"Missing is Useful": Missing Values in Cost-sensitive Decision Trees [1]

Shichao Zhang, *senior Member, IEEE*

Zhenxing Qin, Charles X. Ling, Shengli Sheng

Abstract. Many real-world datasets for machine learning and data mining contain missing values, and much previous research regards it as a problem, and attempts to impute missing values before training and testing. In this paper, we study this issue in cost-sensitive learning that considers both test costs and misclassification costs. If some attributes (tests) are too expensive in obtaining their values, it would be more cost-effective to miss out their values, similar to skipping expensive and risky tests (missing values) in patient diagnosis (classification). That is, "missing is useful" as missing values actually reduces the total cost of tests and misclassifications, and therefore, it is not meaningful to impute their values. We discuss and compare several strategies that utilize only known values and that "missing is useful" for cost reduction in cost-sensitive decision tree learning.

[1] This work is partially supported by Australian large ARC grants (DP0343109 and DP0559536), a China NSFC major research Program (60496321), and a China NSFC grant (60463003).

- Shichao Zhang is with the Department of Computer Science at Guangxi Normal University, Guilin, China; and with the Faculty of Information Technology at University of Technology Sydney, PO Box 123, Broadway, Sydney, NSW 2007, Australia; zhangsc@ it.uts.edu.au.
- Zhenxing Qin is with the Faculty of Information Technology at University of Technology Sydney, PO Box 123, Broadway, Sydney, NSW 2007, Australia; zqin@ it.uts.edu.au.
- Charles X. Ling, Shengli Sheng are with the Department of Computer Science at The University of Western Ontario, London, Ontario N6A 5B7, Canada; {cling, ssheng}@ csd.uwo.ca.

1. Introduction

Machine learning and data mining reply heavily on a large amount of data to build learning models and make predictions, and thus, the quality of data is ultimately important. Though there is no formal measure on the quality of data, it can be intuitively quantified by the inclusion of relevant attributes, the errors in attribute values, and the amount of missing values in datasets. This paper studies the issue of missing attribute values in training and test datasets.

Indeed, many real-world datasets contains missing values, and it is often regarded as a difficult problem to cope with. Sometimes values are missing due to unknown reasons, or errors and omissions when data are recorded and transferred. As many statistical and learning methods cannot deal with missing values directly, examples with missing values are often deleted. However, deleting cases can result in a loss of a large amount of valuable data. Thus much previous research has focused on filling or imputing the missing values before learning and testing is applied to.

In this paper, we study missing data in cost-sensitive learning in which both misclassification costs and test costs are considered. That is, there is a known cost associated with each attribute (variable or test) when obtaining its values. This is true in most real-world application where it costs money to obtain new information. For example, in medical diagnosis, it costs money (to the patient, lab, or health insurance) to request blood tests, X-ray, or other types of tests, some of which can be quite expensive and even risky to patient life (which can also be converted to cost). Doctors often have to balance the cost effectiveness of the tests and the accuracy of the diagnosis (prediction) to decide what tests should be performed. That is, if a test is too expensive compared to the potential reduction in misclassification cost, it is desirable to skip the

test. In other words, if the goal is to minimize the total cost of tests and misclassifications, some attribute values *should* be missing, and doctors did not need to know the missing values in their diagnosis (prediction or classification).

Thus, cost-sensitive learning algorithms should make use of only known values. Of course, the learners may not know exactly how the known values were acquired – were all of them necessary for prediction? In any case, we can assume that the known values may be useful for prediction, but the unknown values are certainly not. Thus, under cost-sensitive learning, there is no need to impute values of any missing data, and the learning algorithms should make use of only known values and that "missing is useful" to minimize the total cost of tests and misclassifications.

The rest of the paper is organized as follows. In Section 2 we review previous techniques for dealing with missing values, and a recent cost-sensitive decision tree algorithm based on which we will discuss our missing-value strategies. We will discuss and compare four missing-value strategies that utilize only known data in Section 3. We experimentally compare the four strategies using real-world datasets in Section 4. Our conclusions and future work occupy Section 5.

