2012

Minimizing Recommended Error Costs Under Noisy Inputs in Rule-Based Expert Systems

Forest D. Thola
Nova Southeastern University, lrnck@cox.net

Minimizing Recommended Error Costs Under

Noisy Inputs in Rule-Based Expert Systems

By

Forest D. Thola

A dissertation submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
In
Computer Information Systems

Graduate School of Computer and Information Sciences
Nova Southeastern University

2012

An Abstract of a Dissertation Submitted to Nova Southeastern University in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

Minimizing Recommended Error Costs Under Noisy Inputs in Rule-Based Expert Systems

By
Forest D. Thola

October 2012

This dissertation develops methods to minimize recommendation error costs when inputs to a rule-based expert system are prone to errors. The problem often arises in web-based applications where data are inherently noisy or provided by users who perceive some benefit from falsifying inputs. Prior studies proposed methods that attempted to minimize the probability of recommendation error, but did not take into account the relative costs of different types of errors. In situations where these differences are significant, an approach that minimizes the expected misclassification error costs has advantages over extant methods that ignore these costs.

Building on the existing literature, two new techniques – Cost-Based Input Modification (CBIM) and Cost-Based Knowledge-Base Modification (CBKM) were developed and evaluated. Each method takes as inputs (1) the joint probability distribution of a set of rules, (2) the distortion matrix for input noise as characterized by the probability distribution of the observed input vectors conditioned on their true values, and (3) the misclassification cost for each type of recommendation error. Under CBIM, for any observed input vector *v*, the recommendation is based on a modified input vector *v'* such that the expected error costs are minimized. Under CBKM the rule base itself is modified to minimize the expected cost of error.

The proposed methods were investigated as follows: as a control, in the special case where the costs associated with different types of errors are identical, the recommendations under these methods were compared for consistency with those obtained under extant methods. Next, the relative advantages of CBIM and CBKM were compared as (1) the noise level changed, and (2) the structure of the cost matrix varied.

As expected, CBKM and CBIM outperformed the extant Knowledge Base Modification (KM) and Input Modification (IM) methods over a wide range of input distortion and cost matrices, with some restrictions. Under the control, with constant misclassification costs, the new methods performed equally with the extant methods. As misclassification costs increased, CBKM outperformed KM and CBIM outperformed IM. Using different cost matrices to increase misclassification cost asymmetry and order, CBKM and CBIM performance increased. At very low distortion levels, CBKM and CBIM underperformed as error probability became more significant in each method's estimation. Additionally, CBKM outperformed CBIM over a wide range of input distortion as its technique of modifying an original knowledge base outperformed the technique of modifying inputs to an unmodified decision tree.

Acknowledgements

This dissertation, and the multi-year journey leading to its beginning, would never have succeeded without the steadfast devotion, sacrifice, and support of my wife, Nadine. No part of this experience would have worked, or been worthwhile, without her 26 years of encouragement, inspiration, and support.

I would like to thank my proofreaders, Dr. Lori Herold-Jacobson and SSgt Ja'nae E. Benest, USMC, for generously finding time in their busy schedules to contribute their skill toward my success.

Especially, I would like to thank my committee chairman, Dr. Sumitra Mukherjee, and committee members Dr. Francisco J. Mitropoulos, and Dr. Michael J. Laszlo for their guidance and expertise throughout this two year journey.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Statement of the Problem to be Investigated

Problem solving computer programs traditionally use well-structured algorithms, data structures, and detailed reasoning strategies to make decisions. Such programs frequently use human expert knowledge in the form of rules, or stored data, to solve real-world problems without the need for real-time human intelligence. A *rule-based expert system* is defined as one whose *knowledge base* contains domain knowledge coded in the form of rules (Robin, 2010). In these systems, the knowledge base contains expert acquired information, data, rules, cases, and relationships the system requires. They employ an *inference engine* to seek information and relationships from the knowledge base and provide answers, predictions, and suggestions. The inference engine finds the applicable data, interpretations and rules, and relates them correctly. Because the knowledge base is separate from the inference engine, it is easier to update or modify for a new domain, as the data are not hard coded into the system. A *user interface*, including several commercially available graphic packages, provides convenient system access for users, developers, and administrators. Additional components include an *explanation facility* that provides information to the user about how the system arrived at its result, and a *knowledge base acquisition facility* that provides a convenient and efficient means for capturing and storing new or updated information (Abraham, 2005). The basic components of a rule-based expert system are shown in Figure 1.

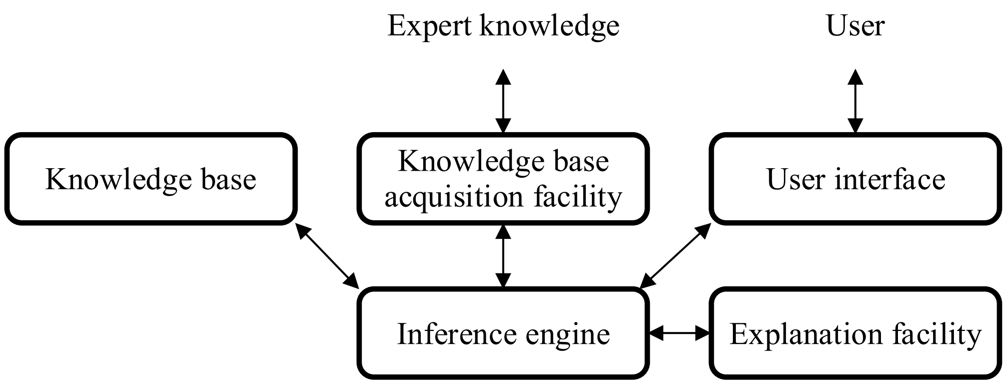


Figure 1: A simple rule-based expert system

Some of the most important contributions of rule-based expert systems are:

1. An ability to capture and preserve perishable human knowledge based on experience;
2. An ability to provide greater consistency in decision making than by humans;
3. An ability to use knowledge in multiple distant locations simultaneously from multiple human experts with a minimal on-site human presence; and
4. Faster response times than by humans.

These significant contributions are instrumental in the many business applications that use expert systems because they are more efficient and reliable than human decision makers. For example, expert systems are used in water quality control, automatic train operation systems, automatic container crane operation systems, elevator control, automobile transmission control, nuclear reactor control (Lee, 1990), fault diagnosis in semiconductor manufacturing, bank failure prediction (Mookerjee, Mannino, & Gibson, 1995), object detection in computer vision (Viola & Jones, 2002), fraud analysis (Viaene, Derrig, & Dedene, 2004), and medical diagnosis (Park, Hwang, & Zhang, 2003).

Rule-based systems are also used in web-based applications with decisions made based on user input on online forms. These applications, however, are vulnerable to input noise or distortion coming from several sources, including deliberate falsification by the user. Some popular columnists even recommend lying on online forms as a reasonable security measure (Bradley, 2011). Some college admissions applicants have paid thousands of dollars hiring professionals to assist in fabricating or embellishing input data (Porter, 2007). Another example is an online credit card application where an applicant intentionally enters false information for education level, employment status, bankruptcy history, and so on. The problem of intentional data falsification has been known for many years. In 1999, as many as 40% of Internet users admitted purposely falsifying such inputs as name, age, income, address, and gender to gain a perceived benefit (Tapscott, 1999). Reasons for falsifying or omitting input data include a lack of trust in the information collector (63%), a belief that the benefits of revealing personal information do not outweigh the risks (65%), and uncertainty about how personal information might be used (69%) (Hoffman, Novak, & Peralta, 1999). Additionally, characteristics including poor interface design and inadequate typing skills cause random, unintentional mistakes. In many cases, truthfulness is not easily checked using secondary sources. Credit scores, for example, are manipulatable within the law to exaggerate an applicant's credit worthiness (Boylu, Aytug, & Koehler, 2010). These mistakes and falsifications then introduce errors into a system's decision-making process.

Noise handling becomes necessary in the absence of any reasonable method to eliminate incorrect user input. Further, a user's level of online experience can influence their behavior with less experienced users showing more trust in online transactions than others (Bewsell, 2008). In the end, while a company may use a carrot-and-stick approach, offering incentives for accurate input, there is no way to guarantee total user motivation toward truthfulness or to guarantee mistake-free input.

In their paper *Lying on the Web: Implications for Expert Systems Redesign,* Jiang, Mookerjee, and Sarkar (2005) developed two techniques for dealing with input noise called Knowledge Base Modification (KM) and Input Modification (IM). KM modifies a knowledge base in the form of a decision tree to account for input noise. Its inputs are an original knowledge base, the joint input probability distributions, and the distortion matrices for all variables. IM modifies observed inputs to their most likely true value and feeds those into an unmodified knowledge base. Its inputs are an unmodified knowledge base, and the marginal distributions and distortion matrices for all variables. Both methods seek to minimize the probability of error in the output.

In some cases, a decision-maker will have a different goal than simply minimizing decision error. Specifically, when considering information acquisition costs, a focus on prediction error may be economically undesirable compared with maximizing expected utility through minimized overall cost (Boylu, Aytug, & Koehler, 2009). In some cases, noise is expected, or even desirable, leading decision makers to seek lowest cost instead of most probable outcome. Jiang et al. (2005) note the example of inventorying assets that require expensive physical inspection as a case where some noise is less expensive than no noise. Given that there will be errors, the goal becomes to minimize the cost of those errors.

In general, errors derive from output misclassification where a specified output is placed into the wrong category or class. In an online credit card application example, a high credit risk applicant might be misclassified as low or medium risk, while an otherwise creditworthy (low risk) applicant might be misclassified as a medium or high credit risk. Each of these misclassification errors would carry a certain cost. Instead of minimizing decision errors, a banking firm would be most interested in minimizing these costs.

Considering the benefit of including error cost analysis in rule-based expert system design, this research expanded the prior work of Jiang et al. (2005) to include misclassification costs. The general aim was to build on the KM and IM methods by developing and evaluating mechanisms to minimize misclassification error costs associated with noisy input data in rule-based expert systems.

Goal

This research expanded the research of Jiang et al. (2005) by developing new processes termed Cost-Based Knowledge Base Modification (CBKM) and Cost-Based Input Modification (CBIM) that take misclassification error costs into account. Specifically, this research addressed the following questions:

1. Are the recommendations based on the proposed methods consistent with those obtained using extant methods when the costs of the different types of errors are identical?
2. How does CBIM compare with CBKM as the structure of the cost matrix varies?
3. How does CBIM compare with CBKM as the level of input noise varies?

This research provided the following contributions:

1. It developed and evaluated a method for CBKM that minimizes expected misclassification error costs.
2. It developed and evaluated a method for CBIM that minimizes expected misclassification error costs.
3. It empirically compared the performance of CBIM and CBKM as the cost structure varied.
4. It empirically compared the performance of CBIM and CBKM as noise varied.

Barriers and Issues

In their KM and IM methodologies, Jiang et al. (2005) used sample data from the machine learning repository at the University of California at Irvine for their experimental process. The same data are available today at http://archive.ics.uci.edu/ml/datasets/Credit+Approval. This database has two classes, 15 variables, and 500 usable instances. In order to maintain consistency, 190 additional instances were excluded because they included duplicate, missing, or conflicting data.

Resources

This approach required a sample data set of real-world values, which was available from the University of California, Irvine. True decision trees were constructed from these data using See5 (C5.0), available under the GNU General Public License for small datasets from Rulequest Research (2010). Additionally, multiple randomly generated inputs were used for process testing. These inputs were generated without special equipment and were tested on an already available AMD FX8120 eight core processor running Kubuntu Linux 12.04 with 32 GB of RAM.

Chapter 2

Review of the Literature

This section will begin with a general overview of rule-based expert systems, input noise or distortion, and misclassification costs before discussing the details of the established KM and IM noise handling methodologies.

Rule-based Expert Systems

Rule-based expert systems have been researched for many years. According to Clancey in 1983, production rules were already considered "a popular representation for capturing heuristics, 'rules of thumb,' in expert systems" with a goal of providing "expert-level consultative advice in scientific and medical problem solving." In 1981, McDermott referenced prior "10 or 12 years" of artificial intelligence (AI) research into techniques for using domain knowledge in computer-based problem solving. In 1969, Newell (as cited in McDermott) discussed an emphasis of AI research in the 1950's and 60's on "discovering heuristics that could be used in solving ill-structured problems" where a heuristic was defined as "a piece of knowledge that can be used to focus search and guide it along the most promising paths."

Much of the available research focuses on improving system performance. For example, decision tree pruning has been researched for many years (Quinlan, 1986). Mookerjee et al. (1995) and Clark and Niblett (1989) demonstrated the advantage of modifying decision trees as part of an induction process. Clark and Niblett further used more of Quinlan's 1983 and 1987 work in dealing with efficiency and simplifying decision trees.

Rule Induction Based on Noisy Methods

Input noise can come from a variety of sources including deliberate falsification, intrinsic system errors, transaction errors, and data entry errors (Morey, 1982, Parssian, Sarkar, & Jacob, 2009). In some cases, such as with current address or telephone number, data are volatile and therefore deteriorate in quality over time (Parssian et al., 2009).

Several methods have been proposed to deal with noisy input data. Hirsh (1994) showed that version space-based learning strategies can be adapted to work well with noisy data sets. A version space is defined by Mitchell (1982) as the set of all classifiers in the language that correctly classify a given set of data. Hirsh's work was an improvement over traditional version spaces that were limited to consistent, noise-free training data. More recently, Boylu et al. (2010) and Boylu (2006) proposed a system to anticipate data input modification in an inductive process. Their approach sought to determine classification rules for intelligent, self-interested agents engaging in strategic behavior to achieve some perceived advantage. In related research, Mannino and Koushik (2000) used a genetic algorithm to calculate the minimum cost of manipulating input data in order to reclassify decision output as a member of a different, preferred class. Aytug, Boylu, and Koehler (2006) point out that most inductive systems assume no strategic behavior (distortion) in the development of training data sets. Dalvi, Domingos, Mausam, Sanghai, and Verma (2004) addressed the case where noise varies as an agent attempts to preemptively distort input in anticipation of a decision maker's strategy, as in the example of email spammers attempting to outwit ever-changing spam filter strategies.

Misclassification Costs

More recently, researchers have recognized that decision makers not only focus on accuracy, but on the potential implications (cost) of particular decisions (Vadera, 2010). As an example, chemical engineers consider the risk of explosion when evaluating processing plant safety, medical consultants consider the potential consequences of patient misdiagnosis, and bank managers consider the cost of a high risk customer defaulting on a loan. Some research looked at error costs in expert systems including early research by Nunez (1991), who looked at cost/benefit ratios dealing with measurement costs that may or may not be monetary. They could, for example, be in terms of time, labor, energy, or level of danger, and so on, using quantifiable attributes such as distance, time, risk, or danger level. Tan (1993) looked at execution costs in robotics and focused on both accuracy and efficiency during the inductive learning process. In an alternative early approach, Turney (1995) used a genetic algorithm with a fitness function that calculated average classification costs for a decision tree, including both measurement and error costs. Another approach used a data resampling process to generate multiple decision trees (Domingos, 1999; Vadera, 2010). Decisions made by the alternative decision trees were combined to reach a decision that minimized classification costs and provided for a final cost-sensitive decision tree. In other research, Ling, Sheng, and Yang (2006) looked at cost-sensitive learning by restructuring decision trees with known attributes in the higher nodes, depending on the attributes in a designated test example. This process required decision tree generation for each test example. Finally, Esmeir and Markovitch (2008) looked at a method they called ACT

(Anytime Cost-sensitive Tree learner) that traded inductive machine learning time for lower costs.

Research into cost matrix utilization includes Elkan (2001) and Zadrozny, Langford, and Abe (2003) who studied cost matrix procedures and investigated the two-class case where decisions were reached by modifying the proportion of negative examples in a training set.

Methods to Deal with Input Noise in Rule-Based Systems

Overview

In their research paper on *Lying on the Web: Implications for Expert Systems Redesign,* Jiang et al. (2005) addressed the problem of designing expert systems operating with noisy input data. While their focus was on deliberate data falsification, their approach would work equally well with any input noise source. Their goal was to increase expert system decision-making accuracy without accounting for misclassification cost.

Assumptions

Given noisy input data, the probability of *true* input values must be estimated. In this context, “true” refers to “correct” or “accurate” input values, say on a credit card application, whereas “observed” refers to the actual input values entered into the system by the user. Therefore, if an observed input vector (**Observed**) refers to one of the possible inputs for a particular problem or a subset of those possibilities, it must have an underlying true input vector (**True**). The probability of the **True** vector given the **Observed** vector is then given in Equation 1 according to Bayes' theorem.

$$\mathrm{P}(\mathbf{True}|\mathbf{Observed}) = \frac{\mathrm{P}(\mathbf{Observed}|\mathbf{True}) \times \mathrm{P}(\mathbf{True})}{\mathrm{P}(\mathbf{Observed})} \tag{1}$$

Given an **Observed** vector with many input possibilities, estimates become difficult for P(**Observed**|**True**) as a large number of input probability parameters must be estimated. For example, given a simple case with *n* binary inputs, probability estimates would be needed for **True** vectors for each of the 2^n possible **Observed** vectors. Because there are 2^n different **True** vector possibilities for each **Observed** vector, probability estimates would be needed for 2^{2n} different probability parameters to evaluate all possible P(**Observed**|**True**) observations. With even a moderate number of inputs, it quickly becomes impossible for domain experts to provide all required estimates. Consequently, two assumptions limit the otherwise prohibitively high number of required estimates:

Assumption 1. $\mathrm{P}(\mathbf{Observed}|\mathbf{True}) = \prod_i \mathrm{P}(Observed_i|\mathbf{True})$, *where i is an index over the inputs.*

Assumption 2. $\mathrm{P}(Observed_i|\mathbf{True}) = \mathrm{P}(Observed_i|True_i)$, *for all i .*

Under Assumption 1, individual input observations are conditionally independent of *other input observations* given the set of true input values. Put another way, the probability of each input observation is dependent on the true state of the inputs and not on the other noisy inputs. In an online credit card application example, an applicant who lies about their education level, perhaps by claiming a bachelor's degree they do not have, would not necessarily also lie about their current income.

Under Assumption 2, an input's observed state is conditionally independent of the *true states* of other inputs, given the true state of the input. That is, noise in one input is not associated with noise in another input such that each input is independent of the others and the distortion is at the individual input level, and not the aggregate input vector level. In the credit card example, this means that an applicant who lies would do so for each application question individually, and not intuitively perceive an advantage to a future inaccurate answer based on a current question. It implies that an applicant would answer questions one at a time, instead of viewing them all at once on a single form and adjusting ones they have already answered based on other questions on the form.

Inputs

The inputs to the KM technique include the original knowledge base in the form of a decision tree, the joint input probability distributions, and the distortion matrices for all variables. The output is a modified knowledge base.

The inputs to the IM technique include the original knowledge base, and the marginal distributions and distortion matrices for all variables. The output is a modified input vector based on its most likely true state.

Knowledge Base Modification

Based on Assumptions 1 and 2, the term P(**Observed**|**True**) in Equation 1 can be simplified according to Equation 2.

$$\mathrm{P}(\mathbf{Observed}|\mathbf{True}) = \prod_i \mathrm{P}(Observed_i|True_i) \tag{2}$$

Therefore, Equation 1 can be expressed as in Equation 3:

$$\mathrm{P}(\mathbf{True}|\mathbf{Observed}) = \frac{\prod_i \mathrm{P}(Observed_i|True_i) \times \mathrm{P}(\mathbf{True})}{\mathrm{P}(\mathbf{Observed})} \tag{3}$$

Estimating the likelihood of a true vector given an observation (P(**True**|**Observed**)) is simpler as the required parameters are P($Observed_i|True_i$) for each i and P(**True**). To obtain the terms P($Observed_i|True_i$) for each i, Jiang, et al. (2005) used real world data from the University of California, Irvine. The second term in the numerator P(**True**) is the joint probability of a true input vector, which is also obtained from available data. Given this information, the probability that a vector is observed is:

$$\mathrm{P}(\mathbf{Observed}) = \sum_r \mathrm{P}(\mathbf{Observed}|True_r)\mathrm{P}(True_r) \quad (4)$$

where r is the index over all possible **True** vectors. Additionally, the probability distributions for each input and their distortion matrices, showing the probability of an observed state given the true input state, are required.

The KM process organizes the procedures into the following five steps.

Step 1. Determine the probabilities of all possible **True** input vectors given an **Observed** vector. Using Equation 3, calculate the conditional probabilities of all **True** input vectors given an **Observed** vector. Determine P(**Observed**) using Equation 4.

Step 2. Find the KM recommendation for the given **Observed** vector by adding the conditional probabilities of all **True** vectors that share the same recommendation. The recommendation with the highest total probability becomes the KM recommendation.

Step 3. Construct a fully enumerated KM table by completing step 1 and step 2 for all possible **Observed** vectors.

Step 4. Compress the KM table to create a condensed KM table, thus removing redundant input. Jiang et al. (2005) reference procedures established by Reinwald and Soland (1966) and Shwayder (1974) for condensing a fully enumerated decision table.

Step 5. Create the KM decision tree. Using the condensed decision table and the probability distribution for **Observed** vectors, an optimum KM tree can be found using AO* search. An informed optimal search algorithm, AO* uses a heuristic function f that estimates the cost of the best solution at a specified node and generates an optimal result so long as f underestimates the expected cost of the solution at every node (Fawcett, 2004). The output of this final step is the KM tree.

Input Modification

Based on the Jiang et al. (2005) prior research, IM recommendations for observed input vectors are created using the original decision tree, a variable's marginal distribution, and the distortion matrices for all variables. For each variable, most likely true input states are chosen using Bayes formula according to Equation 5.

$$P(X^T|X^O) = \frac{P(X^O|X^T)P(X^T)}{P(X^O)} \tag{5}$$

where X^O and X^T are the observed state and true state of a variable respectively. The procedure continues in three steps.

Step 1. Determine the most likely true state from a variable's observed value represented by the root node, and traverse the tree based on that state.

Step 2. If this node is a decision node, it represents the IM recommendation for the observed inputs. Otherwise, compute the most likely true state for the current variable, and traverse the tree again.

Step 3. Continue with step 2 until a decision node is reached, which will be the IM recommendation.

Chapter 3

Methodology

This section first discusses the extant KM and IM methods that form the basis for the new CBKM and CBIM methodologies. Second, it introduces the cost matrix used by CBKM and CBIM for misclassification cost analysis. Finally, it discusses the completed experimental design and analysis. Throughout this chapter, a simple illustrative example is used, consisting of five binary inputs {*A*, *B*, *C*, *D*, *E*} into a system with three possible outcomes {*X*, *Y*, *Z*}.

Inputs to the Proposed Methods

The inputs to the extant KM method are an original knowledge base in the form of a decision tree, the joint input probability distributions, and the distortion matrices for all variables. As in previous studies, the IM method uses the marginal distributions of the variables to compute the joint input probability distributions based on the assumption of independence.

Example

In the example system, the binary inputs form a five element input vector with a single output value (*X*, *Y*, or *Z*). Each input vector represents the observed values submitted to the system that may include input distortion. Distortion is the probability that an observed input deviates from its true, or correct, value. Each **Observed** vector, therefore, has a corresponding **True** vector used to determine the true output of the system as if there were no distortion. For example, an **Observed** input vector might be ($A^O = 1$, $B^O = 0$, $C^O = 1$, $D^O = 0$, $E^O = 1$) with a **True** vector of ($A^T = 1$, $B^T = 0$, $C^T = 1$,

$D^T = 1$, $E^T = 1$). Notice the distortion in input D where the observed value ($D^O = 0$) differs from its true value ($D^T = 1$).

Table 1 shows the rule set used with its binary inputs and three possible outputs. The table and its joint probabilities are condensed from the fully enumerated table shown in Appendix A. Given five binary inputs, there are $2^5 = 32$ possible inputs. Those inputs are condensed to the minimum possible rule set and the joint probabilities calculated from the marginal distribution table as shown in Table 2.

Rule	Inputs					Output	Joint Probability
	A	B	C	D	E		
1	0	0	0	-	0	Y	0.0144
2	0	0	0	-	1	Z	0.0199
3	0	0	1	0	0	X	0.0083
4	0	0	1	0	1	Y	0.0114
5	0	0	1	1	0	Y	0.0101
6	0	-	1	1	1	X	0.0214
7	0	1	0	0	-	X	0.0083
8	-	1	0	1	0	Y	0.0356
9	0	1	0	1	1	Z	0.0059
10	0	1	1	0	-	Z	0.0106
11	0	1	1	1	0	X	0.0054
12	1	0	0	0	0	Z	0.0476
13	1	0	-	0	1	X	0.1493
14	1	0	-	1	0	X	0.1321
15	1	0	0	1	1	Y	0.0803
16	1	0	1	0	0	Y	0.0605
17	1	0	1	1	1	X	0.1022
18	1	1	-	0	0	X	0.0582
19	1	1	-	0	1	Y	0.0804
20	1	1	0	1	1	X	0.0432
21	1	1	1	1	-	Z	0.0949

Table 1: Condensed True Table

In this case, 21 rules completely define all possible input vectors. The input distortion matrix is shown in Table 3.

