QUALITATIVE TEST-COST SENSITIVE CLASSIFICATION

A THESIS

SUBMITTED TO THE DEPARTMENT OF COMPUTER ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Mümin Cebe

August, 2008

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Çiğdem Gündüz Demir (Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. H. Altay Güvenir

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Tolga Can

Approved for the Institute of Engineering and Science:

Prof. Dr. Mehmet B. Baray
Director of the Institute

ABSTRACT

QUALITATIVE TEST-COST SENSITIVE CLASSIFICATION

Mümin Cebe
M.S. in Computer Engineering
Supervisor: Assist. Prof. Dr. Çiğdem Gündüz Demir
August, 2008

Decision making is a procedure for selecting the best action among several alternatives. In many real-world problems, decision has to be taken under the circumstances in which one has to pay to acquire information. In this thesis, we propose a new framework for test-cost sensitive classification that considers the misclassification cost together with the cost of feature extraction, which arises from the effort of acquiring features. This proposed framework introduces two new concepts to test-cost sensitive learning for better modeling the real-world problems: qualitativeness and consistency.

First, this framework introduces the incorporation of qualitative costs into the problem formulation. This incorporation becomes important for many real world problems, from finance to medical diagnosis, since the relation between the misclassification cost and the cost of feature extraction could be expressed only roughly and typically in terms of ordinal relations for these problems. For example, in cancer diagnosis, it could be expressed that the cost of misdiagnosis is larger than the cost of a medical test. However, in the test-cost sensitive classification literature, the misclassification cost and the cost of feature extraction are combined quantitatively to obtain a single loss/utility value, which requires expressing the relation between these costs as a precise quantitative number.

Second, the proposed framework considers the consistency between the current information and the information after feature extraction to decide which features to extract. For example, it does not extract a new feature if it brings no new information but just confirms the current one; in other words, if the new feature is totally consistent with the current information. By doing so, the proposed framework could significantly decrease the cost of feature extraction, and hence, the overall cost without decreasing the classification accuracy. Such consistency

behavior has not been considered in the previous test-cost sensitive literature.

We conduct our experiments on three medical data sets and the results demonstrate that the proposed framework significantly decreases the feature extraction cost without decreasing the classification accuracy.

Keywords: Cost-sensitive learning, qualitative decision theory, feature extraction cost, feature selection, decision theory.

ÖZET

NİTEL MALİYETE DUYARLI SINIFLANDIRMA

Mümin Cebe
Bilgisayar Mühendisliği, Yüksek Lisans
Tez Yöneticisi: Assist. Prof. Dr. Çiğdem Gündüz Demir
Ağustos, 2008

Karar verme bir çok seçeneğin arasından en iyiyi seçme işidir. Gerçek uygulamalarda, karar vericinin en iyi karara varabilmesi için gerekli olan bilginin bir maliyeti vardır. Bu tezde, karar verme aşamasında ortaya çıkan hatalı kararın maliyeti ile en iyi kararı vermek için kullanılan bilginin maliyetini beraber ele alan yeni bir öğrenme yöntemi önerilmiştir. Önerilen bu yeni yöntem, maliyete duyarlı öğrenmeye nitelliği ve tutarlılığı iki yeni kavram olarak sunmuştur.

Bu çalışmayla ilk olarak, nitel maliyet kavramı makine öğrenmesi sürecine dahil edilmiştir. Verilen kararın hatalı olmasından kaynaklanan maliyet ile bu kararı verebilmek için kullanılacak bilginin maliyeti arasındaki ilişkinin bir çok problemde nicel olarak tanımlanamamasından dolayı nitel maliyet kavramı önemlidir. Örneğin kanser teşhisinde, yanlış teşhis yapmanın maliyetinin teşhis için kullanılan testlerin maliyetinden daha büyük olduğu söylenebilir. Fakat, bu iki kavram arasındaki ilişkinin nicel olarak tanımlanması zordur. Daha önce maliyete duyarlı öğrenmeyle ilgili yapılan çalışmalar bu iki maliyetin birbiriyle olan ilişkisinin nicel olarak tanımlanmasını şart koşmuşlardır. Bu yüzden, önerilen nitel maliyet ilişkisi kavramı bu konu hakkındaki çalışmalara yeni bir boyut kazandırmıştır.

İkinci olarak ise, bu tezde yapılan çalışma yeni elde etmeyi beklediğimiz bilgi ile şimdi sahip olduğumuz bilgi arasındaki tutarlığı göz önüne almıştır. Eğer yeni elde edilecek bilgi şimdiki bilgimize yeni bir şey eklemiyor ya da bir başka deyişle yeni elde edilecek bilgi şimdiki bilgimizle tutarlı ise önerilen yöntem yeni elde edilecek bilgi için gerekli olan maliyetin karşılanmasını reddetmektedir. Böylece önerdiğimiz yöntemle, karar verme aşamasında karar verme sürecini etkilemeyen bilgi için maliyet yapılmamış olmaktadır. Bu kavram daha önceki çalışmalarda hiç kullanılmamıştır.

Üç farklı medikal veri kümesi üzerindeki deneylerimiz, önerdiğimiz yöntemin teşhisteki doğruluk oranını etkilemeden, kullanılan medikal testlerin maliyetini büyük ölçeklerde azaltmayı başardığını göstermiştir.

Anahtar sözcükler: Maliyete duyarlı öğrenme, karar teorisi, nitel karar teorisi, öznitelik çıkarma maliyeti, öznitelik seçme, karar teorisi.

Acknowledgement

To my advisor, Professor Çiğdem Gündüz Demir, thank you for your trust in me. Thank you for your patience to teach me how to do research and how to improve my writing (including hand-writing!). Thank you for your scientific and personal guidance. I learned a lot from your superior advices. Thank you for your generosity, enthusiasm and patience for being a great teacher. Thank you for your careful reading of my thesis. And thank you for your endless support in several key moments. I will always be proud to have been your student.

To Professor H. Altay Güvenir and Professor Tolga Can, thank you for kindly accepting to join to my thesis committee and thank you for your valuable contributions and suggestions about my thesis.

To my research group members: Akif Burak Tosun, Tuncer Erdoğan and Melih Kandemir. It is a pleasure to work with these guys in the same group. Another special thanks to Erkan Okuyan and Özgür Bağlıoğlu for their help about my writing and for their time.

Thank you to my professors in Ege University: Aybars Uğur and Muhammed Cinsdikici, for their motivation and helpful suggestions during my undergraduate study, they are the first people to inspire me in studying artificial intelligence. Thank you to my group members in EgeYZ: Hakan Ensari, Yusuf Aytar and Selen Özgür. Special thanks to Yusuf Aytar, my close friend for his moral support and help during my graduate study. I have learned a lot from his resolution and superior motivations.

Last, but not the least, I thank my family for their understanding and love. You are the motivating force behind me at all times. Thank you for everything you give me.

To My Parents,

Contents

1 Introduction **1**

1.1 Related Work 3

1.1.1 Qualitative Decision Theory 3

1.1.2 Cost-Sensitive Learning 5

1.2 Contribution of This Thesis 7

2 Background **9**

2.1 Cost-Sensitive Learning 9

2.1.1 Extensions of Cost-Sensitive Learning 15

2.2 Qualitative Decision Theory 16

2.2.1 Qualitative Probabilities 18

2.2.2 Qualitative Consequences 19

2.2.3 Qualitative Utilities 20

3 Methodology **22**

3.1 Methodology 22

3.2 Consistency-based loss functions 23

3.3 Qualitative decision making for test-cost sensitive classification . . 27

3.3.1 $\texttt{extract}_k$-vs-$\texttt{extract}_m$ 29

3.3.2 $\texttt{extract}_k$-vs-`classify` 32

3.3.3 $\texttt{extract}_k$-vs-`reject` 34

3.3.4 `classify`-vs-`reject` 36

3.4 Qualitative test-cost sensitive classification algorithm 37

4 Experiments **42**

4.1 Experimental Setup 43

4.1.1 Bupa Liver Disorder Dataset 43

4.1.2 Heart Disease Dataset 44

4.1.3 Thyroid Disease Dataset 45

4.2 Results 47

4.2.1 Decision Tree 48

4.2.2 Hidden Markov Model 55

4.3 Comparisons 62

4.3.1 Comparison by ICET 62

4.3.2 Comparison by POMDP 63

5 Conclusions **66**

5.1 Discussions 66

5.2 Conclusions 67

List of Figures

2.1 Mapping function: From all training samples to all corresponding output labels. 10

2.2 The sequential cost-sensitive classification algorithm 12

2.3 Decision steps for patient 1. 14

2.4 Decision steps for patient 2. 14

2.5 Decision steps for patient 3. 15

3.1 For $\texttt{extract}_k$-vs-$\texttt{extract}_m$ comparison, the cases and the rules to determine what action to take. 32

3.2 For $\texttt{extract}_k$-vs-`classify` comparison, the cases and the rules to determine what action to take. 34

3.3 For $\texttt{extract}_k$-vs-`reject` comparison, the cases and the rules to determine what action to take. 35

3.4 For `classify`-vs-`reject` comparison, the cases and the rules to determine what action to take. 37

3.5 The schematic representation of our test-cost sensitive classification algorithm. 38

3.6 Derivation of the SMALL value: (a) the histogram of the distinct $|X|/|Y|$ ratios of ambiguous cases and the two Gaussian components estimated on these ratios and (b) posteriors obtained using the estimated Gaussians and prior probabilities. 41

4.1 Derivation of the SMALL value: (a),(c), and (e) are the histograms of the distinct $|X|/|Y|$ ratios of ambiguous cases and two Gaussian components estimated on these ratios for the Bupa (fold1, fold2, and fold3, respectively). (b),(d) and (e) are posteriors obtained using estimated Gaussians and prior probabilities for the Bupa (fold1, fold2, and fold3, respectively). Here, decision tree classifiers are used. 53

4.2 Derivation of the SMALL value: (a),(c), and (e) are the histograms of the distinct $|X|/|Y|$ ratios of ambiguous cases and two Gaussian components estimated on these ratios for the Heart (fold1, fold2 and fold3, respectively). (b),(d), and (e) are posteriors obtained using estimated Gaussians and prior probabilities for the Heart (fold1, fold2, and fold3, respectively). Here, decision tree classifiers are used. 54

4.3 Derivation of the SMALL value: (a) is the histogram of the distinct $|X|/|Y|$ ratios of ambiguous cases and two Gaussian components estimated on these ratios for the Thyroid. (b) is posteriors obtained using estimated Gaussians and prior probabilities for the Thyroid. Here, decision tree classifiers are used. 55

4.4 Derivation of the SMALL value: (a),(c) and (e) are the histograms of the distinct $|X|/|Y|$ ratios of ambiguous cases and two Gaussian components estimated on these ratios for the Bupa ((fold1, fold2, and fold3, respectively)). (b),(d), and (e) are posteriors obtained using estimated Gaussians and prior probabilities for the Bupa (fold1, fold2, and fold3, respectively). Here, HMM classifiers are used. 60

4.5 Derivation of the SMALL value: (a),(c) and (e) are the histograms of the distinct $|X|/|Y|$ ratios of ambiguous cases and two Gaussian components estimated on these ratios for the Heart (fold1, fold2, and fold3, respectively). (b),(d), and (e) are posteriors obtained using estimated Gaussians and prior probabilities for the Heart (fold1, fold2, and fold3, respectively). Here, HMM classifiers are used. 61

4.6 Derivation of the SMALL value: (a) is the histogram of the distinct $|X|/|Y|$ ratios of ambiguous cases and two Gaussian components estimated on these ratios for the Thyroid. (b) is posteriors obtained using estimated Gaussians and prior probabilities for the Thyroid. Here, HMM classifiers are used. 62

List of Tables

2.1 Complex Cost Matrix 11

2.2 Simple Cost Matrix 11

2.3 Test Costs 12

2.4 The values of the $h(x)$ function 13

3.1 Definition of the conditioned loss function for feature extraction, classification, and reject actions. 23

4.1 Description of the features and their extraction cost for the Bupa Liver Disorder dataset. 44

4.2 Description of the features and their extraction cost for the Heart Disease dataset. 45

4.3 Description of the features and their extraction cost for the Thyroid Disease dataset. 46

4.4 For the Bupa dataset, the results obtained by our qualitative test-cost sensitive algorithm and those obtained by the baseline classifier, which uses all of the features in its decision tree construction. 49

4.5 For the Bupa dataset, the results are obtained by our qualitative test-cost sensitive algorithm when the consistency is not considered.Here, decision tree classifiers are used 49

4.6 For the Heart dataset, the results obtained by our qualitative test-cost sensitive algorithm and those obtained by the baseline classifier, which uses all of the features in its decision tree construction. 50

4.7 For the Heart dataset, the results are obtained by our qualitative test-cost sensitive algorithm when the consistency is not considered. Here, decision tree classifiers are used 50

4.8 For the Thyroid dataset, the results obtained by our qualitative test-cost sensitive algorithm and those obtained by the baseline classifier, which uses all of the features in its decision tree construction. 51

4.9 For the Thyroid dataset, the results are obtained by our qualitative test-cost sensitive algorithm when the consistency is not considered. Here, decision tree classifiers are used 51

4.10 For the Bupa dataset, the results obtained by our qualitative test-cost sensitive algorithm and those obtained by the baseline classifier, which uses all of the features in its HMM. 56

4.11 For the Bupa dataset, the results are obtained by our qualitative test-cost sensitive algorithm when the consistency is not considered. Here, HMM classifiers are used 56

4.12 For the Heart dataset, the results obtained by our qualitative test-cost sensitive algorithm and those obtained by the baseline classifier, which uses all of the features in its HMM. 57

4.13 For the Heart dataset, the results are obtained by our qualitative test-cost sensitive algorithm when the consistency is not considered. Here, HMM classifiers are used 58

4.14 For the Thyroid dataset, the results obtained by our qualitative test-cost sensitive algorithm and those obtained by the baseline classifier, which uses all of the features in its decision tree construction. 58

4.15 For the Thyroid dataset, the results are obtained by our qualitative test-cost sensitive algorithm when the consistency is not considered. Here, HMM classifiers are used 59

4.16 The results obtained by our qualitative test-cost sensitive algorithm when a decision tree classifier is used and the results of the ICET algorithm; the results of ICET are the best reported ones. . 63

4.17 The results obtained by our qualitative test-cost sensitive algorithm when an HMM classifier is used and the results of the algorithm developed by Ji and Carin [23] 65

Chapter 1

Introduction

Decision making is a process that surrounds the world. Every living faces many situations where they have to make decision among alternative choices, and their benefit strictly depends on the outcomes of this decision. In general, at the time of decision, the outcomes of the decision are uncertain. Thus, one should try to maximize his/her expected benefit considering uncertain environments. Computational models have proved their ability to make rational decisions for the problems that have large amounts of uncertainty.

In literature, Neumann and Morgenstern [28] first represent, rational decision making in uncertain environments. The decisions are performed according to the expected utility concept. Expected utility is being used in decision making and finds large application areas from finance to medicine. In this well-known representation of rational decision making, all parameters that affect decision making must be defined and combined quantitatively. However, some problems may occur when a decision maker has a lack of adequate knowledge and/or the decision maker is incapable of correctly estimating numerical values for his/her preferences. One of such cases arises, when it is not possible to define numerical values for outcomes, for example when expressions such as "I would prefer A rather than B" are present. This expression makes such non-numerical preferences important while making decision. Such preference should be used to define a qualitative utility/loss value. Another case arises when the decision maker

has the ability to define qualitative/non-numerical probabilities such as "A is much more probable than B". These expressions operationally translate quantitative/numerical relations into the qualitative/non-numerical relations. The traditional decision theory fails where environment has qualitative/non-numerical probabilities or utilities/loses. Because of the difficulty of defining probabilities and/or utilities/loses quantitatively/numerically for many problems and because of the necessity of handling these values for making the *best* decisions, the qualitative decision theory attracts our attention.

One of the application areas in which the necessity of using qualitative utilities/losses arises is the test-cost sensitive classification. Test-cost sensitive classification considers the misclassification cost together with the cost of feature extraction to minimize the overall cost of the decision process. Misclassification cost is the cost that occurs when a decision maker decides incorrectly, whereas feature extraction cost is the cost that arises from acquiring the feature. In the test-cost sensitive literature, the misclassification cost and the cost of feature extraction are numerically defined and combined to obtain a quantitative utility/loss value. However, although the feature extraction cost could be typically defined in terms of numbers (most of the time, the amount of money that one should pay to obtain the value of a feature), the misclassification cost could not easily be quantified in terms of numerical values for many applications. Generally, there is a preference relation between the misclassification cost and the cost of feature extraction. Thus, one should balance this relation to take the *best* decision. To understand the importance of expressing generic preferences in test-cost sensitive learning, consider the following two examples: first, let us consider a situation where a decision maker tries to decide whether a patient has a cold or not. In this problem, the decision maker can ask some simple questions, or can perform a blood test on the patient in order to learn whether symptoms of cold exist or not. Although the decision maker knows that the blood test is more reliable test than diagnostic questions, he/she generally avoids to perform such tests, since these tests have some cost and the decision maker has a general belief of the test cost generally being greater than the misclassification cost. For the second example, consider the case of cancer diagnosis. The decision maker tends

to perform a higher number of tests to make a decision, because this example is a situation in which the decision maker has the belief of the misclassification cost generally being greater than the test costs. Thus, for a given application, a rational decision maker has to consider the relation between the misclassification cost and the cost of feature extraction to make the *best* decision.

In this thesis, we define a new test-cost sensitive learning scheme in which we use the qualitativeness concept, for the first time. To this end, we define qualitative conditioned-loss function to consider the generic preferences of the user about different types of costs and apply this representation for test-cost sensitive learning. For the remainder of this chapter, we first review the related work and then explain our contribution to test-cost sensitive learning.