2. Review of Previous Work

The issue of missing values (or missing data) has been studied extensively in the statistical and machine learning literature. According to the missing data mechanisms, statisticians have identified three classes of missing data [16]: *missing completely at random* (*MCAR*), *missing at random (MCR),* and *not missing at random (NMAR)*. MCAR is when the probability of missing a value is the same for all variables; MCR is when the probability of missing a value is only dependent on other variables; and NMAR is when the probability of missing a value is also dependent on the value of the missing variable. MCR has received most attentions, for which various "imputation" methods have been designed to predict the missing values before building models. In machine learning, the missing value issue has been dealt with mostly in decision tree learning and rule learning. Various imputation methods have also been tried, such as imputation by the most common value [6], clustering [7], and other learning models [2]. In C4.5 [19, 20] a different approach is used in which a test example with missing values is distributed into branches probabilistically (see Section 3.4). Comparison of various imputation methods has also been published [15]. The approaches we discuss in this paper do not impute any missing values, as it is regarded as unnecessary for cost-sensitive learning that also considers the test costs.

This paper deals with missing values in cost-sensitive learning. Turney [22] presents an excellent survey on different types of costs in cost-sensitive learning, among which misclassification costs and test costs are singled out as most important. Much work has been done in recent years on non-uniform misclassification costs (alone), such as [9, 10 and 14]. Some previous work, such as [18, 21], considers the test cost alone without incorporating misclassification cost, which is obviously an oversight. A few

researchers [5, 13, 23, 24] consider both misclassification and test costs, but their methods are less computationally efficient as our approach is based on decision trees. Ling et al. [17] propose a decision-tree learning algorithm that uses minimum total cost of tests and misclassifications as the attribute split criterion, and it is the basis of the four missing-value strategies to be presented in Section 3. Basically, given a set of training examples, the total cost without further splitting and the total cost after splitting on an attribute can be calculated, and the difference of the two is called cost reduction. The attribute with the maximum, positive cost reduction is chosen for growing the tree. All examples with missing values of an attribute stay at the internal node of that attribute. The method produces decision trees with the minimal total cost of tests and misclassifications on the training data [17].

In the next Section we will discuss several different missing-value strategies, all of which use the maximum cost reduction strategy described above to build cost-sensitive decision trees.

3. Dealing with Missing Values in Cost-sensitive Decision Trees

As we discussed in the Introduction, in cost-sensitive learning which attempts to minimize the total cost of tests and misclassifications, missing data can be useful for cost reduction, and imputing missing values should be unnecessary. Thus, cost-sensitive decision tree learning algorithms should utilize only known values. In the following subsections we will describe four such missing-value techniques. These strategies have been proposed previously but their performance in cost-sensitive learning has not been studied. In Section 4 we will perform empirical experiments to compare the four strategies on real-world datasets by the total cost.

3.1 The Known Value Strategy

The first tree building and test strategy for "missing is useful" is called the Known Value Strategy. It utilizes only the known attribute values in the tree building for each test example. For each test example, a new (and probably different) decision tree is built from the training examples with only those attributes whose values are known in the test example. That is, the new decision tree only uses attributes with known values in the test example, and thus, when the tree classifies the test example, it will never encounter any missing values.

The Known Value Strategy was proposed in [17] but its ability of handling unknown values was not studied. Clearly, the strategy utilizes all known attributes and avoids any missing data directly. It is a lazy tree method [12] where a tree is built during test process. The main drawback of the Known Value Strategy is its relatively high computation cost as different trees may be built for different test examples. This is usually not a problem as the tree building process is very efficient. In addition, we can save frequent trees and use them directly in testing for test examples with the same subsets of known attributes, because decision trees for the same subsets of known attributes are the same. We can use space to trade-off the speed if necessary.

3.2 The Null Strategy

As values are missing for a certain reason – unnecessary and too expensive to test – it might be a good idea to assign a special value, often called "null" in databases [8], to missing data. The null value is then treated just as a regular known value in the tree building and test processes. This strategy has also been proposed in machine learning [1], but its ability in cost-sensitive learning has not been studied.

One potential problem with the Null Strategy is that it does not deliberately utilize the original known values, as missing values are treated as equally as a known value. Another potential drawback is that there might be more than one situation where values are missing. Replacing all missing values by one value (null) may not be adequate. In addition, subtrees can be built under the "null" branch, suggesting oddly that the unknown is more discriminating than known values. The advantage of this strategy is its simplicity and high efficiency compared to the Known Value Strategy, as only one decision tree is built for all test examples.