	Input				
	A	B	C	D	E
0	0.120	0.650	0.440	0.450	0.420
1	0.880	0.350	0.560	0.550	0.580

Table 2: Marginal Distribution

		Observed (*Obs*)									
		A		B		C		D		E	
		0	1	0	1	0	1	0	1	0	1
True Inputs (*True*)	0	0.710	0.290	0.980	0.020	0.960	0.040	0.840	0.160	0.750	0.250
	1	0.160	0.839	0.163	0.837	0.169	0.831	0.069	0.931	0.019	0.981

Table 3: Distortion Matrix

Assumptions 1 and 2 apply as follows. First, individual input observations are conditionally independent of *other input observations* given the set of true input values. In the example where there is distortion in one variable, say *D*, where the true input is '1', but the observed (*Obs*) input is '0,' the assumption implies that the user inputting the value for *D* would not necessarily distort another input, say *B*, given that they have already distorted *D*.

Second, under Assumption 2, an input's observed state is conditionally independent of the *true states* of other inputs, given the true state of the input. In the same example, this means that while there is distortion in one of the inputs, that distortion is not based on the true state of past or future inputs. For example, when a user inputs a value for *D*, he or she would not strategically say "It would be advantageous to distort my input for this variable because I may be required to provide related input in the future." Similarly, when providing input for *B*, he or she would not say "Because I have already input a value for *D*, I should now distort my input for *B*."

The KM and IM processes are now explained using the binary input example discussed. Given the five inputs, the goal of these techniques is to provide the most probable outcome in $\{X, Y, Z\}$.

Returning to Equation 3, the distortion matrix and marginal distribution are used to determine P(**True**|**Observed**) for each **True** vector. As an example, using the observed input vector **Obs** = ($A^O = 1, B^O = 0, C^O = 1, D^O = 0, E^O = 1$), the probability the **True** input vector is actually **True** = ($A^T = 1, B^T = 0, C^T = 1, D^T = 1, E^T = 1$) is determined using Equation 4 as follows:

$P(A^O = 1, B^O = 0, C^O = 1, D^O = 0, E^O = 1 | A^T = 1, B^T = 0, C^T = 1, D^T = 1, E^T = 1)$

$= P(A^O = 1|A^T = 1) \cdot P(B^O = 0|B^T = 0) \cdot P(C^O = 1|C^T = 1) \cdot$

$P(D^O = 0|D^T = 1) \cdot P(E^O = 1|E^T = 1)$

$= 0.839 \cdot 0.980 \cdot 0.831 \cdot 0.069 \cdot 0.981 = 0.04625$

Based on the marginal distribution table (Table 2), the probability the **True** vector is as stated is:

$P(A^T = 1, B^T = 0, C^T = 1, D^T = 1, E^T = 1)$

$= 0.880 \cdot 0.650 \cdot 0.560 \cdot 0.550 \cdot 0.580 = 0.1022$, and

$P(A^O = 1, B^O = 0, C^O = 1, D^O = 0, E^O = 1)$

$= [P(A^O = 1|A^T = 0) \cdot P(A^T = 0) + P(A^O = 1|A^T = 1) \cdot P(A^T = 1)] +$

$[P(B^O = 0|B^T = 0) \cdot P(B^T = 0) + P(B^O = 0|B^T = 1) \cdot P(B^T = 1)] +$

$[P(C^O = 1|C^T = 0) \cdot P(C^T = 0) + P(C^O = 1|C^T = 1) \cdot P(C^T = 1)] +$

$[P(D^O = 0|D^T = 0) \cdot P(D^T = 0) + P(D^O = 0|D^T = 1) \cdot P(D^T = 1)] +$

$[P(E^O = 1|E^T = 0) \cdot P(E^T = 0) + P(E^O = 1|E^T = 1) \cdot P(E^T = 1)]$

$= (0.290 \cdot 0.120) + (0.839 \cdot 0.880) \cdot (0.980 \cdot 0.650) + (0.163 \cdot 0.350) \cdot$

$(0.040 \cdot 0.440) + (0.831 \cdot 0.560) \cdot (0.840 \cdot 0.450) + (0.069 \cdot 0.550) \cdot$

$(0.250 \cdot 0.420) + (0.981 \cdot 0.580) = 0.07265$

From these,

$P(A^T=1, B^T=0, C^T=1, D^T=1, E^T=1|A^O=1, B^O=0, C^O=1, D^O=0, E^O=1) =$

$0.4625 \times 0.1022 / 0.07265 = 0.6506$

All other **True** probabilities are calculated similarly.

Knowledge Base Modification

The observed input vector **Obs** = (A^O=1, B^O=0, C^O=1, D^O=0, E^O=1), true input vector **True** = (A^T=1,B^T=0,C^T=1,D^T=1,E^T=1), and marginal distribution and distortion matrices in Tables 2 and 3 are used for the following KM and IM examples, proceeding through each step in turn. The calculation results are rounded for display.

Step 1. The conditional probabilities for all **True** vectors are calculated given the vector **Obs**. The results are shown in Table 4.

Step 2. The conditional probabilities are then added for all **True** vectors that result in the same output; that is, the sum of P(**True|Obs**) for all **True** vectors that recommend either *X*, *Y*, or *Z*. After consolidating rows, the results are shown in Table 5, where the sum for all vectors with an output of *X* is 0.770502, the sum for output *Y* is 0.213559, and the sum for output *Z* is 0.015939. The *X* output is then chosen as the KM recommendation because it has the highest probability of the three options (highlighted in Table 5).

Step 3. Steps 1 and 2 are repeated for all possible **Observed** vectors to create the KM table as shown in Table 6.

True Vector									
A	B	C	D	E	Output	P(O|T)	P(T)	P(O)	P(T|O)
0	0	0	0	0	Y	0.0024	0.0065	0.0727	0.000213
0	0	0	0	1	Z	0.0094	0.0090	0.0727	0.001154
0	0	0	1	0	Y	0.0002	0.0079	0.0727	0.000021
0	0	0	1	1	Z	0.0008	0.0109	0.0727	0.000116
0	0	1	0	0	X	0.0496	0.0083	0.0727	0.005632
0	0	1	0	1	Y	0.1947	0.0114	0.0727	0.030520
0	0	1	1	0	Y	0.0041	0.0101	0.0727	0.000566
0	0	1	1	1	X	0.0160	0.0139	0.0727	0.003068
0	1	0	0	0	X	0.0004	0.0035	0.0727	0.000019
0	1	0	0	1	X	0.0016	0.0048	0.0727	0.000103
0	1	0	1	0	Y	0.0000	0.0043	0.0727	0.000002
0	1	0	1	1	Z	0.0001	0.0059	0.0727	0.000010
0	1	1	0	0	Z	0.0082	0.0044	0.0727	0.000504
0	1	1	0	1	Z	0.0324	0.0061	0.0727	0.002731
0	1	1	1	0	X	0.0007	0.0054	0.0727	0.000051
0	1	1	1	1	X	0.0027	0.0075	0.0727	0.000275
1	0	0	0	0	Z	0.0069	0.0476	0.0727	0.004520
1	0	0	0	1	X	0.0271	0.0657	0.0727	0.024492
1	0	0	1	0	X	0.0006	0.0581	0.0727	0.000454
1	0	0	1	1	Y	0.0022	0.0803	0.0727	0.002462
1	0	1	0	0	Y	0.1437	0.0605	0.0727	0.119565
1	0	1	0	1	X	0.5637	0.0836	0.0727	0.647929
1	0	1	1	0	X	0.0118	0.0740	0.0727	0.012020
1	0	1	1	1	X	0.0464	0.1022	0.0727	0.065136
1	1	0	0	0	X	0.0011	0.0256	0.0727	0.000404
1	1	0	0	1	Y	0.0045	0.0354	0.0727	0.002192
1	1	0	1	0	Y	0.0001	0.0313	0.0727	0.000041
1	1	0	1	1	X	0.0004	0.0432	0.0727	0.000220
1	1	1	0	0	X	0.0239	0.0326	0.0727	0.010699
1	1	1	0	1	Y	0.0937	0.0450	0.0727	0.057978
1	1	1	1	0	Z	0.0020	0.0398	0.0727	0.001076
1	1	1	1	1	Z	0.0077	0.0550	0.0727	0.005828

Table 4: True vector conditional probabilities

True Vector						
A	B	C	D	E	Output	P(T|O)
0	0	1	0	0	X	0.005632
0	-	1	1	1	X	0.003343
0	1	0	0	-	X	0.000122
0	1	1	1	0	X	0.000051
1	0	-	0	1	X	0.672421
1	0	-	1	0	X	0.012474
1	0	1	1	1	X	0.065136
1	1	-	0	0	X	0.011103
1	1	0	1	1	X	0.000220
					Total:	0.770502
0	0	0	-	0	Y	0.000234
0	0	1	0	1	Y	0.030520
0	0	1	1	0	Y	0.000566
-	1	0	1	0	Y	0.000043
1	0	0	1	1	Y	0.002462
1	0	1	0	0	Y	0.119565
1	1	-	0	1	Y	0.060170
					Total:	0.213559
0	0	0	-	1	Z	0.001270
0	1	0	1	1	Z	0.000010
0	1	1	0	-	Z	0.003235
1	0	0	0	0	Z	0.004520
1	1	1	1	-	Z	0.006904
					Total:	0.015939

Table 5: Conditional probabilities by outcome

Step 4. In this step, the fully enumerated KM table is condensed as shown in Table 7.

Step 5. In the final step, the KM tree is created from the condensed KM table created in step 4. The KM tree is shown in Figure 2. For comparison, the corresponding True tree is shown in Figure 3.

Observed Input					P(X|O)	P(Y|O)	P(Z|O)	KM Output
A	B	C	D	E				
0	0	0	0	0	0.1926	0.3810	0.4264	Z
0	0	0	0	1	0.5031	0.1844	0.3125	X
0	0	0	1	0	0.5152	0.4099	0.0749	X
0	0	0	1	1	0.3031	0.4220	0.2749	Y
0	0	1	0	0	0.4118	0.5379	0.0503	Y
0	0	1	0	1	0.5737	0.3810	0.0452	X
0	0	1	1	0	0.5855	0.3646	0.0499	X
0	0	1	1	1	0.8274	0.1138	0.0588	X
0	1	0	0	0	0.8052	0.1047	0.0901	X
0	1	0	0	1	0.4175	0.4759	0.1066	Y
0	1	0	1	0	0.2027	0.6818	0.1155	Y
0	1	0	1	1	0.4789	0.1879	0.3332	X
0	1	1	0	0	0.5848	0.0462	0.3690	X
0	1	1	0	1	0.1559	0.4744	0.3697	Y
0	1	1	1	0	0.4034	0.0488	0.5478	Z
0	1	1	1	1	0.3706	0.0719	0.5575	Z
1	0	0	0	0	0.1833	0.1846	0.6321	Z
1	0	0	0	1	0.7177	0.1502	0.1321	X
1	0	0	1	0	0.7643	0.1365	0.0992	X
1	0	0	1	1	0.3851	0.5578	0.0571	Y
1	0	1	0	0	0.2112	0.7505	0.0382	Y
1	0	1	0	1	0.7705	0.2136	0.0159	X
1	0	1	1	0	0.7897	0.1397	0.0706	X
1	0	1	1	1	0.8731	0.0582	0.0687	X
1	1	0	0	0	0.8430	0.1053	0.0517	X
1	1	0	0	1	0.2521	0.7167	0.0311	Y
1	1	0	1	0	0.1771	0.6705	0.1525	Y
1	1	0	1	1	0.5975	0.2271	0.1754	X
1	1	1	0	0	0.8148	0.0657	0.1195	X
1	1	1	0	1	0.1723	0.7091	0.1186	Y
1	1	1	1	0	0.1825	0.0397	0.7778	Z
1	1	1	1	1	0.1189	0.1022	0.7789	Z

Table 6: Fully enumerated KM table

Observed Input					KM Output
A	B	C	D	E	
-	0	0	0	0	Z
-	0	-	0	1	X
-	0	-	1	0	X
-	0	0	1	1	Y
-	0	1	0	0	Y
-	0	1	1	1	X
-	1	-	0	0	X
-	1	-	0	1	Y
-	1	0	1	0	Y
-	1	0	1	1	X
-	1	1	1	-	Z

Table 7: Condensed KM Table

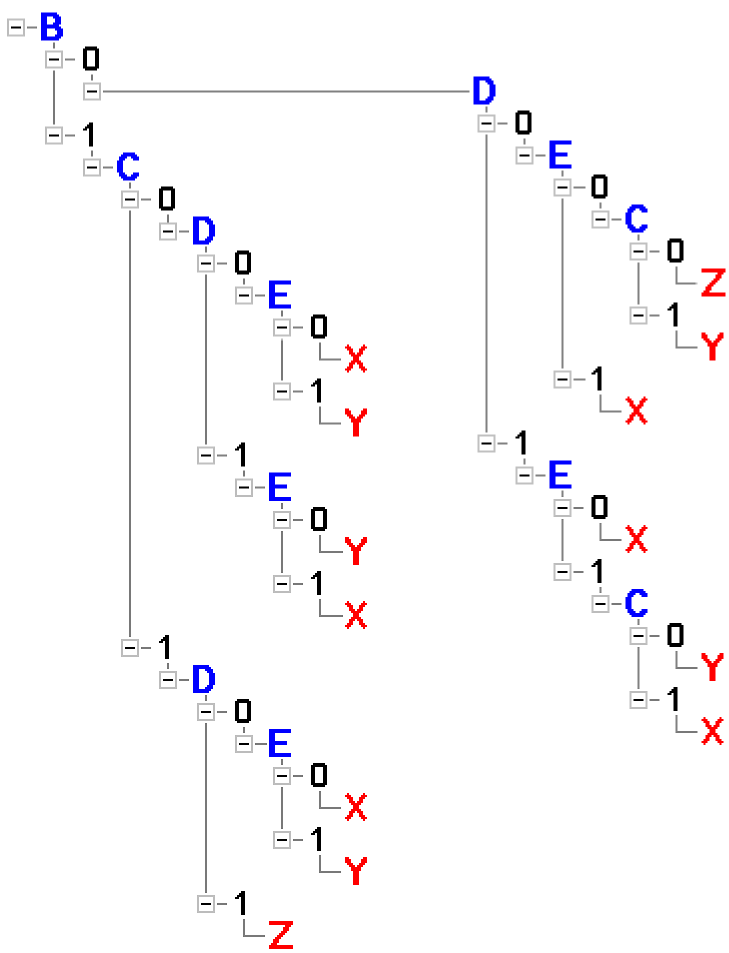


Figure 2: KM Tree

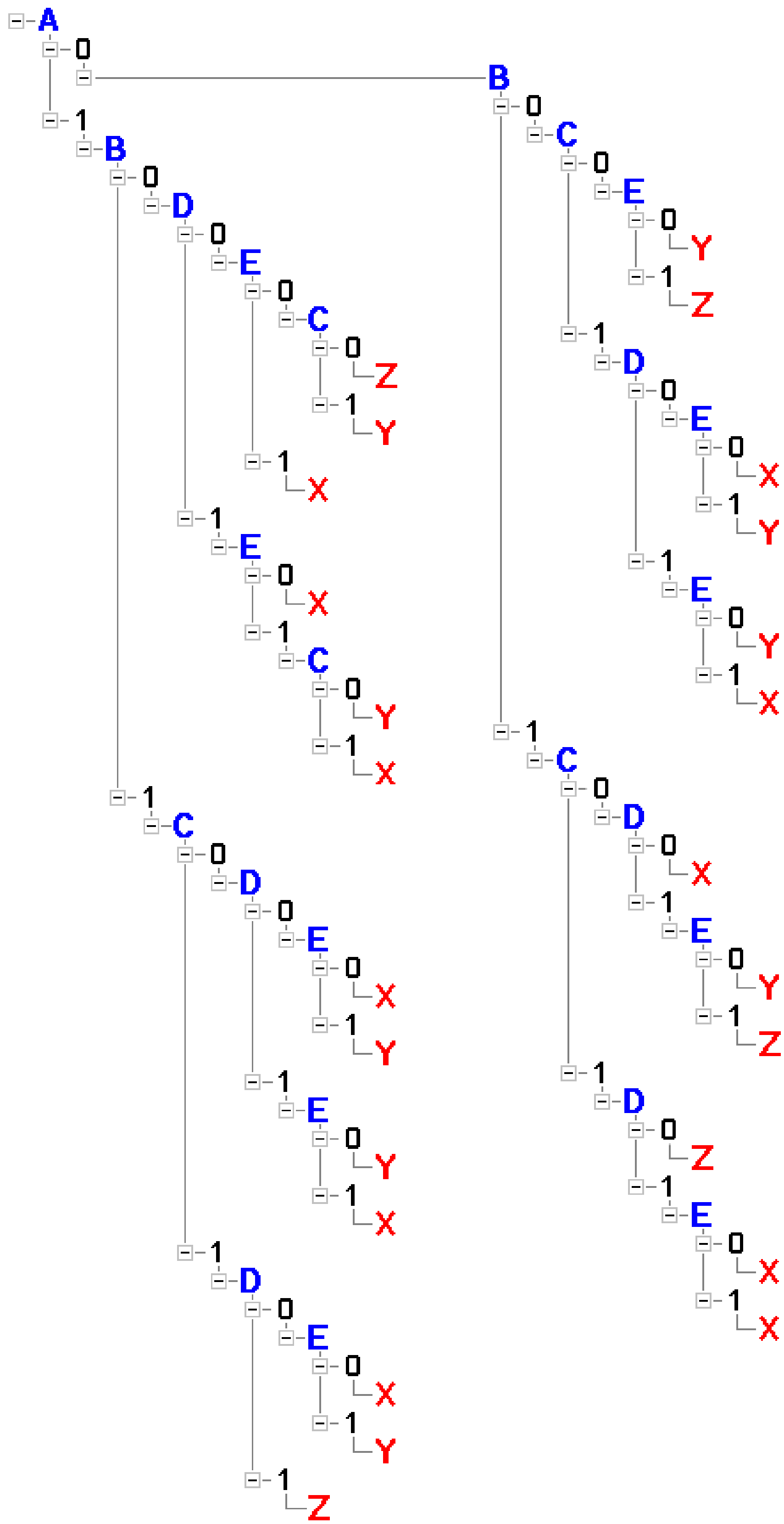


Figure 3: True Tree

Input Modification

Continuing with the example, the same true tree is used from Figure 3 and the input vector remains **Obs** = (A^O=1, B^O=0, C^O=1, D^O=0, E^O=1).

Step 1. Given the observed input of A^O=1 for the True Tree root node variable, the conditional probabilities for the two possible true values for *A* are calculated using Equation 5. The results are shown in Table 8.

True		Observed	ΣP(T|O)
A	0	1	0.0450
	1	1	0.9550

Table 8: Input Modification Input *A* conditional probabilities

Step 2. Given that the conditional probability for P(A^T=1\|A^O=1) = 0.9550 is the highest conditional probability (highlighted in Table 8), the IM process selects 1 as the most likely true state for *A*.

Step 3. Repeating step 2 until a decision node is reached, the process moves according to the true tree in Figure 3 to the variable *B*. Given the observed state of 0 for *B*, the conditional probabilities for P(B^T\|B^O) are calculated as shown in Table 9.

True		Observed	ΣP(T|O)
B	0	0	0.9179
	1	0	0.0821

Table 9: Input Modification Input *B* conditional probabilities

Since $P(B^T=0|B^O=0) = 0.9179$ is the higher value (highlighted in Table 9), the IM selected most likely true state for *B* is 0. Repeating step 2 for the next variable in the true tree, the conditional probabilities for *D* are calculated as displayed in Table 10.

True		Observed	ΣP(T|O)
D	0	0	0.9087
	1	0	0.0913

Table 10: Input Modification Input *D* conditional probabilities

Since $P(D^T=0|D^O=0) = 0.9087$ is the higher value (highlighted in Table 10), the IM selected most likely true state for *D* is 0. Repeating step 2 for the next variable in the true tree, the conditional probabilities for *E* are calculated as displayed in Table 11.

True		Observed	ΣP(T|O)
E	0	1	0.1558
	1	1	0.8442

Table 11: Input Modification Input E conditional probabilities

Since $P(E^T=1|E^O=1) = 0.8442$ is the higher value (highlighted in Table 11), the IM selected most likely true state for *E* is 1. Since the 1 branch of the *E* node leads to a decision node in the true tree, the IM recommendation for the observed vector **Obs** = ($A^O=1$, $B^O=0$, $C^O=1$, $D^O=0$, $E^O=1$) is *X*.

Cost Matrix

General

In this research, the extant KM and IM methodologies were extended to misclassification cost analysis using a cost matrix. The cost matrix records the misclassification error cost $C(i, j)$ where decision i is the decision made given noisy input while decision j is the decision made given true input. The cost matrix takes the format shown in Table 12. By convention, rows correspond to alternative predicted classes, while columns correspond to true classes as shown (Elkan, 2001).

	Actual: D1	Actual: D2	Actual: D3
Decision Made: D1	$C(0,0)$	$C(0,1)$	$C(0,2)$
Decision Made: D2	$C(1,0)$	$C(1,1)$	$C(1,2)$
Decision Made: D3	$C(2,0)$	$C(2,1)$	$C(2,2)$

Table 12: General cost matrix format

An element in the cost matrix $C(i , j)$ represents the misclassification cost of making decision i given the true (actual) decision j. In each case where the decision made (i) is the same as the decision made given true input (j), no error exists and the misclassification cost is zero. In the case where all misclassification costs were equal, this research verified that classification methodologies using misclassification costs produced results identical to methodologies that do not take those costs into account.

In this research, parameters α and β were used in the cost matrix as shown in Table 13. There, α captures the relative importance of asymmetry in misclassification costs while β captures the relative costs based on the degree of difference between the true and the predicted values for a class (referred to in this context as class decision order). The assumption that a lesser misclassification error carries a smaller misclassification cost necessitates the restriction that $\beta \geq 1$ while $\alpha > 0$. For the experiments, asymmetry is restricted to one side of the control value of 1 ($0 < \alpha \leq 1$) as testing on both sides would be duplicative.

	Actual: D1	Actual: D2	Actual: D3
Decision Made: D1	0	C	βC
Decision Made: D2	αC	0	C
Decision Made: D3	αβC	αC	0

Table 13: Cost matrix format

In some cases, misclassification costs are ordered so they consistently increase with the magnitude of the error. That is, a "small" error is less expensive than a "large" error, given a true decision. Looking at the cost matrix, this special case is expressed as in Equation 6.

$$C(i,k) \leq C(j,k), \text{ if } |i-k| \leq |j-k| \quad (6)$$

As an example, consider an online credit card application where a bank's decision to grant or not grant credit to a particular applicant using actual (true) input places that applicant in a high risk class. Equation 6 implies that the misclassification cost to the bank of deciding the applicant is of medium risk (small error) is always less than or equal to the cost of deciding that high risk applicant is of low risk (large error). The parameter β (restricted to $\beta \geq 1$) permits testing of this case over a range of magnitudes of the difference between misclassification costs.

Additionally, in some cases, misclassification costs are asymmetrical within the cost matrix. This equates to the special case as shown in Equation 7.

$$C(j,i) < C(i,j), for\ all\ i\ and\ j\ where\ i > j \tag{7}$$

Consider the case where a bank evaluates two credit card applicants where the first applicant is of high risk, but is misclassified as low risk, whereas the second applicant is of low risk, but is decided to be of high risk. The first error might lead the bank to grant credit at a favorable interest rate to a customer who ultimately defaults on a loan, leaving the bank to cover the cost. In the second case, the bank may fail to grant credit to an otherwise creditworthy customer, causing the bank to lose that customer and their potential future business. In these two examples, the monetary costs to the bank are different. The parameter α (restricted to $0 < \alpha \leq 1$) permits testing of this case over a range of asymmetrical misclassification costs.

Decision Making Using a Cost Matrix

The cost matrix allows decision-making based on lowest expected misclassification cost instead of lowest probability of error as used by the KM and IM methodologies. Following Elkan (2001), using a cost matrix with misclassification costs $C(i, j)$, the lowest cost decision for an example x is the class i the minimizes Equation 8.

$$L(x,i) = \sum_j P(j|x)C(i,j) \tag{8}$$

For each i, $L(x, i)$, is the sum over the alternative possibilities for the true class of x.

Cost-Based Knowledge Base Modification (CBKM)

General

The original KM methodology modifies a knowledge base (decision tree) to account for user input distortion. Its inputs include an original decision tree, the joint input probability distributions, and the distortion matrices for all variables. The output is a modified decision tree referred to as a *KM tree*. In the process, a fully enumerated KM table, including the KM recommendations for all possible **Observed** vectors, is computed. The KM recommendation for a specific **Observed** input vector depends on the conditional probabilities of all **True** vectors, given the **Observed** vector. The process proceeds in five steps as previously outlined.

The CBKM method differs from the KM method in its selection of lowest-cost vectors over highest total **True** vector probability in step 2. It does this by using a cost matrix and the conditional probability of each class given the **Observed** vector. The CBKM steps proceed as follows.

Step 1. Determine the probabilities of all possible **True** input vectors given an **Observed** vector. Using Equation 3, calculate the conditional probabilities of all **True** input vectors given an **Observed** vector. Determine P(**Observed**) using Equation 4.

