuristics used in the iterated-discrim algorithm and presented a theoretical algorithm, MULTINST. With a new approach, Maron and Lozano-Perez (1998) proposed one of the most famous multi-instance learning algorithms, Diverse Density (DD), where the diverse density of a point, p, in the feature space was defined as a probabilistic measure which considered how many different positive bags had an instance near p, and how far the negative instances were from p. This algorithm was combined with the Expectation Maximization (EM) algorithm, appearing as EM-DD (Zhang & Goldman, 2001). Another study that extended the DD algorithm to maintain multilearning regression data sets was the EM-based multi-instance regression algorithm (Amar, Dooly, Goldman, & Zhang, 2001).

In 1998, Long and Tan (1998) described a polynomial-time theoretical algorithm and showed that if instances in the bags were independently drawn from product distribution, then the APR was PAC-learnable. Following with PAC-learnable research, Kalai and Blum (1998) described a reduction from the problem of PAC-learning under the MIL framework to PAC-learning with one-sided random classification noise, and presented a theoretical algorithm with less complexity than the algorithm described in Auer (1997).

The first approaches using lazy learning, decision trees and rule learning were researched during the year 2000. In the lazy learning context, Whang and Zucker (2000) proposed two variants of the k nearest-neighbour algorithm (KNN) that they referred to as Citation-KNN and Bayesian-KNN; these algorithms extended the k-nearest neighbor algorithm for MIL adopting Hausdorff distance. With respect to decision trees and learning rules, Zucker and Chevaleyre (2000) implemented ID3-MI and RIPPER-MI, which are multi-instance versions of decision tree algorithm ID3 and rule learning algorithm RIPPER, respectively. At that time, Ruffo (2000) presented a multi-instance version of the C4.5 decision tree, which was known as RELIC. Later, Zhou et al. (2005) presented the Fretcit-KNN algorithm, a variant of Citation-KNN that modified the minimal Hausdorff distance for measuring the distance between text vectors and using multiple instance perspective. There are also many other practical multiple instance (MI) algorithms, such as the extension of standard neural networks to MIL (Zhang & Zhou, 2006). Also there are proposals about adapting Support Vector Machines to multi-instance framework (Andrews et al., 2002; Qi and Han, 2007) and the use of ensembles to learn multiple instance concepts, (Zhou & Zhang, 2007).

We can see that a variety of algorithms have been introduced to learn in multi-instance settings. Many of them are based on well-known supervised learning algorithms following works such as Ray and Craven's (2005) who empirically studied the relationship between supervised and multiple instance learning, or Zhou (2006) who showed that multi-instance learners can be derived from supervised learners by shifting their focuses from the discrimination on the instances to the discrimination on the bags. Although almost all popular machine learning algorithms have been applied to solve multiple instance problems, it is remarkable that the first proposals to adapt Evolutionary Algorithm to this scenario have not appeared until 2007 (Zafra, Ventura, Herrera-Viedma, & Romero 2007; Zafra & Ventura, 2007) even though these algorithms have been applied successfully in many problems in supervised learning.

MAIN FOCUS

Genetic Programming is becoming a paradigm of growing interest both for obtaining classification rules (Lensberg, Eilifsen, & McKee, 2006), and for other tasks related to prediction, such as characteristic selection (Davis, Charlton, Oehlschalager, & Wilson, 2006) and the generation of discriminant functions. The major considerations when applying GP to classification tasks are that a priori knowledge is not needed about the statistical distribution of the data (data distribution free). It can operate directly on the data in their original form, can detect unknown relationships that exist among data, expressing them as a mathematical expression and can discover the most important discriminating features of a class. We can find different proposals that use the GP paradigm to evolve rule sets for different classification problems, both two-class ones and multiple-class ones. Results show that GP is a mature field that can efficiently achieve low error rates in supervised learning, hence making it feasible to adapt to multiple instance learning to check its performance.