3.3 The Internal Node Strategy

This strategy, as proposed in [17] and reviewed in Section 2, keeps examples with missing values in internal nodes, and does not build branches for them during tree building. When classifying a test example, if the tree encounters an attribute whose value is unknown, then the class probability of training examples falling at the internal node is used to classify it. As unknown values are dealt with by internal nodes, we call this strategy the Internal Node Strategy.

As there might be several different situations where values are missing, leaving the classification to the internal nodes may be a natural choice. This strategy is also quite efficient as only one tree is built for all test examples.

3.4 The C4.5 Strategy

C4.5 [19, 20] does not impute missing values explicitly, and it is shown to be quite effective [4]. Here C4.5's missing-value strategy is applied directly in cost-sensitive trees. During training, an attribute is chosen by the maximum cost reduction discounted by the probability of missing values of that attribute. During testing, a test example with

missing value is split into branches according to the portions of training examples falling into those branches, and goes down to leaves simultaneously. The class of the test example is the weighted classification of all leaves.

4. Experiment Comparisons

In this section we will compare the four missing-value strategies discussed in Section 3. We start with a description of the datasets used in the experiments.

4.1. Datasets

We choose five real-world datasets from UCI Machine Learning Repository [3] compare the four missing-value strategies discussed earlier. These datasets are chosen because they have at least some discrete attributes, binary class, and a good number of examples. The original datasets have only a few missing values and we will select values to be missing (see later) to simulate different situations with missing values. The numerical attributes in the datasets are discretized first using minimal entropy method [11] as the cost-sensitive decision tree learning can currently only deal with discrete attributes. This limitation can be moved easily. The datasets are listed in Table 1.

Table 1. Datasets used in the experiments.

	No. of Attributes	No. of Examples	Class distribution (N/P)
Ecoli	6	332	230/102
Breast	9	683	444/239
Heart	8	161	98/163
Thyroid	24	2000	1762/238
Australia	15	653	296/357

The five original datasets do not have test costs and misclassification costs, so we simply make assumptions on the costs. We assume that test costs and misclassification costs are based on the same unit, such as US dollars. We randomly assign random numbers between 0 and 100 to each attribute as test costs. We also assign 200 for false positive, and 600 for false negative misclassification costs. The cost of true positives and true negatives is set to 0. These assumptions are reasonable as attributes do have some costs in real world, and we compare the four missing-value strategies based on the same test and misclassification costs.

4.2. Comparing the Four Missing-value Strategies

To simulate missing values in datasets, we randomly select certain percentages (20%, 40%, 60%, and 80%) of attribute values in the whole dataset to be missing, and those missing values are distributed into each attribute proportional to its cost, as more expensive attributes usually have more missing values. Each dataset is then split into training and test sets using 10-fold cross validation (thus test sets also have the same percentages of missing values). For each split, a decision tree is built from the training dataset, and is applied to the test examples, using the Null Strategy, the Internal Node Strategy, and the C4.5 Strategy. For the Know Value Strategy, a lazy tree is built for each test example.

The performance of the four missing-value strategies is measured by the average total cost of tests and misclassifications of test examples in the 10-fold cross-validation. Here the test cost is the total cost of the tests (attributes) in actually classifying test examples. That is, it is the "effective" test cost, not the sum of test costs of known attributes in test examples. As we discussed in Section 1, some tests may be unnecessary for prediction, as doctors may subscribe more tests than needed for

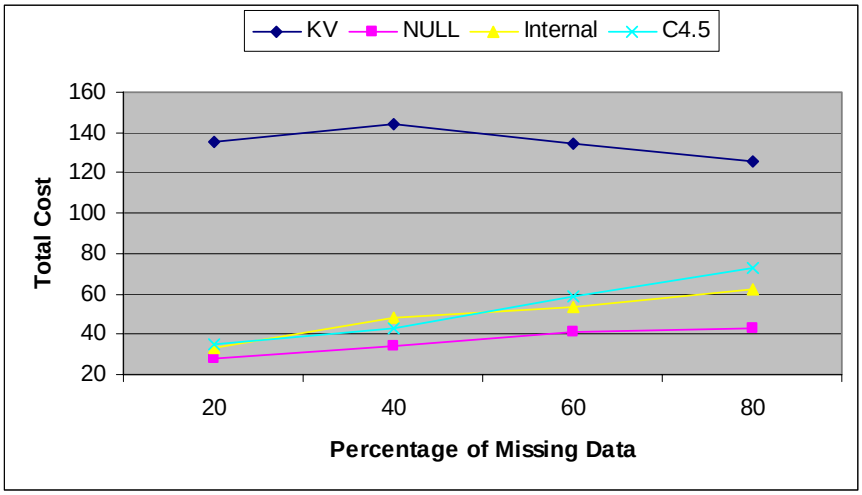
diagnosis. Therefore we use the "effective" test cost to better measure each strategy's actual performance. The misclassification cost is calculated as usual: if the prediction is correct, the misclassification cost is 0; otherwise, it is either the false positive cost or false negative cost, depending on the true class of the test examples. Table 2 lists the average total cost with different missing-value strategies under different percentages of missing values in the datasets. Figures 1 (a) to (e) illustrate the results of Table 2 visually.