1.1 Related Work

1.1.1 Qualitative Decision Theory

Qualitative decision theory studies the incorporation of qualitative knowledge to decision making problems [24]. As opposed to the classical approach postulated by von Neumann and Morgenstern [28], where probabilities and utilities should be defined as exact numerical values, the qualitative decision theory enables to define probabilities, and/or utilities/losses as qualitative values; i.e the qualitative decision theory relaxes the strict requirement that both probabilities and utilities/losses should be defined and combined quantitatively. The main issues about the use and the need of qualitativeness in machine learning are discussed in [4] and [24]. In literature, previous studies related to qualitativeness in machine learning appear in two groups. One group of studies works on the construction of qualitative probabilistic Bayesian networks [29, 31, 32, 33]. The other one focuses on the decision making problem when the utility/loss values are defined qualitatively and in an ordinal scale, reflecting the generic preference. We first review studies about qualitative Bayesian networks then mention the studies about qualitative decision making.

Bayesian networks represent a set of variables and their probabilistic relationships to use for quantitative reasoning [26]. The first attempt to extend the quantitative Bayesian network to qualitative one is done by [29, 30]. In these studies, Wellmann has defined the relationships between variables in network as positive(+), negative(-), null(0) or ambiguous(?). A positive (+) relation between two variables implies that a high value of one of variables makes more likely that the other variable also has a high value. A negative(-) relation between two variables implies that a high value of one variable makes more likely that the other variable has a low value. A null(0) relation between two variables implies that there is no correlation between these variables. The unknown or ambiguous (?) relation is defined when positive and negative relations are combined. The difference between ambiguous(?) and null(0) signs is that the ambiguous sing(?) implies that the real value of sign could be a positive(+), negative(-) or null(0), whereas a null(0) sign implies no correlation between variables. The reasoning in qualitative networks is accomplished by combining two or more variables using additive [29, 30] or multiplicative relations [27]. The difficulty arises when combining positive and negative relations that leads a relation type of ambiguity (?). The ambiguity causes uninformative signs during inference. In [31], Renooif and von der Goag associate a relative strength to relations in order to avoid the ambiguous signs, which appear in qualitative networks. Their combination algorithm is similar to the previous combination algorithms defined in [29, 30], except that they allow the the strong variable to dominate the relatively weaker one for overcoming ambiguous signs. They also extend their previous work with situational sign concept in [25]. The situational sign depends on the whole network state; they determine the situational sign of the current ambiguous sign as positive, negative or null according to the network state, and thus, they impede ambiguous signs in qualitative networks.

The other group of studies works on the qualitative decision making problem. Studies in this group could mainly be grouped into two. One group focuses on building symbolic models for decision making [2, 5, 6]. They allow representation of probabilities and preferences of being in the form of human like expressions such as “if we are going out tonight, I would prefer to go to a restaurant for dinner”.

This class of studies has the common main idea that compares actions on the most plausible states of the world. For instance in [2], the preferences are modeled as $I(\beta|\alpha)$, which means that if the α is the case, the most preferred action is β. A desired decision is given according to the most preferred action conditioned by the most plausible state. The other class of studies mainly depends on the ordering of probabilities and preferences [1, 3, 37, 38]. These approaches use the ordered scale of probabilities and preferences to come with a decision by applying maximin and minimax criteria on the ordinal scale. In practice, the use of ordinal scales translates the quantitative probabilities and preferences (utility/loses) into the qualitative ones.

1.1.2 Cost-Sensitive Learning

There are many studies related to cost-sensitive learning that investigate different types of cost [13]. In literature, the most commonly investigated cost type is 0/1 cost. The classifiers sensitive to 0/1 cost aim at minimizing number of errors during classification. However, these classifier are not adequate for the problems in which some classification errors are more important than the others. To overcome this deficiency, different costs are defined for different types of errors; this types of cost is called as misclassification cost [9, 10, 12]. One common way for making classifier to be sensitive to different misclassification costs is to rebalance the proportion of class samples in training set according to the ratio of misclassification cost values [10]. As another way, MetaCost has been proposed [9]. MetaCost, first, learns the associated class probabilities for each instance in the training set by using any of the classification algorithms. After the probabilities have been learned, MetaCost relabels each sample according to the associated probabilities and misclassification cost. Then, it learns another model for the new modified training set.

Another type of cost is the cost of computation. The computational cost includes both static complexity, which arises from the size of a computer program [7], and dynamic complexity, which is incurred during training and testing a classifier [8]. The computational cost is important in training when the data size

is large and/or the data dimension is high and in testing, especially for real-world applications, when the response time is critical (e.g., in the case of handwritten-character recognition in a personal digital assistant).

The other cost type is the cost of feature extraction, which occurs during acquiring features. This type of cost is important in especially real-world applications. In literature, only a few studies have investigated for the cost of feature extraction. A large group of these studies focus on constructing decision trees in a most accurately but, at the same time, a least costly manner [14, 15, 16, 17, 18, 19]. In [14, 15, 16], the test costs is considereddduring classification. In these studies, the splitting criteria of these decision trees, which selects attributes greedily, combine the information gain and test costs to build cost-sensitive decision trees. In [17], Turney also builds decision trees using criterion described in [14, 15]. However, Turney employs a genetic search by modifying the test costs empirically (assigns random test costs and use these costs in decision trees) to build a population of decision trees. The population is then evaluated according to a utility function that uses the real test costs (not random ones) and misclassification costs. The method in [17] is an influential method that sets the fundamentals of cost sensitive learning by considering both misclassification costs and test costs. Davis and Yang, in [18, 19], change the splitting criterion described in [14, 15] by defining a utility function that additionally considers considers misclassification cost together with test costs.

Another group of studies uses a sequential selection procedure based on the utility that a feature will introduce [11, 19, 20, 21]. The utility of a feature is computed by considering the information gain and cost of extracting feature. The information gain of the feature is obtained by taking difference between the current information and the information to be obtained after extracting the feature. These studies have to estimate information gain for the feature to be extracted. Estimation is done by either estimate the value of the feature [11, 19, 20] or estimating the posterior probabilities when the feature is used [21].

The other group of studies uses a Markov decision process model and selects features according to an optimal policy that is learned on this model with the goal

of minimizing the expected total cost [22, 23]. While a state is defined for each possible combination of features in [22], the states are tied to mixture components of particular features and only partially observable in [23]. However, in this method, the learning process may take higher computational costs comparing the most of other aforementioned cost-sensitive learning algorithms.

1.2 Contribution of This Thesis

In this thesis, we introduce a novel test-cost sensitive learning approach that considers the misclassification cost together with the cost of feature extraction. In this approach, we introduce two new concepts to test-cost sensitive learning: qualitativeness and consistency. By introducing these concepts, we address two important issues, which are commonly the cases in real-world, as opposed to the previous studies. As the first issue, for the definition of a utility/loss function, the previous studies have combined the misclassification cost and the cost of feature extraction quantitatively. To do so, the misclassification cost is expressed as a precise quantitative value that is selected by considering cost of feature extraction (please note that most of the time, the feature extraction cost is easily expressed as a quantitative values; e.g., in medical diagnosis, this cost is commonly the amount that one should pay for corresponding medical test) and its importance over the misclassification cost. However, in real-world applications, most of the time, decision makers cannot express such importance in terms of precise quantitative values. Instead, they only express it roughly (typically in terms of ordinal relations); for instance, in cancer diagnosis, it could be expressed that the cost of a medical test is smaller than that of misdiagnosis. As the second issue, all of the previous studies have selected features based on the current information and the estimated information obtained after feature extraction. None of them considers the consistency between this information. On the other hand, in real-world applications, the consistency is commonly important. For example, in the case of medical diagnosis, a doctor may not order an expensive test for a patient, if the doctor is confident enough that the test confirms the current decision about this patient. Instead, the doctor would like to order a test for which he/she thinks

that it could change his/her decision. By doing so, the cost of extra tests, and hence, the overall cost could significantly be decreased without decreasing the diagnosis accuracy.

In order to successfully address these aforementioned issues, we propose to use a Bayesian decision theoretical framework in which *1.*) the misclassification cost and the cost of feature extraction are combined *qualitatively* and *2.*) the loss function is conditioned with the decisions taken using current and estimated information as well as the *consistency* among them. By combining misclassification and feature extraction costs qualitatively, the proposed algorithm eliminates the major requirement that the user should determine exact quantitative constants to combine the misclassification and feature extraction costs. By using consistency between current and estimated information, the proposed algorithm tends to extract features that are expected to change the current decision (i.e., yield inconsistent decisions) and to stop the extraction if all possible features are expected to confirm the current decision (i.e., yield consistent decisions). This leads to less costly but equally accurate decisions.

Chapter 2

Background

This chapter formally defines the test-cost sensitive learning and qualitative decision theory. Test-cost sensitive learning is an approach to learning the machine learning problems with the objective of minimizing the expected cost of the learning process where both features and classification errors have costs. Qualitative decision theory is the extension of classical decision theory which enables qualitative probability and/or utility/loss function definition during decision process.

2.1 Cost-Sensitive Learning

Test-cost sensitive learning is based on supervised learning. An instance in supervised learning is represented as $< x_i, c_i >$, where x_i represents an input vector of features of the instance and c_i represents the class of instance x_i. The aim of supervised learning is to learn a mapping function $h(x)$ from x_i to c_i for all training samples as illustrated in Figure 2.1.

This mapping function $h(x)$ is selected to minimize the expected risk, which is written in terms of loss function λ and the probabilities of state of nature c_i as given in Equation 2.1.

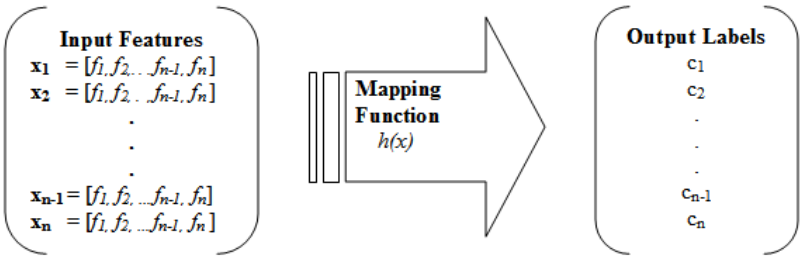


Figure 2.1: Mapping function: From all training samples to all corresponding output labels.

$$R(\lambda(\alpha_i|x)) = \sum_{j=1}^{N} P(c_j|x)\ \lambda(\alpha_i|c_j) \tag{2.1}$$

In this equation, $c_1, ..., c_N$ is the set of N states of nature,$\alpha_1, ..., \alpha_K$ is the set of K possible actions and $\lambda(\alpha_i|c_j)$ is the loss value incurred when the action α_i is selected when the state of nature is c_j. The best action α^* is selected in a way that minimizes the expected risk $R(\lambda(\alpha_i|x))$. If the loss function $\lambda(\alpha_i|c_j)$ has a value of 1 for incorrect mapping and has value of 0 for correct mapping, Equation 2.1 focuses on to minimize the number of errors ignoring the feature extraction costs. However, in cost sensitive learning, the objective is not only finding best mapping function from inputs to classes, but also minimizing the total cost of the learning process. Cost-sensitive learning allows us to minimize the feature extraction costs via minimizing the expected risk. Here, we will define basic notation and terms in cost sensitive learning and show these notations and concepts on a simple problem.

The loss function used in cost sensitive learning generally defined as n by m cost matrix. The actions and outcomes determine the size of the cost matrix. As an example, let us present a cost matrix of a problem that a doctor try to minimize the expected risk during cold diagnosis. Table 2.1 shows the cost matrix

of this problem.

Predicted Classes	**Correct Classes**	
	Cold	*No Cold*
Cold	-100	200
No Cold	1000	0

Table 2.1: Complex Cost Matrix

The table illustrates that if a patient has a cold and the doctor misdiagnoses him/her, cost of such decision is 1000 penalty points because of further risk to the patient health. If the doctor diagnoses him/her correctly the given reward for this decision is -100 point. If the patient has not got a cold and doctor misdiagnoses him/her, the penalty in this case is 200 point because of dissatisfaction of patient. But it is not as high as before because there is no risk for the health status of patient (of course we ignore the side effects of the treatment). In the case where the patient has no cold and the doctor decides correctly, there is no penalty or reward for the decision.

This example shows the effect of the cost matrix/loss function during decision making. In addition to loss function, there are two more parameters that have to be considered. One of these is *h(x)* function and the other one is feature costs. We will expand the above example by adding probabilities and feature costs to the problem. For simplicity, we redefine our cost matrix in Table 2.2, where there are the same cost for misdiagnosis and no reward for correct decision.

Predicted Classes	**Correct Classes**	
	Cold	*No Cold*
Cold	0	200
No Cold	200	0

Table 2.2: Simple Cost Matrix

Also, we define a sequential process that selects the actions (action of feature selection or classification) according to costs of all actions (feature costs and misclassification costs) to use in our example. This sequential process is a common

General Health Control (GHC)	Blood Test (BT)	CT Scan
$1.00	$50.00	$80.00

Table 2.3: Test Costs

way to make decision process cost-sensitive and steps of such sequential algorithm illustrated in Figure 2.2.

[1] Repeat until all features are extracted or classify action be taken
[2] Compute cost of all actions (including extracting of each feature and classify action)
[3] Select action that has minimum cost
[4] If selected action is classify finish sequential process
[5] If that action is extract $feature_k$, extract that feature and compute a new mapping function $h(x)$ using extracted feature
[6] End of loop

Figure 2.2: The sequential cost-sensitive classification algorithm

Suppose that we try to solve cost-sensitive classification problem for diagnosing cold by using the algorithm in Figure 2.2 and simple cost matrix in Table 2.2. In this example problem, we have three patients and three different tests: general health control (GHC) (by asking questions), a blood test (BT) and a CT scan. We also have an $h(x)$ function which gives probabilities for prediction of cold or no-cold. Table 2.3 shows the test costs and the Table 2.4 shows the reliability of the prediction according to test results of patients.

One of the key points in Table 2.4 is the estimation of test results. A decision algorithm should decide on which test should be performed next without performing the test. Thus, $h(x)$ function should also estimate the result of the test by considering already performed tests. For example, after performing GHC, $h(x)$ function estimates the reliability level of BT and CT without performing these tests. For example, reliability of GHC results for patient 1 is $P(GHC) = 0.51$, after performing GHC, the estimated reliability of $P(BT|GHC)$ is equal to 0.76. Keeping in mind that, the BT results are not known in advance, so it

	Patient1	**Patient2**	**Patient3**
$P(GHC)$	0.51(Cold)	0.55(No-cold)	0.60(Cold)
$P(BT\|GHC)$	0.76(No-cold)	0.80(No-cold)	0.80(Cold)
$P(CT\|GHC)$	0.90(No-cold)	0.85(No-cold)	0.85(Cold)
$P(BT\|GHC+CT)$	0.80(No-cold)	0.85(No-cold)	0.90(Cold)
$P(CT\|GHC+BT)$	0.95(No-cold)	0.90(No-cold)	0.92(Cold)
$P(CT+GHC)$	0.90(No-cold)	0.85(No-cold)	0.95(Cold)
$P(BT+GHC)$	0.78(No-cold)	0.80(No-cold)	0.90(Cold)
$P(BT+CT+GHC)$	0.98(Cold)	0.90(No-cold)	0.98(No-cold)

Table 2.4: The values of the $h(x)$ function

should be estimated using the current GHC result.

The Figures 2.3, 2.4 and 2.5 show the steps of cost-sensitive classification algorithm given in Figure 2.2, by using cost matrixes in Tables 2.2, 2.3 and the result of $h(x)$ function in Table 2.4.

Figure 2.3 illustrates the simple decision steps for patient 1 (P1). In this figure, the concrete squares represent the risks of diagnosis using already performed tests and dashed squares represent the risks of diagnosis using already performed tests and estimation of following test. For the P1, GHC results are obtained (starting with cheapest feature). The probability value of $h(x)$ function is 0.51, so the expected risk for this patient in this step is $P(GHC)*0+(1-P(GHC))*200+cost(GHC) = 0.51*0+0.49*200+1 = 99$, according (2.1). After this step, there are three possible actions, first one is diagnosing as cold by just using GHC results. The expected risk of this option is 99. Second one is the performing additional BT test on the P1. The $h(x)$ estimates $P(BH+GHC|GHC)$ by not actually performing test. The expected risk after extraction BT in next step is calculated as 98. The third and the last one is computing the estimated risk of $P(CT+GHC|GHC)$, and that is calculated as 100. By considering expected risk of all three actions, the sequential algorithm in Figure 2.2 selects the action that has the minimum risk. Thus, the following action is actually performing BT test on the P1. In the following step, we have two actions such that diagnosing using already performed $BT+GHC$ results and making another additional test to the

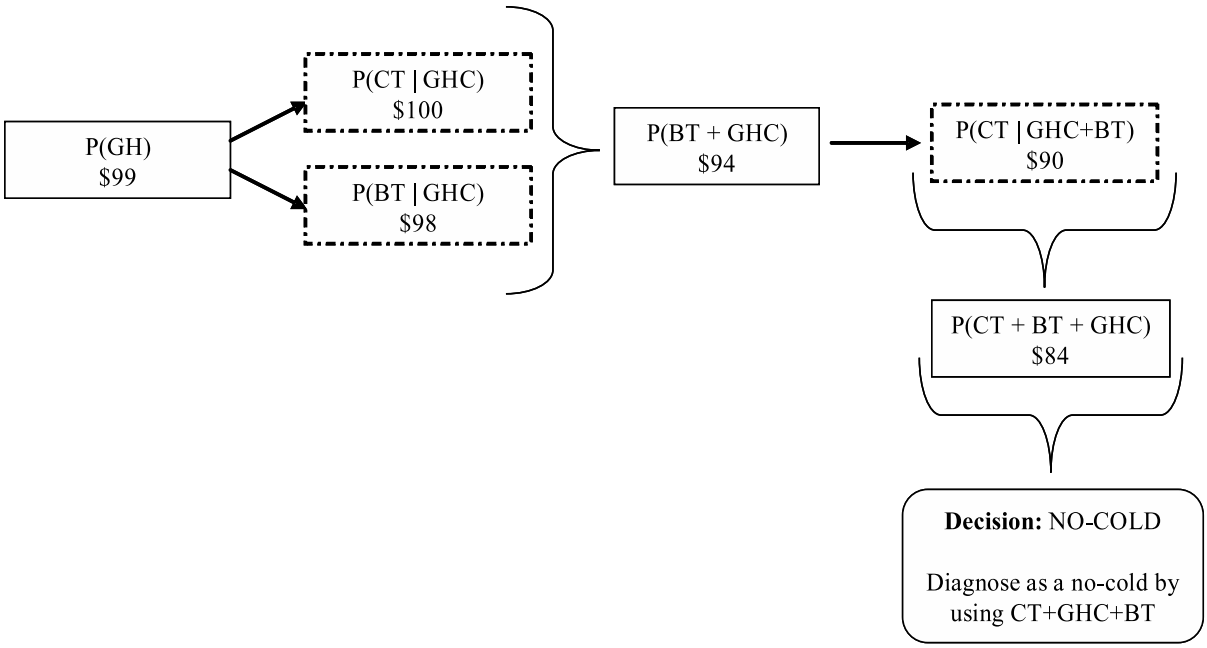


Figure 2.3: Decision steps for patient 1.