Step 2. Find the CBKM recommendation for the given **Observed** vector. Given the probabilities in step 1 and error costs from the cost matrix, calculate the cost of each **True** vector given the **Observed** vector for all **True** vectors with the same decision outcome. The recommendation with the lowest predicted cost becomes the CBKM recommendation.

Step 3. Construct a fully enumerated CBKM table by completing step 1 and step 2 for all possible **Observed** vectors.

Step 4. Compress the CBKM table to create a condensed CBKM table, thus removing redundant input.

Step 5. Create the CBKM decision tree. Using the condensed decision table and the error costs for **Observed** vectors, an optimum CBKM tree can be found. The output of this final step is the CBKM tree.

CBKM differs from KM first with its added requirement for the cost matrix as discussed. In step 1, both CBKM and KM use the joint input probability distributions and distortion matrices to calculate the probabilities of all possible **True** input vectors. The methodologies differ in step 2 because both probabilities and expected costs are required to calculate lowest costs. The goal of CBKM is to minimize the cost whereas the goal of KM is to maximize the sum of conditional probabilities for all **True** vectors. Given the CBKM recommendation from step 2, the final three steps are the same as for KM.

Example

Revisiting the KM example previously discussed, step 1 is the same for CBKM as for KM. Step 2 differs with the introduction of the cost matrix and resulting calculation of lowest misclassification cost. A cost matrix with example misclassification costs is shown in Table 14. In this case, α and β are selected such that $\alpha = 0.1$ and $\beta = 5$. Possible outcomes (decisions) are X, Y, and Z.

	Actual: X	Actual: Y	Actual: Z
Decision Made: X	0	1	5
Decision Made: Y	0.1	0	1
Decision Made: Z	0.5	0.1	0

Table 14: Cost matrix format

In step 2, the conditional probabilities already calculated in Table 7 are used with the cost matrix to satisfy Equation 8. Here, $L(x, i)$ must be evaluated for the three possible values of i, (X, Y, and Z), given the observed input vector **Obs** = (A^O=1, B^O=0, C^O=1, D^O=0, E^O= 1).

$C(\boldsymbol{Obs}, X) =$

$= P(X|\boldsymbol{Obs})C(X,X) + P(Y|\boldsymbol{Obs})C(X,Y) + P(Z|\boldsymbol{Obs})C(X,Z)$

= (0.7705 · 0.0) + (0.2136 · 1.0) + (0.0159 · 5.0) = 0.2931

$C(\boldsymbol{Obs}, Y) =$

$= P(X|\boldsymbol{Obs})C(Y,X) + P(Y|\boldsymbol{Obs})C(Y,Y) + P(Z|\boldsymbol{Obs})C(Y,Z)$

= (0.7705 · 0.1) + (0.2136 · 0.0) + (0.0159 · 1.0) = 0.0929

$C(\boldsymbol{Obs}, Z) =$

$= P(X|\boldsymbol{Obs})C(Z,X) + P(Y|\boldsymbol{Obs})C(Z,Y) + P(Z|\boldsymbol{Obs})C(Z,Z)$

= (0.7705 · 0.5) + (0.2136 · 0.1) + (0.0159 · 0.0) = 0.4066

Because the Y class minimizes the overall cost, it becomes the CBKM recommendation for the given input vector.

The CBKM process proceeds through the remaining steps using the values generated in step 2. These remaining steps are unchanged from the original KM process with the CBKM condensed table as shown in Table 15 and the CBKM tree as shown in Figure 4.

Observed Input					CBKM Output
A	B	C	D	E	
0	0	0	0	-	Z
-	0	-	1	0	Y
0	-	0	1	1	Z
-	0	1	0	-	Y
-	0	1	1	1	Y
-	1	0	0	-	Y
0	1	0	1	0	Y
0	1	1	-	-	Z
1	0	0	0	0	Z
1	0	0	-	1	Y
1	1	-	1	0	Z
1	1	0	1	1	Y
1	1	1	0	-	Y
1	1	1	1	1	Z

Table 15: Condensed CBKM table

Cost-Based Input Modification (CBIM)

General

The CBIM method similarly differs from the IM method in the selection of lowest misclassification error cost over most probable **True** vector. The error cost matrix and true tree are used in three steps that proceed as follows.

Step 1. Using the cost matrix, determine the lowest cost **True** vector from a variable's **Observed** vector represented by the true tree root node, and traverse the tree based on that state.

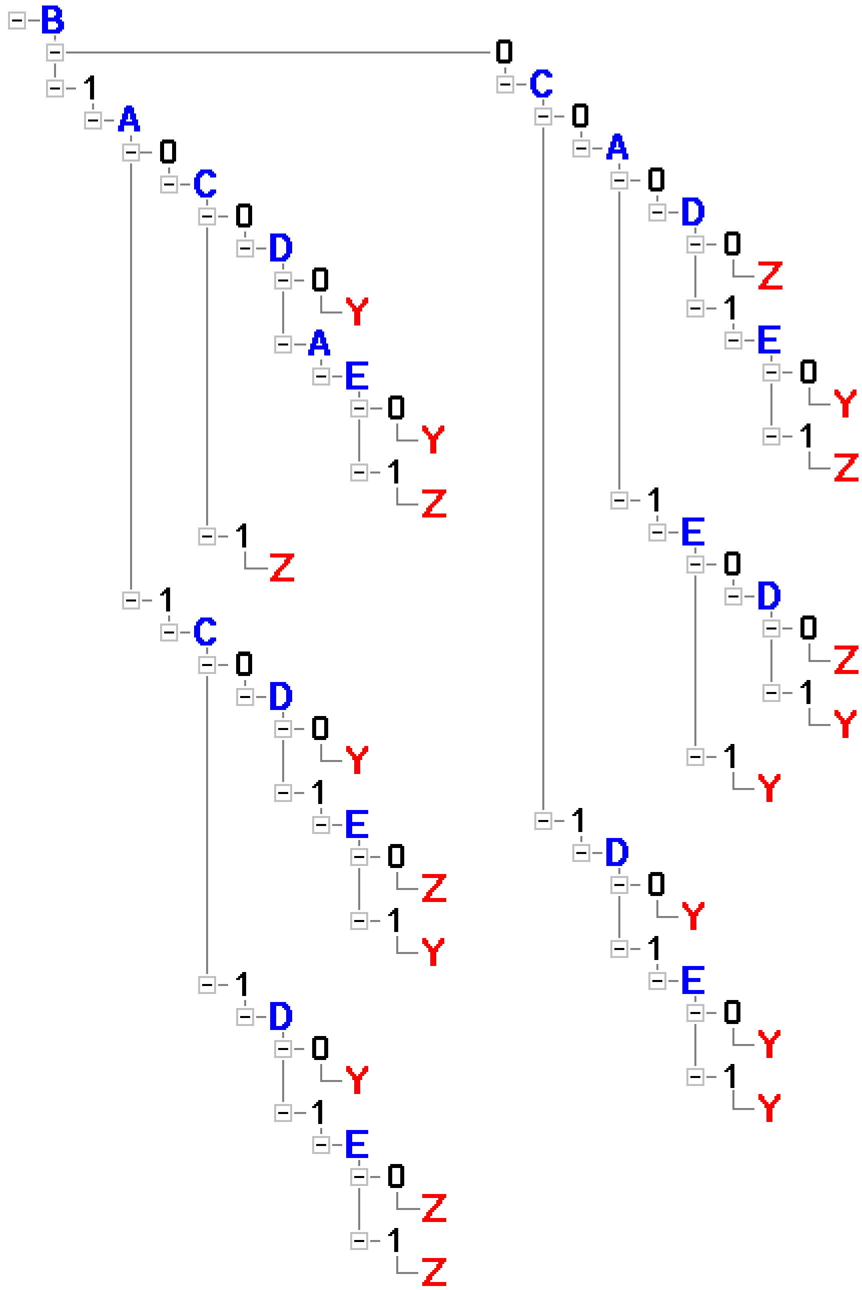


Figure 4: CBKM Tree

Step 2. If this node is a decision node, it represents the CBIM recommendation for the observed inputs. Otherwise, compute the lowest cost true state for the current variable, and traverse the tree again.

Step 3. Continue with step 2 until a decision node is reached, which will be the CBIM recommendation.

Example

Step 1 of the CBIM process uses the original, unaltered true tree as shown in Figure 3. Given an initial *A* root node, the cost of each observed class is calculated using Equation 8. Table 16 shows the required values for each potential **Observed** vector. In the table, the 32 possible **Observed** vectors are listed on the left side from ***Obs*** = (A^O=0, B^O=0, C^O=0, D^O=0, E^O=0) to ***Obs*** = (A^O=1, B^O=1, C^O=1, D^O=1, E^O=1). Given the true tree from Figure 3, *P(X|O)*, *P(Y|O)*, and *P(Z|O)* are the same values already calculated for the CBKM process as previously discussed and displayed in Table 6.

Additional required values are the costs *C(X|O)*, *C(Y|O)*, and *C(Z|O)*. Using the example ***Obs*** = (A^O=1, B^O=0, C^O=1, D^O=0, E^O=1) which is highlighted in Table 16 and the cost matrix in Table 14, the calculations are as follows using Equation 8 (note the calculations are rounded to 4 decimal places while the computer generated results in Table 16 are not):

$$C(X|O) = P(X)C(X^O|X^T) + P(Y)C(X^O|Y^T) + P(Z)C(X^O|Z^T)$$

$$= (0.7705)(0.0) + (0.2135)(1.0) + (0.0159)(5.0) = 0.2930$$

$$C(Y|O) = P(X)C(Y^O|X^T) + P(Y)C(Y^O|Y^T) + P(Z)C(Y^O|Z^T)$$

$$= (0.7705)(0.1) + (0.2135)(0.0) + (0.0159)(1.0) = 0.0930$$

$$C(Z|O) = P(X)C(Z^O|X^T) + P(Y)C(Z^O|Y^T) + P(Z)C(Z^O|Z^T)$$

$$= (0.7705)(0.5) + (0.2135)(0.1) + (0.0159)(0.0) = 0.4066$$

Observed Input					P(T|O)	P(X|O)	P(Y|O)	P(Z|O)	C(X|O)	C(Y|O)	C(Z|O)	Cost
A	B	C	D	E								
0	0	0	0	0	0.000213	0.1926	0.3810	0.4264	2.5130	0.4456	0.1344	3.0930
0	0	0	0	1	0.001154	0.5031	0.1844	0.3125	1.7469	0.3628	0.2700	2.3797
0	0	0	1	0	0.000021	0.5152	0.4099	0.0749	0.7844	0.1264	0.2986	1.2095
0	0	0	1	1	0.000116	0.3031	0.4220	0.2749	1.7966	0.3052	0.1937	2.2956
0	0	1	0	0	0.005632	0.4118	0.5379	0.0503	0.7895	0.0915	0.2597	1.1406
0	0	1	0	1	0.030520	0.5737	0.3810	0.0452	0.6072	0.1026	0.3250	1.0348
0	0	1	1	0	0.000566	0.5855	0.3646	0.0499	0.6140	0.1084	0.3292	1.0517
0	0	1	1	1	0.003068	0.8274	0.1138	0.0588	0.4078	0.1415	0.4251	0.9744
0	1	0	0	0	0.000019	0.8052	0.1047	0.0901	0.5553	0.1707	0.4131	1.1391
0	1	0	0	1	0.000103	0.4175	0.4759	0.1066	1.0088	0.1483	0.2563	1.4135
0	1	0	1	0	0.000002	0.2027	0.6818	0.1155	1.2592	0.1358	0.1695	1.5645
0	1	0	1	1	0.000010	0.4789	0.1879	0.3332	1.8538	0.3811	0.2582	2.4931
0	1	1	0	0	0.000504	0.5848	0.0462	0.3690	1.8910	0.4274	0.2970	2.6155
0	1	1	0	1	0.002731	0.1559	0.4744	0.3697	2.3229	0.3853	0.1254	2.8336
0	1	1	1	0	0.000051	0.4034	0.0488	0.5478	2.7878	0.5882	0.2066	3.5826
0	1	1	1	1	0.000275	0.3706	0.0719	0.5575	2.8595	0.5946	0.1925	3.6465
1	0	0	0	0	0.004520	0.1833	0.1846	0.6321	3.3452	0.6504	0.1101	4.1057
1	0	0	0	1	0.024492	0.7177	0.1502	0.1321	0.8105	0.2038	0.3739	1.3882
1	0	0	1	0	0.000454	0.7643	0.1365	0.0992	0.6326	0.1757	0.3958	1.2040
1	0	0	1	1	0.002462	0.3851	0.5578	0.0571	0.8434	0.0956	0.2483	1.1874
1	0	1	0	0	0.119565	0.2112	0.7505	0.0382	0.9417	0.0593	0.1807	1.1817
1	0	1	0	1	0.647929	0.7705	0.2136	0.0159	0.2933	0.0930	0.4066	0.7928
1	0	1	1	0	0.012020	0.7897	0.1397	0.0706	0.4925	0.1495	0.4088	1.0509
1	0	1	1	1	0.065136	0.8731	0.0582	0.0687	0.4017	0.1560	0.4424	1.0000
1	1	0	0	0	0.000404	0.8430	0.1053	0.0517	0.3637	0.1360	0.4320	0.9317
1	1	0	0	1	0.002192	0.2521	0.7167	0.0311	0.8724	0.0563	0.1977	1.1265
1	1	0	1	0	0.000041	0.1771	0.6705	0.1525	1.4327	0.1702	0.1556	1.7585
1	1	0	1	1	0.000220	0.5975	0.2271	0.1754	1.1042	0.2352	0.3215	1.6608
1	1	1	0	0	0.010699	0.8148	0.0657	0.1195	0.6633	0.2010	0.4140	1.2782
1	1	1	0	1	0.057978	0.1723	0.7091	0.1186	1.3020	0.1358	0.1571	1.5949
1	1	1	1	0	0.001076	0.1825	0.0397	0.7778	3.9289	0.7961	0.0952	4.8202
1	1	1	1	1	0.005828	0.1189	0.1022	0.7789	3.9966	0.7908	0.0697	4.8570

Table 16: CBIM required data

The cost of ***Obs*** is then $C(X|O) + C(Y|O) + C(Z|O)$ = 0.2930 + 0.0930 + 0.4066 = 0.7926. The cost of each other ***Obs*** is calculated analogously and displayed in Table 16.

In step 2, returning to the true tree in Figure 3, and its root node of A, the costs for $A = 0$ and $A = 1$ are derived from Table 16 using Equation 8. For example, the first 16 observed input entries correspond to all **Observed** vectors where $A^O = 0$ and the last 16 entries correspond to all **Observed** vectors where $A^O = 1$. The sum of all P(T|O) for $A^O = 0$ is then 0.0450, the sum of all P(T|O) for $A^O = 1$ is 0.9950, the sum of all costs for $A^O = 0$ is 32.468, and the sum of all costs for $A^O = 1$ is 29.939.

The calculation for the root node is then:

$$C(A{=}0) = P(A{=}0)C(A{=}0|\,A{=}0) + P(A{=}1)C(A{=}0|\,A{=}1)$$

$$= (0.0450)(0.0) + (0.9950)(32.468) = 31.007$$

$$C(A{=}1) = P(A{=}0)C(A{=}1|\,A{=}0) + P(A{=}1)C(A{=}1|\,A{=}1)$$

$$= (.0450)(29.939) + (0.9950)(0.0) = 1.3468$$

Because C(A=1) < C(A=0), A = 1 is selected as the CBIM recommendation for this node and the process proceeds to the next node (B) accordingly repeating the step until a decision node is reached. The results for each observed vector are shown in Table 17. From Table 17, for B, C(B=0) = 24.0242 and C(B=1) = 1.5853. Therefore, B = 1 is selected as the lowest cost alternative, and the process advances to the next true tree node (C). From Table 17, for C, C(C=0) = 27.8964 and C(C=1) = 1.2186. Therefore, C = 1 is selected as the lowest cost alternative, and the process advances to the next true tree node (D). From Table 17, for D, C(D=0) = 24.5915 and C(D=1) = 4.2358. Therefore, D = 0 is selected as the lowest cost alternative.

Observed					Cost
A	B	C	D	E	
0	0	0	0	0	3.0930
0	0	0	0	1	2.3797
0	0	0	1	0	1.2095
0	0	0	1	1	2.2956
0	0	1	0	0	1.1406
0	0	1	0	1	1.0348
0	0	1	1	0	1.0517
0	0	1	1	1	0.9744
0	1	0	0	0	1.1391
0	1	0	0	1	1.4135
0	1	0	1	0	1.5645
0	1	0	1	1	2.4931
0	1	1	0	0	2.6155
0	1	1	0	1	2.8336
0	1	1	1	0	3.5826
0	1	1	1	1	3.6465
1	0	0	0	0	4.1057
1	0	0	0	1	1.3882
1	0	0	1	0	1.2040
1	0	0	1	1	1.1874
1	0	1	0	0	1.1817
1	0	1	0	1	0.7928
1	0	1	1	0	1.0509
1	0	1	1	1	1.0000
1	1	0	0	0	0.9317
1	1	0	0	1	1.1265
1	1	0	1	0	1.7585
1	1	0	1	1	1.6608
1	1	1	0	0	1.2782
1	1	1	0	1	1.5949
1	1	1	1	0	4.8202
1	1	1	1	1	4.8570

	C(0)	C(1)
A	31.0071	1.3468
B	24.0242	1.5853
C	27.8964	1.2186
D	24.5915	4.2358

Table 17: CBIM cost calculations

Referencing the true tree, because *D* is a decision node, in step 3 the CBIM recommendation is Z.

Experimental Design

General

CBKM and CBIM performance were evaluated over varying degrees of distortion (noise), asymmetry (α) and class order (β). The control case is a symmetrical, unordered cost matrix with equal misclassification costs for all class decisions. That is, the control is the cost matrix where misclassification costs are (a) equal ($C(i_1,j_1) = C(i_2,j_2)$ for all i and j), (b) symmetrical ($\alpha = 1$), and (c) unordered ($\beta = 1$). Selecting C = 1 for all (meeting the condition that C > 0), produces the control cost matrix shown in Table 18. This cost matrix implies that all "correct" decisions carry a zero misclassification cost while all incorrect decisions carry a non-zero, but equal cost regardless of the decision.

	Actual: D1	Actual: D2	Actual: D3
Decision Made: D1	0	1	1
Decision Made: D2	1	0	1
Decision Made: D3	1	1	0

Table 18: Control case cost matrix

The CBKM and KM methodologies both modify a knowledge base to achieve an optimal solution. The difference is that KM does not take misclassification costs into account and consequently represents the CBKM case were those costs are consistently equal. Similarly, CBIM and IM both use an original, unmodified knowledge base to achieve optimal performance and differ only in the inclusion of misclassification costs.

CBKM was therefore compared with KM and CBIM was compared with IM using the same control cost matrix.

Both CBKM and CBIM were evaluated under varying degrees of input distortion (noise), asymmetry ($0 < \alpha \leq 1$), and degree of order in the decision classes ($\beta \geq 1$) using this control.

Distortion

Distortion in an input variable (γ) is defined as the probability its observed state differs from its true state. Distortion in any input is computed from a distortion matrix and marginal distribution for that input according to Equation 9 (Jiang et al., 2005).

$$\gamma = \sum_i P(True_i)(1 - P(Observed_i | True_i)) \tag{9}$$

Given a sample distortion matrix and marginal distribution of a ternary variable as in Table 19, the calculation of γ is as shown in Equation 10.

		Observed			
		X	Y	Z	Marginal Probability
True	X	P_{11}	P_{12}	P_{13}	P(X)
	Y	P_{21}	P_{22}	P_{23}	P(Y)
	Z	P_{31}	P_{32}	P_{33}	P(Z)

Table 19: Distortion matrix and marginal distribution of a ternary variable

$$\gamma = P(X)(1 - P_{11}) + P(Y)(1 - P_{22}) + P(Z)(1 - P_{33}) \tag{10}$$

In the experiments, the distortion level evaluated was set such that $0 < \gamma < 0.5$ and the distortion matrices formed as follows: First, the first (n – 1) diagonal elements in each ($n \times n$) distortion matrix were generated randomly. For a binary input, this equated to the single matrix element P_{11}. Element P_{12} was calculated as $P_{12} = (1 - P_{11})$. Given P_{11} and the selected distortion level, P_{22} was calculated from Equation 10 as shown in Equation 11.

$$P_{22} = 1 - \frac{\gamma - P(X)(1 - P_{11})}{P(Y)} \tag{11}$$

The final element P21 was calculated as $P_{21} = (1 - P_{22})$.

Asymmetry

Asymmetry is defined as the degree to which class error type costs differ. It captures the case where misclassification costs in the cost matrix vary by type such that $C(i, j) \neq C(j, i)$. It is represented by α as shown in Table 13 and varies in the range $0 < \alpha \leq 1$. In the experiments, it represented the case where the cost of making decision *X,* given that the true decision was *Z*, was different than making decision *Z*, given that the true decision was *X*.

Order

Ordered class decisions are those whose misclassification costs increase with the magnitude of the error. In a cost matrix, an ordered class is one whose misclassification costs $C(i, j)$ increase such that $C(i, j) \geq C(i-1, j)$ and $C(i, j) \geq C(i, j+1)$ where $i > j$, and $C(i, j) \geq C(i+1, j)$ and $C(i, j) \geq C(i, j-1)$ where $i < j$. In the experiments, this meant that the cost of making decision *X*, given that the true decision was *Z* (a "large error") was greater than making decision *Y*, given that the true decision was *Z* (a "small error").

Procedure

In the experiments, each simulated knowledge base contained five binary input variables with three possible outcomes. Because the experimental inputs were binary, there was no added value from evaluating distortion levels above 0.5, so distortion levels were selected in the range [0, 0.5] and held equal for all inputs.

A program was written to accomplish the following steps. First, a true knowledge base was generated by randomly choosing one of the three possible decisions for all possible combinations of the five input variables. The marginal distributions for the input vectors were determined by randomly setting some values and calculating the remainder. A true decision tree was generated from the true knowledge base. Next, a distortion level was chosen from the interval [0, 0.5] and applied either to all variables equally, or to one variable while holding all others at a constant positive value, to simulate both even and uneven distortion across the inputs. An asymmetry value (α) was chosen in the range [0.1, 1] at a 0.1 interval, and class order (β) was chosen in the range [1, 10]. For each experiment, one parameter was varied at a time within its range while the others were held constant in their range. The generated true tree was used in the CBIM method and a CBKM tree was generated using the five steps already discussed.

For each simulated test case, a **True** input vector was randomly generated. The true outcome for the test case was determined by running the **True** vector through the generated true tree. The **Observed** vector was generated by applying the selected distortion and the process was repeated 10,000 times for each set of parameters to simulate 10,000 different knowledge bases and user experiences.

The CBKM and CBIM methodologies were also evaluated using real world data taken from the University of California, Irvine machine learning repository. The data were preprocessed to remove duplicate and conflicting instances in addition to all instances with missing values, leaving 500 instances for testing. Continuous attributes were converted to ternary attributes and a true tree was generated using the data mining tool See5 (C5.0). Using these data, CBKM and CBIM were evaluated with distortion levels varying in 0.1 intervals with α and β set to constant values. Both techniques were similarly evaluated with α and β incrementally increased with other values remaining constant.

Data Analysis

The analysis first explored the performance difference between CBKM and KM. Because KM does not consider misclassification costs, it represents the case where the cost of making an incorrect decision is no different than the cost of making a correct decision. In the analysis, CBKM was compared with KM using a cost matrix with nonzero constant costs for all off-diagonal elements while diagonal elements where $i = j$ remained zero. CBIM was similarly compared with IM to determine if that method improved performance with respect to misclassification costs. Because neither KM nor IM incorporated any method for misclassification cost analysis, it was expected that CBKM would outperform KM over a wide range of misclassification cost parameters (α and β) and input distortion, and CBIM would similarly outperform IM. The following hypotheses summarize these expectations with respect to various input distortion levels.

Hypothesis 1. CBKM outperforms KM over a wide range of input distortion.

Hypothesis 2. CBKM outperforms the True Tree over a wide range of input distortion.

Hypothesis 3. CBIM outperforms IM over a wide range of input distortion.

Hypothesis 4. CBIM outperforms the True Tree over a wide range of input distortion.

Similarly, CBKM and CBIM were expected to outperform KM and IM with respect to differing cost matrix parameters (α and β) where performance was measured against misclassification costs. The follow hypotheses summarize these expectations.

Hypothesis 5. CBKM outperforms KM over a wide range of cost matrices.

Hypothesis 6. CBIM outperforms IM over a wide range of cost matrices.

The analysis was additionally interested in the performance difference between CBKM and CBIM with respect to misclassification costs. CBKM and CBIM were expected to respond differently because they handle input distortion differently. CBIM uses an original knowledge base without modification, whereas CBKM is permitted to modify its knowledge base to its advantage. In CBIM, a noisy input in the root node, or any node subsequently traversed, remains in the CBIM process. CBKM, on the other hand, can eliminate an especially noisy input in its restructured knowledge base. Therefore, CBKM was expected to outperform CBIM as input distortion levels increased. This expectation with respect to noise is summarized in the final hypothesis.