We propose, MOG3P-MI, a multiobjective grammar guided genetic programming algorithm. Our main motivations to introduce genetic programming into this field are: (a) grammar guided genetic programming (G3P) is considered a robust tool for classification in noisy and complex domains where it achieves to extract valuable information from data sets and obtain classifiers which contain simple rules which add comprehensibility and simplicity in the knowledge discovery process and (b) genetic programming with multiobjective strategy allows us to obtain a set of optimal solutions that represent a trade-off between different rule quality measurements, where no one can be considered to be better than any other with respect to all objective functions. Then, we could introduce preference information to select the solution which offers the best classification guarantee with respect to new data sets.

In this section we specify different aspects which have been taken into account in the design of the MOG3P-MI algorithm, such as individual representation, genetic operators, fitness function and evolutionary process.

Individual Representation

The choice of adequate individual representation is a very important step in the design of any evolutionary learning process to determine the degree of success in the search for solutions. In the proposed algorithm the representation is given by two components: a phenotype and a genotype. An individual phenotype represents a full classifier which is applied to bags. This classifier labels a bag as being a positive bag if it contains at least one instance which satisfies the antecedent, otherwise it is labelled as a negative bag. The representation has the structure shown in Figure 1.

The antecedent consists of tree structures and is applied to instances. It represents the individual genotype which can contain multiple comparisons attached by conjunction or disjunction according to a grammar to enforce syntactic constraints and satisfy the closure property (see Figure 2).

Genetic Operators

The elements of the following population are generated by means of two operators: mutation and crossover, designed to work in grammar guided genetic programming systems.

Mutation

The mutation operator randomly selects a node in the tree and the grammar is used to derive a new subtree which replaces the subtree in this node. If the new offspring is too large, it will be eliminated to avoid having invalid individuals. Figure 3 shows an example of this mutation.

Figure 1. Classifier applied to multi-instance learning

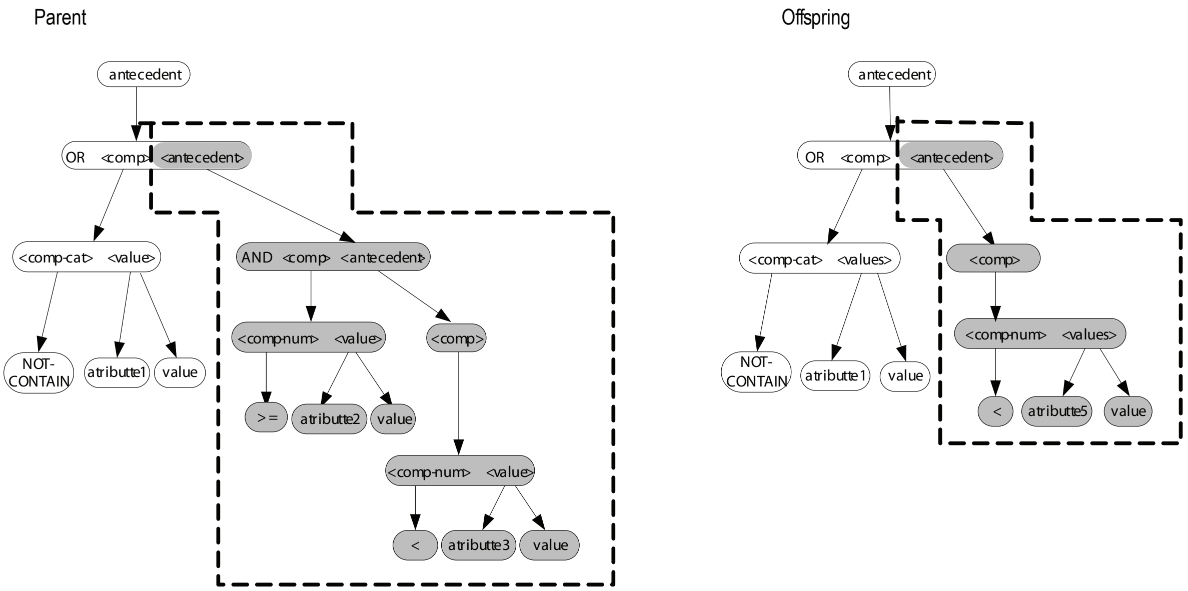
Cover$_{bag}$(bag$_i$) → *IF* $\exists_.$ instance$_j$ $\in$ bag$_i$ where Coverinstance(instance$_j$) is positive
THEN The bag is positive.
ELSE The bag is negative.