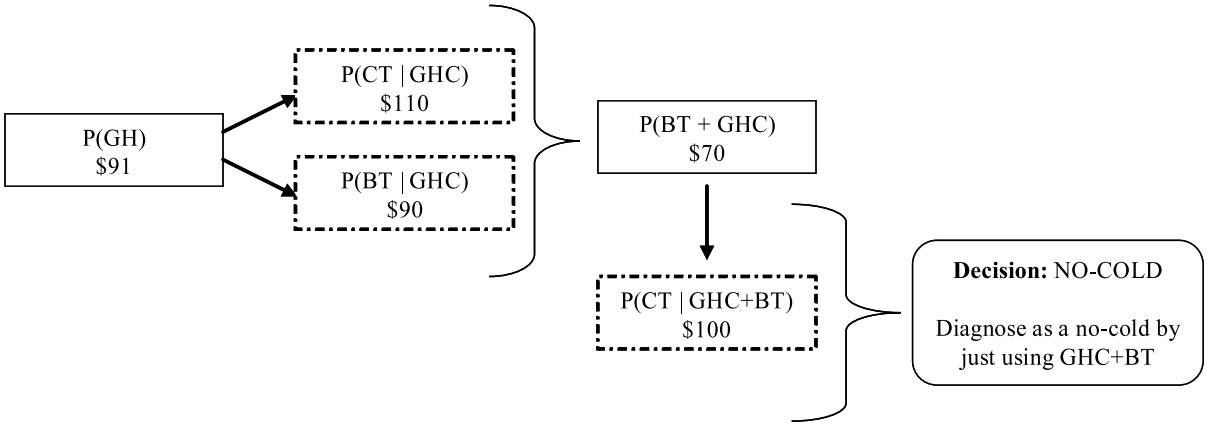


Figure 2.4: Decision steps for patient 2.

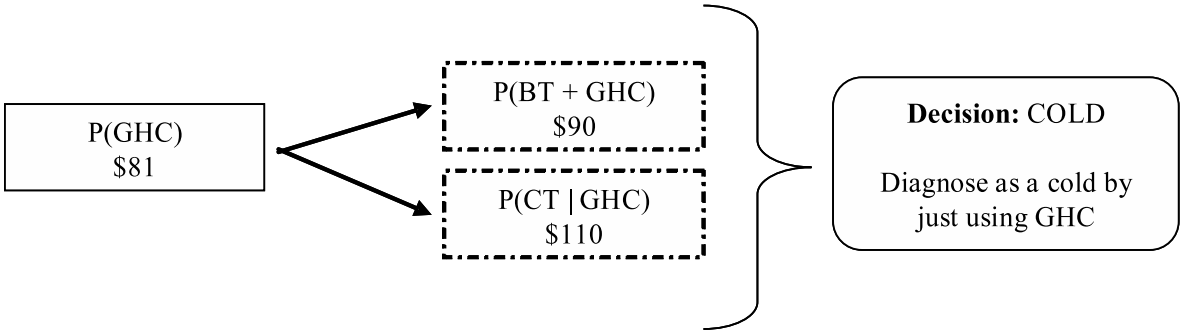


Figure 2.5: Decision steps for patient 3.

P1. After computing the expected risk according to equation 2.1, the action that has the minimum risk is making the additional CT test. After performing the CT test, the algorithm diagnoses the patient as no-cold.

One of the interesting properties of cost-sensitive learning is the highest reliable action does not have to be the best action for cost sensitive classification. In this problem, $P(CT|GHC) = 0.9$ and $P(BT|GHC) = 0.76$, the expected risk by including test cost associated for BT and CT are $R(P(CT + GHC|GHC)) = 0.76 * 0 + 0.24 * 200 + 50 = 98$ and $R(P(CT + GHC|GHC)) = 0.9 * 0 + 0.10 * 200 + 80 = 100$. Although, the most reliable action is performing CT test, BT test becomes minimum risk action after considering associated test costs.

2.1.1 Extensions of Cost-Sensitive Learning

In addition to examined cost-sensitive algorithms in 1.1.2, there are additional behaviors that should be considered in cost-sensitive learning. We show some possible examples belows.

- Conditioned Feature Cost

  Turney, in [7], introduced the conditioned test costs. According the performed action, the test costs may vary. This does not fit the assumption

that the test costs are constant prior to learning. The volatility of feature costs should be considered in cost-sensitive algorithms. One case for conditioned test costs occurs when the test have a common cost. For example, collecting blood for different tests is a common cost. If one of these test is performed, the following tests with the same common cost will not have the collecting blood cost. One cost-sensitive algorithm should consider this kind of cost volatility during both learning and execution case. Test costs also vary according to current state of the problem. A test cost may have different values depending on the patient age or patient health status. A test may be much more expensive for a patient who has critical medical condition.

- Delayed Test Results

Most of cost-sensitive algorithms in cost-sensitive learning ignore the delay in test results. For example, in medical diagnosis, a test may require a time limit. The cost-sensitive algorithms should consider time limits and should decide on which test is performed. A doctor may order a blood test to measure patient uric acid level in her/his blood, which usually takes one hour and after considering this test the doctor can also order additional tests. According to this scenario the patient must wait another one hour to obtain the other test result. This case is impractical in real life. Thus, the cost-sensitive algorithms should also consider tests that are whether immediate or delayed.

2.2 Qualitative Decision Theory

The fundamentals of decision theory was founded by von Neumann and Morgenstern's utility theory [28]. This utility theory models the actions by a probability distribution over the consequences. Using that model, preference over actions are ranked by a function called as expected utility. A decision problem in this model has three parameters as follows:

1. $\Omega = (\omega_1, ..., \omega_n)$
2. $X = (x_1, ..., x_2)$
3. $A : \Omega \rightarrow x$

Ω contains final set of the possible state of natures. X represents the consequences of actions. A represents the set of actions that each action is a mapping function from each state of nature, $\omega \in \Omega$, to possible consequences $x \in X$. The decision maker ranks the actions according to quantitative utility function $U(\alpha) \in \mathbf{R}$. Uncertainty comes from the possibility of being in any of the state in Ω. Thus, we can see the action α as a vector that maps possible states to different consequences of an action.

Then, according to probability distribution of π on Ω, the decision maker makes an orders the actions in A, based on expected utility function:

$$EU(\alpha) = \sum_{\omega \in \Omega} \pi(\omega) U(\alpha(\omega)) \tag{2.2}$$

The preference order of an action α_1 to action α_2 determined by the relation between $EU(\alpha_1)$ and $EU(\alpha_2)$. However this classic model assumes that utilities and probabilities should be in the form of numerical values. This restriction make rational decision making impossible in a case where parameters can not be described in quantitative manner. Here the need for qualitative decision theory arises.

Studies in qualitative decision theory, focus on adapting a utility function to qualitative probabilities and qualitative outcomes. These studies showed that there exists a utility function, if following axioms hold:

1. **Orderability:** Among all possible outcomes or probabilities there has to be either a preference or a indifference relation.
2. **Reflexibility:** Any preference on P is at least as preferred as itself.

3. Transitivity: If P is more preferred than Q and Q is more preferred than Y, then P is more preferred than Y

Thus, to define a utility function in qualitative problems, the probabilities and outcomes should be modified to fit these axioms. Following two sections describe studies for fitting qualitative probability and outcomes to these axioms, respectively.

2.2.1 Qualitative Probabilities

The probability distribution π on Ω is a mapping from Ω to unit interval [0,1]. This scale can be thought in two different ways. One is quantitative where the values in the unit interval have real values and the other one is qualitative where values in the unit interval just represent an ordering between different states of nature. In first case, multiplication and summation operation can be applied like in equation 2.2. However in the second case, instead of applying multiplication and summation operation the max *or* min operations are applicable. The properties of qualitative probability distribution of π is as follows:

1. $\pi(\omega_i)= 1$ if only if ω_i is normal/expected
2. $\pi(\omega_i)= 0$ if only if ω_i is impossible
3. $\forall i \in \pi(\omega_i) = \gamma_i$ where $\gamma_i \in$ [0,1]

The qualitative probability is making a complete mapping of each $\pi(\omega_i)$ where $\forall\omega_i \in \Omega$, $\pi(\omega_i) \in [0, 1]$. To see the preference relation over qualitative π, let assume that we have a subset $\Delta \subset \Omega$ and the probability measure of this subset defined as follows:

$$\beta = \max_{\omega_i \in \Delta}(\pi(\omega_i))$$

For any $\omega_i \notin \Delta$ is at least plausible/normal as the subset Δ if only if $\pi(\omega_i) = \gamma_i \geq \beta$. The plausibility/preference is determined by max operator over Ω.

2.2.2 Qualitative Consequences

People tend to express their decisions over consequences in terms of generic preferences. This tendency leads to researchers to formulate decision theory that can handle such a human-derived expression style. In literature, they showed preference relation over possible two outcomes P and Q as follows:

1. $P \succeq Q$ represents the case where P more preferred than Q
2. $P \preceq Q$ represents the case where P less preferred than Q
3. $P \sim Q$ represents the case where P and Q has equal preference

This type of preference expressions is common in human reasoning. The decision maker is not able to quantify his/her preference easily over possible outcomes. However, generally, he/she easily define a preference order for them. To see a generic preference definition for a problem, let us consider an example borrowed from [2].

Example: Consider yourself in a trip, your options include carrying an umbrella u, not carrying an umbrella $\neg u$, being dry d and being wet $\neg d$. And you prefer not carrying an umbrella to carrying an umbrella $\neg u \succeq u$, being dry to being wet $d \succeq \neg d$ and being dry to carrying an umbrella $d \succeq u$.

In this example, the decision maker defines a general preference order among all possible outcomes. The difficulty of describing relations among options quantitatively for this problem is obvious. Thus, classical approaches are not able to bring a model for rational decision making. However, by describing the preference in qualitative order scale, previous studies showed that a qualitative model can be defined for decision making [4, 24].

Although described generic preference is the common way used in qualitative decision theory, Lehmann, in [37], came with a smart method by redefining qualitative preference ordering. He expands the qualitative preference order beyond the usual order. In his work, he postulated that to define a qualitative preference order between two qualitative outcomes P and Q, P and Q should carry following properties:

1. $P \; and \; Q \notin \mathbf{R}$ (they are qualitative numbers)
2. $r \in \mathbf{R}$ (r is a standard number)
3. $P \succ Q$ iff there is a positive r such that $(P - Q)/P \geq r$

Described qualitative preference ordering used in [37] allows decision makers to use quantitative probabilities with qualitative preference orders. The clearest advantage of preference order proposed by Lehmann is that preference order is presented in forms of numbers (of course they are qualitative numbers), and not in forms of matrices as in usual order. These qualitative numbers could be used in an utility function as normal numbers, but they are not real numbers, they just represent preference order.

2.2.3 Qualitative Utilities

After accomplishing previous treatments on qualitative probabilities and consequences. The qualitative decision theory shows that there exists a utility function U such that:

1. $P \succeq Q$ if only if $U(P) \geq U(Q)$
2. $P \sim Q$ if only if $U(P) = U(Q)$

This utility function definition for qualitative probabilities and consequences/outcomes makes von Neumann and Morgenstern's utillity concept is

avaliable for decision making problems where probabilities and/or outcomes is not in forms of numerical values.

Chapter 3

Methodology

3.1 Methodology

In our approach, we define the loss function qualitatively and condition it with the current and estimated decisions as well as their consistency. Using our loss function $\lambda(\alpha_i|C_j)$, we define the conditional risk of taking action α_i for instance x is as follows:

$$R(\alpha_i|x) = \sum_{j=1}^{N} P(C_j|x)\ \lambda(\alpha_i|C_j) \tag{3.1}$$

In this equation $\{C_1, C_2, ..., C_N\}$ is the set of N possible states of nature and $\lambda(\alpha_i|C_j)$ is the loss incurred for taking action α_i when the actual state of nature is C_j. In our approach, we consider C_j as the class that an instance can belong to and α_i as one of the following actions:

(a) `extract`$_k$: extract feature F_k,

(b) `classify`: stop the extraction and classify the instance using the current information, and

	$\texttt{extract}_k$	classify	reject
Case 1: $C_{\texttt{actual}} = C_{\texttt{curr}} = C_{\texttt{est}_k}$	$\texttt{cost}_k$	$-$REWARD	PENALTY
Case 2: $C_{\texttt{actual}} \neq C_{\texttt{curr}} \neq C_{\texttt{est}_k}$	$\texttt{cost}_k + $PENALTY	PENALTY	$-$REWARD
Case 3: $C_{\texttt{curr}} = C_{\texttt{est}_k} \neq C_{\texttt{actual}}$	$\texttt{cost}_k + $PENALTY	PENALTY	$-$REWARD
Case 4: $C_{\texttt{actual}} = C_{\texttt{curr}} \neq C_{\texttt{est}_k}$	$\texttt{cost}_k + $PENALTY	$-$REWARD	PENALTY
Case 5: $C_{\texttt{actual}} = C_{\texttt{est}_k} \neq C_{\texttt{curr}}$	$\texttt{cost}_k - $REWARD	PENALTY	PENALTY

Table 3.1: Definition of the conditioned loss function for feature extraction, classification, and reject actions.

(c) `reject`: stop the extraction and reject the classification of the instance.

In this section, we first define our loss function by conditioning it with the current and estimated decisions together with their *consistency* and derive the equations for conditional risks using this loss function definition (in Section 3.2). Then, we incorporate the *qualitativeness* into this conditioned-loss function definition and explain how to *qualitatively* compare the conditional risks for each pair of actions (in Section 3.3). Finally, we provide the details of our test-cost sensitive algorithm that uses this qualitative loss function definition (in Section 3.4).

3.2 Consistency-based loss functions

We define our loss function for the $\texttt{extract}_k$, `classify`, and `reject` actions in Table 3.1. In this table, $C_{\texttt{actual}}$ is the actual class that an instance belongs to; $C_{\texttt{curr}}$ is the class estimated by the current classifier (which uses only the features that have been extracted thus far); $C_{\texttt{est}_k}$ is the class estimated by the classifier that uses the extracted features plus feature F_k (the one to be extracted next). The actual class $C_{\texttt{actual}}$ and the $C_{\texttt{est}_k}$ should be estimated using the current information since it is not possible to know these values in advance; we explain the details of this estimation in Section 3.4.

In our loss function definition, for the $\texttt{extract}_k$ action (Table 3.1), the extraction cost ($\texttt{cost}_k$) that should be paid for acquiring feature F_k is always included. Additionally, the extraction of F_k is penalized with a qualitative amount of PENALTY if this extraction does not yield correct classification (i.e.,

$C_{\texttt{est}_k} \neq C_{\texttt{actual}}$). On the contrary, the extraction is rewarded with a qualitative amount of `REWARD` (by adding $-$`REWARD` to the loss function), if it yields correct classification by changing our current decision (i.e., $C_{\texttt{est}_k} = C_{\texttt{actual}}$ but $C_{\texttt{curr}} \neq C_{\texttt{actual}}$). If it just confirms the current decision which has already been correct (i.e., $C_{\texttt{est}_k} = C_{\texttt{actual}}$ and $C_{\texttt{curr}} = C_{\texttt{actual}}$), the extraction is not rewarded since it brings an additional cost without providing any new information. Thus, we force our algorithm not to extract additional features when they are expected to confirm the correct current decision. This leads to less costly but equally accurate results. Note that in this loss function definition, as well as in those defined for the `classify` and `reject` actions, `PENALTY` and `REWARD` are defined as *positive qualitative values*. The next section provides the details of the use of these qualitative values in making our decisions.

For the `classify` action (Table 3.1), the classification is rewarded with `REWARD` if the current decision is correct (i.e., $C_{\texttt{curr}} = C_{\texttt{actual}}$) and penalized with `PENALTY` otherwise (i.e., $C_{\texttt{curr}} \neq C_{\texttt{actual}}$). Thus, in the case of current decision being correct, we force our algorithm to classify the instance without extracting any additional features.