Hypothesis 7. The performance of CBIM compared with CBKM will deteriorate with increases in noise.

Chapter 4

Results

General

The experimental results from the CBKM and CBIM methods were compared with the same inputs to the extant KM and IM methods, and the True Tree. As is to be expected, CBKM consistently outperformed KM and the True Tree over a wide range of input distortion and cost matrix α and β when the evaluation criterion was error cost. The performance difference between CBIM, and IM and the True Tree was less pronounced. In this chapter, the cost matrix will be discussed first, followed by the CBKM and CBIM results using experimental data. Results using real world data will follow before an explanation of the findings.

Control Cost Matrix

The control cost matrix is as shown in Table 18. Using the control, the CBKM and CBIM methodologies produced the same results as the KM and IM implementations, because the control guarantees zero cost for correct classifications and a uniform cost for misclassifications. From Equation 8, the CBKM and CBIM methodologies seek minimized misclassification costs through a summation of products of $P(j|x)C(i,j)$. Where $i = j$, there are no misclassification costs and the product resolves to zero. In the control, where $i \neq j$, $C(i, j) = 1$ and the product resolves to include only probabilities. Using the control cost matrix, in attempting to minimize these errors (costs) CBKM and CBIM minimize the probability of an error, which is logically equivalent to the KM and IM goal of maximizing the probability of *not* making an error. Experimental results with varying levels of distortion in the range [0, 0.5] are shown in Table 20. As indicated, there are no cost differences between KM and CBKM, and IM and CBIM.

Distortion	KM	CBKM	IM	CBIM
0.0	0.3761	0.3761	0.3093	0.3093
0.1	0.4726	0.4726	0.4548	0.4548
0.2	0.5420	0.5420	0.5505	0.5505
0.3	0.5835	0.5835	0.5980	0.5980
0.4	0.6187	0.6187	0.6380	0.6380
0.5	0.6270	0.6270	0.6465	0.6465

Table 20: Misclassification error costs — control cost matrix result

In the experiments, various cost matrices were used. Starting from the control cost matrix where both α and β were equal to one, α and β were tested at varying degrees as they deviated from one. The magnitude of β was increased from the control value of one in the sequence {1, 2, 3, …}, with larger values representing greater degrees of order. The magnitude of α was also increased from the control in the sequence {1.0, 0.9, 0.8, …} with a greater deviation from the control representing greater asymmetry.

Experimental Data

Hypothesis One

The following *t*-test was conducted to test hypotheses one, that CBKM outperforms KM over a wide range of input distortion.

$$\begin{cases} H_0: \ \Delta C_{KM-CBKM} = 0 \\ H_1: \ \Delta C_{KM-CBKM} > 0 \end{cases}$$

Here, $\Delta C_{KM\text{-}CBKM}$ is the misclassification cost difference between KM and CBKM. That is, the null hypothesis (H_0) claimed that the misclassification cost difference between KM and CBKM was zero, while the research hypothesis (H_1) claimed that the difference was not zero. For this test, 10,000 iterations were recorded at each input distortion level, simulating that many users' inputs to the expert system. Distortion was varied in 0.1 increments in the range [0, 0.5] across all inputs with α constant at 0.1 and β constant at 10. The *t*-test was repeated at each input distortion level, providing an assessment of statistical significance at each input distortion level individually. The experimental results are shown in Figure 5 and the *t*-test results are shown in Table 21.

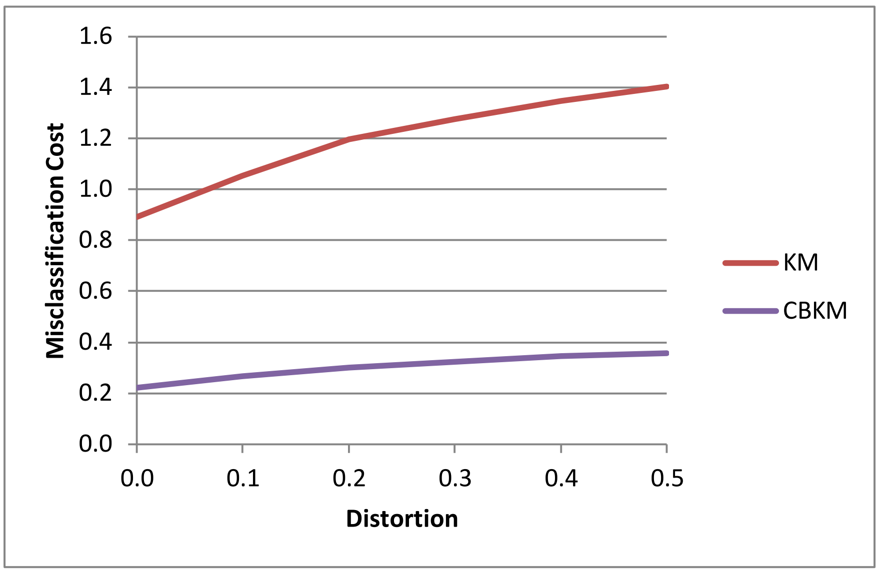


Figure 5: CBKM vs KM — varying distortion, all inputs

Distortion	0	0.1	0.2	0.3	0.4	0.5
Asymmetry (α)	0.1	0.1	0.1	0.1	0.1	0.1
Order (β)	10	10	10	10	10	10
Sample Average ($\bar{X}$)	0.2215	0.2755	0.3108	0.3275	0.3433	0.3621
Reference (μ_0)	0.8426	1.0516	1.2197	1.2625	1.3171	1.3964
Standard Error ($S_{\bar{X}}$)	0.0222	0.0244	0.0261	0.0264	0.0269	0.0276
t-statistic	-27.96	-31.80	-34.77	-35.43	-36.24	-37.46

Table 21: CBKM vs KM — varying distortion, all inputs

In this case, H_0 was strongly rejected with a p-value less than 0.001 ($|t\text{-statistic}| > 3.291$—all t-statistic critical values provided in Siegel (1990)) for all distortion levels. As expected, CBKM outperformed KM over a wide range of input distortion.

In the experiment, because distortion was set equal for each of the five binary inputs, it evaluated a system where no single input was more prone to errors than another. A separate experiment was conducted with all inputs but one held at a constant distortion level of 0.4. The final input was varied in [0, 0.5] in 0.1 increments to simulate a system with uneven input distortion. The experimental results are shown in Figure 6. The t-test was repeated with the results shown in Table 22. As expected, CBKM outperformed KM where distortion varied unevenly across the inputs and H_0 was strongly rejected with a p-value less than 0.001.

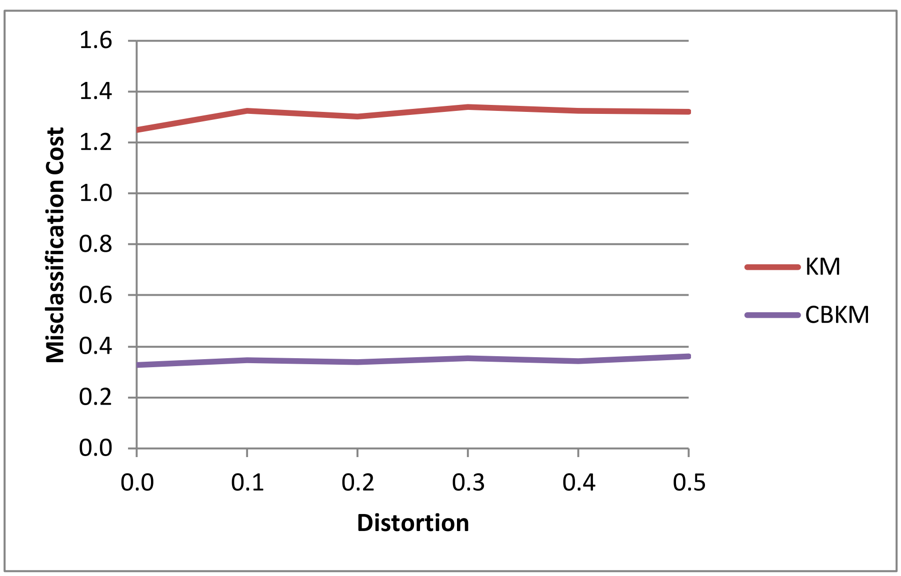


Figure 6: CBKM vs KM — varying distortion, one input

Distortion	0	0.1	0.2	0.3	0.4	0.5
Asymmetry (α)	0.1	0.1	0.1	0.1	0.1	0.1
Order (β)	10	10	10	10	10	10
Sample Average ($\bar{X}$)	0.3323	0.3268	0.3467	0.3421	0.3457	0.3553
Reference (μ_0)	1.3174	1.2927	1.4049	1.3018	1.3299	1.3943
Standard Error ($S_{\bar{X}}$)	0.0270	0.0267	0.0278	0.0266	0.0269	0.0277
t-statistic	-36.52	-36.22	-38.03	-36.08	-36.55	-37.56

Table 22: CBKM vs KM — varying distortion, one input

Hypothesis Two

The following *t*-test was conducted to test hypothesis two, that CBKM outperforms the True Tree over a wide range of input distortion.

$$\begin{cases} H_0\colon\ \Delta C_{True-CBKM} = 0 \\ H_1\colon\ \Delta C_{True-CBKM} > 0 \end{cases}$$

Here, $\Delta C_{True\text{-}CBKM}$ is the misclassification cost difference between the True Tree and CBKM. The null hypothesis (H_0) claimed that the difference in misclassification costs between CBKM and a True Tree was zero, while the research hypothesis (H_1) claimed the difference was not zero. As in the previous test, distortion was first varied in 0.1 increments in the range [0. 0.5] across all inputs with α constant at 0.1 and β constant at 10. Ten thousand iterations were recorded at each distortion level, with the misclassification cost summary shown in Figure 7 and the *t*-test results shown in Table 23. As expected, CBKM outperformed the True Tree for all distortion levels greater than zero and H_0 was strongly rejected with a *p*-value less than 0.001 for all distortion levels. In each case, a negative *t*-statistic indicates CBKM outperforming KM $\left(\overline{C(CBKM)} < \overline{C(KM)}\right)$ while a positive *t*-statistic indicates underperformance.

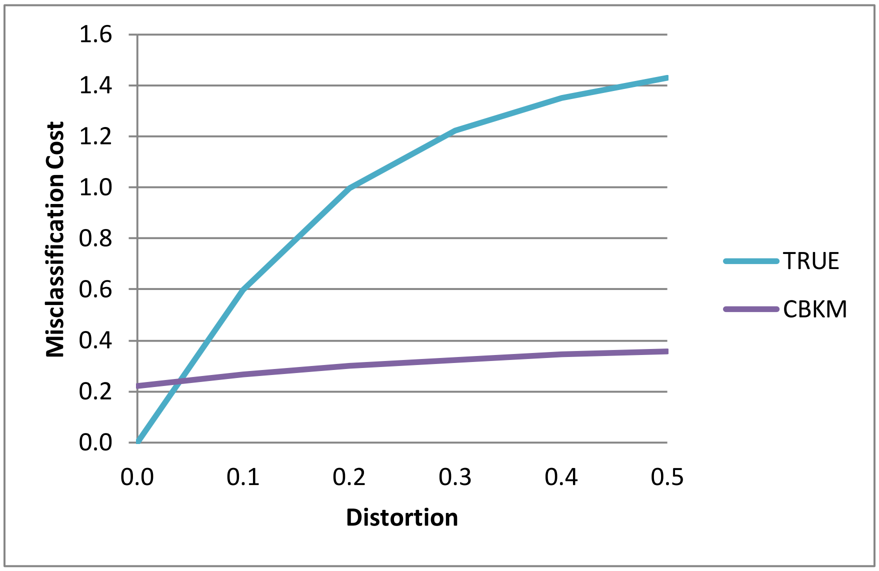


Figure 7: CBKM vs True Tree — varying distortion, all inputs

Distortion	0	0.1	0.2	0.3	0.4	0.5
Asymmetry (α)	0.1	0.1	0.1	0.1	0.1	0.1
Order (β)	10	10	10	10	10	10
Sample Average ($\bar{X}$)	0.2215	0.2755	0.3108	0.3275	0.3433	0.3621
Reference (μ_0)	0.0000	0.6185	1.0078	1.2202	1.3171	1.4506
Standard Error ($S_{\bar{X}}$)	0.0045	0.0207	0.0253	0.0272	0.0282	0.0293
t-statistic	49.47	-16.54	-27.57	-32.77	-35.28	-37.20

Table 23: CBKM vs True Tree — varying distortion, all inputs

As shown in the graph, in the absence of distortion (zero distortion), there are no misclassification errors, so the True Tree produces correct results for every input and zero misclassification costs.

CBKM was further compared with True Tree results with distortion held constant at 0.4 across all binary inputs except one. The final input was varied in 0.1 increments in the range [0, 0.5] with α remaining at 0.1 and β at 10. As expected, H_0 was strongly rejected ($p < 0.001$) as CBKM outperformed the True Tree with uneven distortion across the inputs. The results are summarized in the graph in Figure 8 and the *t*-test results are shown in Table 24.

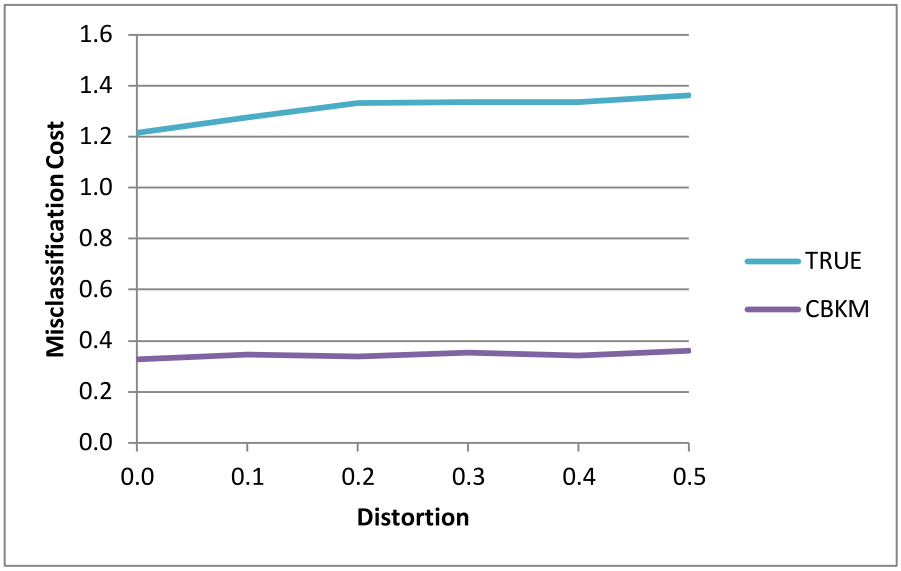


Figure 8: CBKM vs True Tree — varying distortion, one input

Distortion	0	0.1	0.2	0.3	0.4	0.5
Asymmetry (α)	0.1	0.1	0.1	0.1	0.1	0.1
Order (β)	10	10	10	10	10	10
Sample Average ($\bar{X}$)	0.3323	0.3268	0.3467	0.3421	0.3458	0.3553
Reference (μ_0)	1.2798	1.2658	1.3575	1.3589	1.3299	1.3977
Standard Error ($S_{\bar{X}}$)	0.0278	0.0277	0.0286	0.0286	0.0285	0.0288
t-statistic	-34.13	-33.94	-35.34	-35.54	-35.57	-36.14

Table 24: CBKM vs True Tree — varying distortion, one input

Hypothesis Three

The following *t*-test was conducted to test hypotheses three, that CBIM outperforms IM over a wide range of input distortion.

$$\begin{cases} H_0: \ \Delta C_{IM-CBIM} = 0 \\ H_1: \ \Delta C_{IM-CBIM} > 0 \end{cases}$$

Here, $\Delta C_{IM\text{-}CBIM}$ is the misclassification cost difference between IM and CBIM. The null hypothesis (H_0) claimed that the misclassification cost difference between CBIM and IM was zero while the research hypothesis (H_1) claimed the difference was not zero. The same experiments were conducted as with hypotheses one and two, with varying distortion, α, and β, with varying results. As before, 10,000 iterations were recorded at each step and *t*-tests conducted for each level of distortion, α, or β.

In the first case, distortion for all inputs was varied in 0.1 increments in the range [0, 0.5] with α constant at 0.1 and β constant at 10.

For hypothesis three, H_0 was rejected for all distortion values ($p < 0.05$, |*t*-statistic| > 1.960) and strongly rejected for all values greater than zero ($p < 0.001$, |*t*-statistic| > 3.291) as CBIM outperformed IM. The results are shown in Figure 9 and the *t*-test results displayed in Table 25.

As shown in Figure 9, the difference between CBIM and IM misclassification costs lessens as distortion approaches zero as both methodologies use the original true tree to produce similar results. The greater the distortion, however, the greater the difference in misclassification costs.

Hypothesis three was also tested using uneven distortion in the inputs. For this test, distortion was held constant at 0.4 for all inputs except one, which was incremented in the range [0, 0.5] while α remained constant at 0.1 and β remained at 10. For all

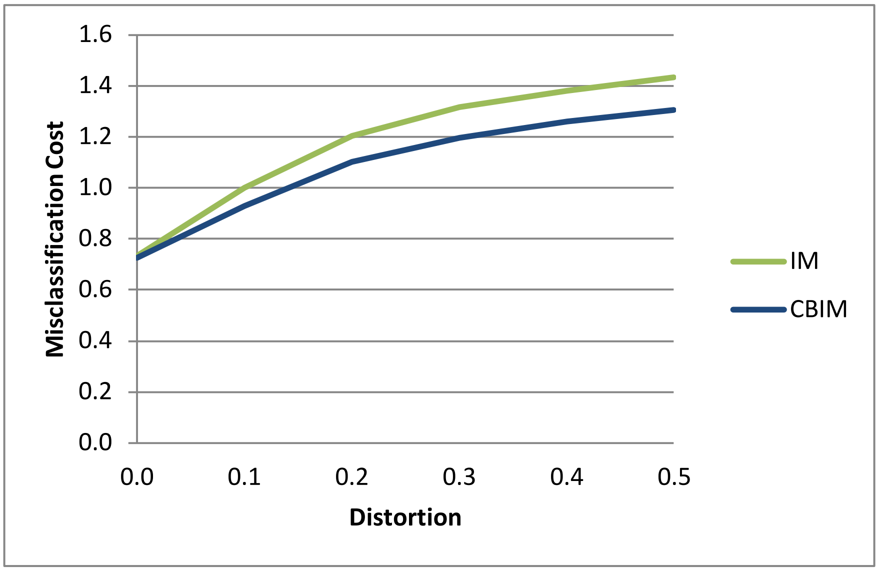


Figure 9: CBIM vs IM — varying distortion, all inputs

Distortion	0	0.1	0.2	0.3	0.4	0.5
Asymmetry (α)	0.1	0.1	0.1	0.1	0.1	0.1
Order (β)	10	10	10	10	10	10
Sample Average ($\bar{X}$)	0.6566	0.9334	1.1295	1.1659	1.2555	1.2770
Reference (μ_0)	0.6819	1.0035	1.2203	1.3099	1.3824	1.4197
Standard Error ($S_{\bar{X}}$)	0.0121	0.0128	0.0136	0.0139	0.0137	0.0147
t-statistic	-2.10	-5.48	-6.69	-10.37	-9.26	-9.72

Table 25: CBIM vs IM — varying distortion, all inputs

distortion levels, H_0 was strongly rejected ($p < 0.001$) as CBIM outperformed IM. The experimental results are shown in Figure 10 and the *t*-test results are shown in Table 26.

As shown in Figure 10, the CBIM and IM estimations do not converge with uneven distortion because, with four inputs set at a distortion level of 0.4, total input distortion never approaches zero.

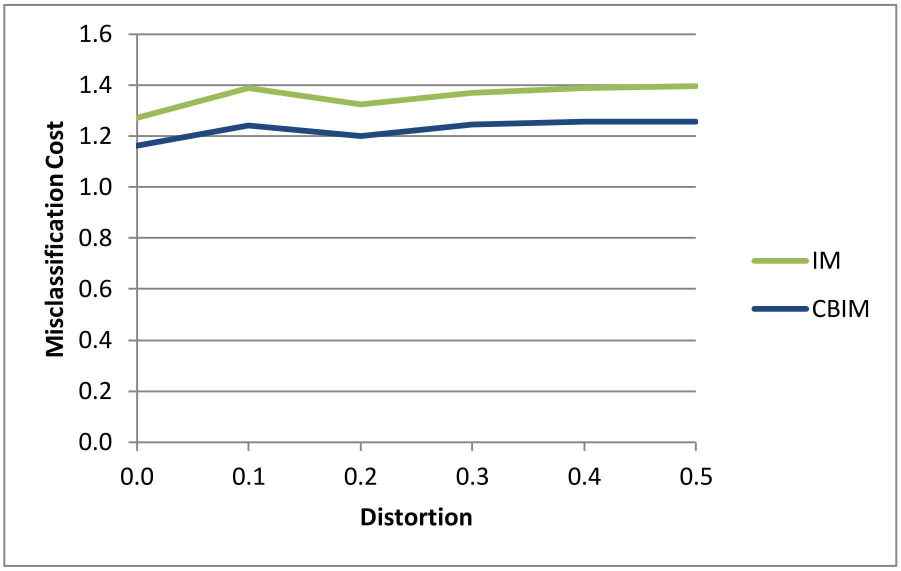


Figure 10: CBIM vs IM — varying distortion, one input

Distortion	0	0.1	0.2	0.3	0.4	0.5
Asymmetry (α)	0.1	0.1	0.1	0.1	0.1	0.1
Order (β)	10	10	10	10	10	10
Sample Average ($\bar{X}$)	1.2043	1.2016	1.2971	1.2680	1.2724	1.2975
Reference (μ_0)	0.0000	1.3212	1.4170	1.3818	1.3849	1.4260
Standard Error ($S_{\bar{X}}$)	0.0157	0.0143	0.0138	0.0141	0.0141	0.0136
t-statistic	-9.78	-8.37	-8.72	-8.07	-8.00	-9.44

Table 26: CBIM vs IM — varying distortion, one input

Hypothesis Four

The following *t*-test was conducted to test hypothesis four, that CBIM outperforms a True Tree over a wide range of input distortion.

$$\begin{cases} H_0\colon \ \Delta C_{True-CBIM} = 0 \\ H_1\colon \ \Delta C_{True-CBIM} > 0 \end{cases}$$

Here, $\Delta C_{True\text{-}CBIM}$ is the misclassification cost difference between the True Tree and CBIM. The null hypothesis (H_0) claimed that the misclassification cost difference between CBIM and a True Tree was zero while the research hypothesis (H_1) claimed the difference was not zero. The same experiments were performed with varying distortion, α, and β, with 10,000 iterations recorded at each level. The experimental results are summarized in Figure 11 and the *t*-test results are shown in Table 27. In this case, H_0 was rejected for all distortion values ($p < 0.05$, $|t\text{-statistic}| > 1.960$) and strongly rejected for all values above and below 0.3 ($p < 0.001$, $|t\text{-statistic}| > 3.291$). Looking at Figure 11, a distortion level of 0.3 is the level at which CBIM performance surpasses the True Tree, indicating CBIM consistently outperforms the True Tree only at higher distortion levels (distortion > 0.3).

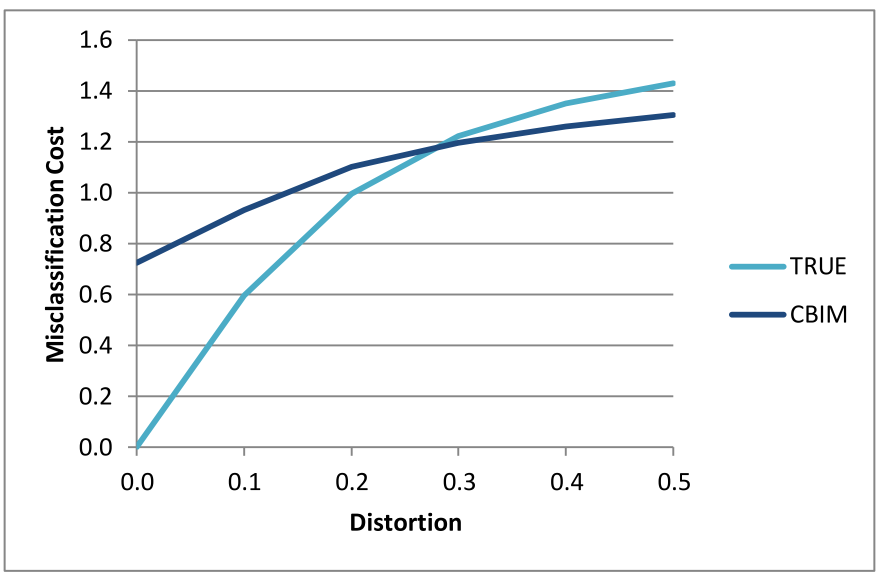


Figure 11: CBIM vs True — varying distortion, all inputs

Distortion	0	0.1	0.2	0.3	0.4	0.5
Asymmetry (α)	0.1	0.1	0.1	0.1	0.1	0.1
Order (β)	10	10	10	10	10	10
Sample Average ($\bar{X}$)	0.6844	0.9334	1.1295	1.1659	1.2334	1.2770
Reference (μ_0)	0.0000	0.6185	1.0078	1.2202	1.3408	1.4506
Standard Error ($S_{\bar{X}}$)	0.0215	0.0241	0.0254	0.0254	0.0254	0.0269
t-statistic	31.86	13.10	4.80	-2.14	-3.32	-6.46

Table 27: CBIM vs True Tree — varying distortion, all inputs

The *t*-test results indicate there is a statistically significant difference between CBIM and True Tree misclassification costs at each input distortion level in the range [0.1, 0.5]. Positive *t*-statistic values correspond to distortion levels where CBIM underperforms the True Tree while negative values indicate CBIM outperforms. This result is explained in the *Findings* section below.

