Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 176 (2020) 653–659

Procedia Computer Science

www.elsevier.com/locate/procedia

24th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems

Representation of Knowledge by Decision Trees for Decision Tables with Multiple Decisions

Mohammad Azad[a,*], Igor Chikalov[b], Mikhail Moshkov[c]

[a] *Jouf University, College of Computer and Information Sciences, Sakaka 72441, Saudi Arabia*
[b] *Intel Corporation, 5000 W Chandler Blvd, Chandler, AZ 85226, USA*
[c] *King Abdullah University of Science and Technology, Computer, Electrical and Mathematical Sciences and Engineering Division, Thuwal 23955-6900, Saudi Arabia*

Abstract

In this paper, we study decisions trees for decision tables with multiple decisions as a means for knowledge representation. To this end, we consider three methods to design decision trees and evaluate the number of nodes, and local and global misclassification rates of constructed trees. The considered methods are based on a dynamic programming algorithm for bi-objective optimization of decision trees. The goal of this study is to construct trees with reasonable number of nodes and at the same time reasonable accuracy. Previously, it was mentioned that the consideration of only the global misclassification rate of the decision tree is not enough and it is necessary to study also the local misclassification rate. The reason is that even if the global misclassification rate related to the whole tree is enough small, the local misclassification rate related to the terminal nodes of the tree can be too big. One of the considered methods allows us to construct the decision trees with moderate number of nodes as well as moderate global and local misclassification rates. These decision trees can be used for the knowledge representation.

Keywords: Knowledge representation; Decision trees; Decision tables with multiple decisions; Bi-objective optimization; Pareto optimal points

1. Introduction

In conventional decision tables, a single decision (or label) is attached to each object (row). However, there are situations where a set of decisions is associated with each row like in semantic annotation of images [4], music categorization into emotions [19], functional genomics [18], and text categorization [20].

* Corresponding author.
E-mail address: mmazad@ju.edu.sa

10.1016/j.procs.2020.09.037

In decision tables containing experimental data, we can meet often groups of equal rows with, probably, different decisions. Instead of such a group, we can keep one row from the group and label it with the set of decisions attached to rows from the group [1, 2].

The rough sets theory [14, 15] works with inconsistent decision tables that have equal rows with different decisions. The set of decisions attached to equal rows is called the generalized decision for each of these equal rows [14, 15]. Usually, the aim is to find for a given row its generalized decision. In some situations, the problem of finding an arbitrary decision from the generalized one has also sense.

Decision trees are widely used as classifiers, as a means of knowledge representation, and as algorithms [5, 12, 16]. Depending on the goal, we can (i) minimize time complexity of decision trees (depth or average depth), or (ii) minimize space complexity of decision trees (number of nodes, or number of terminal/nonterminal nodes), or (iii) maximize accuracy of decision tree (by minimizing the number of misclassifications).

Previously, we created a dynamic programming approach to bi-objective optimization of decision trees for decision tables with multiple decisions relative to the number of nodes and the number of misclassifications [1, 2]. Bi-objective optimization is vital since we often need to optimize simultaneously two criteria. In literature, it is also mentioned in different names like multiple objective discrete optimization, multiple criteria decision making, multiple objective combinatorial optimization [8, 11], etc.

This paper is devoted to the investigation of decision tables with multiple decisions in which each row is labeled with a set of decisions. Our aim is to find a decision from the set attached to a given row. We concentrate on the consideration of decision trees as a means of knowledge representation. To be understandable, the decision tree should have a smaller number of nodes. To represent properly knowledge from the decision table, the decision tree should have a high accuracy (i.e., minimum number of misclassifications). We have shown previously [3] that the consideration of only the global misclassification rate is not enough. It is because the misclassifications may not be uniformly distributed. Also, for some nonterminal nodes, the fraction of misclassifications can be big enough. The solution is that we should consider also the local misclassification rate. Therefore, in [3], we designed three methods for decision tree construction which are applicable to medium-sized decision tables with single-valued decisions and allowed us to work with both rates.

In this paper, we extended these three methods to the case of decision tables with multiple decisions. We apply the considered algorithms to ten modified decision tables from the UCI ML Repository [7] which are obtained after applying some procedures. For the constructed decision trees, we study three parameters mentioned in [3].