For the `reject` action (Table 3.1), the rejection of both classification and feature extraction is rewarded with `REWARD`, if both the current and estimated decisions yield misclassification (i.e., $C_{\texttt{curr}} \neq C_{\texttt{actual}}$ and $C_{\texttt{est}_k} \neq C_{\texttt{actual}}$ for every $C_{\texttt{est}_k}$ in $\mathcal{C}$). The rejection is penalized with `PENALTY` if either the current decision or any of the estimated decisions after feature extraction yields the correct classification (i.e., $C_{\texttt{curr}} = C_{\texttt{actual}}$ or $C_{\texttt{est}_k} = C_{\texttt{actual}}$ for at least one $C_{\texttt{est}_k} \in \mathcal{C}$). Thus, we force our algorithm to stop and reject the classification only when the correct classification is not possible, since there could be a cost associated with the `reject` action (e.g., the dissatisfaction of a patient about his/her doctor). Therefore, our algorithm takes this action only it believes that no correct classification is possible.

Using these definitions, we derive the conditional risks for the $\texttt{extract}_k$ action. For a particular instance x, we express the conditional risk of each action using the definition of loss function above and take the action with the minimum

risk. With $\mathcal{C} = \{C_{\texttt{curr}}, C_{\texttt{est}_1}, C_{\texttt{est}_2}, ..., C_{\texttt{est}_M}\}$ being the set of the current class and the classes estimated after extracting each feature, the conditional risk of extracting feature F_k ($\texttt{extract}_k$ action) is defined as follows. Here, $P(C_{\texttt{curr}} = j|x)$ is the probability of the current class being equal to j and $P(C_{\texttt{est}_k} = j|x)$ is the probability of class estimated after extracting feature F_k being equal to j.

$$R\ (\ \texttt{extract}_k|x, \mathcal{C}) = \sum_{j=1}^{N} P(C_{\texttt{actual}} = j|x) \times \tag{3.2}$$
$$\left[\begin{array}{l} P(C_{\texttt{curr}} = j|x)\ P(C_{\texttt{est}_k} = j|x)\ \texttt{cost}_k + \\ P(C_{\texttt{curr}} \neq j|x)\ P(C_{\texttt{est}_k} \neq j|x)\ P(C_{\texttt{curr}} = C_{\texttt{est}_k}|x)\ [\texttt{cost}_k + \texttt{PENALTY}] + \\ P(C_{\texttt{curr}} \neq j|x)\ P(C_{\texttt{est}_k} \neq j|x)\ P(C_{\texttt{curr}} \neq C_{\texttt{est}_k}|x)\ [\texttt{cost}_k + \texttt{PENALTY}] + \\ P(C_{\texttt{curr}} = j|x)\ P(C_{\texttt{est}_k} \neq j|x)\ [\texttt{cost}_k + \texttt{PENALTY}] + \\ P(C_{\texttt{curr}} \neq j|x)\ P(C_{\texttt{est}_k} = j|x)\ [\texttt{cost}_k - \texttt{REWARD}] \end{array}\right]$$

$$R\ (\ \texttt{extract}_k|x, \mathcal{C}) = \sum_{j=1}^{N} P(C_{\texttt{actual}} = j|x) \times \tag{3.3}$$
$$\left[\begin{array}{l} P(C_{\texttt{curr}} = j|x)\ P(C_{\texttt{est}_k} = j|x)\ \texttt{cost}_k + \\ P(C_{\texttt{curr}} \neq j|x)\ P(C_{\texttt{est}_k} \neq j|x)\ [\texttt{cost}_k + \texttt{PENALTY}] + \\ P(C_{\texttt{curr}} = j|x)\ P(C_{\texttt{est}_k} \neq j|x)\ [\texttt{cost}_k + \texttt{PENALTY}] + \\ P(C_{\texttt{curr}} \neq j|x)\ P(C_{\texttt{est}_k} = j|x)\ [\texttt{cost}_k - \texttt{REWARD}] \end{array}\right]$$

$$R\ (\ \texttt{extract}_k|x, \mathcal{C}) = \sum_{j=1}^{N} P(C_{\texttt{actual}} = j|x) \times \tag{3.4}$$
$$\left[\begin{array}{lll} P(C_{\texttt{curr}} = j|x) & P(C_{\texttt{est}_k} = j|x) & \texttt{cost}_k + \\ [1 - P(C_{\texttt{curr}} = j|x)] & [1 - P(C_{\texttt{est}_k} = j|x)] & [\texttt{cost}_k + \texttt{PENALTY}] + \\ P(C_{\texttt{curr}} = j|x) & [1 - P(C_{\texttt{est}_k} = j|x)] & [\texttt{cost}_k + \texttt{PENALTY}] + \\ [1 - P(C_{\texttt{curr}} = j|x)] & P(C_{\texttt{est}_k} = j|x) & [\texttt{cost}_k - \texttt{REWARD}] \end{array}\right]$$

$$R\ (\ \texttt{extract}_k|x, \mathcal{C}) = \sum_{j=1}^{N} P(C_{\texttt{actual}} = j|x) \times \tag{3.5}$$
$$\left[\begin{array}{l} \texttt{cost}_k + \\ [1 - P(C_{\texttt{est}_k} = j|x)]\ \texttt{PENALTY} + \\ P(C_{\texttt{est}_k} = j|x)\ [1 - P(C_{\texttt{curr}} = j)|x]\ [-\texttt{REWARD}] \end{array}\right]$$

Equation 3.5 implies that the extraction of feature F_k requires paying for its cost ($\texttt{cost}_k$). Furthermore, it implies that the $\texttt{extract}_k$ action is penalized with PENALTY if the class estimated after feature extraction is incorrect and is rewarded with REWARD if this estimated class is correct but it is different than the currently estimated class.

For a particular instance x, the conditional risk of the classify action is given in Equation 3.8.

$$R\ (\ \texttt{classify}|x,\mathcal{C}) = \sum_{j=1}^{N} P(C_{\texttt{actual}} = j|x) \times \tag{3.6}$$
$$\left[\begin{array}{l} P(C_{\texttt{curr}} = j|x)\ P(C_{\texttt{est}_k} = j|x)\ (-\texttt{REWARD}) + \\ P(C_{\texttt{curr}} \neq j|x)\ P(C_{\texttt{est}_k} \neq j|x)\ P(C_{\texttt{curr}} = C_{\texttt{est}_k}|x)\ \texttt{PENALTY} + \\ P(C_{\texttt{curr}} \neq j|x)\ P(C_{\texttt{est}_k} \neq j|x)\ P(C_{\texttt{curr}} \neq C_{\texttt{est}_k}|x)\ \texttt{PENALTY} + \\ P(C_{\texttt{curr}} = j|x)\ P(C_{\texttt{est}_k} \neq j|x)\ (-\texttt{REWARD}) + \\ P(C_{\texttt{curr}} \neq j|x)\ P(C_{\texttt{est}_k} = j|x)\ \texttt{PENALTY} \end{array}\right]$$

$$R\ (\ \texttt{classify}|x,\mathcal{C}) = \sum_{j=1}^{N} P(C_{\texttt{actual}} = j|x) \times \tag{3.7}$$
$$\left[\begin{array}{l} P(C_{\texttt{curr}} = j|x)\ (-\texttt{REWARD} +) \\ P(C_{\texttt{curr}} \neq j|x)\ P(C_{\texttt{est}_k} \neq j|x)\ \texttt{PENALTY} + \\ P(C_{\texttt{curr}} \neq j|x)\ P(C_{\texttt{est}_k} = j|x)\ \texttt{PENALTY} \end{array}\right]$$

$$R\ (\ \texttt{classify}|x,\mathcal{C}) = \sum_{j=1}^{N} P(C_{\texttt{actual}} = j|x) \times \tag{3.8}$$
$$\left[\ P(C_{\texttt{curr}} = j|x)\ [-\texttt{REWARD}] + [1 - P(C_{\texttt{curr}} = j|x)]\ \texttt{PENALTY}\ \right]$$

Equation 3.8 implies that classifying the instance with the current classifier (classify action) is rewarded with REWARD if this is a correct classification and is penalized with PENALTY otherwise.

Similarly, for a particular instance x, we derive the conditional risk of the reject action in Equation 3.9.

$$R \quad (\quad \texttt{reject}|x, \mathcal{C}) = \sum_{j=1}^{N} P(C_{\texttt{actual}} = j|x) \times \tag{3.9}$$
$$\left[\begin{array}{l} \left[[1 - P(C_{\texttt{curr}} = j|x)] \prod_{m=1}^{M} [1 - P(C_{\texttt{est}_m} = j|x)] \right] \; [-\texttt{REWARD}] + \\ \left[1 - [1 - P(C_{\texttt{curr}} = j|x)] \prod_{m=1}^{M} [1 - P(C_{\texttt{est}_m} = j|x)] \right] \; \texttt{PENALTY} + \end{array} \right]$$

This equation implies that rejecting the classification is only rewarded with REWARD if neither the estimated classes nor the current class is correct; otherwise, it is penalized with PENALTY.

With this loss function formalization, we introduce the consistency concept to test-cost sensitive learning. Here, we define the REWARD and PENALTY values qualitatively, which causes the conditional risks being also defined qualitatively. Thus, there is no need to know the exact values of these parameters in the computation of the conditional risks associated with each of our actions. In the following section, we explain how these qualitative values and consistency are used in decision making.

3.3 Qualitative decision making for test-cost sensitive classification

Qualitative-reasoning concerns with the development of methods that allow to design systems without precise quantitative information. It primarily uses the ordinal relations between quantities, especially at particular locations ("landmark values"). The numerical value of a landmark may or may not be known, but the ordinal relations with this landmark, reflecting the generic preferences, are known [40].

In our test-cost sensitive classification, our landmark values are the extraction cost for each feature F_k ($\texttt{cost}_k$) and the values of PENALTY and REWARD. For

qualitative decision making, in order to take an action with the minimum conditional risk, we should qualitatively compare the conditional risks in which these landmark values are used. To this end, we should specify the ordering among these landmarks. In this thesis, we specify such an ordering focusing on a medical diagnosis problem. Please note that depending on the application, one can change these assumptions and specify a new ordering for the comparison of the conditional risks. In this ordering we make the following assumptions:

1. We assume that the cost of acquiring a feature (the price of a medical test) is expressed quantitatively and is exactly known. Thus, costs for different features are quantitatively compared among themselves.

2. `PENALTY` and `REWARD` are defined as positive numbers, but their precise values are not known. Here, `PENALTY` is considered as the amount that we pay in the case of misdiagnosis and `REWARD` is considered as the amount that we earn in the case of correct diagnosis. In our system, we assume that the amount that we pay for misdiagnosis is always greater than the amount that we earn for correct diagnosis (`PENALTY` $>$ `REWARD`). Therefore, we force our system to have a higher tendency in preventing misdiagnosis compared to resulting in correct diagnosis.

3. The extraction costs (the prices of medical tests) are always smaller than any partial amounts of `PENALTY` and `REWARD`. Thus, here we assume that all tests are affordable to prevent misdiagnosis and lead to the correct one.

In our decision making, we compare the conditional risks for each pair of actions and select the action for which the conditional risk is qualitatively minimum. In the following subsections, by using the aforementioned assumptions, we explain how to make qualitative comparisons for the $\texttt{extract}_k$-vs-$\texttt{extract}_m$, $\texttt{extract}_k$-vs-`classify`, $\texttt{extract}_k$-vs-`reject`, and `classify`-vs-`reject` actions. In these subsections, we also explain how to select what action to take as a result of these comparisons.

3.3.1 $\texttt{extract}_k$-vs-$\texttt{extract}_m$

We compute the net conditional risk for comparing the conditional risks of the $\texttt{extract}_k$ and $\texttt{extract}_m$ actions, which are defined for extracting features F_l and F_n, respectively.

$$NetRisk = R(\texttt{extract}_k|x, C_{\texttt{curr}},) - R(\texttt{extract}_m|x, C_{\texttt{curr}},) \tag{3.10}$$

Using Equation 3.5 in Equation 3.10, the net conditional risk is expressed as

$$\begin{aligned} NetRisk = & \qquad (3.11)\\ & (\texttt{cost}_k - \texttt{cost}_m) + \\ & \sum_{j=1}^{N} P(C_{\texttt{actual}} = j|x) \left[P(C_{\texttt{est}_m} = j|x) - P(C_{\texttt{est}_k} = j|x)\right] \texttt{ PENALTY} + \\ & \sum_{j=1}^{N} P(C_{\texttt{actual}} = j|x) \left[P(C_{\texttt{est}_m} = j|x) - P(C_{\texttt{est}_k} = j|x)\right] \times \\ & \qquad \left[1 - P(C_{\texttt{curr}} = j|x)\right] \texttt{ REWARD} \qquad (3.12) \end{aligned}$$

With

$NetCost = (\texttt{cost}_k - \texttt{cost}_m)$,

$$X = \sum_{j=1}^{N} P(C_{\texttt{actual}} = j|x) \left[P(C_{\texttt{est}_m} = j|x) - P(C_{\texttt{est}_k} = j|x)\right],$$

and

$$Y = \sum_{j=1}^{N} P(C_{\texttt{actual}} = j|x) \left[P(C_{\texttt{est}_m} = j|x) - P(C_{\texttt{est}_k} = j|x)\right] \left[1 - P(C_{\texttt{curr}} = j|x)\right],$$

we rewrite this equation as

$$NetRisk = NetCost + X\texttt{PENALTY} + Y\texttt{REWARD} \tag{3.13}$$

Negative values of $NetRisk$ imply that the conditional risk of $\texttt{extract}_k$ is

smaller than that of $\texttt{extract}_m$. Therefore, we take the $\texttt{extract}_k$ action for negative $NetRisk$s and the $\texttt{extract}_m$ action for nonnegative ones.

In Equation 3.13, we simply neglect $NetCost$ since the feature extraction cost is always less than any partial amounts of PENALTY and REWARD (the third assumption). As PENALTY and REWARD are defined as positive values, the sign of $NetRisk$ depends on the signs of X and Y which are computed using posterior probabilities[1]. Therefore, we can have four different cases:

- **Case 1** ($X \geq 0$ and $Y \geq 0$): It implies that the values of $X\texttt{PENALTY}$ and $Y\texttt{REWARD}$ are greater than or equal to zero, and consequently, the value of $NetRisk$ is nonnegative. Therefore, we take the $\texttt{extract}_m$ action. If both $X = 0$ and $Y = 0$, we take the $\texttt{extract}_k$ action for which the cost ($\texttt{cost}_k$) is minimum; note that here we know the ordering among the feature extraction costs (the first assumption).

- **Case 2** ($X < 0$ and $Y < 0$): It implies that the values of $X\texttt{PENALTY}$ and $Y\texttt{REWARD}$ are less than zero, and consequently, the value of $NetRisk$ is negative. Therefore, we take the $\texttt{extract}_k$ action.

- **Case 3** ($X \geq 0$ and $Y < 0$): The sign of $NetRisk$ depends on the magnitudes of X and Y. If $|X| \geq |Y|$ then $|X\texttt{PENALTY}| > |Y\texttt{REWARD}|$, since PENALTY is greater than REWARD (the second assumption). Thus, the value of $NetRisk$ is nonnegative and the $\texttt{extract}_m$ action is taken.

  If $|X| < |Y|$, we then qualitatively compare $|X\texttt{PENALTY}|$ and $|Y\texttt{REWARD}|$. For that, we use the following definition, which is given in [37].

  Definition 1 *Let A and B be positive. A is qualitatively larger than B if and only if there is a strictly positive real number r such that $(A-B)/A \geq r$.*

  With r being a strictly positive real number, $|Y\texttt{REWARD}|$ is qualitatively

[1] Once again, note that $P(C_{\texttt{actual}} = j|x)$ and $P(C_{\texttt{est}_k} = j|x)$ are not known in advance and they should be estimated using the current information beforehand. We provide the details of this estimation in Section 3.4.

larger than $|X\texttt{PENALTY}|$ if and only if

$$|Y\ \texttt{REWARD}| \succ |X\ \texttt{PENALTY}| \ \Leftrightarrow\ \frac{|Y\ \texttt{REWARD}| - |X\ \texttt{PENALTY}|}{|Y\ \texttt{REWARD}|} \geq r$$

$$|Y\ \texttt{REWARD}| \succ |X\ \texttt{PENALTY}| \ \Leftrightarrow\ 1 - \frac{|X\ \texttt{PENALTY}|}{|Y\ \texttt{REWARD}|} \geq r$$

$$|Y\ \texttt{REWARD}| \succ |X\ \texttt{PENALTY}| \ \Leftrightarrow\ \frac{|X|\ \texttt{PENALTY}}{|Y|\ \texttt{REWARD}} \leq 1 - r$$

Selecting r in between 0 and 1, $1 - r$ gives another strictly positive real number p, which is also in between 0 and 1.

$$|Y\ \texttt{REWARD}| \succ |X\ \texttt{PENALTY}| \Leftrightarrow \frac{|X|}{|Y|} \leq p\ \frac{\texttt{REWARD}}{\texttt{PENALTY}}$$

We define another strictly positive real number $\texttt{SMALL} = p \times (\texttt{REWARD}/\texttt{PENALTY})$. [2]. This number is also in between 0 and 1 since REWARD is less than PENALTY which implies $\texttt{REWARD}/\texttt{PENALTY} < 1$.

$$|Y\texttt{REWARD}| \succ |X\texttt{PENALTY}| \ \Leftrightarrow\ \frac{|X|}{|Y|} \leq \texttt{SMALL} \tag{3.14}$$

Therefore, if $|X| < |Y|$, we check whether or not Equation 3.14 holds. If $|X/Y| \leq \texttt{SMALL}$ then $|Y\texttt{REWARD}|$ is qualitatively larger than $|X\texttt{PENALTY}|$, and hence, the value of $NetRisk$ is negative and the $\texttt{extract}_k$ action is taken. Otherwise (if $|X/Y| > \texttt{SMALL}$), the value of $NetRisk$ is nonnegative and the $\texttt{extract}_m$ action is taken. Obviously, the value of the SMALL affects our decision. Here, its derivation could be considered as determining a parameter. However, in this work, we propose to determine its value automatically from the training data rather than having the user select this value. Thus, its derivation does not require the user to express his/her belief in terms of quantitative numbers. In Section 3.4, we explain how to automatically determine its value in detail.