	20%	40%	60%	80%
The Ecoli Dataset				
The Known Value Strategy	135.1	144.5	134.2	125.8
The Null Strategy	28.3	33.9	41.4	42.5
The Internal Node Strategy	33.2	48.6	53.5	62.3
The C4.5 Strategy	35.0	42.5	58.9	72.6
The Breast Dataset				
The Known Value Strategy	67.6	91.9	111.3	116.2
The Null Strategy	53.3	61.4	69.8	74.2
The Internal Node Strategy	51.0	59.6	63.5	77.3
The C4.5 Strategy	52.2	57.8	72.6	71.4
The Heart Dataset				
The Known Value Strategy	146.6	126.0	98.2	121.9
The Null Strategy	90.3	88.6	103.7	98.8
The Internal Node Strategy	86.6	85.3	83.2	88.2
The C4.5 Strategy	88.2	87.6	83.2	88.9
The Thyroid Dataset				
The Known Value Strategy	169.4	153.7	138.9	108.5
The Null Strategy	66.6	72.7	76.1	73.3
The Internal Node Strategy	64.4	70.7	71.8	71.7
The C4.5 Strategy	64.4	72.3	90.5	72.4
The Australia Dataset				
The Known Value Strategy	174.2	143.0	106.3	107.4
The Null Strategy	115.3	99.2	121.0	113.1
The Internal Node Strategy	97.1	90.7	94.4	96.8
The C4.5 Strategy	98.1	94.0	109.4	96.2

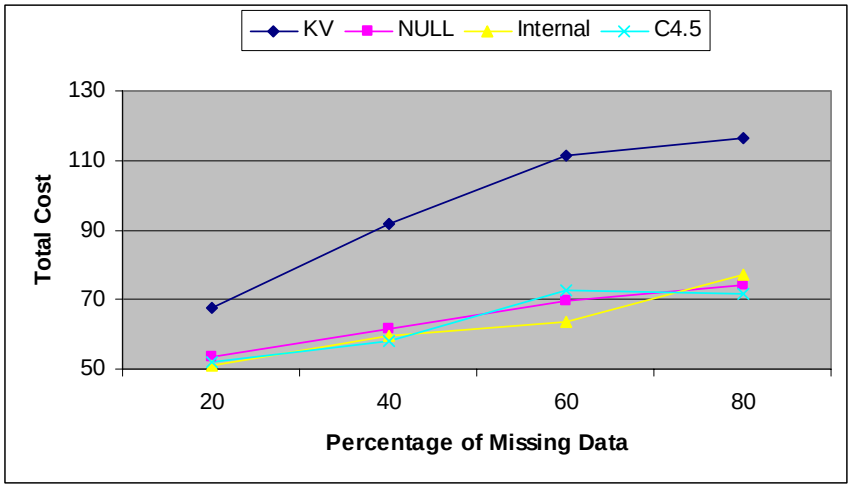
We can draw the following interesting conclusions from the results. First of all, the Known Value Strategy (KV) is almost always the worst. This is because deleting

attributes with missing values in the test example loses useful information in the datasets. Thus, this strategy should be avoided in the future. Second, in only one dataset (Ecoli) the Null Strategy is slightly better than others; in other datasets, it is either similar (in Breast and Thyroid) or worse (in Heart and Australia). This shows that the Null Strategy, although very simple, is often not suitable. Third, the Internal Node Strategy is often comparable with the C4.5 Strategy (in Ecoli, Breast, and Heart) and is better than C4.5 in Thyroid and Australia. This indicates that overall the Internal Node Strategy is better than the C4.5 Strategy. Thus, we can conclude from our experiments that the Internal Node Strategy is the best, followed closely by the C4.5 Strategy, and followed by the Null Strategy. The Known Value Strategy is the worst.