The obtained results show that at least one of the considered algorithms (*GL*-method) can be useful for the extraction of knowledge from medium-sized decision tables and for its representation by decision trees. This algorithm can be used in different areas of data analysis including rough set theory [14, 15]. One of the limitations of our approach is that the time complexity of the algorithms in the worst case is exponential.

The rest of the paper is organized as follows. In Sect. 2, we discuss briefly the literature regarding decision tables with multiple decisions. In Sect. 3, we define the notion of a decision table with multiple decisions and also the notion of a decision tree for this type of tables. In Sect. 4, we describe three methods for decision tree construction. In Sect. 5, we discuss results of experiments with modified decision tables from the UCI ML Repository [7]. Section 6 contains short conclusions.

2. Related Work

If we look at the literature, we often meet papers related to the study of prediction problems for decision tables with multiple decisions, i.e., multi-label learning [17] and multi-instance learning [21]. We should mention also semi-supervised learning [22], partial learning [6], ambiguous learning [9], and multiple label learning [10].

We can meet decision tables with multiple decisions in studies related to the decision trees as algorithms: (i) in combinatorial optimization (for example, for traveling salesman problem), (ii) in computational geometry (for example, for nearest post office problem), (iii) in diagnosis of faults in circuits [1, 2, 13], etc.

In this paper, we focus on the problem of knowledge representation.

3. Decision Tables with Multiple Decisions and Decision Trees

A decision table with multiple decisions is a table T filled with nonnegative integers. Columns of this table are labeled with conditional attributes. Rows of the table are pairwise different and they are labeled with nonempty finite sets of natural numbers that are interpreted as sets of decisions. An example of a decision table with multiple decisions T_0 is given in Fig. 1.

$T_0 =$

f_1	f_2	
0	0	$\{1,2\}$
0	1	$\{3\}$
1	0	$\{1,3\}$
1	1	$\{2\}$

Fig. 1. Decision table T_0 with multiple decisions

A decision tree for the table T is a finite directed tree with root in which nonterminal nodes are labeled with conditional attributes, terminal nodes are labeled with nonnegative integers, and edges starting in each nonterminal node are labeled with pairwise nonnegative integers. An example of exact decision tree for the decision table T_0 is given in Fig. 2.

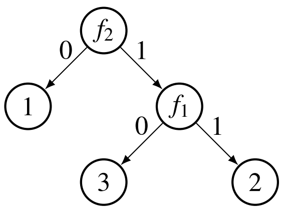


Fig. 2. Exact decision tree for decision table T_0

In this paper, we study not only exact but also approximate decision trees. Let T be a decision table with multiple decisions and Γ be a decision tree for the table T. We consider the following three parameters of the tree Γ:

- $N = N(\Gamma)$ – the number of nodes in Γ.
- $G = G(T,\Gamma)$ – the global misclassification rate. We count the number of misclassifications of Γ on T and divide it by the number of rows in T.
- $L = L(T,\Gamma)$ – the local misclassification rate. For each terminal node, we count the number of misclassifications among rows accepted by this node, divide it by the number of all rows accepted by the node, and consider the maximum of the obtained value among all terminal nodes. One can prove that $G(T,\Gamma) \leq L(T,\Gamma)$.

4. Design of Decision Trees

In this section, we use the following framework to work with decision trees. The first step is to create a directed acyclic graph (DAG) for the given decision table with multiple decisions, nodes of which are some subtables of the considered decision table. For this purpose, we use the algorithm $\mathcal{A}_1$ from the book [1]. The considered DAG for the decision table T_0 can be found in Fig. 3. After that, we use the algorithm $\mathcal{A}_7$ from [1] which builds the set of Pareto optimal points (POPs) relative to two criteria for each node (subtable) of the constructed DAG. Next, for each POP, we can derive a decision tree with values of the considered parameters equal to the coordinates of this point.

In this paper, we use the algorithm $\mathcal{A}_7$ to solve two bi-objective optimization problems: relative to the parameters N and G (see examples in Fig. 4 (a), (c), (e), (g)) and relative to the parameters N and L (see examples in Fig. 4 (b), (d), (f), (h)).