- **Case 4** ($X < 0$ and $Y \geq 0$): Similar to Case 3, the sign of $NetRisk$ depends on the magnitudes of X and Y. Similarly, if $|X| \geq |Y|$ then $|X\texttt{PENALTY}| > |Y\texttt{REWARD}|$, since PENALTY is greater than REWARD (the second

[2]Considering our second assumption ($\texttt{PENALTY} \succ \texttt{REWARD}$), we assume that only a small portion of PENALTY could be smaller than REWARD, so we call this real number as SMALL

assumption). Thus, the value of $NetRisk$ is negative and the $\texttt{extract}_k$ action is taken.

If $|X| < |Y|$, we qualitatively compare $|X\texttt{PENALTY}|$ and $|Y\texttt{REWARD}|$ using Equation 3.14 (and Definition 1). Likewise, if $|X/Y| \leq \texttt{SMALL}$ then $|Y\texttt{REWARD}|$ is qualitatively larger than $|X\texttt{PENALTY}|$, and hence, the value of $NetRisk$ is nonnegative and the $\texttt{extract}_m$ action is taken. Otherwise (if $|X/Y| > \texttt{SMALL}$), the value of $NetRisk$ is negative and the $\texttt{extract}_k$ action is taken.

In Figure 3.1, we provide the summary of these four different cases and the rules to determine what action to take.

Case 1: $X \geq 0$, $Y \geq 0$		$\texttt{extract}_m$
Case 2: $X < 0$, $Y < 0$		$\texttt{extract}_k$
Case 3: $X \geq 0$, $Y < 0$	if $X \geq \lvert Y \rvert$	$\texttt{extract}_m$
	else if $\frac{X}{\lvert Y \rvert} \leq \texttt{SMALL}$	$\texttt{extract}_k$
	else	$\texttt{extract}_m$
Case 4: $X < 0$, $Y \geq 0$	if $\lvert X \rvert \geq Y$	$\texttt{extract}_k$
	else if $\frac{\lvert X \rvert}{Y} \leq \texttt{SMALL}$	$\texttt{extract}_m$
	else	$\texttt{extract}_k$

Figure 3.1: For $\texttt{extract}_k$-vs-$\texttt{extract}_m$ comparison, the cases and the rules to determine what action to take.

3.3.2 $\texttt{extract}_k$-vs-classify

Similar to the $\texttt{extract}_k$-vs-$\texttt{extract}_m$ comparison, we compute the net conditional risk for the comparison of the conditional risks of the $\texttt{extract}_k$ and classify actions.

$$NetRisk = R(\texttt{extract}_k|x, C_{\texttt{curr}},) - R(\texttt{classify}|x, C_{\texttt{curr}},) \tag{3.15}$$

Using Equations 3.5 and 3.8 in Equation 3.15, the net conditional risk is expressed as

$$
\begin{aligned}
NetRisk = & \qquad (3.16)\\
& \texttt{cost}_k + \\
& \sum_{j=1}^{N} P(C_{\texttt{actual}} = j|x)\,[P(C_{\texttt{curr}} = j|x) - P(C_{\texttt{est}_k} = j|x)]\ \texttt{PENALTY} + \\
& \sum_{j=1}^{N} P(C_{\texttt{actual}} = j|x)\ \times \\
& \qquad [P(C_{\texttt{curr}} = j|x) - P(C_{\texttt{est}_k} = j|x)\ \ [1 - P(C_{\texttt{curr}} = j|x)]]\ \texttt{REWARD}
\end{aligned}
$$

With

$$X = \sum_{j=1}^{N} P(C_{\texttt{actual}} = j|x)\,[P(C_{\texttt{curr}} = j|x) - P(C_{\texttt{est}_k} = j|x)],$$

and

$$Y = \sum_{j=1}^{N} P(C_{\texttt{actual}} = j|x)\,[P(C_{\texttt{curr}} = j|x) - P(C_{\texttt{est}_k} = j|x)\ \ [1 - P(C_{\texttt{curr}} = j|x)]],$$

we rewrite this equation as

$$NetRisk = \texttt{cost}_k + X\texttt{PENALTY} + Y\texttt{REWARD} \qquad (3.17)$$

Here we take the $\texttt{extract}_k$ action if $NetRisk$ is negative and the `classify` action otherwise. Similar to the $\texttt{extract}_k$-vs-$\texttt{extract}_m$ comparison, we neglect the $\texttt{cost}_k$ term in Equation 3.17 and have four different cases depending on the signs of X and Y. For each of these cases, we derive the comparison rules in a similar way that we do in the case of the $\texttt{extract}_k$-vs-$\texttt{extract}_m$ comparison. These four cases and the associated rules are given in Figure 3.2.

Case 1: $X \geq 0$, $Y \geq 0$		`classify`
Case 2: $X < 0$, $Y < 0$		`extract`$_k$
Case 3: $X \geq 0$, $Y < 0$	if $X \geq \lvert Y \rvert$	`classify`
	else if $\frac{X}{\lvert Y \rvert} \leq$ `SMALL`	`extract`$_k$
	else	`classify`
Case 4: $X < 0$, $Y \geq 0$	if $\lvert X \rvert \geq Y$	`extract`$_k$
	else if $\frac{\lvert X \rvert}{Y} \leq$ `SMALL`	`classify`
	else	`extract`$_k$

Figure 3.2: For `extract`$_k$-vs-`classify` comparison, the cases and the rules to determine what action to take.

3.3.3 `extract`$_k$-vs-`reject`

Likewise, we compute the net conditional risk for comparing the conditional risks of the `extract`$_k$ and `reject` actions.

$$NetRisk = R(\texttt{extract}_k|x, C_{\texttt{curr}},) - R(\texttt{reject}|x, C_{\texttt{curr}},) \tag{3.18}$$

Using Equations 3.5 and 3.9 in Equation 3.18, the net conditional risk is expressed as

$$\begin{aligned} NetRisk = & \qquad (3.19)\\ & \texttt{cost}_k + \\ & \sum_{j=1}^{N} P(C_{\texttt{actual}} = j|x) \times \\ & \quad \left[\begin{array}{l} [1 - P(C_{\texttt{curr}} = j|x)] \prod_{C_{\texttt{est}_m} \in \mathcal{C}} [1 - P(C_{\texttt{est}_m} = j|x)] - \\ P(C_{\texttt{est}_k} = j|x) \end{array} \right] \texttt{PENALTY} + \\ & \sum_{j=1}^{N} P(C_{\texttt{actual}} = j|x) \times \\ & \quad \left[\begin{array}{l} [1 - P(C_{\texttt{curr}} = j|x)] \prod_{C_{\texttt{est}_m} \in \mathcal{C}} [1 - P(C_{\texttt{est}_m} = j|x)] - \\ P(C_{\texttt{est}_k} = j|x)[1 - P(C_{\texttt{curr}} = j|x)] \end{array} \right] \texttt{REWARD} \end{aligned}$$

With

$$X = \sum_{j=1}^{N} P(C_{\texttt{actual}} = j|x) \left[\begin{array}{l} [1 - P(C_{\texttt{curr}} = j|x)] \prod_{C_{\texttt{est}_m} \in \mathcal{C}} [1 - P(C_{\texttt{est}_m} = j|x)] - \\ P(C_{\texttt{est}_k} = j|x) \end{array} \right],$$

and

$$Y = \sum_{j=1}^{N} P(C_{\texttt{actual}} = j|x) \left[\begin{array}{l} [1 - P(C_{\texttt{curr}} = j|x)] \prod_{C_{\texttt{est}_m} \in \mathcal{C}} [1 - P(C_{\texttt{est}_m} = j|x)] - \\ P(C_{\texttt{est}_k} = j|x)[1 - P(C_{\texttt{curr}} = j|x)] \end{array} \right],$$

we rewrite this equation as

$$NetRisk = \texttt{cost}_k + X\texttt{PENALTY} + Y\texttt{REWARD} \tag{3.20}$$

Here we take the $\texttt{extract}_k$ action if $NetRisk$ is negative and the `reject` action otherwise. Similarly, we neglect the $\texttt{cost}_k$ term in Equation 3.20 and have four different cases depending on the signs of X and Y. For each of these cases, we derive the comparison rules in a similar way that we do in the case of the $\texttt{extract}_k$-vs-$\texttt{extract}_m$ comparison. These four cases and the associated rules are given in Figure 3.3.

Case 1: $X \geq 0$, $Y \geq 0$		`reject`
Case 2: $X < 0$, $Y < 0$		$\texttt{extract}_k$
Case 3: $X \geq 0$, $Y < 0$	if $X \geq \lvert Y \rvert$	`reject`
	else if $\frac{X}{\lvert Y \rvert} \leq \texttt{SMALL}$	$\texttt{extract}_k$
	else	`reject`
Case 4: $X < 0$, $Y \geq 0$	if $\lvert X \rvert \geq Y$	$\texttt{extract}_k$
	else if $\frac{\lvert X \rvert}{Y} \leq \texttt{SMALL}$	`reject`
	else	$\texttt{extract}_k$

Figure 3.3: For $\texttt{extract}_k$-vs-`reject` comparison, the cases and the rules to determine what action to take.

3.3.4 classify-vs-reject

For the comparison of the conditional risks of the `classify` and `reject` actions, we similarly compute the net conditional risk as follows:

$$NetRisk = R(\texttt{reject}|x, C_{\texttt{curr}},) - R(\texttt{classify}|x, C_{\texttt{curr}}, \mathcal{C}) \tag{3.21}$$

Using Equations 3.8 and 3.9 in Equation 3.21, the net conditional risk is expressed as

$$\begin{aligned} NetRisk = & \\ & \sum_{j=1}^{N} P(C_{\texttt{actual}} = j|x) \times \\ & \quad \left[\begin{array}{l} P(C_{\texttt{curr}} = j|x) - \\ [1 - P(C_{\texttt{curr}} = j|x)] \prod_{C_{\texttt{est}_m} \in \mathcal{C}} [1 - P(C_{\texttt{est}_m} = j|x)] \end{array} \right] \texttt{PENALTY} + \\ & \sum_{j=1}^{N} P(C_{\texttt{actual}} = j|x) \times \\ & \quad \left[\begin{array}{l} P(C_{\texttt{curr}} = j|x) - \\ [1 - P(C_{\texttt{curr}} = j|x)] \prod_{C_{\texttt{est}_m} \in \mathcal{C}} [1 - P(C_{\texttt{est}_m} = j|x)] \end{array} \right] \texttt{REWARD} \end{aligned} \tag{3.22}$$

With

$$X = \sum_{j=1}^{N} P(C_{\texttt{actual}} = j|x) \left[\begin{array}{l} P(C_{\texttt{curr}} = j|x) - \\ [1 - P(C_{\texttt{curr}} = j|x)] \prod_{C_{\texttt{est}_m} \in \mathcal{C}} [1 - P(C_{\texttt{est}_m} = j|x)] \end{array} \right],$$

we rewrite this equation as

$$NetRisk = X\texttt{PENALTY} + X\texttt{REWARD} \tag{3.23}$$

We take the `reject` action if $NetRisk$ is negative and the `classify` action otherwise. Here, we have the same multiplier for the `PENALTY` and `REWARD` values. As both `PENALTY` and `REWARD` are positive, we can have two different cases depending on the sign of the multiplier X. If $X \geq 0$, $NetRisk$ is nonnegative, and

consequently, the `classify` action is taken. Otherwise (if $X < 0$), $NetRisk$ is negative, and hence, the `reject` action is taken. These cases and the associated rules are given in Figure 3.4.

Case 1: $X \geq 0$	`classify`
Case 2: $X < 0$	`reject`

Figure 3.4: For `classify`-vs-`reject` comparison, the cases and the rules to determine what action to take.

3.4 Qualitative test-cost sensitive classification algorithm

For a given instance x, our algorithm dynamically selects a subset of features for its classification. At a given time, it qualitatively compares the conditional risks of possible actions, and subsequently, takes the action with the minimum conditional risk using the rules given in Figures 3.1–3.4. For that, first, the $\texttt{extract}_k$ action with the minimum conditional risk is selected by comparing the conditional risks of `extract` actions for every non-extracted feature F_i (using Figure 3.1). Then, the conditional risks of the `classify` and `reject` actions are compared (using Figure 3.4) and the one with smaller risk is selected. Finally, if the `classify` action is selected, the conditional risk of the selected $\texttt{extract}_k$ action is compared with that of the `classify` action (using Figure 3.2). Otherwise, the conditional risk of the selected $\texttt{extract}_k$ action is compared with that of the `reject` action (using Figure 3.3). Our algorithm sequentially conducts these comparisons as well as the selection of the actions until either the `classify` or the `reject` action is taken. The schematic representation of this algorithm is given in Figure 3.5.

To qualitatively compare the conditional risks of the actions using Figures 3.1–3.4, the values of X, Y, and `SMALL` should be computed. For computing X and Y, we use posterior probabilities as given in Equations 3.13, 3.17, 3.20, and 3.23. In

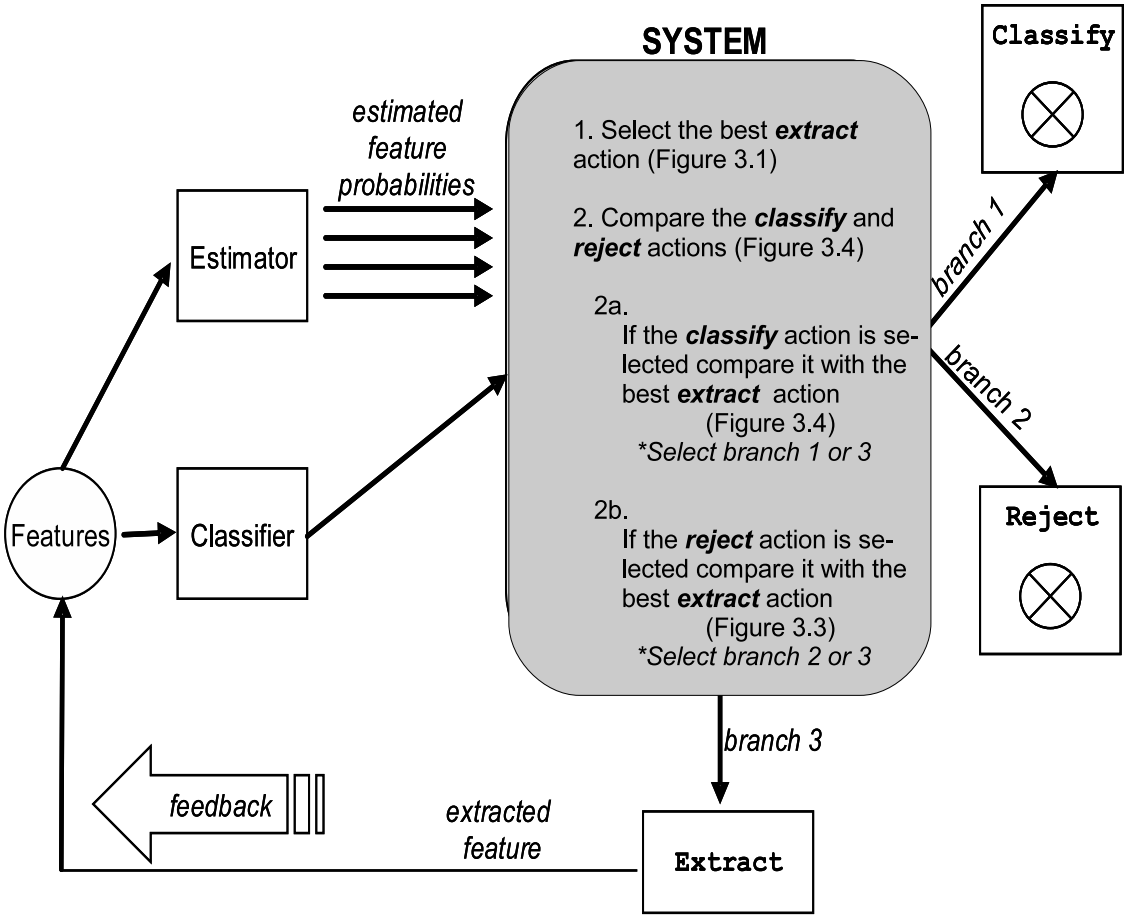


Figure 3.5: The schematic representation of our test-cost sensitive classification algorithm.

these equations, posterior probabilities $P(C_{\texttt{curr}} = j|x)$ are computed by the current classifier using the current information (i.e., using the previously extracted features). However, posteriors $P(C_{\texttt{est}_k} = j|x)$ and $P(C_{\texttt{actual}} = j|x)$ could not exactly be known prior to extracting feature F_k, and hence, they should be estimated using the current information (i.e., using the previously extracted features).