Coverinstance(instance$_i$) → *IF* (antecedent is satisfied by instance$_i$)
THEN The instance is positive.
ELSE The instance is negative.

Figure 2. Grammar used for individual representation

```
<antecedent> →          <comp>
                       | OR    <comp> <antecedent>
                       | AND <comp> <antecedent>

<comp> →       <comp-num>  <values>
              | <comp-cat>  <values>

<comp-num> →        <
                   | ≥

<comp-cat > →              CONTAIN
                           | NOT_CONTAIN

<values> →  attribute value
```

Figure 3. Example of mutation process



Crossover

The crossover is performed by swapping the sub-trees of two parents for two compatible points randomly selected in each parent. Two tree nodes are compatible if their subtrees can be swapped without producing an invalid individual according to the defined grammar. If any of the two offspring is too large, they will be replaced by one of their parents. Figure 4 shows an example of the crossover operator.

Fitness Function

The fitness function evaluates the quality of each individual according to two indices that are normally used to evaluate the accuracy of algorithms in supervised classification problems. These are sensitivity and specificity. Sensitivity is the proportion of cases correctly identified as meeting a certain condition and specificity is the proportion of cases correctly identified as not meeting a certain condition.

The adaptation of these measures to the MIL field needs to consider the bag concept instead of the instance concept. In this way, their expression would be:

$$specificity = \frac{t_n}{t_n + t_p}\ ;\ sensitivity = \frac{t_p}{t_p + f_n}$$

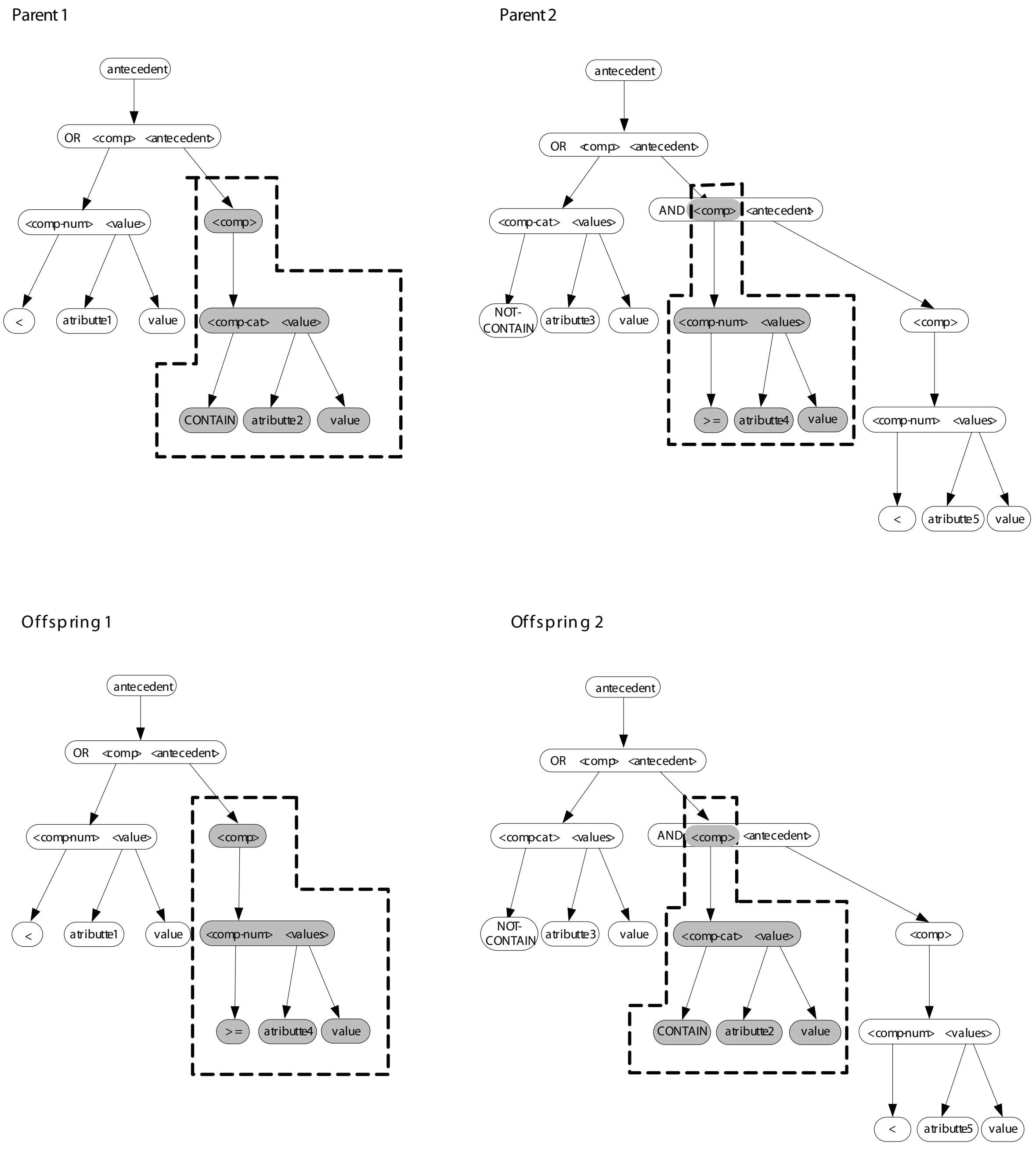
where *true positive* (t_p) represents the cases where the rule predicts that the bag has a given class and the bag does have that class. *True negative,* (t_n), are cases where the rule predicts that the bag does not have a given class, and indeed the bag does not have it. *False negative,* (f_n) cases are where the rule predicts that the bag does not have a given class but the bag does have it. Finally, *P*, is the number of positive bags and *N*, is the number of negative bags.