Hypothesis Five

CBKM was evaluated with varying levels of α and β in the cost matrix to test hypothesis five, that CBKM outperforms KM over a wide range of cost matrices. The same *t*-tests were conducted as for hypothesis one.

$$\begin{cases} H_0\colon \Delta C_{KM-CBKM} = 0 \\ H_1\colon \Delta C_{KM-CBKM} > 0 \end{cases}$$

The null hypothesis (H_0) claimed that the misclassification cost difference between CBKM and KM was zero in the case of varying α and β in the cost matrix. The research hypothesis (H_1) claimed the misclassification cost difference was not zero. Figure 12 shows the experimental results with 10,000 iterations at each step with α varying in increments of 0.1 in [0.1, 1.0]. Distortion was held constant across all inputs at 0.4 and β was held at 10.

H_0 was strongly rejected with $p < 0.001$ for all values of α as CBKM outperformed KM, as shown in Table 28.

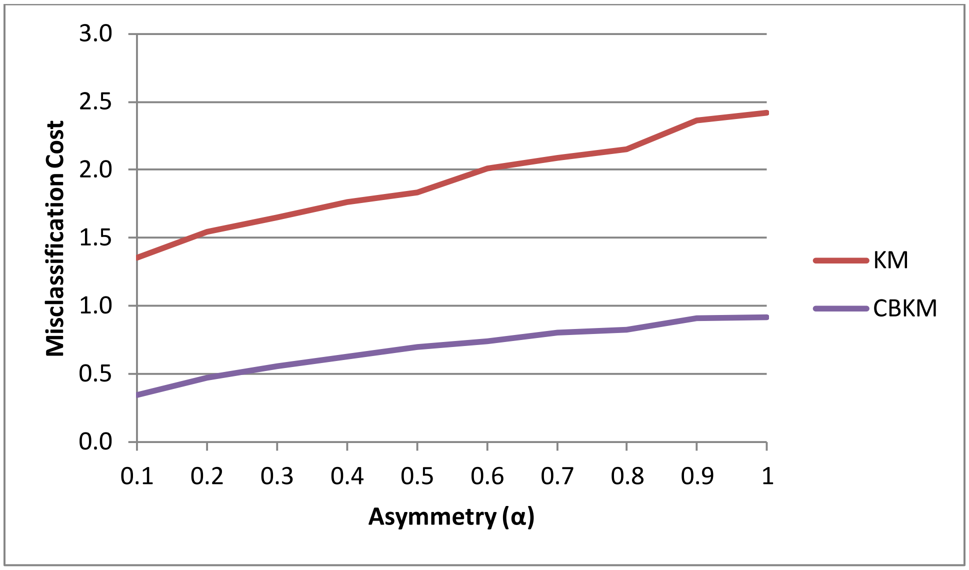


Figure 12: CBKM vs KM — varying asymmetry

Distortion	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4
Asymmetry (α)	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
Order (β)	10	10	10	10	10	10	10	10	10	10
Sample Average ($\bar{X}$)	0.3400	0.4768	0.5685	0.6278	0.6985	0.7407	0.7758	0.8346	0.9090	0.9374
Reference (μ_0)	1.3006	1.4461	1.6155	1.7404	1.8283	1.9586	2.0629	2.2013	2.3225	2.4776
Standard Error ($S_{\bar{X}}$)	0.0266	0.0269	0.0277	0.0280	0.0281	0.0293	0.0302	0.0318	0.0330	0.0351
t-statistic	-36.12	-36.05	-37.85	-39.70	-40.18	-41.58	-42.65	-43.00	-42.89	-43.88

Table 28: CBKM vs KM — varying asymmetry

Figure 13 shows the results with α held constant at 0.1 and β varied in increments of 1.0 in the range [1, 10]. As expected, H_0 was strongly rejected for all levels of varying β ($p < 0.001$) as CBKM outperformed KM. The results for the *t*-tests are shown in Table 29.

Figure 13: CBKM vs KM — varying order

Distortion	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4
Asymmetry (α)	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
Order (β)	1	2	3	4	5	6	7	8	9	10
Sample Average ($\bar{X}$)	0.0945	0.1450	0.1787	0.2151	0.2402	0.2695	0.2919	0.3118	0.3331	0.3541
Reference (μ_0)	0.3339	0.4466	0.5433	0.6728	0.7993	0.8787	1.0187	1.1367	1.2441	1.3722
Standard Error ($S_{\bar{X}}$)	0.0045	0.0064	0.0086	0.0112	0.0139	0.0162	0.0190	0.0219	0.0245	0.0273
t-statistic	-53.72	-46.97	-42.48	-40.76	-40.33	-37.69	-38.18	-37.72	-37.19	-37.28

Table 29: CBKM vs KM — varying order

Hypothesis Six

The following *t*-test was conducted to test hypothesis six, that CBIM outperforms IM over a wide range of cost matrices.

$$\begin{cases} H_0\colon \Delta C_{IM-CBIM} = 0 \\ H_1\colon \Delta C_{IM-CBIM} > 0 \end{cases}$$

This is the same *t*-test as for hypothesis three where the null hypothesis (H_0) claimed the misclassification cost difference between CBIM and IM was zero and the research hypothesis (H_1) claimed the misclassification cost difference was not zero. To test the hypothesis, α was varied in [0.1, 1.0] while β was held constant at 10. The experimental results are shown in Figure 14 and the *t*-test results are in Table 30.

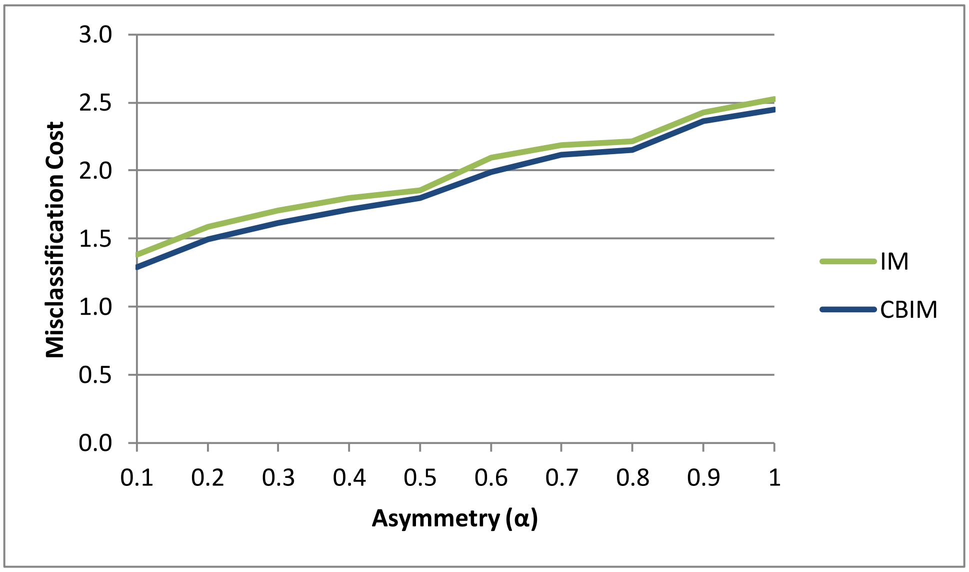


Figure 14: CBIM vs IM — varying asymmetry

Distortion	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4
Asymmetry (α)	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
Order (β)	10	10	10	10	10	10	10	10	10	10
Sample Average ($\bar{X}$)	1.2357	1.4152	1.5808	1.7041	1.8289	1.9373	2.0721	2.1774	2.3253	2.4387
Reference (μ_0)	1.3598	1.5041	1.6502	1.7915	1.8880	2.0090	2.1588	2.2643	2.4241	2.5396
Standard Error ($S_{\bar{X}}$)	0.0141	0.0132	0.0136	0.0128	0.0126	0.0135	0.0130	0.0146	0.0150	0.0159
t-statistic	-8.78	-6.73	-5.11	-6.82	-4.68	-5.31	-6.67	-5.95	-6.58	-6.35

Table 30: CBIM vs IM — varying asymmetry

As expected, CBIM consistently outperforms IM with varying asymmetry in the cost matrix and H_0 is strongly rejected ($p < 0.001$).

Hypothesis six was further tested with β varied in [1, 10] and α held constant at 0.1. The experimental results are shown in Figure 15 and the *t*-test results are shown in Table 31.

Figure 15: CBIM vs IM — varying order

As expected, CBIM outperforms IM with varying order in the cost matrix, and H_0 was strongly rejected for all values.

Hypothesis Seven

Hypothesis seven is concerned with the performance difference between the new CBKM and CBIM methodologies as input distortion varies. The following hypothesis was tested to assess the statistical significance of the difference between the two techniques.

Distortion	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4
Asymmetry (α)	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
Order (β)	1	2	3	4	5	6	7	8	9	10
Sample Average ($\bar{X}$)	0.3406	0.4440	0.5426	0.6468	0.7865	0.8187	0.9639	1.0600	1.1795	1.3160
Reference (μ_0)	0.3483	0.4562	0.5685	0.6812	0.8390	0.8914	1.0339	1.1535	1.2673	1.4117
Standard Error ($S_{\bar{X}}$)	0.0012	0.0019	0.0031	0.0044	0.0064	0.0081	0.0089	0.0112	0.0120	0.0138
t-statistic	-6.49	-6.38	-8.46	-7.75	-8.14	-9.00	-7.88	-8.36	-7.31	-6.96

Table 31: CBIM vs IM — varying order

$$\begin{cases} H_0:\ \Delta C_{CBIM-CBKM} = 0 \\ H_1:\ \Delta C_{CBIM-CBKM} > 0 \end{cases}$$

Here, $\Delta C_{CBIM-CBKM}$ is the misclassification cost difference between the results of CBIM and CBKM. The null hypothesis (H_0) claimed that the misclassification cost difference between CBKM and CBIM as distortion varied was zero while the research hypothesis (H_1) claimed that the misclassification cost difference was not zero. An experiment was conducted with distortion varied across all inputs at 0.1 intervals in the range [0, 0.5], with α constant and 0.1 and β constant at 10. The experimental results are shown in Figure 16 and the *t*-test results are shown in Table 32.

As expected, CBKM outperformed CBIM over a wide range of input distortion with the most pronounced rate of change for smaller values. As hypothesis seven deals with the rate of performance degradation as distortion increases, the misclassification cost difference between CBKM and CBIM at zero distortion are selected as a reference. The percentage cost increase from the reference at each distortion level is shown in Table 33.

As indicated, CBIM performance deteriorated at a higher rate than CBKM over the input distortion range [0, 0.5] and H_0 was strongly rejected ($p < 0.001$).

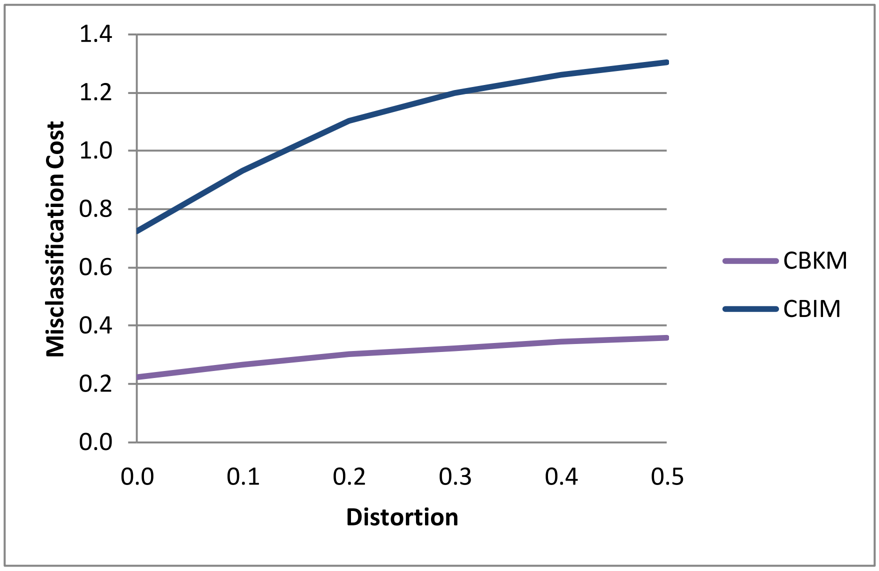


Figure 16: CBKM vs CBIM — varying distortion, all inputs

Distortion	0	0.1	0.2	0.3	0.4	0.5
Asymmetry (α)	0.1	0.1	0.1	0.1	0.1	0.1
Order (β)	10	10	10	10	10	10
Sample Average ($\bar{X}$)	0.6566	0.9334	1.1295	1.1659	1.2334	1.2770
Reference (μ_0)	0.2167	0.2755	0.3108	0.3275	0.3313	0.3621
Standard Error ($S_{\bar{X}}$)	0.0194	0.0230	0.0252	0.0253	0.0259	0.0264
t-statistic	-22.71	-28.56	-32.54	-33.12	-34.82	-34.70

Table 32: CBKM vs CBIM — varying distortion, all inputs

Distortion	CBKM Cost	CBIM Cost	CBKM % increase	CBIM % increase
0.0	0.2230	0.7242	0.00	0.00
0.1	0.2656	0.9311	4.26	20.69
0.2	0.3023	1.1033	7.92	37.91
0.3	0.3238	1.1975	10.08	47.33
0.4	0.3445	1.2609	12.14	53.67
0.5	0.3581	1.3054	13.51	58.13

Table 33: CBKM vs CBIM degradation

Real World Data

Real world data were provided from the University of California, Irvine Machine Learning Repository at http://archive.ics.uci.edu/ml/datasets/Credit+Approval. The dataset was robust in that it offered discrete and continuous input variables, inputs with from two to 14 variables, missing values, and conflicting data. For the purpose of these experiments, all continuous variables were categorized as ternary inputs and assigned values of H, M, or L based on the distribution of continuous values within the range of all values for that variable. For example, an input dataset of the five values {1, 1, 1, 3, 5} would have mapped to the set {L, L, L, M, H}. Duplicate, missing and conflicting (identical inputs with disparate results) were ignored. With this preprocessing, the dataset resolved to 500 inputs.

The True Tree and decision table were generated using the data mining tool See5, provided by Rulequest Research under the GNU General Public License. This scaled down (non-commercial) version further limited the input dataset to 400 instances. The generated True Tree is shown in Appendix B and the corresponding decision table is in Appendix C. This True Tree/decision table provided results for 99.93% of the

10,450,944 possible inputs. The 0.07% of inputs not solvable by the decision tree were ignored.

The seven hypotheses were tested using the same experiments as for the experimental data. In each case, 10,000 user inputs were randomly generated and fed to each of the methodologies for each of the selected variables and parameters, simulating that many users' inputs to each system. As before, *t*-tests were conducted at each evaluated variable or parameter level.

Hypothesis One

The null hypothesis (H_0) claimed that the difference in misclassification costs between CBKM and KM with varying distortion was zero while the research hypothesis (H_1) claimed that the difference was not zero. The experimental results with varying input distortion in [0.1, 0.5], and constant α (0.1) and β (10) are shown in Figure 17 while the *t*-test results are shown in Table 34. Consistent with the experimental data, H_0 was strongly rejected ($p < 0.001$) at all distortion levels as CBKM outperformed KM.

Hypothesis Two

The null hypothesis (H_0) claimed that the difference in misclassification costs between CBKM and a True Tree with varying distortion was zero while the research hypothesis (H_1) claimed that the difference was not zero. The experimental results with varying distortion are shown in Figure 18 and the results of the *t*-tests are in Table 35. As expected, H_0 was strongly rejected ($p < 0.001$) for all levels of input distortion above and below 0.1. Identical to the experimental results and as expected, the real data True Tree produced zero errors and associated misclassification costs in the absence of distortion, so outperformed all other methods at that point. The single distortion level (0.1) where

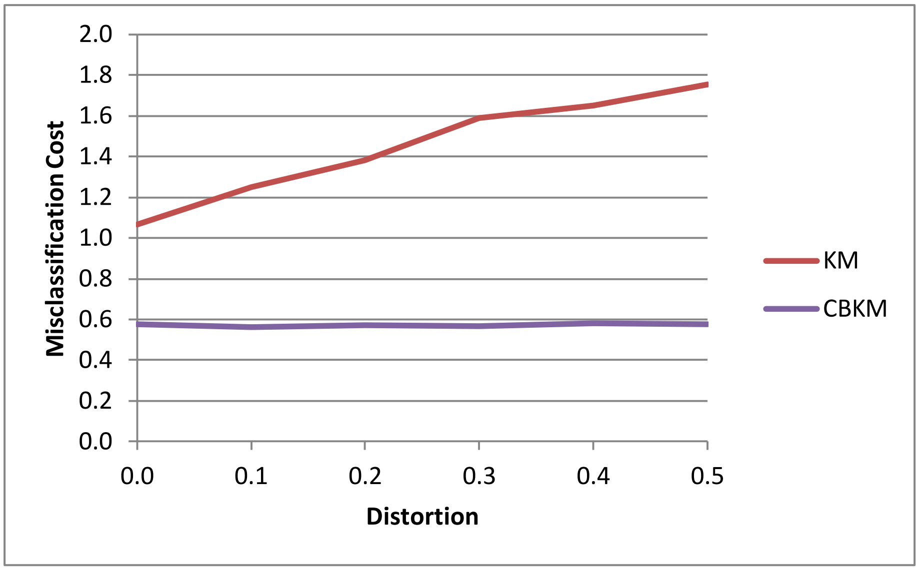


Figure 17: CBKM vs KM — real world data, varying distortion

Distortion	0	0.1	0.2	0.3	0.4	0.5
Asymmetry (α)	0.1	0.1	0.1	0.1	0.1	0.1
Order (β)	10	10	10	10	10	10
Sample Average ($\bar{X}$)	0.5800	0.5890	0.5710	0.5860	0.5680	0.5610
Reference (μ_0)	1.1170	1.1420	1.2870	1.4440	1.7330	1.9310
Standard Error ($S_{\bar{X}}$)	0.0936	0.0946	0.1006	0.1047	0.1155	0.1208
t-statistic	-5.74	-5.85	-7.12	-8.20	-10.09	-11.34

Table 34: CBKM vs KM — real world data, varying distortion

there was no statistically significant difference (|*t*-statistic| < 1.960) between CBKM and True Tree estimations was at the crossover point between CBKM underperformance and CBKM outperforming the True Tree.

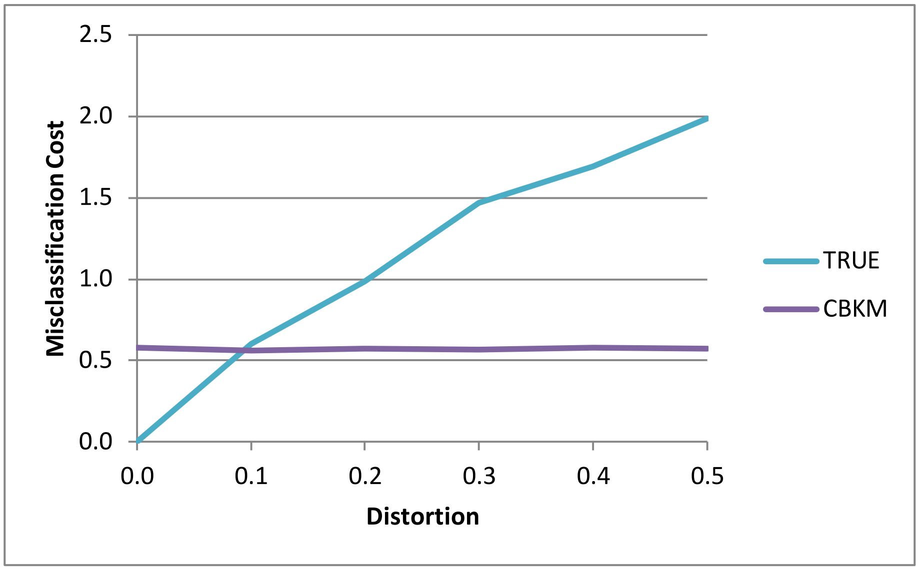


Figure 18: CBKM vs True Tree — real world data, varying distortion

.

Distortion	0	0.1	0.2	0.3	0.4	0.5
Asymmetry (α)	0.1	0.1	0.1	0.1	0.1	0.1
Order (β)	10	10	10	10	10	10
Sample Average ($\bar{X}$)	0.5800	0.5890	0.5710	0.5860	0.5680	0.5610
Reference (μ_0)	0.0000	0.6100	1.0630	1.4690	1.7330	2.1780
Standard Error ($S_{\bar{X}}$)	0.0156	0.0784	0.0997	0.1134	0.1275	0.1325
t-statistic	37.14	-0.27	-4.93	-7.78	-10.71	-12.20

Table 35: CBKM vs True Tree — real world data, varying distortion

Hypothesis Three

The null hypothesis (H_0) claimed that the difference in misclassification costs between CBIM and IM given varying distortion was zero while the research hypothesis (H_1) claimed that the difference was not zero. The experimental results with varying input distortion in [0.1, 0.5], with constant α (0.1) and β (10) are shown in Figure 19 while the *t*-test results are shown in Table 36. Consistent with the experimental results, H_0 was strongly rejected ($p < 0.001$) as CBIM outperformed IM at all distortion levels.

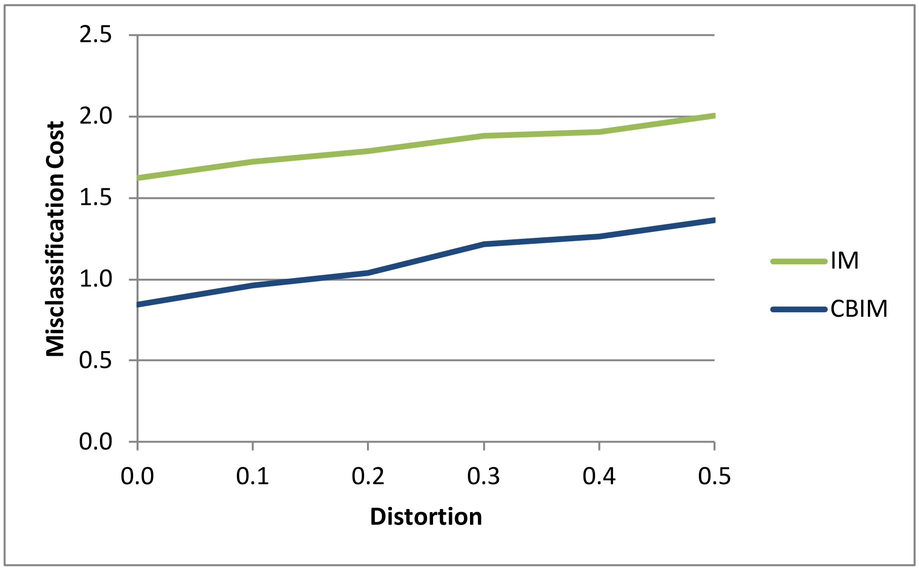


Figure 19: CBIM vs IM — real world data, varying distortion

Distortion	0	0.1	0.2	0.3	0.4	0.5
Asymmetry (α)	0.1	0.1	0.1	0.1	0.1	0.1
Order (β)	10	10	10	10	10	10
Sample Average ($\bar{X}$)	0.9060	0.8840	0.9820	1.1220	1.4000	1.5170
Reference (μ_0)	1.5780	1.7280	1.6940	1.8220	1.9880	2.0190
Standard Error ($S_{\bar{X}}$)	0.0865	0.0954	0.0917	0.0951	0.0888	0.0910
t-statistic	-7.77	-8.84	-7.77	-7.36	-6.62	-5.51

Table 36: CBIM vs IM — real world data, varying distortion

Hypothesis Four

The null hypothesis (H_0) claimed that the difference in misclassification costs between CBIM and a True Tree given varying distortion was zero while the research hypothesis (H_1) claimed that the difference was not zero. The experimental results with varying input distortion are shown in Figure 20 and the *t*-test results are shown in Table 37. Consistent with the experimental data, in the absence of input distortion, the True Tree produced correct results and zero misclassification costs.