In the estimation of $P(C_{\texttt{est}_k} = j|x)$, we take the following steps. For each unextracted feature F_k, we train a classifier on training samples $D = \{x_t\}_{t=1}^{T}$. In this classifier, the inputs are the features of the training samples including the ones that have already been extracted plus feature F_k and the outputs are the class labels of these training samples. Then, for each training sample x_t, we generate the posterior probabilities $P(C_{\texttt{est}_k} = j|x_t)$ using this classifier. Finally, we train an estimator to learn these generated posteriors from only the features that have already been extracted (but not feature F_k) on the training samples. These estimators are then used to estimate $P(C_{\texttt{est}_k} = j|x)$ for a given test instance x, without using feature F_k. In this work, we use a Parzen window estimator whose kernel function $\rho(u)$ defines a unit hypercube

$$\rho(u) = \begin{cases} 1 & \text{if } |u_i| \leq 1/2 \text{ for all feature dimensions } i = 1, 2, ..., d \\ 0 & \text{otherwise} \end{cases} \tag{3.24}$$

Using this kernel function, the posterior $P(C_{\texttt{est}_k} = j|x)$ is estimated as

$$\widehat{P(C_{\texttt{est}_k} = j|x)} = \frac{\sum_{t=1}^{T} \rho\left(\frac{x-x_t}{h}\right) \cdot P(C_{\texttt{est}_k} = j|x_t)}{\sum_{t=1}^{T} \rho\left(\frac{x-x_t}{h}\right)} \tag{3.25}$$

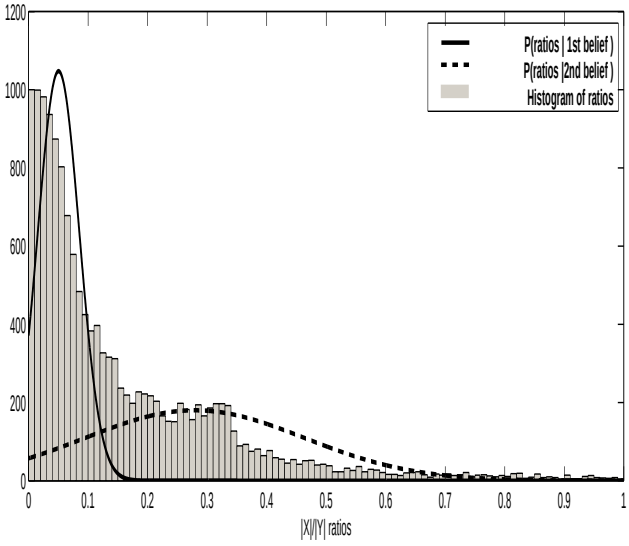
where h is the length of an edge of the hypercube and selected using a leave-one out maximum likelihood estimation.

In the computation of $P(C_{\texttt{actual}} = j|x)$ for the $\texttt{extract}_k$ action, we make use of both the posteriors computed by the current classifier and those estimated by the Parzen window estimators. To do so, for each class j, we multiply the corresponding posteriors ($P(C_{\texttt{curr}} = j|x)$ and $P(C_{\texttt{est}_k} = j|x)$), and then normalize them such that $\sum_{j=1}^{N} P(C_{\texttt{actual}} = j|x) = 1$. For the `classify` and `reject` actions, we use only the posteriors computed by the current classifier

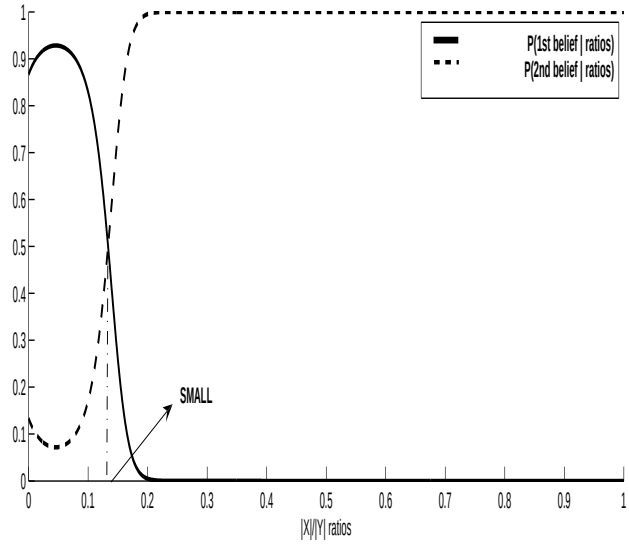
($P(C_{\texttt{curr}} = j|x)$) since these actions stop further feature extractions and no more features are used to classify the given instance x.

Next, we automatically determine the value of SMALL using the training samples. For that, on the training samples, we first determine the distinct cases where the ambiguity arises (e.g., when $|X| < |Y|$ in Case 3 of the $\texttt{extract}_k$-vs-classify comparison). For such cases, we record the $|X|/\ |Y|$ ratios and continue the algorithm by taking the SMALL value as zero. Note that in the learning phase of SMALL, we quantitatively compare $|X\texttt{PENALTY}|$ and $|Y\texttt{REWARD}|$ rather than comparing them qualitatively using Definition 1. Here we suppose that such ambiguous cases arise due to the possibility of two different beliefs (e.g., when $|X| < |Y|$ in Case 3 of the $\texttt{extract}_k$-vs-classify comparison, one belief says to take the $\texttt{extract}_k$ action whereas the other one says to take the classify action). Thus, we assume that these cases, and hence their $|X|/\ |Y|$ ratios, are drawn from a mixture density of two Gaussian components, each representing a different belief. Therefore, on the recorded $|X|/\ |Y|$ ratios, we estimate these two Gaussian components as well as the priors of the beliefs using an expectation-maximization algorithm. Then we determine the SMALL value as the point, where the posterior of the first belief always yields smaller values compared to that of the second belief. For an exemplary data set, the histogram of the $|X|/\ |Y|$ ratios of ambiguous cases and the two Gaussian components estimated on these ratios are shown in Figure 3.6(a). The posteriors which are obtained using the estimated Gaussians and the priors, are shown in Figure 3.6(b); in this figure, the derivation of the SMALL value is also illustrated.

This derivation process of SMALL value is an important issue in our framework for deciding the correct action for ambiguous cases. If the SMALL value is selected to be 1, then no further features are extracted for the ambiguous cases. On the other hand, if the SMALL value is selected to be 0, features are always extracted. Fortunately, in our proposed framework, we determine the SMALL value automatically from the samples of the training set. Thus, no external optimization is necessary for the SMALL value derivation.



(a)



(b)

Figure 3.6: Derivation of the `SMALL` value: (a) the histogram of the distinct $|X|/\ |Y|$ ratios of ambiguous cases and the two Gaussian components estimated on these ratios and (b) posteriors obtained using the estimated Gaussians and prior probabilities.

Chapter 4

Experiments

This chapter describes our experimental study on medical datasets for test-cost sensitive learning. The goal of our experiments is to investigate the benefit of the following two issues:

- *Consistency Behavior*: We investigate whether or not the proposed consistency behavior provides aforementioned benefits for reducing the cost during learning

- *Qualitative Representation*: We also investigate whether or not the qualitative representation is actually robust and efficient in test-cost sensitive learning

To investigate the benefits of these two issues, we conduct our experiments on three real medical datasets which are taken from the UCI repository [41]. We also compare our results with those of the previous studies to investigate the robustness and effectiveness of qualitativeness and consistency concepts.

In this chapter, first, we give a summary of the datasets and explain how we make our evaluations. Then, we report the results of our algorithm using two different types of classifiers and illustrate the derivation of the `SMALL` value

for each dataset. Finally, we compare our algorithm with different well-known algorithms.

4.1 Experimental Setup

We conduct our experiments on three medical datasets taken from (UCI) repository [41]. We choose these datasets for two reasons. First, they are all real medical diagnosis datasets. Second, feature extraction costs have been provided for all of them. Before describing the these dataset in detail, we describe some common properties of them. All of these data sets consist of features extracted by asking questions (*question-based-features*) to a patient as well as those extracted from medical tests (*medical-test-based-features*). A nominal cost of $1 is assigned to the former features and the cost of the corresponding medical test is assigned to the latter ones. Some of the medical tests are in the same group and according to their groups, they share a common cost. For such medical tests, the cost is decreased by an amount of the common cost after conducting the first test of its corresponding group. For instance, suppose that two tests are in the same medical test group. Their costs are $5 and $7 and they have a common cost of $2. This common cost is deducted from amount of these test cost, when one of them is extracted. Thus, if both of these tests are conducted, the total cost will be $10 ($5 + $7 - $2). At the UCI repository, for each dataset, the costs for medical tests and the common costs are given; we also provide their summary below (4.1 - 4.3).

4.1.1 Bupa Liver Disorder Dataset

First dataset that we use in our experiments is the Bupa liver disorder dataset. This dataset includes features to diagnose whether or not a patient has a liver disorder. There are 345 instances, each of which has 5 features and one of two classes (healthy liver or sick). All of the features are *medical-test-based-features* with the costs of {$7.27, $7.27, $7.27, $7.27, $9.86}. All features in Bupa dataset

Table 4.1: Description of the features and their extraction cost for the Bupa Liver Disorder dataset.

	Description	Cost	Group
1	Alkaline phosphotase	\$7.27	A
2	Mean corpuscular volume	\$7.27	A
3	Alamine aminotransferase	\$7.27	A
4	Aspartate aminotransferase	\$7.27	A
5	Gamma-glutamyl transpeptidase	\$9.86	A

belong to the same group that has a common cost of \$2.10. The description of features and their costs are summarized in Table 4.1.

For the Bupa Liver Disorder, there is a single dataset (no separate training and test sets) in the UCI repository. As the size of this dataset is relatively small, we divide dataset into 3 random folds and perform 3-fold cross validation to evaluate the performance our system.

4.1.2 Heart Disease Dataset

This dataset includes features to diagnose whether or not a patient has a heart disease. This dataset has a total of 303 instances. In our experiments, we eliminate six of them that have missing values and use the remaining 297 instances. [1] This dataset includes two classes (sick and healthy). It has 13 features and four of these features are *question-based-features* with \$1 nominal cost and the remaining nine of them are *medical-test-based-features* with the costs of {\$7.27, \$5.20, \$102.90, \$102.90, \$87.30, \$87.30, \$87.30, \$15.50, \$100.90}. In this dataset, there are three feature groups. The group A has a common cost of \$2.10. The group B has a common cost of \$101.90. And the last group C has a common cost of \$86.30. The description of the features and their costs are summarized in

[1] Our algorithm handles missing values depending on the type of a classifier used in our framework. For instance, if the classifier is a decision tree, our framework can handle the missing values. Note that, if such missing values exist, we do not consider the corresponding features in our feature selection algorithm. However, in this thesis, to compare our work with the others we exclude the missing values

Table 4.2: Description of the features and their extraction cost for the Heart Disease dataset.

	Description	Cost	Group
1	Serum cholesterol	$7.27	A
2	Fasting blood sugar	$5.20	A
3	Resting electrocardiograph	$102.90	B
4	Maximum heart rate achieved	$102.90	B
5	Exercise induced angina	$87.30	C
6	ST depression induced by exercise relative to rest	$87.30	C
7	Slope of peak exercise ST	$87.30	C
8	Number of major vessels coloured by fluoroscopy	$15.50	–
9	3 = normal; 6 = fixed defect 7 = reversible defect	$100.90	–
10	Resting blood pressure	$1	–
11	Chest pain type	$1	–
12	Patients gender	$1	–
13	Age in years	$1	–

Table 4.2.

Similarly, for the Heart Disease, there is a single dataset (no separate training and test sets) in the UCI repository. As the size of this dataset is relatively small, we divide the dataset into 3 random folds and perform 3-fold cross validation to evaluate the performance our system.

4.1.3 Thyroid Disease Dataset

The last dataset that we use in our experiments is the Thyroid Dataset. This dataset includes features to diagnose whether a patient has hypothyroid, hyperthyroid or healthy.

It has three classes and 21 features. The first 16 features are *question-based-features* with $1 nominal cost. The next four features are obtained from the blood tests and the assigned costs of these blood tests are

Table 4.3: Description of the features and their extraction cost for the Thyroid Disease dataset.

	Description	Cost	Group
1	Age in years	$1	–
2	Gender	$1	–
3	Patient on thyroxine	$1	–
4	Maybe on thyroxine	$1	–
5	On antithyroid medication	$1	–
6	Patient reports malaise	$1	–
7	Patient pregnant	$1	–
8	History of thyroid surgery	$1	–
9	Patient on I131 treatment	$1	–
10	Maybe hypothyroid	$1	–
11	Maybe hyperthyroid	$1	–
12	Patient on lithium	$1	–
13	Patient has goiter	$1	–
14	Patient has tumor	$1	–
15	Patient hypopituitary	$1	–
16	Psychological symptoms	$1	–
17	TSH value	$22.78	A
18	T3 value	$11.41	A
19	TT4 value	$14.51	A
20	T4U value	$11.41	A
21	FT1 – calculated from TT4 and T4U	Uses features 19 and 20	

{$22.78, $11.41, $14.51, $11.41}. These features belong to the same group and have a common cost of $2.10. In addition, there is another feature which is calculated from the nineteenth and twentieth features. We use this last feature in classification only if both nineteenth and twentieth features are already extracted. The description of the features as well as their extraction costs are summarized in Table 4.3.

For the Thyroid Disease dataset, there are two separate training and test sets in the UCI repository. Thus, we use the test set to evaluate the performance of our system. Note that, the training and test sets consist of 3772 and 3428 instances, respectively; the size of this dataset is very high compared to the other two datasets.

4.2 Results

In our experiments, our algorithm starts with the cheapest feature and sequentially selects a subset of the other features until the `classify` or the `reject` action is taken. If there are more than one feature with the same minimum cost, we select the starting feature based on the distinctive power of these cheapest features; for that we select the one that has the smallest ratio between its within-class and between-class scatter values. In the estimation of the posterior probabilities for non-extracted features, we use Parzen windows. According to the selected window size using a leave one-out likelihood estimation, we compute the posterior probabilities for each non-extracted feature.

Both classifiers and estimators are trained on the training set. In Parzen window estimation, for some of the test samples, there are no training samples falling in its window. For such cases, the estimators do not provide any information, and thus, we do not penalize any feature extraction and consider only the posteriors obtained by the current classifier to compute the conditional risks.

In this section, we first use decision trees as the classifiers and show the results obtained by our proposed framework. Then, we use hidden Markov Models (HMMs) as the classifiers and show the obtained results. Finally, we compare our results with other studies in section 4.3. For both of the classifiers, we investigate the benefits of the consistency concept. To this end, we also conduct our experiments without considering the consistency; we always reward the feature extraction (with an amount of REWARD) if it yields correct classification, regardless of whether or not this classification would be consistent with our current decision. For these classifiers, we also show the derivation process of the `SMALL` value for the three datasets. At the end of the this chapter, we compare our results with those of the previous studies to understand the benefits of the use of qualitativeness and consistency concepts.

4.2.1 Decision Tree

In this section, we give the results of our framework when the classifier is a decision tree. The results of our framework are presented in Sections 4.2.1.1, 4.2.1.2 and 4.2.1.3, for the Bupa, Heart, and Thyroid datasets, respectively.

4.2.1.1 Results of the BUPA dataset

We report the test results of our proposed algorithm for each fold of the Bupa dataset in Table 4.4. In this table, we provide the accuracy, the percentage of the reduction in the overall feature extraction cost, and the number of samples for which the reject action is taken. We also provide the results obtained by a baseline classifier, which uses all of the available features in its decision tree construction. The results in Table 4.4 show that our algorithm yields significant cost reductions for each fold. Also they show that the results of our algorithm are better than those of the baseline classifier for fold1 and fold3.

In Table 4.5, we give the results when the consistency is not considered. Comparing the results considering consistency and without considering consistency (Tables 4.4 and 4.5), we observe that the consistency provides more amount of cost reduction. This is because of our algorithm avoiding to pay for tests that are not going to change the current decision.

4.2.1.2 Results of the Heart dataset

For the Heart dataset, we report the test results of our framework for each fold in Table 4.6. This table demonstrates that the proposed algorithm significantly decreases cost expenditure without decreasing the accuracy. Furthermore, similar to the case of the Bupa dataset, the accuracy for fold2 and fol3 increases. This is attributed to the effect of curse of dimensionality, since these datasets include a less amount of samples and the baseline classifier uses all of the features whereas the proposed algorithm uses only a subset of them. In Table 4.7, we report the

Table 4.4: For the Bupa dataset, the results obtained by our qualitative test-cost sensitive algorithm and those obtained by the baseline classifier, which uses all of the features in its decision tree construction.

	Decision Tree Baseline	Our algorithm		
	Accuracy	Accuracy	Cost red. percentage	Number of reject cases
Fold1	57.50	59.17	70.34	-
Fold2	55.08	53.39	65.86	-
Fold3	57.63	64.41	69.81	-
Avg.	56.74	58.99	68.67	-
Std.	1.43	5.51	2.45	-

Table 4.5: For the Bupa dataset, the results are obtained by our qualitative test-cost sensitive algorithm when the consistency is not considered.Here, decision tree classifiers are used

	Consistency-off		
	Accuracy	Cost red. percentage	Number of reject cases
Fold1	55.83	23.14	-
Fold2	55.08	7.61	-
Fold3	63.79	22.74	2
Avg.	57.66	17.08	2
Std.	3.83	8.86	-

results obtained when the consistency is not considered. Similar to the Bupa dataset, without considering consistency the cost reduction percentages severely decreases.

4.2.1.3 Results of the Thyroid Dataset

In Table 4.8, we report the test results obtained by our algorithm for the Thyroid dataset. The results when the consistency is not considered are given in Table 4.9. Similarly, these results demonstrate that the proposed algorithm leads to a close accuracy to the baseline classifier and reduces the cost expenditure greatly. When

Table 4.6: For the Heart dataset, the results obtained by our qualitative test-cost sensitive algorithm and those obtained by the baseline classifier, which uses all of the features in its decision tree construction.

	Decision Tree Baseline	Our algorithm		
	Accuracy	Accuracy	Cost red. percentage	Number of reject cases
Fold1	78.79	77.78	62.15	-
Fold2	73.74	81.82	69.96	-
Fold3	73.74	74.75	70.21	-
Avg.	75.42	78.11	67.44	-
Std.	2.92	3.55	4.58	-

Table 4.7: For the Heart dataset, the results are obtained by our qualitative test-cost sensitive algorithm when the consistency is not considered. Here, decision tree classifiers are used

	Consistency-off		
	Accuracy	Cost red. percentage	Number of reject cases
Fold1	76.77	0.34	-
Fold2	73.74	$\tilde{0}$	-
Fold3	73.74	0.09	-
Avg.	74.75	0.14	-
Std	1.75	0.18	-

Table 4.8: For the Thyroid dataset, the results obtained by our qualitative test-cost sensitive algorithm and those obtained by the baseline classifier, which uses all of the features in its decision tree construction.