We now describe three methods to design decision trees using the algorithm $\mathcal{A}_7$. We assume that the DAG for the considered decision table is already created by the algorithm $\mathcal{A}_1$. Note that the time complexity of the algorithms $\mathcal{A}_1$

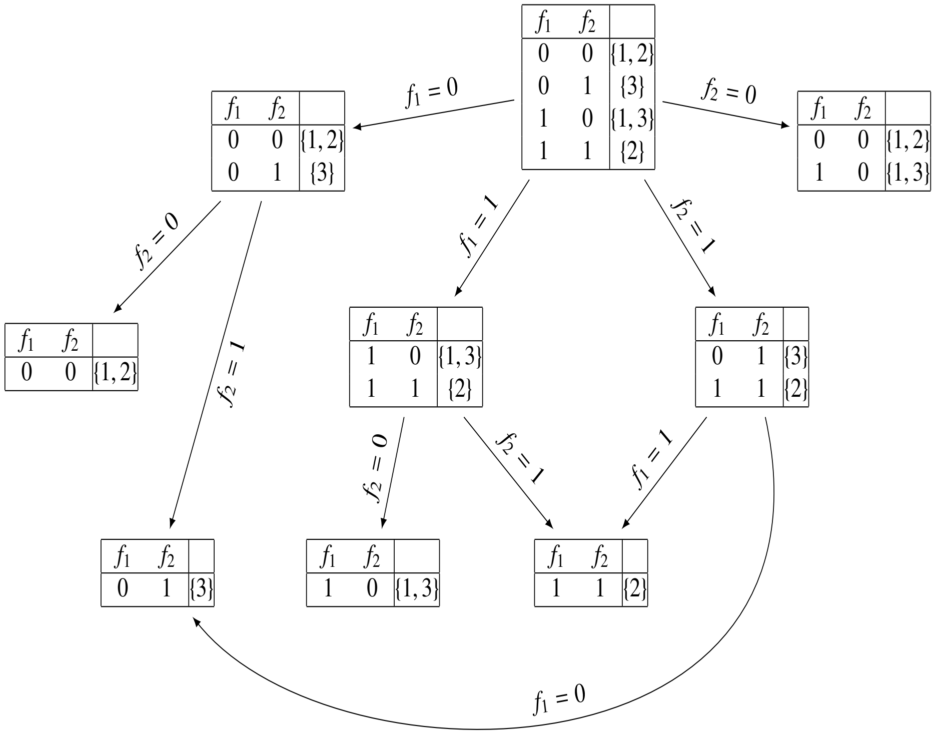


Fig. 3. DAG for decision table T_0

and $\mathcal{A}_7$ is exponential in the worst case depending on the size of the decsision tables. Therefore, the following three methods have exponential time complexity in the worst case as well.