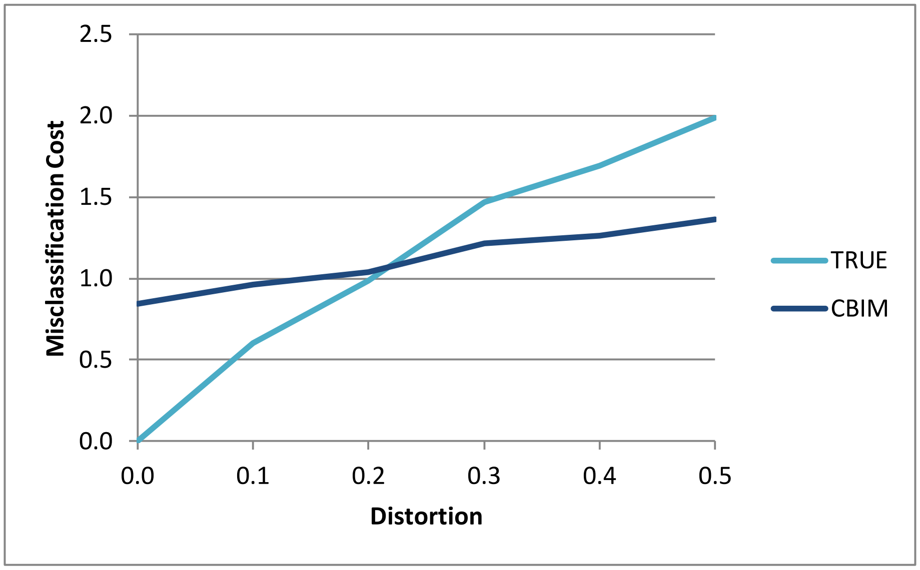


Figure 20: CBIM vs True Tree — real world data, varying distortion

Distortion	0	0.1	0.2	0.3	0.4	0.5
Asymmetry (α)	0.1	0.1	0.1	0.1	0.1	0.1
Order (β)	10	10	10	10	10	10
Sample Average ($\bar{X}$)	0.9060	0.8840	0.9820	0.7512	1.4000	1.5170
Reference (μ_0)	0.0000	0.6100	1.0630	1.2072	1.7330	2.1780
Standard Error ($S_{\bar{X}}$)	0.0704	0.0834	0.1034	0.0712	0.1154	0.1213
t-statistic	12.88	3.29	-0.78	-6.40	-4.63	-5.45

Table 37: CBIM vs True Tree — real world data, varying distortion

Therefore, with the real world data, CBIM underperformed the True Tree at very low distortion levels (< 0.2) and outperformed at all others. There is a statistically significant difference ($p < 0.002$) between CBIM and True Tree misclassification costs at all distortion levels except the crossover point, just as with the experimental data.

Hypothesis Five

The null hypothesis (H_0) claimed that the misclassification cost difference between CBKM and KM was zero over a wide range of cost matrices while the research hypothesis (H_1) claimed that the difference was not zero. This hypothesis was tested for both varying α in the range [0.1, 1.0] with distortion held constant at 0.4 and β constant at 10, and for varying β with distortion held constant at 0.4 and α constant at 0.1. The experimental results are shown in Figures 21 and 22 while the *t*-test results are shown in Tables 38 and 39.

As shown, the real world CBKM results were more pronounced with greater deviation from the control cost matrix α and β values of one. H_0 was rejected for all α values less than 0.6 ($p < 0.001$) and also for all values of β, all of which are greater than one.

Hypothesis Six

The null hypothesis (H_0) claimed that the misclassification cost difference between CBIM and IM was zero over a wide range of cost matrices while the research hypothesis (H_1) claimed that the difference was not zero. This hypothesis was similarly tested for both varying α in the range [0.1, 1.0] with distortion held constant at 0.4 and β constant at 10, and for varying β with distortion held constant at 0.4 and α constant at 0.1. The experimental results are shown in Figures 23 and 24 while the *t*-test results are shown in Tables 40 and 41.

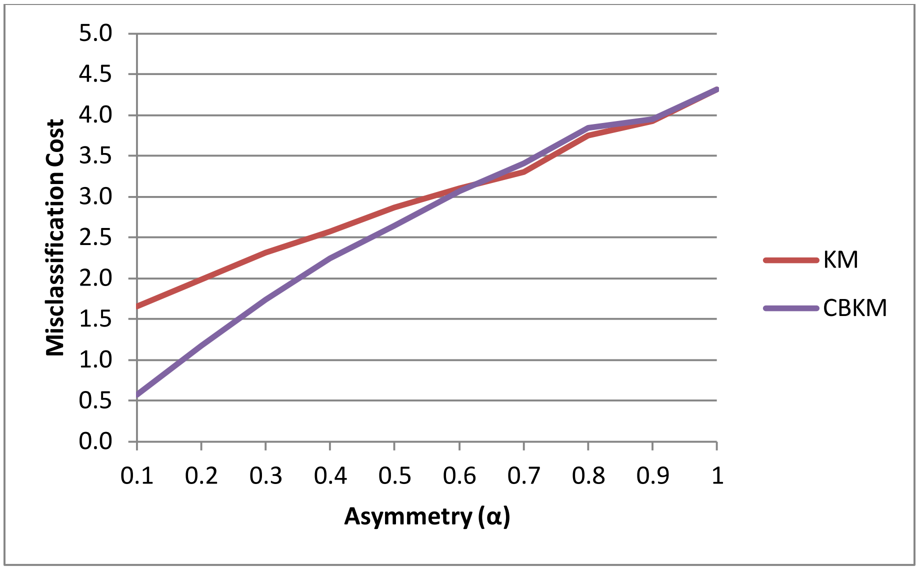


Figure 21: CBKM vs KM — real world data, varying asymmetry

Figure 22: CBKM vs KM — real world data, varying order

Distortion	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4
Asymmetry (α)	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
Order (β)	10	10	10	10	10	10	10	10	10	10
Sample Average ($\bar{X}$)	0.5694	1.1698	1.7348	2.2434	2.6500	3.0732	3.4044	3.8426	3.9560	4.3210
Reference (μ_0)	1.6564	1.9876	2.3148	2.5764	2.8675	3.1038	3.3066	3.7500	3.9267	4.3210
Standard Error ($S_{\bar{X}}$)	0.0358	0.0365	0.0357	0.0345	0.0343	0.0327	0.0313	0.0287	0.0211	0.0000
t-statistic	-30.41	-22.42	-16.26	-9.66	-6.35	-0.94	3.13	3.23	1.39	0.00

Table 38: CBKM vs KM — real world data, varying asymmetry

Distortion	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4
Asymmetry (α)	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
Order (β)	1	2	3	4	5	6	7	8	9	10
Sample Average ($\bar{X}$)	0.0570	0.1133	0.1716	0.2266	0.2866	0.3418	0.3938	0.4538	0.5107	0.5694
Reference (μ_0)	0.1684	0.3360	0.4746	0.6592	0.8417	1.0218	1.1768	1.3367	1.4727	1.6564
Standard Error ($S_{\bar{X}}$)	0.0036	0.0072	0.0105	0.0143	0.0181	0.0218	0.0252	0.0287	0.0320	0.0358
t-statistic	-30.91	-30.92	-28.92	-30.31	-30.72	-31.20	-31.06	-30.77	-30.08	-30.41

Table 39: CBKM vs KM — real world data, varying order

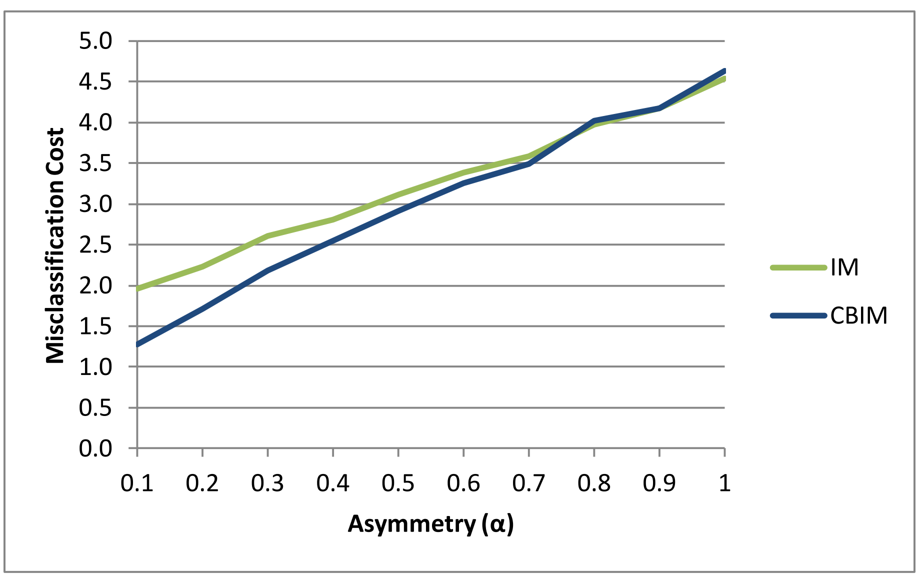


Figure 23: CBIM vs IM — real world data, varying asymmetry

Figure 24: CBIM vs IM — real world data, varying order

Distortion	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4
Asymmetry (α)	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
Order (β)	10	10	10	10	10	10	10	10	10	10
Sample Average ($\bar{X}$)	1.2716	1.7098	2.1863	2.5506	2.9115	3.2504	3.4913	4.0236	4.1764	4.6340
Reference (μ_0)	1.9575	2.2284	2.6105	2.8076	3.1140	3.3808	3.5817	3.9736	4.1750	4.5430
Standard Error ($S_{\bar{X}}$)	0.0295	0.0290	0.0299	0.0294	0.0309	0.0309	0.0322	0.0333	0.0337	0.0354
t-statistic	-23.26	-17.91	-14.17	-8.74	-6.56	-4.22	-2.80	1.50	0.04	2.57

Table 40: CBIM vs IM — real world data, varying asymmetry

Distortion	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4
Asymmetry (α)	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
Order (β)	1	2	3	4	5	6	7	8	9	10
Sample Average ($\bar{X}$)	0.1284	0.2549	0.3664	0.5068	0.6536	0.7671	0.8764	1.0044	1.1430	1.2716
Reference (μ_0)	0.2020	0.3907	0.5659	0.7678	0.9688	1.1946	1.3927	1.5578	1.7482	1.9575
Standard Error ($S_{\bar{X}}$)	0.0030	0.0059	0.0086	0.0116	0.0143	0.0179	0.0211	0.0236	0.0263	0.0295
t-statistic	-24.26	-22.92	-23.13	-22.51	-21.98	-23.94	-24.46	-23.42	-22.97	-23.26

Table 41: CBIM vs IM — real world data, varying order

As with the experimental data, the performance improvement using real world data was more pronounced the farther β deviated from the control value of one. H_0 was rejected for all α values less than 0.8 ($p < 0.05$) and for all values of β.

Hypothesis Seven

Finally, in hypothesis seven, the null hypothesis (H_0) claimed that the difference in performance deterioration between CBKM and CBIM was zero as distortion increased while the research hypothesis claimed that the difference was not zero. Distortion was varied in the range [0, 0.5] while α was held constant at 0.1 and β was held constant at 10. As expected, CBIM's performance deteriorated at a higher rate than CBKM as distortion increased and H_0 was strongly rejected ($p < 0.001$) for all levels of distortion. The results are shown in Figure 25 and the *t*-test results are shown in Table 42.

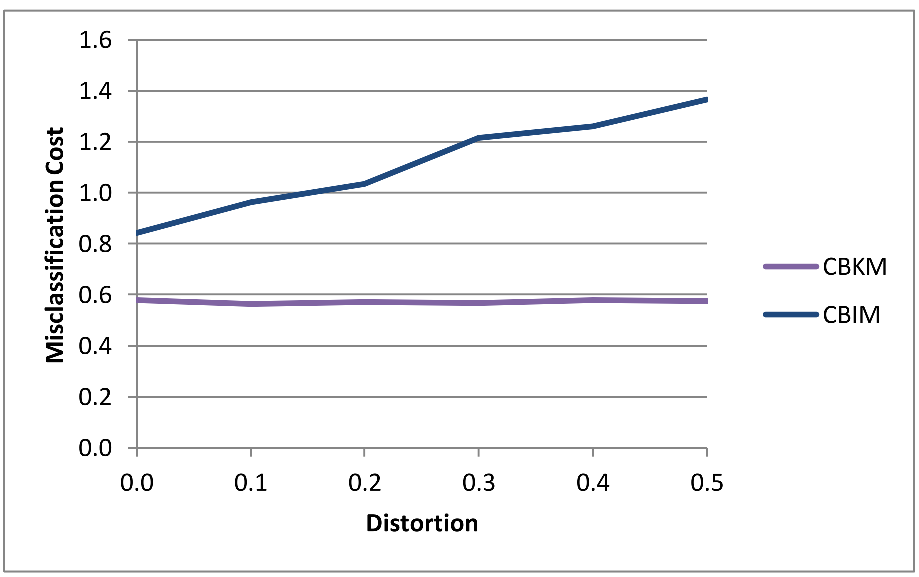


Figure 25: CBKM vs CBIM — real world data, varying distortion

Distortion	0	0.1	0.2	0.3	0.4	0.5
Asymmetry (α)	0.1	0.1	0.1	0.1	0.1	0.1
Order (β)	10	10	10	10	10	10
Sample Average ($\bar{X}$)	0.8416	0.9630	1.0365	1.2158	1.2616	1.3653
Reference (μ_0)	0.2167	0.2755	0.3108	0.3275	0.3313	0.3621
Standard Error ($S_{\bar{X}}$)	0.0227	0.0249	0.0248	0.0287	0.0291	0.0305
t-statistic	-11.62	-16.04	-18.72	-22.55	-23.36	-25.90

Table 42: CBKM vs CBIM — real world data, varying distortion

Findings

General

Three significant data input characteristics influence the new CBKM and CBIM techniques: (a) distortion level, (b) joint input probability distribution, and (c) cost matrix structure. Distortion level and cost matrix structure provide a threshold beyond which CBKM and CBIM either outperform or underperform the extant techniques. In general, and as expected, CBKM is less susceptible to these characteristics than CBIM and outperforms all methodologies over a wide range of distortion, and variations in α and β. Because CBIM does not alter its original decision table, it outperforms IM, but over a different range of distortion, α, and β.

Distortion

With the new methodologies, distortion levels are constant and "unknown." That is, CBKM and CBIM seek lowest cost irrespective of distortion magnitude—the procedure is the same for distortion of 0.1, 0.4, or even 0.0. Likewise, KM and IM seek most probable outcome regardless of inherent system distortion. In contrast, the True Tree assumes perfect distortion knowledge. That is, it assumes distortion of zero and therefore seeks its solution by ignoring input variation. Consequently, in the absence of distortion, the True Tree always produces perfect (lowest cost, correct) results. Because CBKM and CBIM do not make this same zero distortion assumption, their selection of lowest cost solutions are never "perfect." The CBKM result displayed in Figure 7 demonstrates this characteristic. CBKM outperforms KM and the True Tree at all distortion levels, except zero, which is the only point where the True Tree produces perfect results. The same effect is shown for CBIM in Figure 11. The CBKM results differ in Figure 8 because only one input variable's distortion level is varied. Even when that one is at zero, the other inputs have some distortion.

Even with perfect misclassification cost knowledge, CBKM and CBIM will not produce perfect results because joint input probabilities influence their estimation.

Joint Input Probability Distribution

Joint input probability distribution and cost matrix structure combine to influence CBKM and CBIM performance at very low misclassification cost levels. This is because, from Equation 8, CBKM and CBIM seek minimized misclassification costs through a summation of products of cost and probability ($P(j|x)C(i,j)$). Even with perfect misclassification cost knowledge, CBKM and CBIM consider the estimated probability $P(j|x)$ that a class j is the true class of x. Because of this, the methodologies are sensitive to differences in the magnitudes of the costs and probabilities. Given a set of probabilities, lower cost differences between classes have a lower influence on the result than do higher cost differences. Further, the CBKM and KM methodologies create a new, modified decision tree taking costs (CBKM) and probabilities (CBKM/KM) into account, whereas CBIM and IM both use the same, unmodified decision tree. Because of this, CBIM is more sensitive to cost magnitude differences than CBKM, and the performance difference is less between CBIM and IM (Figures 14 and 15), than between CBKM and KM (Figures 12 and 13).

A CBIM example illustrates this phenomenon. Returning to the True Tree example from Figure 3, and the previous example input vector ***Obs*** = (A^O=1, B^O=0, C^O=1, D^O=0, E^O=1), the calculations and results for CBIM and IM are shown in Figure 26. In the figure, the CBIM decision process is highlighted in red boxes and the IM process is highlighted in green boxes for comparison. Estimations rejected by CBIM (higher costs) are shown in blue. At each node, the probabilities used by IM and the additional cost figures used by CBIM are shown next to that node. For both CBIM and IM, at each node, probabilities and costs are calculated for all possible values of the variable at that node, regardless of the node level. That is, for the variable being evaluated, *all* calculated probabilities and costs are summed, even if that node is several levels down and the selected states of previous nodes are known. Those summations are shown in Figure 26 in addition to the misclassification costs for each node's decisions. Following the CBIM decision process, the selections are as shown in Table 43. In this example, the lowest misclassification cost leads to a "1" decision at each node (as shown in red) and a final decision of "Z."

For comparison, following the IM decision process, the selections are as shown in Table 44. The highest probabilities at each node lead to the same "1" decision at each node (as shown in green) and a final decision of "Z."

Table 45 provides a comparison of probability and cost values at each node. As can be seen, the ratio of probabilities (P(1)/P(0)) and costs (C(1)/C(0)) show how

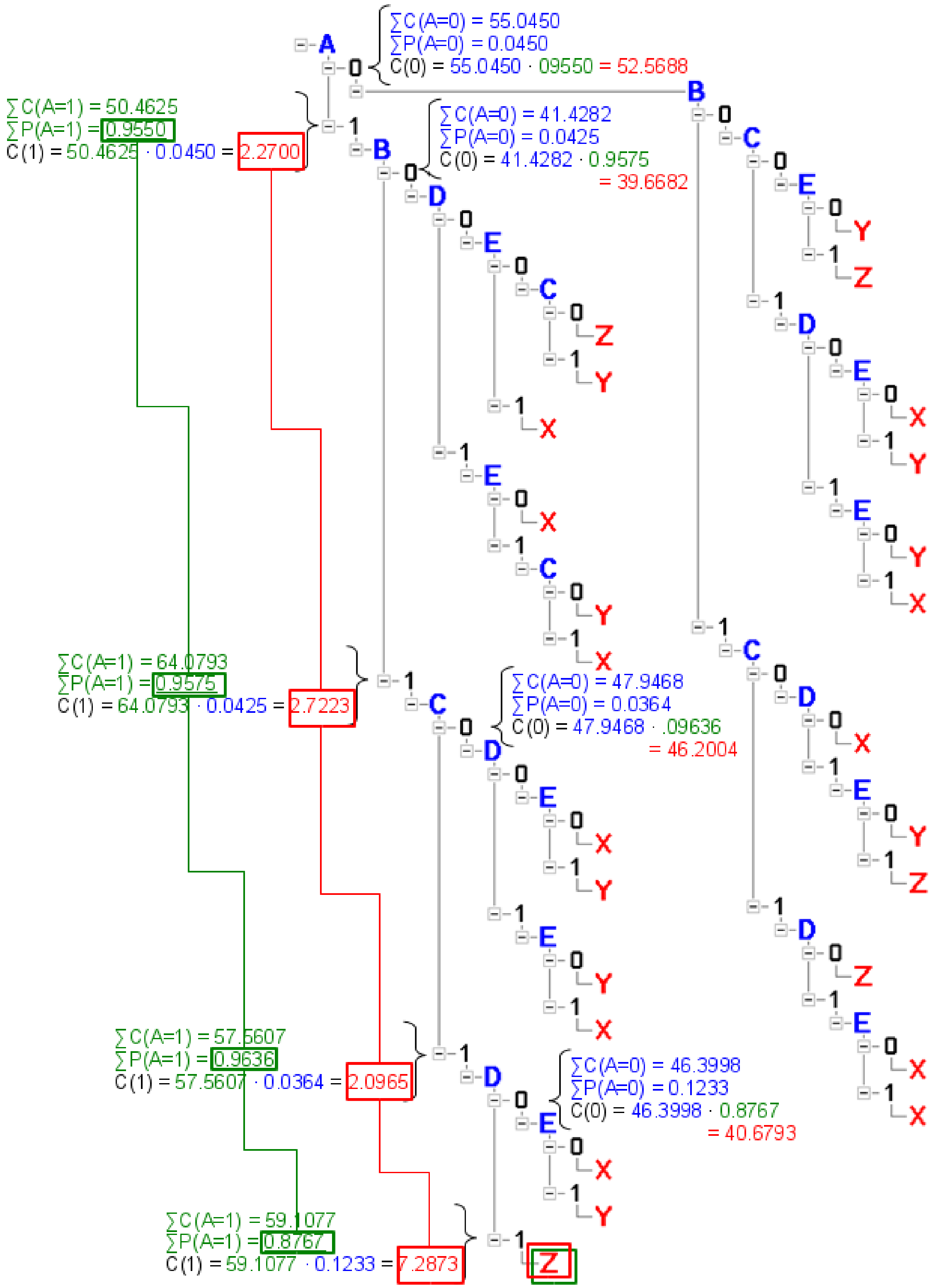


Figure 26: CBIM and IM Example

Node	C(0)	C(1)
A	52.5688	2.2700
B	39.6682	2.7223
C	46.2004	2.0965
D	40.6793	7.2873
Decision		Z

Table 43: CBIM misclassification cost example

Node	C(0)	C(1)
A	0.0450	0.9550
B	0.0425	0.9575
C	0.0364	0.9636
D	0.1233	0.8767
Decision		Z

Table 44: IM probability example

Node	P(0)	P(1)	C(0)	C(1)	P(1)/P(0)	C(1)/C(0)
A	0.0450	0.9550	55.0450	50.4625	21.2222	0.9167
B	0.0425	0.9575	41.4282	64.0793	22.5294	1.5468
C	0.0364	0.9636	47.9468	57.5607	26.4725	1.2005
D	0.1233	0.8767	46.3998	59.1077	7.1103	1.2739

Table 45: Probability and cost ratio comparison

probabilities dominate the decision process over selecting lower cost. This influence of probability over cost persists at all lower misclassification cost magnitudes.

Cost Matrix Structure

CBKM and CBIM performance are influenced by the cost matrix structure shown in Table 13. Remembering that the control cost matrix sets both α and β equal to one, misclassification cost magnitude *differences* increase as α and β deviate from one. Also, because Equation 6 requires that β remain greater than or equal to one, misclassification costs due to increasing β (increasing order) proportionally increase numerically from one (1, 2, 3, and so on). It is also true that deviations from the control value increase in magnitude in the same manner for values between zero and one. Because α is restricted to values greater than zero, its influence on misclassification costs *increases* as it gets numerically *smaller* than one (1.0, 0.9, 0.8, and so on). Because of this, evaluations of α show misclassification costs between the new and extant methods converging to the right of the graphs, while evaluations of β show costs converging toward the left. This is true both for experimental data (for example, Figures 12, 13, 14, and 15) and for real world data (for example, Figures 21, 22, 23 and 24). In all cases, this indicates increasing misclassification costs as α and/or β magnitudes increase from the control.

Additionally, within the specified limits placed on α and β by Equations 6 and 7, α and β were varied in the range $10^{-1} – 10^{0}$ (α) and $10^{0} – 10^{1}$ (β). All graph scales and data generated for the *t*-tests were generated on a linear scale with 10 increments in the range [0.1, 1.0] (α) and [1, 10] (β). While a quick look at the graphs might indicate CBIM is more sensitive to variations in α than in β, this is not the case, but the result of the selected scales for each variable.

CBKM and CBIM vs Extant Methods

Based on the experiments, one can conclude that, not only does CBKM outperform CBIM, but the degree to which CBKM outperforms KM exceeds the degree to which CBIM outperforms IM. That is, CBKM provides a greater improvement over its extant model than does CBIM. The difference is most pronounced using the experimental, as opposed to real world, data. A summary of the experimental data for CBKM from Figure 5 and for CBIM from Figure 11 illustrate the point and are shown in Table 46.

	Average Cost		Cost Ratio	Average Cost		Cost Ratio
Distortion	KM	CBKM	CBKM/KM	IM	CBIM	CBIM/IM
0	0.8902	0.2230	0.2506	0.7331	0.7242	0.9878
0.1	1.0524	0.2656	0.2524	1.0007	0.9311	0.9304
0.2	1.1962	0.3023	0.2527	1.2045	1.1033	0.9160
0.3	1.2769	0.3238	0.2536	1.3173	1.1975	0.9090
0.4	1.3456	0.3445	0.2560	1.3806	1.2609	0.9133
0.5	1.4030	0.3581	0.2552	1.4350	1.3054	0.9097

Table 46: CBKM/CBIM improvement over extant methods

As can be seen for these parameters ($\alpha = 0.1$, $\beta = 10$), for various distortion levels, CBKM consistently reduces misclassification costs by about 75% over KM while CBIM reduces costs by about 10% when compared with IM. This difference between CBKM and its reference methodology (KM) and CBIM and its reference methodology (IM) is similar to the differences between KM and IM compared with their reference methodology (a True Tree). With respect to classification accuracy, Jiang et al. (2005) showed KM consistently outperformed IM and a True Tree while IM was no better than its reference True Tree for distortion values less than 0.5. Figure 27 (reproduced from

Jiang et al.) shows the reported results for KM and IM as distortion increased using binary input experimental data, and Figure 28 shows their result using the same real world data used in this research.