	Decision Tree Baseline	Our algorithm		
	Accuracy	Accuracy	Cost red. percentage	Number of reject cases
Test Set	98.57	98.08	53.01	1

Table 4.9: For the Thyroid dataset, the results are obtained by our qualitative test-cost sensitive algorithm when the consistency is not considered. Here, decision tree classifiers are used

	Consistency-off		
	Accuracy	Cost red. percentage	Number of reject cases
Thyroid	98.19	5.89	3

consistency is not considered, the cost reduction severely decreases although the accuracy is still near to that of the baseline classifier. Please note that, we automatically derive the `SMALL` value. Thus, we do not expect the best possible results. However, the results given in Table 4.8 show that our algorithm selects the `SMALL` value in a good range, and hence, its accuracy is close to that of baseline classifier while its cost reduction is high (more than 50 percent).

4.2.1.4 Derivation of the `SMALL` Value

In this section, we illustrate how to derive the `SMALL` value for a dataset on the plots given in Figures 4.1, 4.2, and 4.3 for the Bupa, Heart and Thyroid datasets, respectively. For that, on the histogram of the ratios we estimate two Gaussian distributions (the likelihoods) and their priors. Then we combine them into the posterior probabilities as shown these Figures. The point where the posteriors

are equal to each other is determined as the SMALL value.

In our experiments, the derived SMALL values for the Bupa dataset are 0.335, 0.325, and 0.215 for fold1, fold2 and fold3, respectively (Figure 4.1). The SMALL values for the Heart dataset are 0.212, 0.265 and 0.277 for fold1, fold2, and fold3, respectively (Figure 4.2). The SMALL for the Thyroid dataset is 0.06 (Figure 4.3).

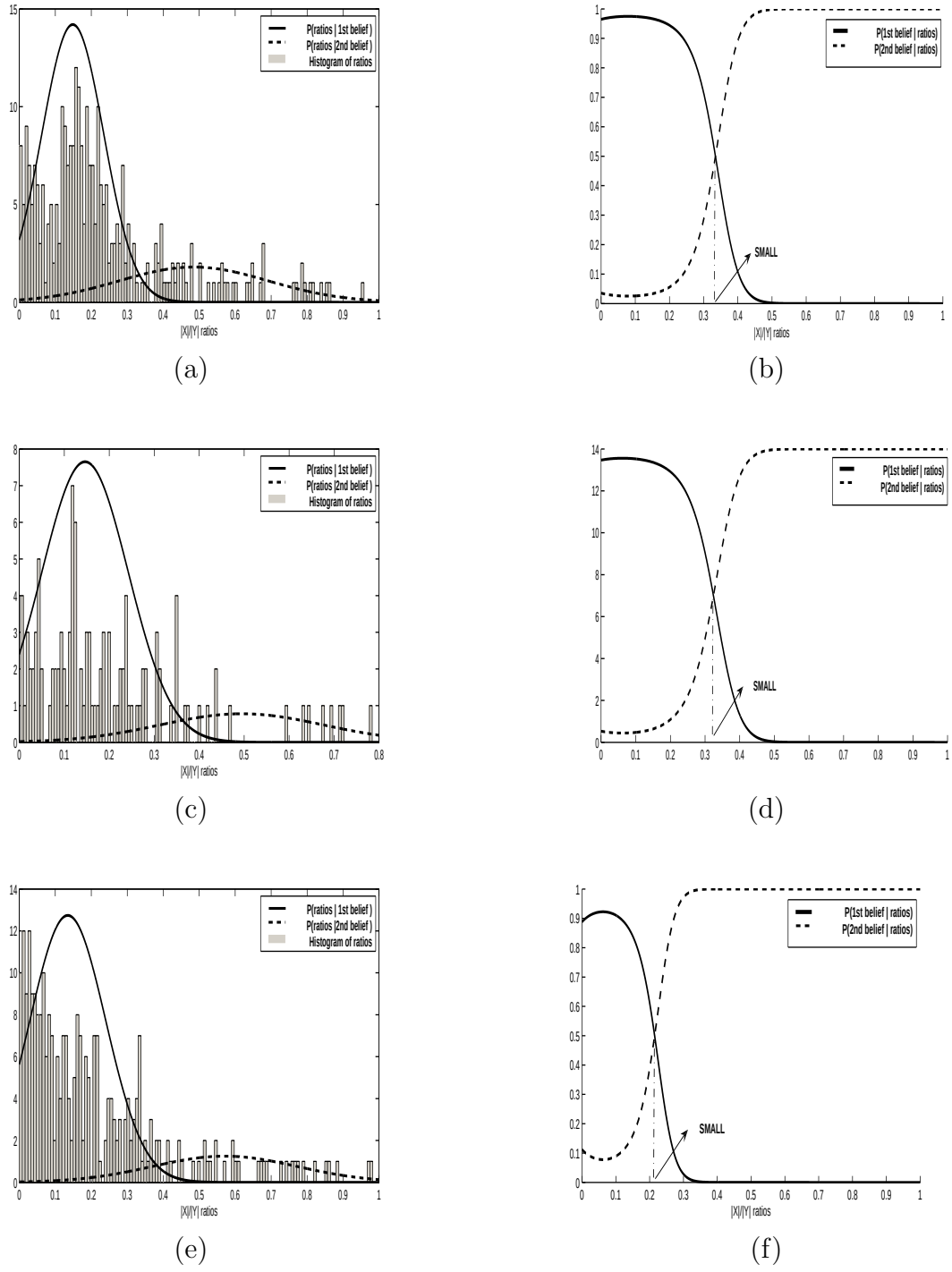


Figure 4.1: Derivation of the `SMALL` value: (a),(c), and (e) are the histograms of the distinct $|X|/|Y|$ ratios of ambiguous cases and two Gaussian components estimated on these ratios for the Bupa (fold1, fold2, and fold3, respectively). (b),(d) and (e) are posteriors obtained using estimated Gaussians and prior probabilities for the Bupa (fold1, fold2, and fold3, respectively). Here, decision tree classifiers are used.

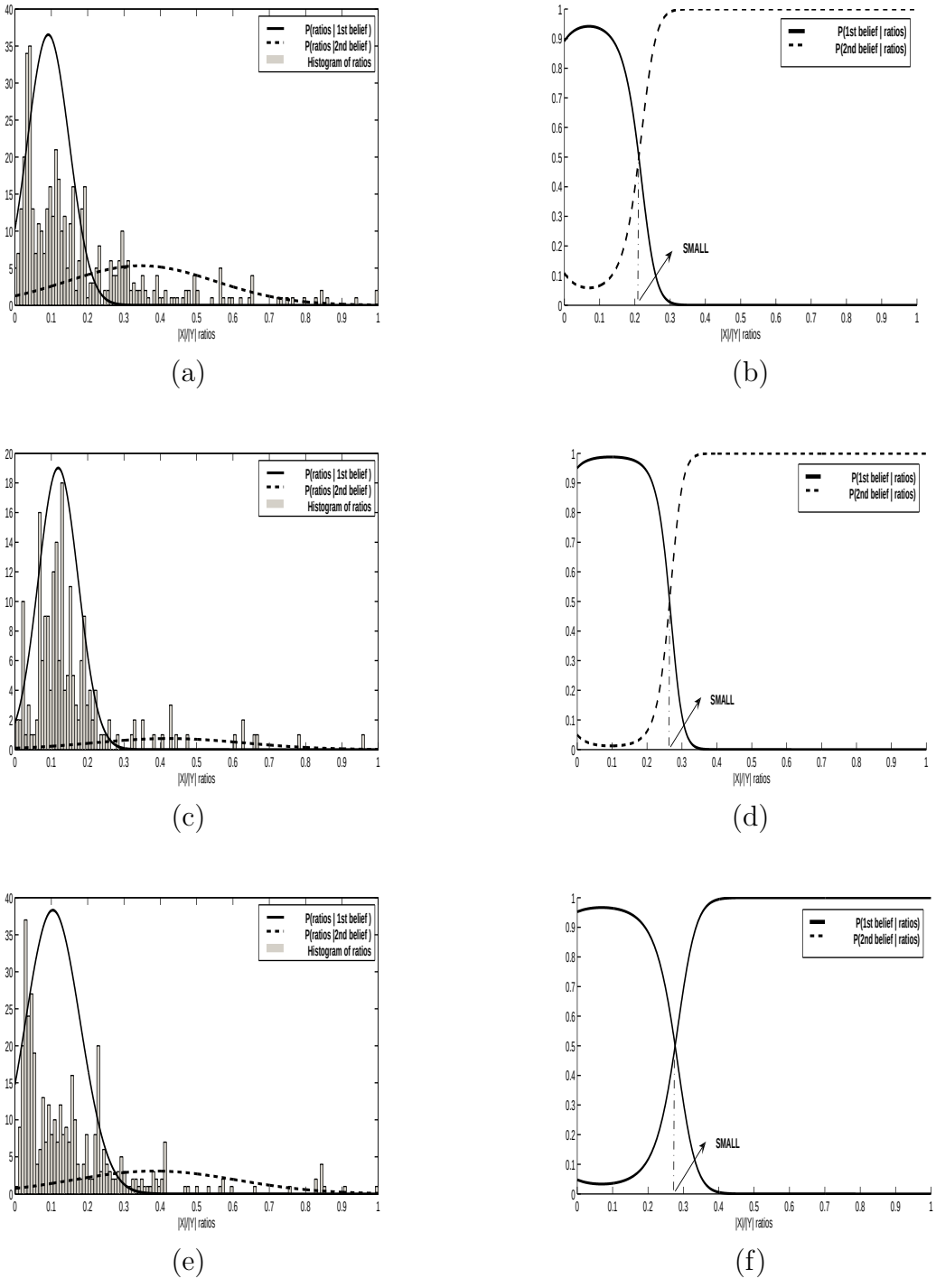
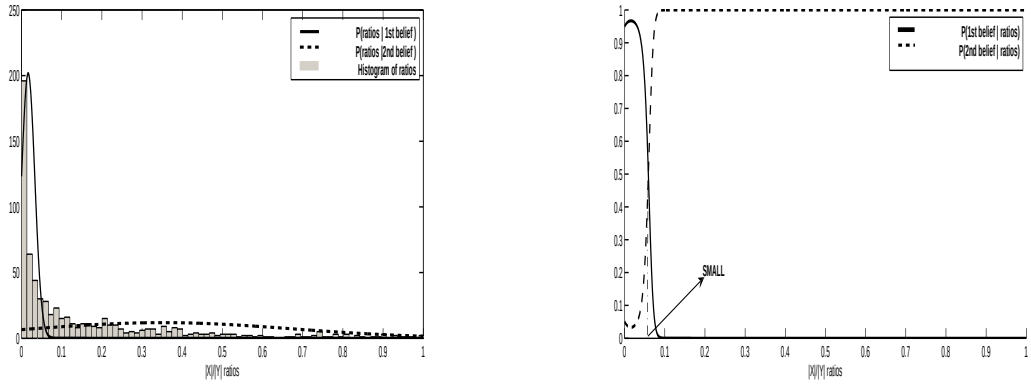


Figure 4.2: Derivation of the `SMALL` value: (a),(c), and (e) are the histograms of the distinct $|X|/|Y|$ ratios of ambiguous cases and two Gaussian components estimated on these ratios for the Heart (fold1, fold2 and fold3, respectively). (b),(d), and (e) are posteriors obtained using estimated Gaussians and prior probabilities for the Heart (fold1, fold2, and fold3, respectively). Here, decision tree classifiers are used.



(a) (b)

Figure 4.3: Derivation of the SMALL value: (a) is the histogram of the distinct $|X|/|Y|$ ratios of ambiguous cases and two Gaussian components estimated on these ratios for the Thyroid. (b) is posteriors obtained using estimated Gaussians and prior probabilities for the Thyroid. Here, decision tree classifiers are used.

4.2.2 Hidden Markov Model

In this section, we provide the results of our framework when a Hidden Markov Model classifier is used. The results of our framework are presented in sections 4.2.2.1, 4.2.2.2, and 4.2.2.3 for the Bupa, Heart, and Thyroid datasets, respectively.

4.2.2.1 Results of the BUPA dataset

In Table 4.10, we present the results of our algorithm when a HMM classifier is used. Similarly, we obtain high percentages of cost reduction with no decrease in the diagnosis accuracy. This demonstrates that our algorithm is independent from the classifier type and any of the classifier could be used in the system. For this dataset, the accuracies and cost reductions are very similar to those in the case of the use of decision trees. In Table 4.11, we also report our results when consistency is not considered. Similarly, this table shows that without consistency, the cost expenditure is high.

Table 4.10: For the Bupa dataset, the results obtained by our qualitative test-cost sensitive algorithm and those obtained by the baseline classifier, which uses all of the features in its HMM.

	HMM Baseline	Our algorithm		
	Accuracy	Accuracy	Cost red. percentage	Number of reject cases
Fold1	57.50	56.67	68.86	-
Fold2	56.78	59.32	64.14	-
Fold3	61.02	61.86	67.01	-
Avg.	58.43	59.28	66.67	-
Std.	2.60	2.27	2.38	-

Table 4.11: For the Bupa dataset, the results are obtained by our qualitative test-cost sensitive algorithm when the consistency is not considered. Here, HMM classifiers are used

	Consistency-off		
	Accuracy	Cost red. percentage	Number of reject cases
Fold1	55.83	6.21	-
Fold2	56.78	7.75	-
Fold3	59.32	13.99	-
Avg.	57.32	9.32	-
Std.	1.84	4.12	-

Table 4.12: For the Heart dataset, the results obtained by our qualitative test-cost sensitive algorithm and those obtained by the baseline classifier, which uses all of the features in its HMM.

	HMM Baseline	Our algorithm		
	Accuracy	Accuracy	Cost red. percentage	Number of reject cases
Fold1	84.85	86.87	43.11	-
Fold2	85.86	82.83	51.78	-
Fold3	77.78	76.77	52.14	-
Fold2	82.83	82.15	49.01	-
Fold3	4.40	5.08	5.11	-

4.2.2.2 Results of the Heart dataset

In Table 4.12, we report the results of the Heart dataset, when the classifier used is an HMM. In this table, the results show that the accuracy of the baseline classifier is much higher than that of the decision tree. Thus, we obtain higher values compared to the case of decision trees. Our results are again close to the accuracy of the baseline classifier. This shows that the accuracy of the proposed algorithm depends on the accuracy of the baseline classifier. Furthermore it also shows that the use of more accurate baseline classifiers improves the performance of our algorithm. Similar to the previous case, our algorithm still yields high percentages of cost reduction. These values are lower than those reported in Table 4.12 since the algorithm believes that with an HMM classifier, it could reach higher accuracy values when more number of features are used. Similar to the decision tree case, without considering the consistency, we obtain lower cost reductions (Table 4.13).

4.2.2.3 Results of the Thyroid dataset

In Table 4.14, we report the test results for the thyroid dataset when a HMM classifier is used. In this table, the proposed algorithm leads to significant amount

Table 4.13: For the Heart dataset, the results are obtained by our qualitative test-cost sensitive algorithm when the consistency is not considered. Here, HMM classifiers are used

	Consistency-off		
	Accuracy	Cost red. percentage	Number of reject cases
Fold1	84.85	0.75	-
Fold2	85.86	2.95	-
Fold3	76.77	0.40	-
Avg.	82.49	1.37	-
Std.	4.98	1.38	-

Table 4.14: For the Thyroid dataset, the results obtained by our qualitative test-cost sensitive algorithm and those obtained by the baseline classifier, which uses all of the features in its decision tree construction.

	HMM Baseline	Our algorithm		
	Accuracy	Accuracy	Cost red. percentage	Number of reject cases
Test Set	95.62	96.03	46.58	26

of cost reduction without decreasing accuracy. Similarly without the consistency behavior, there is almost no cost reduction (Table 4.15). For the thyroid dataset, there are more rejection cases when the HMM classifier is used. When we analyze the rejection cases, we see that nearly half of them are the samples that are misclassified by the baseline classifier. The rejection action is selected to prevent misclassification as there is no way to classify these samples correctly even all of the features are used. Here note that in the computation of the accuracy, we consider the reject cases as incorrect classifications (i.e., the accuracy is computed by dividing the number of correct classifications by the size of the dataset).

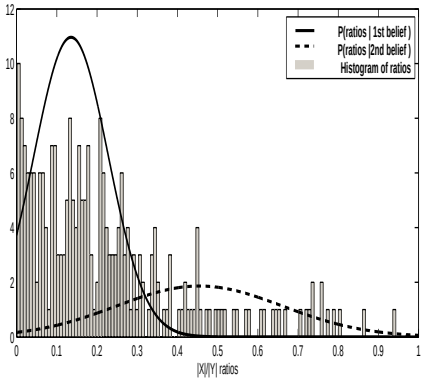
Table 4.15: For the Thyroid dataset, the results are obtained by our qualitative test-cost sensitive algorithm when the consistency is not considered. Here, HMM classifiers are used

	Consistency-off		
	Accuracy	Cost red. percentage	Number of reject cases
Thyroid	94.78	0.34	44

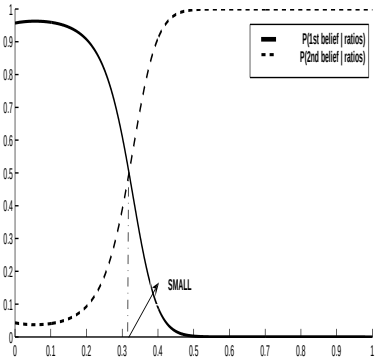
4.2.2.4 Derivation of the SMALL

Similar to case of the decision trees, we illustrate how to select the value of SMALL for three datasets in Figures in Figures 4.3, 4.4, and 4.5 for the Bupa, Heart and Thyroid datasets, respectively. Similarly, this derivation is done automatically on the training sets and the test samples are not used in this process at all.