4.1. G-method

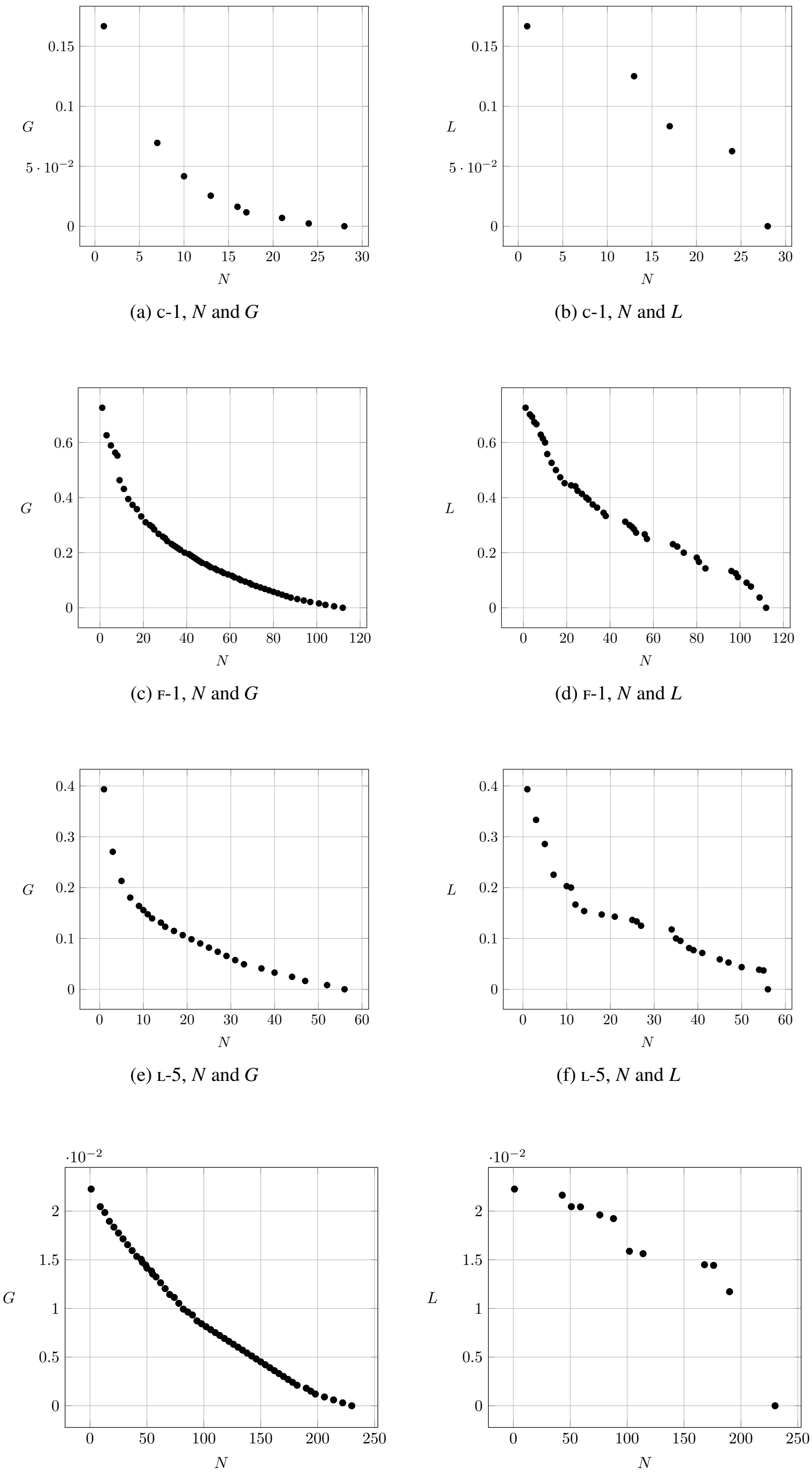
For a given decision table T, we construct by the algorithms $\mathcal{A}_1$ and $\mathcal{A}_7$ the set of POPs for the parameters N and G. We normalize coordinates of POPs and choose a normalized POP with the minimum Euclidean distance from the origin. After that, we derive a decision tree, for which the values of the parameters N and G are equal to the initial coordinates of this POP. This decision tree is the output of G-method.

4.2. L-method

L-method works in the same way as G-method but instead of the parameters N and G it uses the parameters N and L.

4.3. GL-method

We apply G-method to the decision table T and construct a decision tree Γ_1. After that, using the algorithm $\mathcal{A}_7$ we construct the set of POPs for the parameters N and L, and choose a POP for which the value of the coordinate N is closest to $N(\Gamma_1)$. At the end, we derive a decision tree Γ_2 for which the values of the parameters N and L are equal to the coordinates of the chosen POP. The tree Γ_2 is the output of GL-method.



(a) c-1, *N* and *G*

(b) c-1, *N* and *L*

(c) f-1, *N* and *G*

(d) f-1, *N* and *L*

(e) l-5, *N* and *G*

(f) l-5, *N* and *L*

(g) p-5a, *N* and *G*

(h) p-5a, *N* and *L*

Fig. 4. Sets of POPs for decision tables c-1, f-1, l-5, and p-5a for pairs of parameters *N*, *G* and *N*, *L*

5. Results of Experiments

We made experiments with ten modified decision tables from the UCI ML Repository [7] described in Table 1. For a table from [7], which name can be found in the column 'Original data set - removed columns', we fill missing values and remove some conditional attributes. Indices of the removed attributes can be found in the mentioned column. The obtained inconsistent decision table is transformed into a decision table with multiple decisions. The name of the new decision table is mentioned in the column 'Decision table T'. Columns 'Rows', 'Attr', and 'Dec' gives us information about the number of rows, attributes, and decisions in the new table. The number $\#i$ (see the column 'Spectrum') is equal to the number of rows for which the set of decisions has cardinality i.