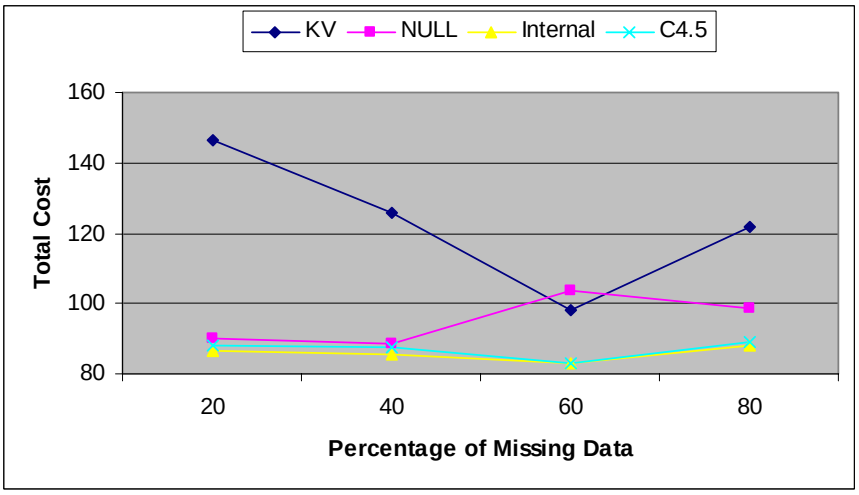
It might be slightly counterintuitive why the C4.5 Strategy, which obtains weighted classifications from leaves, is not better than the Internet Node Strategy that uses the internal node directly. This is because when it weighs leave's classifications, there is a loss of information. If it weighs the leaves' probabilities, it can be shown easily that the result is equivalent the class probability in the internal node in the Internal Node Strategy. Thus, the Internal Node Strategy is better than the C4.5 Strategy.



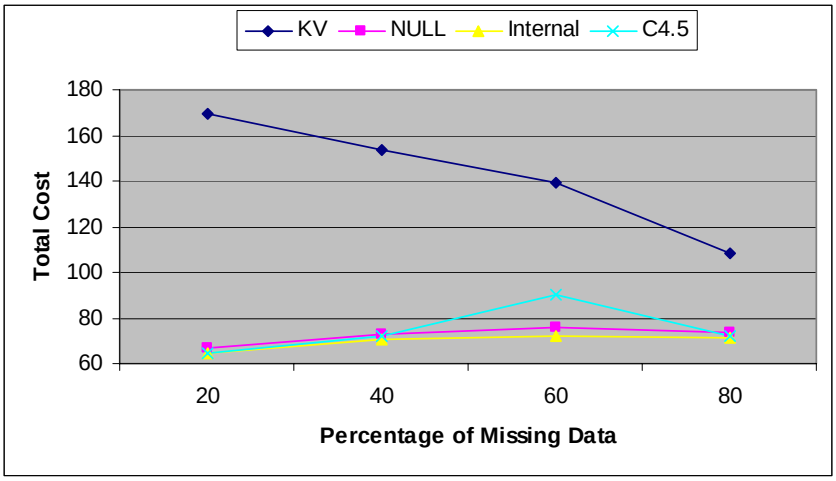
Figures 1 (a) Total costs for Ecoli. In this and the following figures, “KV” stands for the Known Value Strategy, “NULL” for the Null Strategy, “Internal” for the Internal Node Strategy, and “C4.5” for the C4.5 Strategy.