The evaluation involves a simultaneous optimization of these two conflicting objectives where a value of 1 in both measurements represents perfect classification. Normally, any increase in sensitivity will be accompanied by a decrease in specificity. Thus, there is no single optimal solution, and the interaction among different objectives gives rise to a set of compromised solutions, largely known as the Pareto-optimal solutions. Since none of these Pareto-optimal solutions can be identified as better than any others without further consideration, the goal is to find as many Pareto-optimal solutions as possible and include preference information to choose one of them as the final classifier.

Evolutionary Algorithm

The main steps of our algorithm are based on the well-known Strength Pareto Evolutionary Algorithm 2 (SPEA2). This algorithm was designed by Zitzler, Laumanns and Thiele (2001). It is a Pareto Front based multiobjective evolutionary algorithm that introduces some interesting concepts, such as an external elitist

Figure 4. Example of recombination process



set of non-dominated solutions, a fitness assignment schema which takes into account how many individuals each individual dominates and is dominated by, a nearest neighbour density estimation technique and a truncation method that guarantees the preservation of boundary solutions. The general outline of SPEA2 algorithm is shown in Figure 5.

FUTURE TRENDS

During these years, significant research efforts have been dedicated to MI learning and many approaches have been proposed to tackle MI problems. Although some very good results have been reported, the study of MI learning still requires topics which should be addressed. First, more datasets would have to be avail-

Figure 5. Main steps of MOG3P-MI algorithm

```
Algorithm MOG3P-MI

BEGIN
    Generate initial population of rules, $P_0$ and empty archive (external set) $A_0$. Set t = 0.

    DO
        Calculate fitness values of individuals in $P_t$ and $A_t$.
        $A_{t+1}$ = non-dominated individuals in $P_t$ and $A_t$.
        IF (size of $A_{t+1} > N$)
            Reduce $A_{t+1}$.
        ELSE IF (size of $A_{t+1} < N$)
            Fill $A_{t+1}$ ith dominated individuals in $P_t$ and $A_t$.
        END-IF

        Fill mating pool by binary tournament selection with replacement on $A_{t+1}$.
        Apply recombination and mutation operators to selected population, $P_{t+1}$.
        Set $P_{t+1}$ as resulting population. Set t = t + 1.

    UNTIL an acceptable classification rule is found or the specified maximum number of generations has
    been reached.

END
```

able for the purpose of evaluation because the lack of information about many of the MI problems tackled limits studies and comparisons with other developed methods. Secondly, studies are needed to establish a general framework for MI methods and applications. Recognizing the essence and the connection between different methods can sometimes inspire new solutions with well-founded theoretical justifications. Thirdly, with respect to our adaptation of the Genetic Programming paradigm, although it has shown excellent results, more optimization of this method is possible: issues such as the stopping criterion, the pruning strategy, the choice of optimal solutions and the introduction of new objectives for further simplification would be interesting issues for future work.