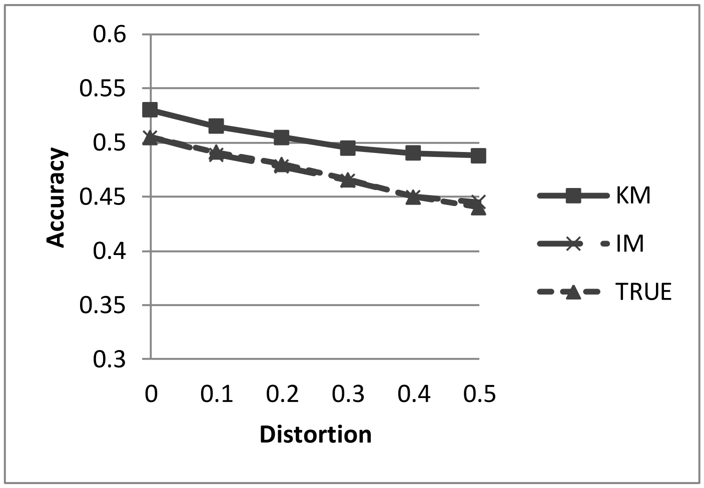


Figure 27: Extant method experimental results — performance vs reference True Tree

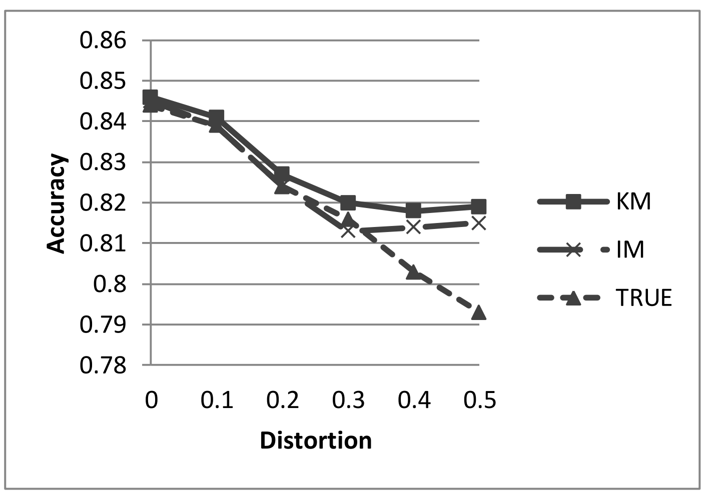


Figure 28: Extant method real world results — performance vs reference True Tree

The reason CBIM more closely parallels IM than CBKM parallels KM derives from the way CBKM modifies its knowledge base while CBIM does not. In step 2 of the CBKM process, given an **Observed** vector, misclassification costs for *every* possible **True** vector are estimated. The CBKM recommendation comes from the lowest predicted cost **True** vector. In contrast, CBIM traverses a True Tree developed without regard to misclassification costs. In step 1, CBIM estimates a lowest cost **True** vector from a *single* variable's **Observed** vector represented by the True Tree root node. In step 2, the **True** vector is estimate again based on a different single variable, without regard for the information available from step 1. A CBIM recommendation is therefore limited by its individual estimation of *single* variables in sequence. This research confirmed the expected result that CBIM's results would more closely parallel IM's results than CBKM's results compared with KM, just as KM provided a greater improvement over the True Tree than IM.

Further, within some parameters, CBIM performance appears similar to, though statistically different from, IM performance with respect to misclassification error costs. Figure 14, for example, seems to indicate that CBIM performance closely parallels IM performance for varying levels of α. This is a result of CBIM's lower improvement over IM (as compared with CBKM's improvement over KM), and the selection of α in the range $0 < \alpha \leq 1$. As discussed, in general, α is restricted to $\alpha > 0$ and increases in magnitude as it deviates from the control value of one. For the experiments, α values were selected in the available range less than one. Selection of the other available range for α, however, is useful in a further investigation of the performance difference between CBIM and IM because is allows for a wider range of α values. For example, instead of restricting α to the range 0 to 1, assigning values in the range 1 to 10 amplifies the significance of the difference in CBIM and IM performance in the results. Figure 29 shows the misclassification cost difference between CBIM and IM with α varied in the wider range.

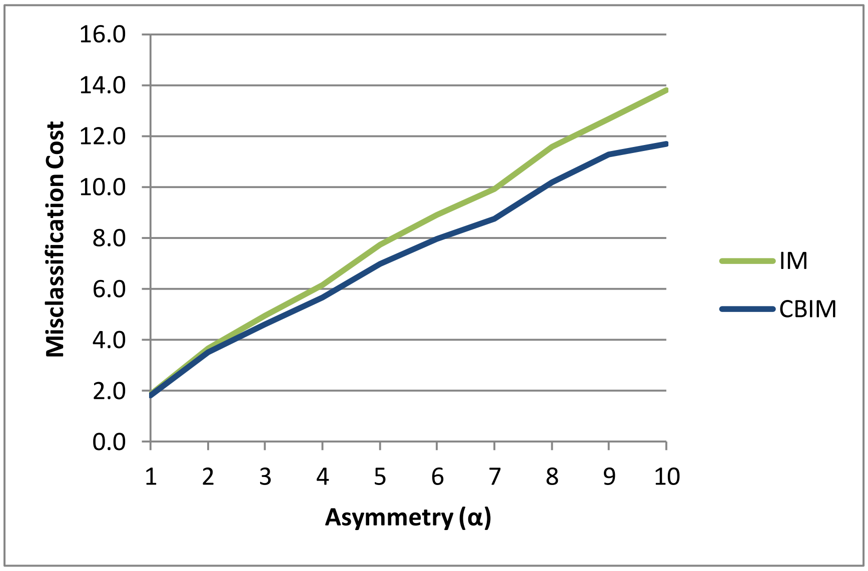


Figure 29: CBIM vs IM — increased asymmetry range

Using the increased range for α in this case highlights the performance difference between CBIM and IM as α increases.

Summary of Results

Experimental results using both computer generated and real world data indicate the new CBKM and CBIM methodologies outperform extant methods over a wide range of input distortion and cost matrices, with some restrictions. As expected, the extant True Tree always outperforms in the total absence of distortion. Likewise, in the absence of misclassification costs, or where all such costs are equal, there is no measurable difference between the methodologies. Because CBIM is restricted to using an unmodified, original True Tree, it underperforms CBKM, which does not have this restriction. CBKM and CBIM performance increase as distortion, or the magnitude of α or β increase with CBKM recording the greatest difference.

In conclusion, the new methodologies are most useful in the presence of measurable input distortion *and* misclassification costs.

Chapter 5

Conclusions, Implications, Recommendations, and Summary

Conclusions

New methodologies can be used to improve performance of rule-based expert systems operating with noisy inputs. The proposed CBKM and CBIM techniques improve performance, but only in the presence of distortion and misclassification costs. In the absence of distortion, a True Tree will produce perfect results and in the absence of misclassification costs, the new methods perform identically to the extant KM and IM methods. The assumption, therefore, is that the new methods would be applied to systems with both distortion and misclassification costs. Experiments conducted on real world data from the credit card application domain indicate there are real world systems that could benefit from the new techniques.

The new techniques differ from previous methods that sought most probable outcomes in that misclassification costs are not calculated as strict cost values. Under the extant KM and IM methods, probabilities are calculated from marginal distributions and input distortion matrices. Even if estimated, one probability value can be numerically compared with another in the process of selecting most probable outcomes. In contrast, lowest misclassification costs are calculated from a cost matrix *and* the misclassification probabilities according to Equation 8. Consequently, while extant methods strictly evaluate probabilities, CBKM and CBIM decisions can be influenced by misclassification probabilities, particularly with small misclassification cost differences between classes.

One shortcoming of the available real world data is the relatively small number of instances, even though they vary greatly in terms including missing, duplicate, and conflicting data. As structured, the fifteen inputs could provide 10,450,944 different input combinations. Because See5 only allowed for 400 input instances, the generated True Tree and decision table were not perfect. As many as 0.07% (7,776) possible inputs were unsolvable using the See5 output. In contrast, the computer generated data for the experiments were produced from five binary variables totaling 2^5 possible inputs. This smaller number of inputs meant that every possible input combination could be tested using a disparate variety of marginal distributions, distortion matrices, and cost matrices. For this reason, the experimental data provided more conclusive evidence of the veracity of the proposed methods.

Given the experimental and real world data results, the new methods work best with a combination of high distortion, high asymmetry, and high order.

Implications

This research contributed two new methodologies for rule-based expert systems operating with noisy input data. To be useful, the techniques require domain knowledge sufficient for; (a) creating a decision table that produces correct results in the absence of input noise, and (b) an accuracy estimate for each possible input data element.

The experiments indicate techniques that use an unmodified True Tree combined with a cost matrix underperform techniques that do not have the same restriction. The new techniques also underperform in the absence of either input distortion or misclassification costs. This observation is not surprising given that the new techniques are designed for distortion-ridden systems. At the same time, the research did not attempt to quantify the distortion or misclassification threshold costs required for any specific domain. An interesting question for future research would be to devise techniques for quantifying breakeven points in various domains.

Recommendations

In the research, the new CBKM technique modified an original decision table while the CBIM technique did not. In the experiments, the CBKM approach outperformed CBIM with respect to misclassification costs over a wide range of input distortion. However, in addition to not modifying an original decision tree, the CBIM technique limited its inputs at each node to misclassification costs associated with that node's decision variable only. This means that, for each node below the root node, CBIM ignores the decision knowledge for all higher nodes. For example, referencing Figure 30, at the decision node "C," CBIM will evaluate all misclassification costs for "C." Given that the "A" node has already been evaluated, there may be an advantage to evaluating only those costs for "C" where A=1. This process is the same as for the most probable decision seeking IM technique. The evaluation of this modification to both CBIM and IM, as well as its comparison with CBKM and KM performance, is left for future research.

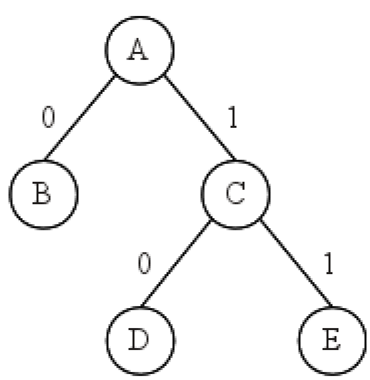


Figure 30: CBIM decision process example

Summary

This dissertation developed methods to minimize recommendation error costs when inputs to rule-based expert systems are prone to errors. Prior studies proposed methods that attempted to minimize the total probability of recommendation error, but did not take relative misclassification costs into account. Given that many real world systems, including web-based applications, are prone to input errors, many organizations are incentivized to seek minimum overall decision cost instead of overall accuracy.

Two techniques proposed by Jiang et al. (2005) are intended to improve decision making accuracy in the presence of deliberately false input data. The first technique, termed Knowledge Based Modification (KM) improves accuracy by modifying a knowledge base in the form of a decision tree. Its inputs are an original knowledge base, the joint input probability distributions, and the distortion matrices for all variables. The second, termed Input Modification (IM), modifies observed inputs to their most likely true value and feeds those inputs into an unmodified knowledge base. Its inputs are an unmodified knowledge base, and the marginal distributions, and distortion matrices for all variables. Both techniques seek to minimize the probability of errors in the output.

This research expanded the KM and IM techniques by adding a cost matrix, allowing new methods to seek minimized error cost. The goal was to develop and empirically test these new methods to determine if minimized misclassification error costs are achievable, and under what input distortion and cost matrix parameters.

The first new method, called Cost-Based Knowledge Base Modification (CBKM), modified an original knowledge base by taking into account the misclassification cost of different types of errors. The second, called Cost-Based Input Modification (CBIM), modified individual inputs to their lowest cost values and fed those into the original decision tree.

The structure of the cost matrix allowed for comparison with the extant methods by providing inputs either with, or without, disparate misclassification costs. By setting all correct decision (no error) misclassification costs equal to zero and all incorrect decision costs equal to one, the cost matrix allow for testing the case where minimized misclassification costs are equal, regardless of the decision. In that special case, as expected, the new CBKM and CBIM techniques performed no better, and no worse, than KM and IM.

As the disparity between different error types increased, however, CBKM outperformed KM with respect to misclassification error costs, as did CBIM when compared with IM. The cost matrix structure provided for testing two different types of disparities. First, by using an asymmetrical cost matrix wherein the misclassification costs for errors with one variable were higher (or lower) than similar errors with another variable, the experiments tested systems with uneven costs between different decision outcomes. Second, by orderly increasing, or decreasing, misclassification costs diagonally across a cost matrix, the experiments tested the case where making “small” errors is always less than or equal to making “large” errors, given a particular output. In both cases, the performance of CBKM and CBIM increased with increased asymmetry and order.

This research also looked at CBKM and CBIM performance with varying levels of input distortion. In general, where no input distortion exists, there is no need for a system seeking minimized misclassification costs as a True Tree that assumes correct output, given any input, would produce correct (lowest cost) results every time. The difficulty arises where real world systems have some level of unknown input distortion whether it be intentional, as in the case of falsified credit card applications, or accidental, as in the case of poor typing skills on online forms. As expected, both CBKM and CBIM underperform at very low (near zero) distortion levels, but outperformed KM and IM as well as a True Tree, as distortion levels increased. This was verified with distortion applied evenly and also unevenly across multiple inputs.

Additionally, CBKM and CBIM performance were compared against each other with varying distortion levels. As expected, CBKM outperformed CBIM over a wide range of input distortion. Because the CBKM process modifies an existing knowledge base, taking misclassification costs into account, its process produces a cost-sensitive knowledge base. In contrast, CBIM selects the lowest cost solution for each input variable, then applies those inputs to an unmodified knowledge base. The CBIM process, therefore, is more likely to get “stuck” with a poor decision early in the decision process, unlike CBKM.

The experiments indicate both new methodologies are susceptible to variations in input distortion, joint input probability distribution, and cost matrix structure. As mentioned, at very low input distortion levels, the systems designed to account for such distortion do no better than a True Tree and can underperform where distortion is zero.

Joint input probability distributions and cost matrix structure combine to influence outcomes where either estimated probabilities are high relative to estimated misclassification costs, or where the cost matrix structure provides little differentiation between the misclassification costs of different decisions. In the first case, a problem arises because cost estimates are calculated based on the misclassification costs indicated by the cost matrix, and the estimated probability that a selected class is a true class. Where the probability magnitudes dominate misclassification costs, those costs can be underrepresented in the ensuing estimation, resulting in class selection based more on probability than misclassification cost. In the second case, where asymmetry and order are small relative to their control value of one, both CBKM and CBIM can underperform a True Tree that assumes zero input distortion in all cases.

In addition, the improvement of CBKM over its reference methodology KM was greater than the improvement of CBIM over IM. This resulted from CBKM's ability to use estimated misclassification costs for all potential true input vectors while CBIM was limited to its original, unaltered decision tree. These results were similar to those reported in the literature for the KM and IM methodologies when compared with their True Tree reference methodology.

In conclusion, the new CBKM and CBIM methodologies outperform extant methods that do not take misclassification costs into account, where performance is measured as the ability to minimize misclassification costs, for most input levels of distortion, and cost matrix asymmetry and order. At very low input distortion levels, a True Tree can outperform all other methods, which would work only if one could ensure zero input distortion. Performance is also impacted by a cost matrix that has very little distinction in misclassification costs between decisions, whether that be caused by a low value for asymmetry or low value for order. In both cases, low values are those that approach the control value of one.

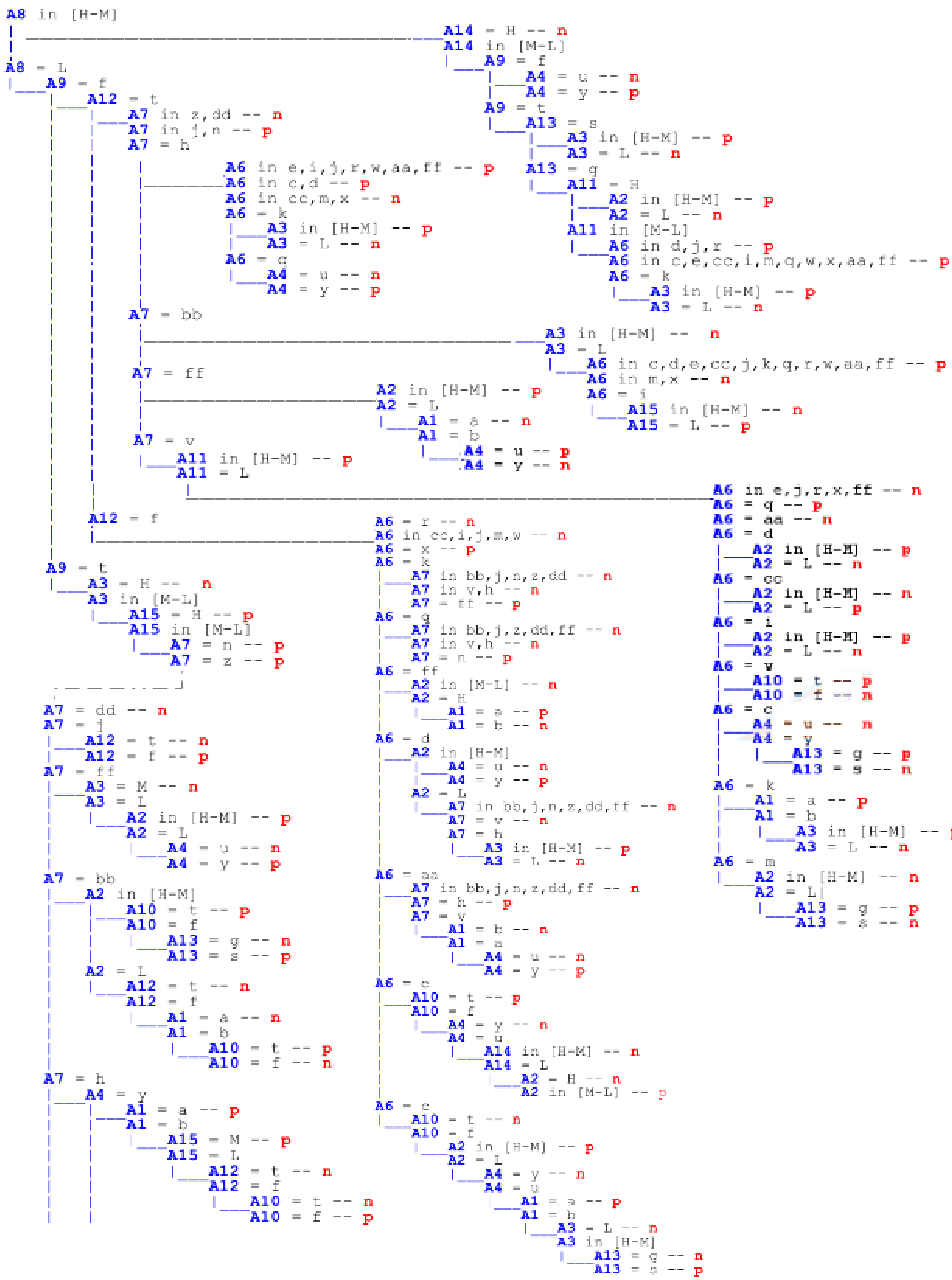
Appendix A

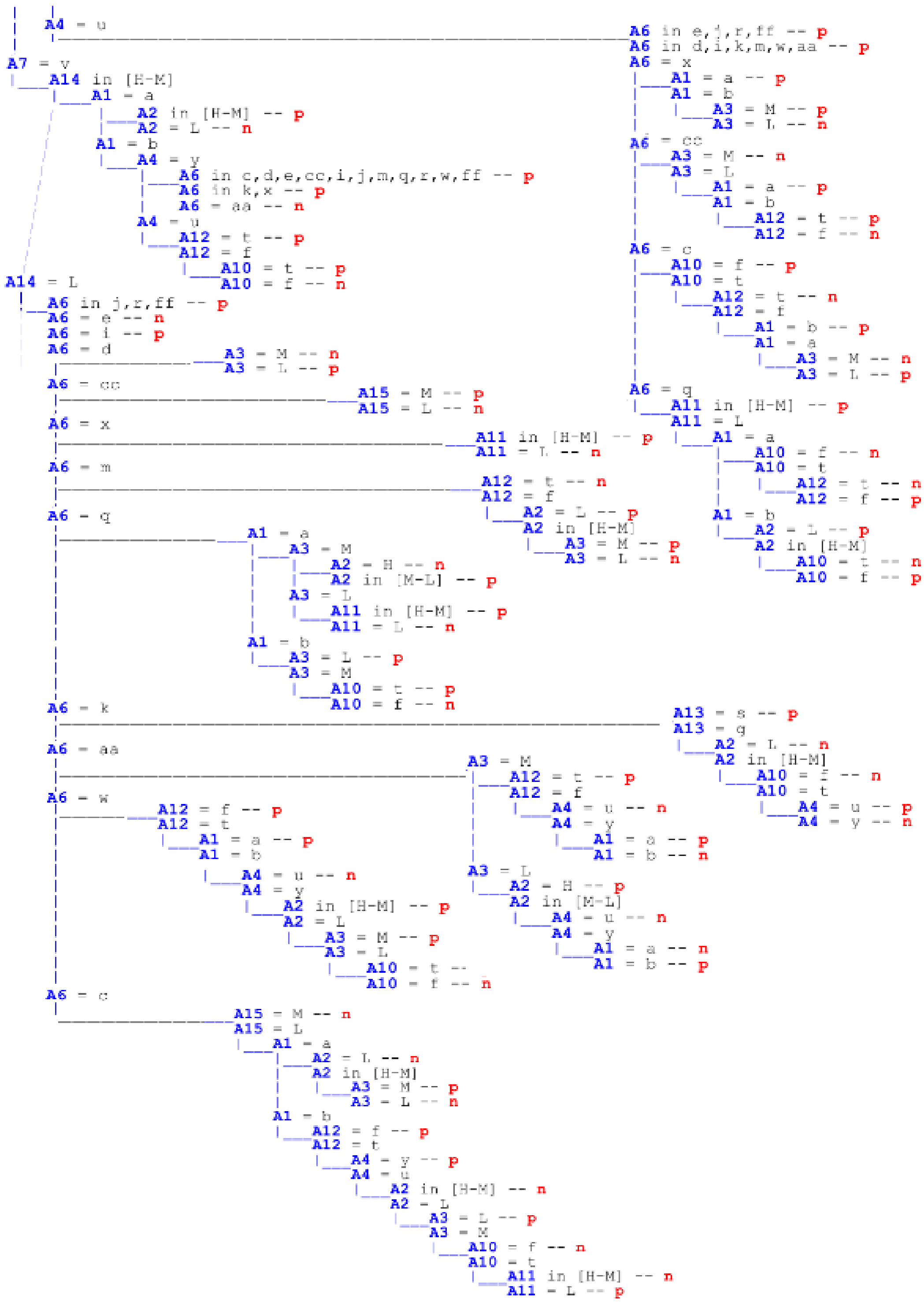
Fully Enumerated True Table Example

Rule	Inputs					Output	Distribution
	A	B	C	D	E		
1	0	0	0	0	0	Y	0.006486
2	0	0	0	0	1	Z	0.008958
3	0	0	0	1	0	Y	0.007928
4	0	0	0	1	1	Z	0.010948
5	0	0	1	0	0	X	0.008256
6	0	0	1	0	1	Y	0.011400
7	0	0	1	1	0	Y	0.010090
8	0	0	1	1	1	X	0.013934
9	0	1	0	0	0	X	0.003493
10	0	1	0	0	1	X	0.004823
11	0	1	0	1	0	Y	0.004269
12	0	1	0	1	1	Z	0.005895
13	0	1	1	0	0	Z	0.004445
14	0	1	1	0	1	Z	0.006139
15	0	1	1	1	0	X	0.005433
16	0	1	1	1	1	X	0.007503
17	1	0	0	0	0	Z	0.047568
18	1	0	0	0	1	X	0.065688
19	1	0	0	1	0	X	0.058138
20	1	0	0	1	1	Y	0.080286
21	1	0	1	0	0	Y	0.060540
22	1	0	1	0	1	X	0.083604
23	1	0	1	1	0	X	0.073994
24	1	0	1	1	1	X	0.102182
25	1	1	0	0	0	X	0.025613
26	1	1	0	0	1	Y	0.035371
27	1	1	0	1	0	Y	0.031305
28	1	1	0	1	1	X	0.043231
29	1	1	1	0	0	X	0.032599
30	1	1	1	0	1	Y	0.045017
31	1	1	1	1	0	Z	0.039843
32	1	1	1	1	1	Z	0.055021