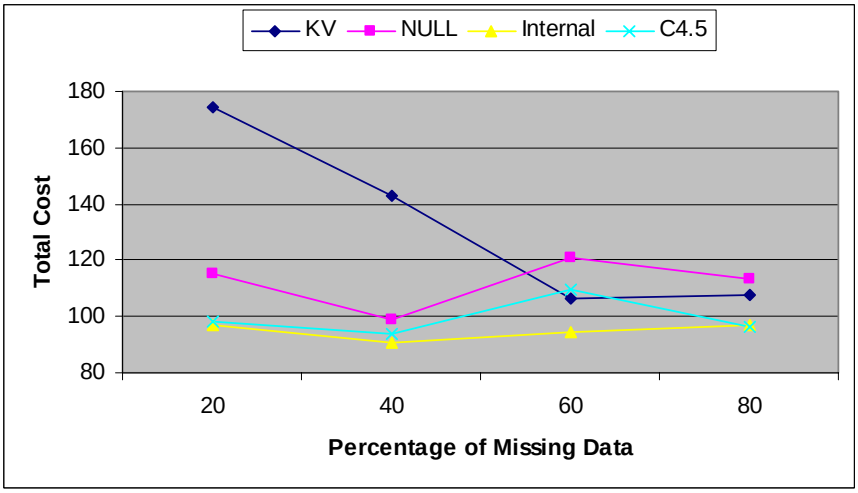
Figures 1 (b) Total costs for Breast



Figures 1 (c) Total costs for Heart



Figures 1 (d) Total costs for Thyroid



Figures 1 (e) Total costs for Australia

5 Conclusions and Future Work

Missing values are traditionally regarded as a tough problem, and must be imputed before learning is applied. In this paper we break away from this tradition, and argue that in cost-sensitive learning that also considers test costs, it is actually desirable to have missing values to reduce the total cost of tests and misclassifications. Thus, cost-sensitive decision tree learning algorithms would only need the known values, and take advantage of "missing is useful" for cost reduction. We compare four such strategies, and conclude that the Internet Node Strategy, originally proposed in [17], is the best. In

our Future work, we plan to apply those strategies to datasets with real costs and missing values.

References

[1] K. M. Ali and M. J. Pazzani, Hydra: A noise-tolerant relational concept learning algorithm. In: R. Bajcsy (Ed.), *Proceedings of the 13th International Joint Conference on Artificial Intelligence* (IJCAI93), pp. 1064-1071. Morgan Kaufmann, 1993.

[2] L. Breiman, J. H. Friedman, R. H. Olshen, and C.J Stone, *Classification and Regression Trees*. Wadsworth, Belmont, California, 1984.

[3] C. L. Blake and C. J. Merz, *UCI Repository of machine learning databases* (See [http://www.ics.uci.edu/~mlearn/MLRepository.html]). Irvine, CA: University of California, Department of Information and Computer Science, 1998.

[4] G. Batista and M. C. Monard, An analysis of four missing data treatment methods for supervised learning. *Applied Artificial Intelligence*, Vol. 17, pp. 519-533, 2003.

[5] X. Chai, L. Deng, Q. Yang, and C. X. Ling, Test-cost sensitive naïve Bayesian classification. In: *Proceedings of the Fourth IEEE International Conference on Data Mining* (ICDM04), Brighton, UK: IEEE Computer Society Press, 2004.

[6] P. Clark and T. Niblett, The CN2 induction algorithm. *Machine Learning*, Vol. 3, pp. 261–283, 1989.

[7] P. Cheeseman, and J. Stutz, Bayesian classification (AutoClass): Theory and results. In: U. Fayyad, G. Piatesky-Shapiro, P. Smyth, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, pp. 153-180. AAAI Press, Menlo Park, CA, 1995.

[8] C. J. Date and H. Darwen, The Default Values Approach to Missing Information. In: *Relational Database Writings 1989-1991*, pp. 343-354, 1989.

[9] P. Domingos, MetaCost: A general method for making classifiers cost-sensitive. In: *Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining* (KDD99), pp. 155-164. San Diego, CA: ACM Press, 1999.

[10] C. Elkan, The foundations of cost-sensitive learning. In: *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence* (IJCAI01), pp. 973-978, 2001.

[11] U. M. Fayyad and K.B. Irani, Multi-interval discretization of continuous-valued attributes for classification learning. In: *Proceedings of the 13th International Joint Conference on Artificial Intelligence* (IJCAI93), pp. 1022-1027. Morgan Kaufmann, 1993.

[12] J. Friedman, Y. Yun and R. Kohavi, Lazy decision trees. In: *proceedings of the 13th National Conference Artificial Intelligence* (AAAI96), pp. 717-724, 1996.

[13] R. Greiner, A. J. Grove and D. Roth, Learning cost-sensitive active classifiers. *Artificial Intelligence*, Vol. 139, No. 2, pp. 137-174, 2002.

[14] M. T. Kai, Inducing Cost-sensitive trees via instance weighting. In: *Proceedings of the Second European Symposium on Principles of Data Mining and Knowledge Discovery*, pp. 23-26. Springer-Verlag, 1998.