CONCLUSION

The problem of MIL is a learning problem which has drawn the attention of the machine learning community. We describe a new approach to solve MIL problems which introduces the Evolutionary Algorithm in this learning. This algorithm is called MOG3P-MI and it is derived from the traditional G3P method and SPEA2 multiobjective algorithm.

MOG3P-MI generates a simple rule-based classifier that increases generalization ability and includes interpretability and simplicity in the knowledge discovered. Computational experiments (Zafra & Ventura, 2007) show that the multiobjective technique applied to G3P is an interesting algorithm for learning from multiple instance examples, finding rules which maintain a tradeoff between sensitivity and specificity and obtaining the best results in terms of accuracy with respect to other existing learning algorithm.

REFERENCES

Amar, R. A., Dooly, D. R., Goldman, S. A., & Zhang, Q. (2001). Multiple-instance learning of real-valued data. *Proceedings of 18th International Conference on Machine Learning*, Massachusetts, USA, 28 June – 1 July, 3-10.

Andrews, S., Tsochantaridis, I., & Hofmann, T. (2002). Support vector machines for multiple-instance learning. *Proceedings of the 2002 Conference of the Neural Information Processing System* 15, Vancouver, Canada, 10-12 December, 561-568.

Auer, P. (1997). On Learning from Multi-Instance Examples: Empirical evaluation of a theoretical approach.

Proceedings of the 14th International Conference on Machine Learning, Nashville, Tennessee, USA, 8-12 July, 21-29.

Chen, Y., Bi, J., & Wang J. Z. (2006). MILES: Multiple-instance learning via embedded instance selection. *IEEE Transactions on Pattern Analysis and Machine Learning*, *28*(12), 1931-1947.

Davis, R. A., Charlton, A. J., Oehlschlager, S., & Wilson, J. C. (2006). Novel feature selection method for genetic programming using metabolomic 1H NMR data. *Chemometrics and Intelligent Laboratory Systems*, *81*(1), 50-59.

Dietterich, T. G., Lathrop, R. H., & Lozano-Perez, T. (1997). Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, *89*(1-2), 31-71.

Kalai, A., & Blum, A. (1998). A note on learning from multiple-instance examples. *Machine Learning*,*30*(1), 23-30.

Lensberg, T., Eilifsen, A., & McKee, T. E. (2006). Bankruptcy theory development and classification via genetic programming. *European Journal of Operational Research*, *169*(2), 677-697.

Long, P.M., & Tan, L. (1998). PAC learning axis-aligned rectangles with respect to product distributions from multiple-instance examples. *Machine Learning*, *30*(1), 7-21.

Maron, O., & Lozano-Perez, T. (1997). A framework for multiple-instance learning. *Proceedings of the 1997 Conference of the Neural Information Processing System 10*, Cambridge, MA, USA, 2-6 December, 570-576.

Qi, X., & Han, Y. (2007). Incorporating multiple svms for automatic image annotation. *Pattern Recognition*, *40*(2), 728-741.

Ray, S., & Craven, M. (2005). Supervised versus multiple instance learning: an empirical comparison. *Proceedings of the 22nd International Conference on Machine learning*, Bonn, Germany, 7-11 August, 697-704.

Ruffo, G. (2000). *Learning Single and Multiple Instance Decision Tree for Computer Security Applications*. PhD thesis, Department of Computer Science. University of Turin, Torino, Italy.

Wang, J., & Zucker, J. D. (2000). Solving the multiple-instance problem: A lazy learning approach. *Proceedings of the 17th International Conference on Machine Learning*, San Francisco, CA, USA, 29 June- 2 July, 1119-1126.

Xue, X, Han J., JianY., & Zhou .Z. (2007). Link recommendation in web index page based on multi-instance leaning techniques. *Computer Research and Development*, 44(3), 106-111.