Appendix B

See5 Generated True Tree – Real World Data




A4 = u
A6 in e,j,r,ff -- p
A6 in d,i,k,m,w,aa -- p
A6 = x
A1 = a -- p
A1 = b
A3 = M -- p
A3 = L -- n
A6 = cc
A3 = M -- n
A3 = L
A1 = a -- p
A1 = b
A12 = t -- p
A12 = f -- n
A6 = c
A10 = f -- p
A10 = t
A12 = t -- n
A12 = f
A1 = b -- p
A1 = a
A3 = M -- n
A3 = L -- p
A6 = q
A11 in [H-M] -- p
A11 = L
A1 = a
A10 = f -- n
A10 = t
A12 = t -- n
A12 = f -- p
A1 = b
A2 = L -- p
A2 in [H-M]
A10 = t -- n
A10 = f -- p
A7 = v
A14 in [H-M]
A1 = a
A2 in [H-M] -- p
A2 = L -- n
A1 = b
A4 = y
A6 in c,d,e,cc,i,j,m,q,r,w,ff -- p
A6 in k,x -- p
A6 = aa -- n
A4 = u
A12 = t -- p
A12 = f
A10 = t -- p
A10 = f -- n
A14 = L
A6 in j,r,ff -- p
A6 = e -- n
A6 = i -- p
A6 = d
A3 = M -- n
A3 = L -- p
A6 = cc
A15 = M -- p
A15 = L -- n
A6 = x
A11 in [H-M] -- p
A11 = L -- n
A6 = m
A12 = t -- n
A12 = f
A2 = L -- p
A2 in [H-M]
A3 = M -- p
A3 = L -- n
A6 = q
A1 = a
A3 = M
A2 = H -- n
A2 in [M-L] -- p
A3 = L
A11 in [H-M] -- p
A11 = L -- n
A1 = b
A3 = L -- p
A3 = M
A10 = t -- p
A10 = f -- n
A6 = k
A13 = s -- p
A13 = g
A2 = L -- n
A2 in [H-M]
A10 = f -- n
A10 = t
A4 = u -- p
A4 = y -- n
A6 = aa
A3 = M
A12 = t -- p
A12 = f
A4 = u -- n
A4 = y
A1 = a -- p
A1 = b -- n
A3 = L
A2 = H -- p
A2 in [M-L]
A4 = u -- n
A4 = y
A1 = a -- n
A1 = b -- p
A6 = w
A12 = f -- p
A12 = t
A1 = a -- p
A1 = b
A4 = u -- n
A4 = y
A2 in [H-M] -- p
A2 = L
A3 = M -- p
A3 = L
A10 = t --
A10 = f -- n
A6 = c
A15 = M -- n
A15 = L
A1 = a
A2 = L -- n
A2 in [H-M]
A3 = M -- p
A3 = L -- n
A1 = b
A12 = f -- p
A12 = t
A4 = y -- p
A4 = u
A2 in [H-M] -- n
A2 = L
A3 = L -- p
A3 = M
A10 = f -- n
A10 = t
A11 in [H-M] -- n
A11 = L -- p

Appendix C

See5 Generated Decision Table – Real World Data

Rule	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	Out
1	-	-	-	-	-	-	-	H,M	-	-	-	-	-	H	-	n
2	-	-	-	u	-	-	-	H,M	f	-	-	-	-	M,L	-	n
3	-	-	-	y	-	-	-	H,M	f	-	-	-	-	M,L	-	p
4	-	-	H,M	-	-	-	-	H,M	t	-	-	-	s	M,L	-	p
5	-	-	L	-	-	-	-	H,M	t	-	-	-	s	M,L	-	n
6	-	H,M	-	-	-	-	-	H,M	t	-	H	-	g	M,L	-	p
7	-	L	-	-	-	-	-	H,M	t	-	H	-	g	M,L	-	n
8	-	-	-	-	-	c,d,e,j,cc,i,r,m,q,w,x,aa,ff	-	H,M	t	-	M,L	-	g	M,L	-	p
9	-	-	H,M	-	-	k	-	H,M	t	-	M,L	-	g	M,L	-	p
10	-	-	L	-	-	k	-	H,M	t	-	M,L	-	g	M,L	-	n
11	-	-	-	-	-	-	z,dd	L	f	-	-	t	-	-	-	n
12	-	-	-	-	-	-	j,n	L	f	-	-	t	-	-	-	p
13	-	-	-	-	-	c,d,e,i,j,r,w,aa,ff	h	L	f	-	-	t	-	-	-	p
14	-	-	-	-	-	cc,m,x	h	L	f	-	-	t	-	-	-	n
15	-	-	H,M	-	-	k	h	L	f	-	-	t	-	-	-	p
16	-	-	L	-	-	k	h	L	f	-	-	t	-	-	-	n
17	-	-	-	u	-	k	h	L	f	-	-	t	-	-	-	n
18	-	-	-	y	-	k	h	L	f	-	-	t	-	-	-	p
19	-	-	H,M	-	-	-	bb	L	f	-	-	t	-	-	-	n
20	-	-	L	-	-	c,d,e,cc,j,k,q,r,w,aa,ff	bb	L	f	-	-	t	-	-	-	p
21	-	-	L	-	-	m,x	bb	L	f	-	-	t	-	-	-	n
22	-	-	L	-	-	l	bb	L	f	-	-	t	-	-	H,M	n
23	-	-	L	-	-	l	bb	L	f	-	-	t	-	-	L	p
24	-	H,M	-	-	-	-	ff	L	f	-	-	t	-	-	-	p
25	a	L	-	-	-	-	ff	L	f	-	-	t	-	-	-	n
26	b	L	-	u	-	-	ff	L	f	-	-	t	-	-	-	p
27	b	L	-	y	-	-	ff	L	f	-	-	t	-	-	-	n
28	-	-	-	-	-	-	v	L	f	-	H,M	t	-	-	-	p
29	-	-	-	-	-	aa,e,j,r,x,ff	v	L	f	-	L	t	-	-	-	n
30	-	-	-	-	-	q	v	L	f	-	L	t	-	-	-	p
31	-	H,M	-	-	-	d	v	L	f	-	L	t	-	-	-	p
32	-	L	-	-	-	d	v	L	f	-	L	t	-	-	-	n
33	-	H,M	-	-	-	cc	v	L	f	-	L	t	-	-	-	n
34	-	L	-	-	-	cc	v	L	f	-	L	t	-	-	-	p
35	-	H,M	-	-	-	i	v	L	f	-	L	t	-	-	-	p
36	-	L	-	-	-	i	v	L	f	-	L	t	-	-	-	n
37	-	-	-	-	-	w	v	L	f	t	L	t	-	-	-	p
38	-	-	-	-	-	w	v	L	f	f	L	t	-	-	-	n
39	-	-	-	u	-	c	v	L	f	-	L	t	-	-	-	n
40	-	-	-	y	-	c	v	L	f	-	L	t	g	-	-	p
41	-	-	-	y	-	c	v	L	f	-	L	t	s	-	-	n
42	a	-	-		-	k	v	L	f	-	L	t	-	-	-	p
43	b	-	H,M	-	-	k	v	L	f	-	L	t	-	-	-	p
44	b	-	L	-	-	k	v	L	f	-	L	t	-	-	-	n
45	-	H,M	-	-	-	m	v	L	f	-	L	t	-	-	-	n
46	-	L	-	-	-	m	v	L	f	-	L	t	g	-	-	p
47	-	L	-	-	-	m	v	L	f	-	L	t	s	-	-	n
48	-	-	-	-	-	r	-	L	f	-	-	f	-	-	-	n
49	-	-	-	-	-	cc,i,j,m,w	-	L	f	-	-	f	-	-	-	n
50	-	-	-	-	-	x	-	L	f	-	-	f	-	-	-	p
51	-	-	-	-	-	k	bb,j,n,v,h,z,dd	L	f	-	-	f	-	-	-	n
52	-	-	-	-	-	k	ff	L	f	-	-	f	-	-	-	p
53	-	-	-	-	-	q	bb,j,v,h,z,dd,ff	L	f	-	-	f	-	-	-	n
54	-	-	-	-	-	q	n	L	f	-	-	f	-	-	-	p
55	-	M,L	-	-	-	ff	-	L	f	-	-	f	-	-	-	n
56	a	H	-	-	-	ff	-	L	f	-	-	f	-	-	-	p
57	b	H	-	-	-	ff	-	L	f	-	-	f	-	-	-	n
58	-	H,M	-	u	-	d	-	L	f	-	-	f	-	-	-	n
59	-	H,M	-	y	-	d	-	L	f	-	-	f	-	-	-	p
60	-	L	-	-	-	d	bb,j,n,v,z,dd,ff	L	f	-	-	f	-	-	-	n
61	-	L	H,M	-	-	d	h	L	f	-	-	f	-	-	-	p
62	-	L	L	-	-	d	h	L	f	-	-	f	-	-	-	n
63	-	-	-	-	-	aa	bb,j,n,z,dd,ff	L	f	-	-	f	-	-	-	n

Rule	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	Out
64	-	-	-	-	-	aa	h	L	f	-	-	f	-	-	-	p
65	b	-	-	-	-	aa	v	L	f	-	-	f	-	-	-	n
66	a	-	-	u	-	aa	v	L	f	-	-	f	-	-	-	n
67	a	-	-	y	-	aa	v	L	f	-	-	f	-	-	-	p
68	-	-	-		-	e	-	L	f	t	-	f	-	-	-	p
69	-	-	-	y	-	e	-	L	f	f	-	f	-	-	-	n
70	-	-	-	u	-	e	-	L	f	f	-	f	-	H,M	-	n
71	-	H	-	u	-	e	-	L	f	f	-	f	-	L	-	n
72	-	M,L	-	u	-	e	-	L	f	f	-	f	-	L	-	p
73	-		-	-	-	c	-	L	f	t	-	f	-	-	-	n
74	-	H,M	-	-	-	c	-	L	f	f	-	f	-	-	-	p
75	-	L	-	y	-	c	-	L	f	f	-	f	-	-	-	n
76	a	L	-	u	-	c	-	L	f	f	-	f	-	-	-	p
77	b	L	L	u	-	c	-	L	f	f	-	f	-	-	-	n
78	b	L	H,M	u	-	c	-	L	f	f	-	f	g	-	-	n
79	b	L	H,M	u	-	c	-	L	f	f	-	f	s	-	-	p
80	-	-	H	-	-	-	-	L	t	-	-	-	-	-	-	n
81	-	-	M,L	-	-	-	-	L	t	-	-	-	-	-	H	p
82	-	-	M,L	-	-	-	n,z	L	t	-	-	-	-	-	M,L	p
83	-	-	M,L	-	-	-	dd	L	t	-	-	-	-	-	M,L	n
84	-	-	M,L	-	-	-	j	L	t	-	-	t	-	-	M,L	n
85	-	-	M,L	-	-	-	j	L	t	-	-	f	-	-	M,L	p
86	-	-	M	-	-	-	ff	L	t	-	-	-	-	-	M,L	n
87	-	H,M	L	-	-	-	ff	L	t	-	-	-	-	-	M,L	p
88	-	L	L	u	-	-	ff	L	t	-	-	-	-	-	M,L	n
89	-	L	L	y	-	-	ff	L	t	-	-	-	-	-	M,L	p
90	-	H,M	M,L	-	-	-	bb	L	t	t	-	-	-	-	M,L	p
91	-	H,M	M,L	-	-	-	bb	L	t	f	-	-	g	-	M,L	n
92	-	H,M	M,L	-	-	-	bb	L	t	f	-	-	s	-	M,L	p
93	-	L	M,L	-	-	-	bb	L	t		-	t	-	-	M,L	n
94	a	L	M,L	-	-	-	bb	L	t		-	f	-	-	M,L	n
95	b	L	M,L	-	-	-	bb	L	t	t	-	f	-	-	M,L	p
96	b	L	M,L	-	-	-	bb	L	t	f	-	f	-	-	M,L	n
97	a	-	M,L	y	-	-	h	L	t	-	-	-	-	-	M,L	p
98	b	-	M,L	y	-	-	h	L	t	-	-	-	-	-	M	p
99	b	-	M,L	y	-	-	h	L	t		-	t	-	-	L	n
100	b	-	M,L	y	-	-	h	L	t	t	-	f	-	-	L	n
101	b	-	M,L	y	-	-	h	L	t	f	-	f	-	-	L	p
102	-	-	M,L	u	-	d,e,i,j,k,r,m,w,aa,ff	h	L	t	-	-	-	-	-	M,L	p
103	a	-	M,L	u	-	x	h	L	t	-	-	-	-	-	M,L	p
104	b	-	M	u	-	x	h	L	t	-	-	-	-	-	M,L	p
105	b	-	L	u	-	x	h	L	t	-	-	-	-	-	M,L	n
106	-	-	M	u	-	cc	h	L	t	-	-	-	-	-	M,L	n
107	a	-	L	u	-	cc	h	L	t	-	-	-	-	-	M,L	p
108	b	-	L	u	-	cc	h	L	t	-	-	t	-	-	M,L	p
109	b	-	L	u	-	cc	h	L	t	-	-	f	-	-	M,L	n
110	-	-	M,L	u	-	c	h	L	t	f	-	-	-	-	M,L	p
111	-	-	M,L	u	-	c	h	L	t	t	-	t	-	-	M,L	n
112	b	-	M,L	u	-	c	h	L	t	t	-	f	-	-	M,L	p
113	a	-	M	u	-	c	h	L	t	t	-	f	-	-	M,L	n
114	a	-	L	u	-	c	h	L	t	t	-	f	-	-	M,L	p
115	-	-	M,L	u	-	q	h	L	t	-	H,M	-	-	-	M,L	p
116	a	-	M,L	u	-	q	h	L	t	f	L	-	-	-	M,L	n
117	a	-	M,L	u	-	q	h	L	t	t	L	t	-	-	M,L	n
118	a	-	M,L	u	-	q	h	L	t	t	L	f	-	-	M,L	p
119	b	L	M,L	u	-	q	h	L	t	-	L	-	-	-	M,L	p
120	b	H,M	M,L	u	-	q	h	L	t	t	L	-	-	-	M,L	n
121	b	H,M	M,L	u	-	q	h	L	t	f	L	-	-	-	M,L	p
122	a	H,M	M,L	-	-	-	v	L	t	-	-	-	-	H,M	M,L	p
123	a	L	M,L	-	-	-	v	L	t	-	-	-	-	H,M	M,L	n
124	b	-	M,L	y	-	c,d,e,k,x,cc,i,j,m,q,r,w,ff	v	L	t	-	-	-	-	H,M	M,L	p
125	b	-	M,L	y	-	aa	v	L	t	-	-	-	-	H,M	M,L	n
126	b	-	M,L	u	-	-	v	L	t	-	-	t	-	H,M	M,L	p

Rule	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	Out
127	b	-	M,L	u	-	-	v	L	t	t	-	f	-	H,M	M,L	p
128	b	-	M,L	u	-	-	v	L	t	f	-	f	-	H,M	M,L	n
129	-	-	M,L	-	-	i,j,r,ff	v	L	t	-	-	-	-	L	M,L	p
130	-	-	M,L	-	-	e	v	L	t	-	-	-	-	L	M,L	n
131	-	-	M	-	-	d	v	L	t	-	-	-	-	L	M,L	n
132	-	-	L	-	-	d	v	L	t	-	-	-	-	L	M,L	p
133	-	-	M,L	-	-	cc	v	L	t	-	-	-	-	L	M	p
134	-	-	M,L	-	-	cc	v	L	t	-	-	-	-	L	L	p
135	-	-	M,L	-	-	x	v	L	t	-	H,M	-	-	L	M.L	p
136	-	-	M,L	-	-	x	v	L	t	-	L	-	-	L	M.L	n
137	-	-	M,L	-	-	m	v	L	t	-	-	t	-	L	M.L	n
138	-	L	M,L	-	-	m	v	L	t	-	-	f	-	L	M.L	p
139	-	H,M	M	-	-	m	v	L	t	-	-	f	-	L	M.L	p
140	-	H,M	L	-	-	m	v	L	t	-	-	f	-	L	M.L	n
141	a	H	M	-	-	q	v	L	t	-	-	-	-	L	M.L	n
142	a	M,L	M	-	-	q	v	L	t	-	-	-	-	L	M.L	p
143	a	-	L	-	-	q	v	L	t	-	H,M	-	-	L	M.L	p
144	a	-	L	-	-	q	v	L	t	-	L	-	-	L	M.L	n
145	b	-	L	-	-	q	v	L	t	-	-	-	-	L	M.L	p
146	b	-	M	-	-	q	v	L	t	t	-	-	-	L	M.L	p
147	b	-	M	-	-	q	v	L	t	f	-	-	-	L	M.L	n
148	-	-	M,L	-	-	k	v	L	t	-	-	-	s	L	M.L	p
149	-	L	M,L	-	-	k	v	L	t	-	-	-	g	L	M.L	n
150	-	H,M	M,L	-	-	k	v	L	t	f	-	-	g	L	M.L	n
151	-	H,M	M,L	u	-	k	v	L	t	t	-	-	g	L	M.L	p
152	-	H,M	M,L	y	-	k	v	L	t	t	-	-	g	L	M.L	n
153	-	-	M	-	-	aa	v	L	t	-	-	t	-	L	M.L	p
154	-	-	M	u	-	aa	v	L	t	-	-	f	-	L	M.L	n
155	a	-	M	y	-	aa	v	L	t	-	-	f	-	L	M.L	p
156	b	-	M	y	-	aa	v	L	t	-	-	f	-	L	M.L	n
157	-	H	L	-	-	aa	v	L	t	-	-	-	-	L	M.L	p
158	-	M,L	L	u	-	aa	v	L	t	-	-	-	-	L	M.L	n
159	a	M,L	L	y	-	aa	v	L	t	-	-	-	-	L	M.L	n
160	b	M,L	L	y	-	aa	v	L	t	-	-	-	-	L	M.L	p
161	-	-	M,L	-	-	w	v	L	t	-	-	f	-	L	M.L	p
162	a	-	M,L	-	-	w	v	L	t	-	-	t	-	L	M.L	p
163	b	-	M,L	u	-	w	v	L	t	-	-	t	-	L	M.L	n
164	b	H,M	M,L	y	-	w	v	L	t	-	-	t	-	L	M.L	p
165	b	L	M	y	-	w	v	L	t	-	-	t	-	L	M.L	p
166	b	L	L	y	-	w	v	L	t	t	-	t	-	L	M.L	p
167	b	L	L	y	-	w	v	L	t	f	-	t	-	L	M.L	n
168	-	-	M,L	-	-	c	v	L	t	-	-	-	-	L	M	n
169	a	L	M,L	-	-	c	v	L	t	-	-	-	-	L	L	n
170	a	H,M	M	-	-	c	v	L	t	-	-	-	-	L	L	p
171	a	H,M	L	-	-	c	v	L	t	-	-	-	-	L	L	n
172	b	-	M,L	-	-	c	v	L	t	-	-	f	-	L	L	p
173	b	-	M,L	y	-	c	v	L	t	-	-	t	-	L	L	p
174	b	H,M	M,L	u	-	c	v	L	t	-	-	t	-	L	L	n
175	b	L	L	u	-	c	v	L	t	-	-	t	-	L	L	p
176	b	L	M	u	-	c	v	L	t	f	-	t	-	L	L	n
177	b	L	M	u	-	c	v	L	t	t	H,M	t	-	L	L	n
178	b	L	M	u	-	c	v	L	t	t	L	t	-	L	L	p

Reference List

Abraham, A. (2005). Rule-based Expert Systems. In P. H. Sydenham & R. Thorn (Eds.), *Handbook of Measuring System Design (*pp. 909-919). Hoboken, NJ: John Wiley & Sons.

Aytug, H., Boylu, F., & Koehler, G. J. (2006). Learning in the presence of self-interested agents. *Proceedings of the 39th Hawaii International Conference on System Sciences,*158b.

Bewsell, G. R. (2008). Mechanisms that impact online auction trust. *19th Australasian Conference on Information Systems (ACIS '08), Paper 37*.

Boylu, F. (2006). *Strategic learning* (Unpublished Doctoral Dissertation). University of Florida, Gainsville, FL.

Boylu, F., Aytug, H., & Koehler, G. J. (2009). Principal-agent learning. *Decision Support Systems, 47*, 75-81.

Boylu, F., Aytug, H., & Koehler, G. J. (2010). Induction over strategic agents. *Information Systems Research, 21*(1), 170-189.

Bradley, T. (2011, July). Protect your data from the breach epidemic. *PCWorld, 29*(7), 35-36.

Clancey, W. J. (1983). The epistemology of a rule-based expert system--a framework for explanation. Artificial Intelligence, 20(3), 215-251.

Clark, P., & Niblett, T. (1989). The CN2 induction algorithm. Machine Learning, 3(4), 261-283.

Dalvi, N., Domingos, P., Mausam, Sanghai, S., & Verma, D. (2004). Adversarial classification. Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), , 99-108.

Domingos, P. (1999). MetaCost: A general method for making classifiers cost-sensitive. Proceedings of the 5th International Conference on Knowledge Discovery and Data Mining, , 155-164.

Elkan, C. (2001). The foundations of cost-sensitive learning. Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI '01), , 973-978.

Esmeir, S., & Markovitch, S. (2008). Anytime induction of low-cost, low-error classifiers: a sampling-based approach. Journal of Artificial Intelligence Research, 33, 1-31.

Fawcett, T. (2004, March 16). ROC graphs: Notes and practical considerations for researchers (HPL-2003-4). Palo Alto, CA: HP Laboratories.

Hirsh, H. (1994). Generalizing version spaces. Machine learning, 17, 5-46.

Hoffman, D. L., Novak, T. P., & Peralta, M. (1999). Building consumer trust online. Communications of the ACM, 42(4), 80-85.

Jiang, Z., Mookerjee, V. S., & Sarkar, S. (2005). Lying on the Web: Implications for expert systems redesign. Information Systems Research, 16(2), 131-148.

Lee, C. C. (1990). Fuzzy logic in control systems: Fuzzy logic controller - Part I. IEEE Transactions on Systems, Man and Cybernetics, 20(2), 404-418.

Ling, C. X., Sheng, V. S., & Yang, Q. (2006). Test strategies for cost-sensitive decision trees. IEEE Transactions on Knowledge and Data Engineering, 18(8), 1055-1067.

Mannino, M. V., & Koushik, M. V. (2000). The cost-minimizing inverse classification problem: A genetic algorithm approach. Decision Support Systems, 29, 283-300.

McDermott, J. (1981). Domain knowledge and the design process. In Proceedings of the 18th Design Automation Conference (pp. 580-588). Nashville, TN: IEEE Press.

Michie, D., Spiegelhalter, D. J., & Taylor, C. C. (Eds.). (1994). Machine Learning, Neural and Statistical Classification. Englewood Cliffs, NJ: Prentice Hall.

Mitchell, T. M. (1982). Generalization as search. Artificial Intelligence, 18(2), 203-226.

Mookerjee, V., Mannino, M., & Gibson, R. (1995). Improving the performance stability of inductive expert systems under input noise. Information Systems Research, 6(4), 328-356.

Morey, R. C. (1982). Estimating and improving the quality of information in a MIS. Communications of the ACM, 25(5), 337-342.

Nunez, M. (1991). The use of background knowledge in decision tree induction. Machine Learning, 6, 231-250.

Park, S. -B., Hwang, S., & Zhang, B. -T. (2003). Mining the risk types of human papillomavirus (hpv) by AdaCost. In V. Marik, W. Retschitzegger, & O. Štepánková (Eds.), Database and Expert Systems Applications (Vol. 2736, pp. 403-412). : Springer.

Parssian, A., Sarkar, S., & Jacob, V. S. (2009). Impact of the union and difference operations on the quality of information products. Information Systems Research, 20(1), 99-120.

Porter, J. (2007, May 24). A booming business in MBA coaches. Retrieved July 9, 2011, from http://www.businessweek.com/bschools/content/may2007/bs20070524_906621.htm

Quinlan, J. R. (1986). Machine Learning: An Artificial Intelligence Approach, Vol. II. Los Altos, CA: Morgan Kaufmann.

Reinwald, L. T., & Soland, R. M. (1966). Conversion of limited-entry decision tables to optimal computer programs: Minimum average processing time. Journal of the ACM, 13(3), 339-358.

Robin (2010, November 1). Rule based expert systems. Retrieved September 26, 2011, from http://intelligence.worldofcomputing.net/expert-systems-articles/rule-based-expert-systems.html

Rulequest Research (2010, December). Free Software Downloads. Retrieved September 4, 2011, from http://www.rulequest.com/download.html

Shwayder, K. (1974). Extending the information theory approach to converting limited-entry decision tables to computer programs. Communications of the ACM, 17(9), 537.

Tan, M. (1993). Cost-sensitive learning of classification knowledge and its applications in robotics. Machine Learning, 7, 7-33.

Tapscott, D. (1999, April 26). IBM is showing leadership on the privacy issue. Computerworld, 33(17), 34.

Turney, P. D. (1995). Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm. Journal of Artificial Intelligence Research, 2, 369-409.

Vadera, S. (2010). CSNL: A cost-sensitive non-linear decision tree algorithm. ACM transactions on Knowledge Discovery from Data, 4(2), 6.1-6.25.

Viaene, S., Derrig, R. A., & Dedene, G. (2004). Cost-sensitive learning and decision making for Massachusetts pip claim fraud data. International Journal of Intelligent Systems, 19, 1197-1215.

Viola, P., & Jones, M. (2002). Fast and robust classification using asymmetric AdaBoost and a detector cascade. Advances in Neural Information Processing Systems, 14(2), 1311-1318.

Zadrozny, B., Langford, J., & Abe, N. (2003). Cost-sensitive learning by cost-proportionate example weighting. Preceedings of the 3rd IEEE International Conference on Data Mining, , 435-442.