[15] K. Lakshminarayan, S. A. Harp and T. Samad, Imputation of missing data in industrial databases. *Applied Intelligence*, Vol. 11, pp. 259-275, 1999.

[16] R. J. A. Little and D. B. Rubin, *Statistical Analysis with Missing Data*, New York: John Wiley, 1987.

[17] C. X. Ling, Q. Yang, J. Wang, and S. Zhang, Decision trees with minimal costs. *In Proceedings of 21st International Conference on Machine Learning* (ICML04), Banff, Alberta, Canada, July 4-8, 2004.

[18] M. Nunez, The use of background knowledge in decision tree induction. *Machine learning*, Vol. 6, pp. 231-250, 1991.

[19] J. R. Quinlan, Unknown attribute values in induction. In Segre A.M. (ed.), *Proceeding Of the Sixth International Workshop on Machin Learning*, pp. 164-168, Morgan Kaufmann, Los Altos, USA, 1989.

[20] J. R. Quinlan, *C4.5: Programs for machine learning*. Morgan Kaufmann, San Mateo, California, 1993.

[21] M. Tan, Cost-sensitive learning of classification knowledge and its applications in robotics. *Machine Learning*, Vol. 13, pp. 7-33, 1993.

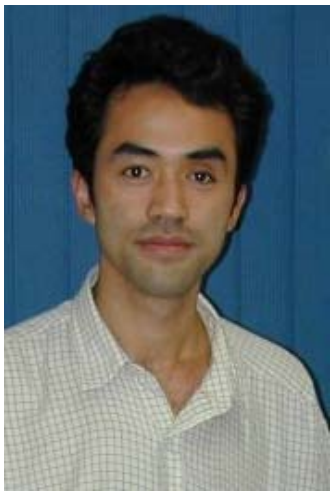
[22] P. D. Turney, Types of cost in inductive concept learning. In: *Proceeding of the Workshop on Cost-Sensitive Learning at the Seventeenth International Conference on Machine Learning*, Stanford University, California, 2000.

[23] P. D. Turney, Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm. *Artificial Intelligence Research*, Vol. 2, pp. 369-409, 1995.

[24] V. B. Zubek and T. G. Dietterich, Pruning improves heuristic search for cost-sensitive learning. In: *Proceedings of the Nineteenth International Conference on Machine Learning* (ICML02), pp. 27-34, Sydney, Australia, 2002.

Shichao Zhang is a principal research fellow in the Faculty of Information Technology at the University of Technology, Sydney, and a professor at the Guangxi Normal University. He received his PhD degree in computer science from Deakin University, Australia. His research interests include data analysis and smart pattern discovery. He has published about 30 international journal papers (including 5 in IEEE/ACM Transactions, 2 in Information Systems, 6 in IEEE magazines) and over 30 international conference papers (including 2 ICML papers and 3 FUZZ-IEEE/AAMAS papers). He has won 4 China NSFC/863 grants, 2

Australian large ARC grants and 2 Australian small ARC grants. He is a senior member of the IEEE, a member of the ACM, and serving as an associate editor for *Knowledge and Information Systems* and *The IEEE Intelligent Informatics Bulletin*.



Zhenxing received the B.S. and M.S. degree from Guangxi Normal University, China. He is currently a Ph.D candidate in Faculty of Information Technology, University of Technology, Sydney, Australia. His research interests are in data mining and machine learning. He has published over 10 papers in international journals and conferences.

Professor Charles X. Ling earned his PhD degree from the Department of Computer Science at the University of Pennsylvania in 1989. Since then he has been a faculty member in

Computer Science at University of Western Ontario. His main research areas include machine learning (theory, algorithms, and applications), cognitive modeling, and AI in general. He has published extensively in journals (such as Machine Learning, Journal of Artificial Intelligence Research, and IEEE TKDE) and international conferences (such as IJCAI and ICML). He is also the Director of Data Mining Lab, leading data mining development in CRM, Bioinformatics, and the Internet. He has managed several data-mining projects for major banks and insurance companies in Canada. See http://www.csd.uwo.ca/faculty/cling for more information.

Shengli Sheng received the B.E. degree from Chongqing University, China, in 1993, and the M.E. degree from Suzhou University, China, in 1999. He received the M.S. degree in computer science from the University of New Brunswick in 2004. He is currently a Ph.D candidate in computer science department of the University of Western Ontario, Canada. He is interested in data mining research and applications.