Zafra, A., Ventura, S., Herrera-Viedma, E., & Romero, C. (2007). Multiple instance learning with genetic programming for web mining. *Proceedings of the 9th International Work-Conference on Artificial Neural Networks*, San Sebastian, Spain, 20-22 June, 919-927, *LNCS 4507*, Springer-Verlag.

Zafra, A., & Ventura, S. (2007). Multi-objective genetic programming for multiple instance learning. *Proceedings the 18th European Conference on Machine Learning*, Warsaw, Poland, 17-21 September, 790-797**,** *LNAI 4701*, Springer Verlag.

Zhang, Q., & Goldman, S. (2001). EM-DD: An improved multiple-instance learning technique. *Proceedings of the 2001 of the Conference of the Neural Information Processing System 14*, Vancouver, Canada, 3-8 December, (2001).

Zhang, M. L., & Zhou, Z. H. (2006). Adapting RBF neural networks to multi-instance learning. *Neural Processing Letters*, 23(1), 1-26.

Zhou, Z.H., Jiang, K., & Li, M. (2005). Multi-instance learning based web mining. *Applied Intelligence*, 22(2), 135-147.

Zhou, Z. H. (2006) Multi-instance learning from supervised view. *Journal Computer Science and Technology*, *21*(5), 800-809.

Zhou, Z.H., & Zhang, M.L. (2007). Solving multi-instance problems with classifier ensemble based on constructive clustering. *Knowledge and Information Systems*, *11*(2), 155-170.

Zucker, J.D., & Chevaleyre, Y. (2001). Solving multiple-instance and multiple-part learning problems with decision trees and decision rules. application to the mutagenesis problem. *Proceedings of the 14th Canadian Conference on Artificial Intelligence,* Ottawa, Canada, 7-9 June, 204-214, *LNAI 2056,* Springer-Verlag.

KEY TERMS

Evolutionary Algorithm (EA): They are search and optimization methodologies based on simulation models of natural selection, which begin with a set of potential solutions and then iteratively generate new candidates and select the fittest from this set. It has been successfully applied to numerous problems from different domains, including optimization, automatic programming, machine learning, economics, ecology, studies of evolution and learning, and social systems.

Genetic Programming (GP): An Evolutionary Algorithm that provides a flexible and complete mechanism for different tasks of learning and optimization. Its main characteristic is that it uses expression tree-based representations or functional program interpretation as its computational model.

Grammar Guided Genetic Programming (G3P): An Evolutionary Algorithm that is used for individual representation grammars and formal languages. This general approach has been shown to be effective for some natural language learning problems, and the extension of the approach to procedural information extraction is a topic of current research in the GP community.

Multi-instance Learning (MIL): It is proposed as a variation of supervised learning for problems with incomplete knowledge about labels of training examples. In MIL the labels are only assigned to *bags of instances*. In the binary case, a bag is labeled positive if *at least* one instance in that bag is positive, and the bag is labeled negative if *all* the instances in it are negative. There are no labels for individual instances. The goal of MIL is to classify unseen bags or instances based on the labeled bags as the training data.

Multiobjective Optimization Problem (MOP): The problem consists of simultaneously optimizing vector functions which maintain conflicting objectives subject to some constrained conditions.

Multiobjective Evolutionary Algorithms (MOEAs): A set of Evolutionary Algorithms suitable for solving multiobjective problems. These algorithms are well suited to multiobjective optimization problems because they are fundamentally based on biological processes which are inherently multiobjective. Multiobjective Evolutionary Algorithms are able to find optimal trade-offs in order to get a set of solutions that are optimal in an overall sense.

Strength Pareto Evolutionary Algorithm 2 (SPEA2): It is an elitist Multiobjective Evolutionary Algorithm. It is an improved version of the Strength Pareto Evolutionary Algorithm (SPEA) which incorporates a fine-grained fitness assignment strategy, a density estimation technique, and an enhanced archive truncation method. SPEA2 operates with a population (archive) of fixed size, from which promising candidates are drawn as parents of the next generation. The resulting offspring then compete with the former ones for inclusion in the population.

Supervised Learning: A machine learning technique for creating a model from training data where every training instance is assigned a discrete or real-valued label. The task of the supervised learner is to classify unseen instances based on the labelled instances of training data.