Decision table T	Original data set - removed columns	Rows	Attr	Dec	Spectrum #1, #2, #3 , ...
C-1	CARS -1	432	5	4	258, 161, 13
F-4	FLAGS -1,2,3,19	176	22	6	168, 8
F-3	FLAGS -1,2,3	184	23	6	178, 6
F-1	FLAGS -1	190	25	6	188, 2
L-5	LYMPHOGRAPHY -1,13,14,15,18	122	13	4	113, 9
N-4	NURSERY -1,5,6,7	240	4	5	97, 96, 47
P-5A	POKER-HAND -1,2,4,6,8	3324	5	10	128, 1877, 1115
P-5B	POKER-HAND -2,3,4,6,8	3323	5	10	130, 1850, 1137
Z-5	ZOO-DATA -2,6,8,9,13	43	11	7	40, 1, 2
Z-2	ZOO-DATA -6,13	46	14	7	44, 2

Table 1. Decision tables used in experiments

Decision	G-method			L-method			GL-method		
table T	N	G	L	N	G	L	N	G	L
C-1	10	0.04	0.17	17	0.01	0.08	13	0.03	0.13
F-4	48	0.27	0.50	56	0.28	0.33	45	0.32	0.38
F-3	51	0.26	0.45	57	0.29	0.33	51	0.29	0.38
F-1	31	0.43	0.50	38	0.25	0.33	30	0.28	0.39
L-5	15	0.12	0.50	14	0.13	0.15	14	0.13	0.15
N-4	4	0.01	0.03	4	0.01	0.03	4	0.01	0.03
P-5A	94	0.01	0.06	102	0.01	0.02	88	0.01	0.02
P-5B	86	0.01	0.06	86	0.01	0.02	86	0.01	0.02
Z-5	7	0.19	0.45	9	0.12	0.25	7	0.21	0.38
Z-2	7	0.24	0.54	7	0.26	0.33	7	0.26	0.33
Average	35.3	0.16	0.33	39	0.14	0.19	34.5	0.16	0.22

Table 2. Results of experiments

We applied G-method, L-method, and GL-method to each of these tables and found values of the parameters N (number of nodes), G (global misclassification rate), and L (local misclassification rate) for the constructed decision trees – see results in Table 2.

Decision trees constructed by G-method have overall moderate values of the parameters N and G but often have high values of the parameter L.

Decision trees constructed by L-method have overall moderate values of the parameters G and L but enough high value of the parameter N on average.

Decision trees constructed by GL-method have overall moderate values of the parameters N, G, and L. We can use GL-method to construct enough understandable and accurate decision trees.

6. Conclusions

In this paper, we studied decision trees constructed by three methods (G-method, L-method, and GL-method) based on the dynamic programming algorithms for bi-objective optimization and evaluated not only the global misclassification rate of the constructed decision trees but also their local misclassification rate. We found that the decision trees constructed by GL-method have moderate number of nodes as well as moderate global and local misclassification rates. The decision trees constructed by this method can be used for the knowledge representation. Later, we are planning to extend the considered technique to the case of decision tables containing numerical attributes. Furthermore, we are planning to study partial DAGs with limiting number of branches. It will reduce the time complexity of the algorithms.

Acknowledgments

Research reported in this publication was supported by Jouf University and by King Abdullah University of Science and Technology (KAUST). The authors are greatly indebted to the anonymous reviewers for useful comments.

References

[1] Alsolami, F., Azad, M., Chikalov, I., Moshkov, M., 2020. Decision and Inhibitory Trees and Rules for Decision Tables with Many-valued Decisions. volume 156 of *Intelligent Systems Reference Library*. Springer.
[2] Azad, M., 2018. Decision and Inhibitory Trees for Decision Tables with Many-Valued Decisions. Ph.D. thesis. King Abdullah University of Science & Technology. Thuwal, Saudi Arabia. URL: http://hdl.handle.net/10754/628023, doi:10.25781/KAUST-E3NLJ.
[3] Azad, M., Chikalov, I., Moshkov, M., 2019. Decision trees for knowledge representation, in: Ropiak, K., Polkowski, L., Artiemjew, P. (Eds.), 28th International Workshop on Concurrency, Specification and Programming, CS&P 2019, Olsztyn, Poland, September 24–26, 2019, CEUR-WS.org. URL: http://ceur-ws.org/Vol-2571/CSP2019_paper_1.pdf.
[4] Boutell, M., Luo, J., Shen, X., Brown, C., 2004. Learning multi-label scene classification. Pattern Recognition 37, 1757–1771.
[5] Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J., 1984. Classification and Regression Trees. Wadsworth and Brooks.
[6] Cour, T., Sapp, B., Jordan, C., Taskar, B., 2009. Learning from ambiguously labeled images, in: 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2009, Miami, Florida, USA, June 20–25, 2009, IEEE Computer Society. pp. 919–926.
[7] Dua, D., Graff, C., 2017. UCI Machine Learning Repository. University of California, Irvine, School of Information and Computer Sciences. URL: http://archive.ics.uci.edu/ml.
[8] Ehrgott, M., Gandibleux, X., 2000. A survey and annotated bibliography of multiobjective combinatorial optimization. OR Spectr. 22, 425–460.
[9] Hüllermeier, E., Beringer, J., 2005. Learning from ambiguously labeled examples, in: Famili, A.F., Kok, J.N., Sánchez, J.M.P., Siebes, A., Feelders, A.J. (Eds.), Advances in Intelligent Data Analysis VI, 6th International Symposium on Intelligent Data Analysis, IDA 2005, Madrid, Spain, September 8–10, 2005. Springer. volume 3646 of *Lecture Notes in Computer Science*, pp. 168–179.
[10] Jin, R., Ghahramani, Z., 2002. Learning with multiple labels, in: Becker, S., Thrun, S., Obermayer, K. (Eds.), Advances in Neural Information Processing Systems 15, NIPS 2002, Vancouver, British Columbia, Canada, December 9–14, 2002, MIT Press. pp. 897–904.
[11] Kouvelis, P., Sayın, S., 2006. Algorithm robust for the bicriteria discrete optimization problem. Ann. Oper. Res. 147, 71–85.
[12] Moshkov, M., 2005. Time complexity of decision trees, in: Peters, J.F., Skowron, A. (Eds.), Trans. Rough Sets III. Springer. volume 3400 of *Lecture Notes in Computer Science*, pp. 244–459.
[13] Moshkov, M., Zielosko, B., 2011. Combinatorial Machine Learning - A Rough Set Approach. volume 360 of *Studies in Computational Intelligence*. Springer.
[14] Pawlak, Z., 1991. Rough Sets - Theoretical Aspect of Reasoning About Data. Kluwer Academic Publishers.
[15] Pawlak, Z., Skowron, A., 2007. Rudiments of rough sets. Inf. Sci. 177, 3–27.
[16] Rokach, L., Maimon, O., 2008. Data Mining with Decision Trees: Theory and Applications. World Scientific Publishing.
[17] Tsoumakas, G., Katakis, I., 2007. Multi-label classification: An overview. IJDWM 3, 1–13.
[18] Vens, C., Struyf, J., Schietgat, L., Džeroski, S., Blockeel, H., 2008. Decision trees for hierarchical multi-label classification. Mach. Learn. 73, 185–214.
[19] Wieczorkowska, A., Synak, P., Lewis, R., Ras, Z., 2005. Extracting emotions from music data, in: Hacid, M., Murray, N., Ras, Z., Tsumoto, S. (Eds.), 15th International Conference on Foundations of Intelligent Systems, ISMIS 2005, Saratoga Springs, NY, USA, May 25–28, 2005, Springer. pp. 456–465.
[20] Zhou, Z.H., Jiang, K., Li, M., 2005. Multi-instance learning based web mining. Appl. Intell. 22, 135–147.
[21] Zhou, Z.H., Zhang, M.L., Huang, S.J., Li, Y.F., 2012. Multi-instance multi-label learning. Artif. Intell. 176, 2291–2320.
[22] Zhu, X., Goldberg, A.B., 2009. Introduction to Semi-Supervised Learning. Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan & Claypool Publishers.