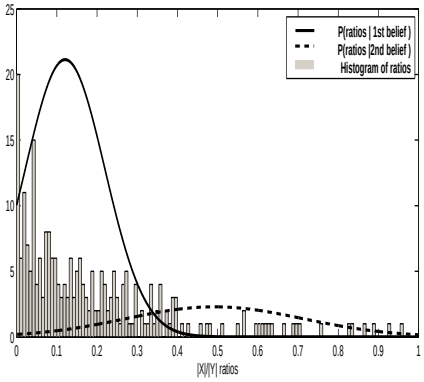
In our experiments, the derived SMALL values for the Bupa dataset are 0.320, 0.183, and 0.174 for fold1, fold2, and fold3, respectively (Figure 4.4). The SMALL values for the Heart are 0.040, 0.051 and 0.040 for fold1, fold2 and fold3, respectively (Figure 4.5). The SMALL for the Thyroid is 0.134 (Figure 4.6).



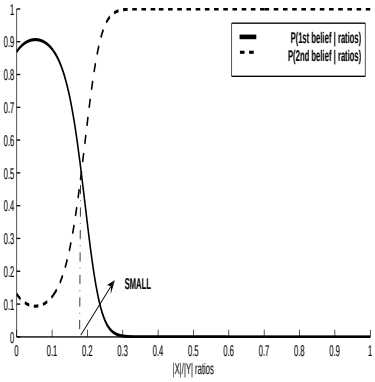
(a)



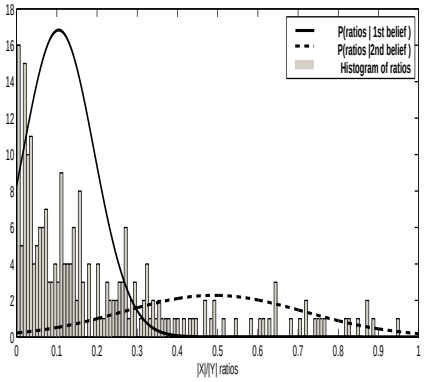
(b)



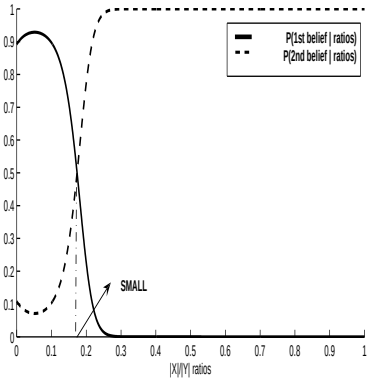
(c)



(d)



(e)



(f)

Figure 4.4: Derivation of the `SMALL` value: (a),(c) and (e) are the histograms of the distinct $|X|/|Y|$ ratios of ambiguous cases and two Gaussian components estimated on these ratios for the Bupa ((fold1, fold2, and fold3, respectively)). (b),(d), and (e) are posteriors obtained using estimated Gaussians and prior probabilities for the Bupa (fold1, fold2, and fold3, respectively). Here, HMM classifiers are used.

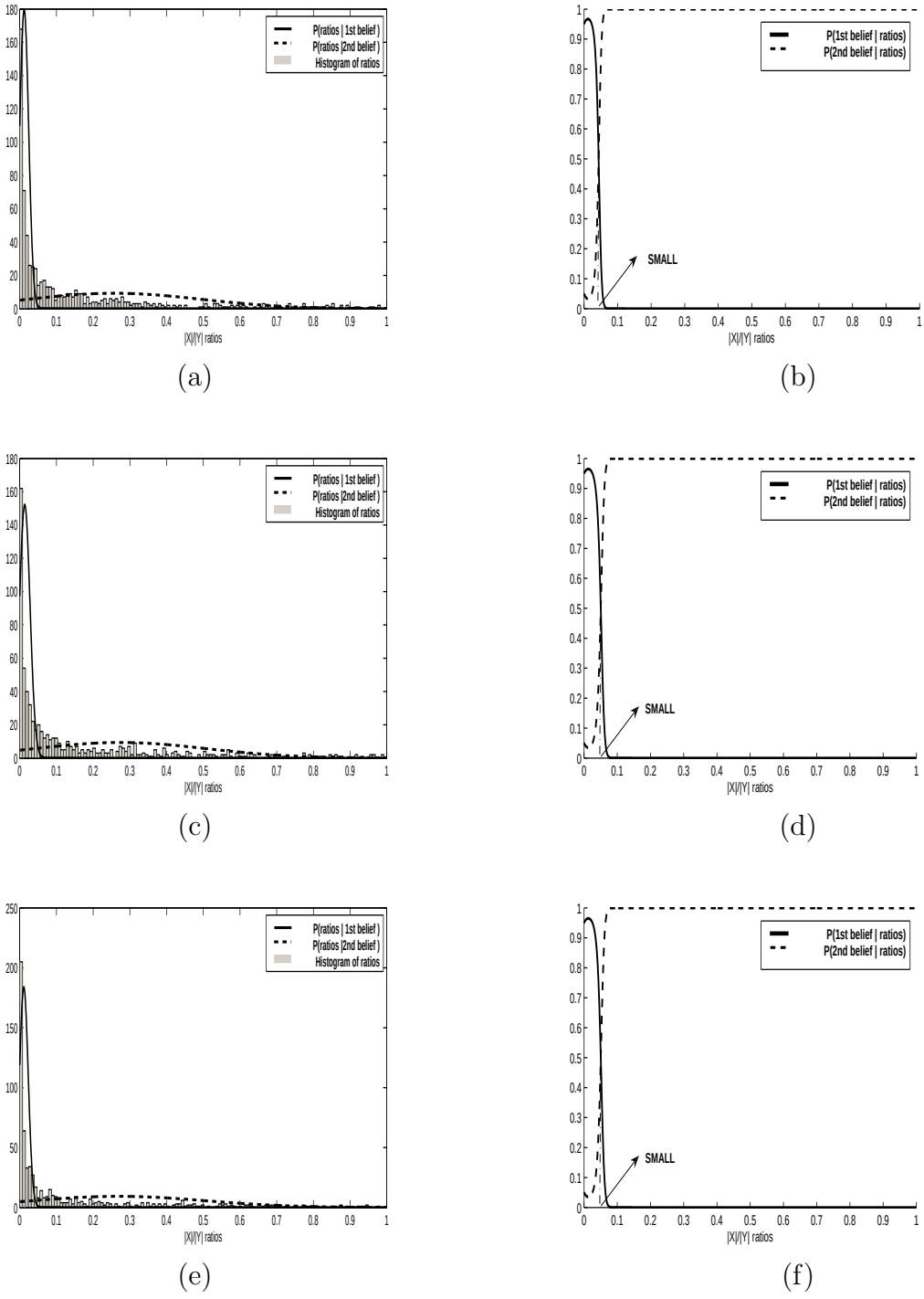
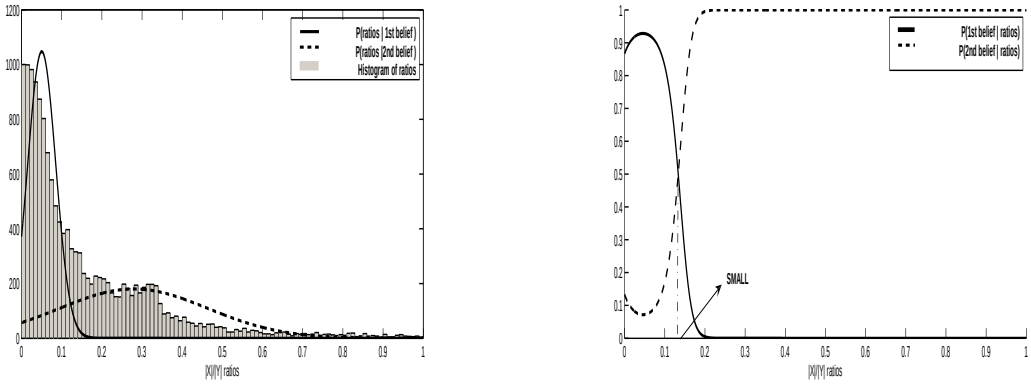


Figure 4.5: Derivation of the `SMALL` value: (a),(c) and (e) are the histograms of the distinct $|X|/|Y|$ ratios of ambiguous cases and two Gaussian components estimated on these ratios for the Heart (fold1, fold2, and fold3, respectively). (b),(d), and (e) are posteriors obtained using estimated Gaussians and prior probabilities for the Heart (fold1, fold2, and fold3, respectively). Here, HMM classifiers are used.



(a) (b)

Figure 4.6: Derivation of the `SMALL` value: (a) is the histogram of the distinct $|X|/|Y|$ ratios of ambiguous cases and two Gaussian components estimated on these ratios for the Thyroid. (b) is posteriors obtained using estimated Gaussians and prior probabilities for the Thyroid. Here, HMM classifiers are used.

4.3 Comparisons

In this section, we compare our results with those of two previous studies. First of these studies uses decision trees and the other one uses hidden Markov models for test-cost sensitive learning. In test-cost sensitive learning, several approaches have been proposed to modify existing decision tree algorithms. One of the most influential one is ICET [7] and we compare our results with the results of ICET in Section 4.3.1. The other study in [23], incorporates feature extraction cost into hidden Markov models (HMM); they use a partially observable Markov decision process (POMDP) to select the subset of features. The comparison of the results of our algorithm and algorithm proposed in [23] is given Section 4.3.2.

4.3.1 Comparison by ICET

ICET is a hybrid of a genetic algorithm and a cost-sensitive decision tree growing algorithm. In this algorithm, ICET first employs a genetic search by modifying the test costs empirically (attaining random costs) to build different decision

Table 4.16: The results obtained by our qualitative test-cost sensitive algorithm when a decision tree classifier is used and the results of the ICET algorithm; the results of ICET are the best reported ones.

	Baseline	Our algorithm		ICET	
	Accuracy	Accuracy	Cost red. percentage	Accuracy	Cost red. percentage
Bupa	56.74	59.59	68.67	54.5	25.5
Heart	75.42	78.11	67.44	74.1	33.8
Thyroid	98.57	98.07	53.01	99.1	22.5

trees. Then, ICET selects the best one among decision trees according to a utility function that uses both the feature extraction costs and the misclassification costs. To fairly compare the performance of ICET and our algorithm, we obtain our results by using decision tree classifiers in our framework and obtain ICET results from [7] with the best reported accuracy and cost-reduction. Note that in [7], the best value of the misclassification is externally selected by trying different values. However in our algorithm, there is no need to externally optimize the parameters, the value of SMALL is automatically determined. Here we should note that, although both of these algorithms use decision trees, our algorithm do not build a test-cost sensitive decision tree. The test-cost sensitivity of our method is obtained during the selection of features. Table 4.16 shows the comparison result against ICET. Our algorithm leads to more amount of cost reduction compared to the algorithm ICET. This indicates the importance of the consistency behavior. Moreover, it leads to more accurate results for the Bupa and Heart data sets. However, it is less accurate for the Thyroid data set, but it is still close to the accuracy of baseline classifier.

4.3.2 Comparison by POMDP

In this section, we also compare our results with those of the study in [23]. In [23], they use a partially observable Markov decision process (POMDP) to solve cost-sensitive classification problem. Their POMDP model sequentially determines the action (classify or extract) at each step according to a utility

function that considers both feature extraction cost and misclassification cost. Once the classify action is selected, their POMDP model reduces to classical HMM model for classification. In order to have a fair comparison, we use the same HMM classifier as the classifier of our model. The feature selection policy of [23] has two free model parameters (namely, the cost of correct classification and the cost of misclassification). In Table 4.17, we report the results obtained when these two model parameters are optimized on the training samples (i.e., when the best combination of the parameters, which yields the best accuracies and cost reductions are selected). On the other hand, the feature selection policy of our algorithm does not require any free model parameters to be externally optimized; there is no need for the user to determine the value of SMALL beforehand since it is automatically determined on the training samples, reflecting two different beliefs of the user.

The results in Table 4.17 show that our algorithm leads to more accurate results, which are closer to those of the baseline classifier than the results of study in [23]. Moreover, it leads to higher amount of cost reduction for the Bupa and Heart data sets, indicating the importance of the consistency behavior. It yields less amount of cost reduction for the Thyroid dataset. However, it obtains higher accuracy, which results in extracting more features. For this data set, our algorithm tries to improve the accuracy at the cost of extracting more and more features as the cost of misclassification (and the benefit of correct classification) is assumed to be always greater than the extraction cost of any features (our third assumption).

Table 4.17: The results obtained by our qualitative test-cost sensitive algorithm when an HMM classifier is used and the results of the algorithm developed by Ji and Carin [23]

.

	Baseline	Our algorithm		Algorithm in [23]	
	Accuracy	Accuracy	Cost red. percentage	Accuracy	Cost red. percentage
Bupa	58.43	59.28	66.67	58.71	23.47
Heart	82.83	82.15	49.01	80.81	47.89
Thyroid	95.68	95.62	46.58	94.81	52.90

Chapter 5

Conclusions

5.1 Discussions

In this section, we discuss whether or not the consistency behavior provides more amount of cost reduction during experiments. Next, we investigate the benefit and necessity of using qualitative information in test-cost sensitive learning. Finally, we mention about the benefits of the automatic derivation of the `SMALL` value.

The results given in Tables 4.10-4.15 in Chapter 4 show that without having the consistency behavior, the algorithm tends to extract almost all of the features. This is most probably because of the assumption of misclassification cost being greater than the extraction cost of any feature (the third assumption). On the other hand, with consistency, our algorithm can stop extracting the features if it ensures that the future decisions are consistent with the current one. This prevents extracting features that bring about no new information.

In Chapter 1, we emphasize the necessity of using qualitative information in decision making in a situation where quantization of probabilities and/or utilities can not be accomplished. In addition to this necessity, we observe the benefit of enabling qualitative information to test-cost sensitive learning in Figures 3.1-3.4.

These figures show the obtained analysis capability of qualitative information in test-cost sensitive learning. In these figures, we qualitatively group the different cases; thus we determine the cases that need further investigation and the cases that can be decided easily. With further investigation, we bring the `SMALL` value into play for the cases where ambiguity arises; therefore, we do not need to wonder about for the other cases. These benefits provide us to cluster the steps of decision making into comprehensive meaningful classes. By doing so, we capture intrinsic points of the problem.

With the help of automatic derivation of the `SMALL` value, there is no need to determine the relational importance of misclassification cost to the feature extraction cost. Instead, we automatically derive it from the training data. Furthermore, the derivation of the `SMALL` value could adaptively be modified when a new test instance arises. After deciding on this instance, the corresponding ratios may be included in the `SMALL` value calculation process. This provides improvement the knowledge about the importance relation between the misclassification cost and the feature extraction cost during the test.

5.2 Conclusions

In this thesis, we introduce a new Bayesian decision theoretical framework for test-cost sensitive classification. In this framework, we use a new loss function definition in which the misclassification cost and the cost of feature extraction are qualitatively combined and the loss function is conditioned with the current information and the information expected after feature extraction as well as the consistency among them. Working with three medical diagnosis problems, our experiments demonstrate that *1.)* our proposed approach significantly decreases the overall feature extraction cost without decreasing the diagnosis accuracy, and *2.)* it overcomes the problem for the user to express his/her prior belief (the relation between the misclassification cost and the cost of feature extraction) as an exact quantitative number.

One of the future research directions is to investigate the incorporation of the qualitative decision theory into other machine learning problems. Another one is the consideration of different misclassification cost for different actions. This modifies the system to handle complex loss functions where different loses associated with misclassifications. Another possibility is to also include the other types of cost (e.g., the computational cost) into the problem formulation.

Bibliography

[1] Dubois D., Fargier H., Prade H.: Decision-making under ordinal preferences and uncertainty. AAAI Spring Symposium on Qualitative Preferences in Deliberation and Practical Reasoning, 4146, 1997.

[2] Boutilier C.: Towards a logic for qualitative decision theory. In Proc. of the 4rd Inter. Conf. on Princ. of Knowl. Repres. and Reason KR'94), 7586, Bonn, Germany.

[3] Giang, P. H.,Shenoy, P. P.: A comparison of axiomatic approaches to qualitative decision making under possibility theory, Proc. of 17th Conf. on Uncertainty in Artificial Intelligence UAI01 162170.

[4] Boutilier C.: Toward a logic for qualitative decision theory, Proc. 4th Inter. Conf. on Principles of Knowledge Representation and Reasoning (KR'94), 75-86, Bonn, Germany.

[5] Tan S.W., Pearl J.: Specification and evaluation of preferences under uncertainty. In Proc. 4th Inter. Conf on Principles of Knowledge Representation and Reasoning (KR'94),530-539, Bonn, Germany. 1994.

[6] Thomason R.: Desires and defaults: A framework for planning with inferred goals:Proc. 7th Internat. Conference on Principles of Knowledge Representation and Reasoning (KR00), 702713,San Mateo, CA.

[7] Turney, P.D.: Low size-complexity inductive logic programming: The East-West Challenge considered as a problem in cost-sensitive classification. Proceedings of the Fifth International Inductive Logic Programming Workshop, 1995.

[8] Demir, C., Alpaydin, E.: Cost-conscious classifier ensembles. Pattern Recognition Letters **26** (2005) 2206-2214.

[9] P Domingos: MetaCost: A general method for making classifiers cost-sensitive , ICKDM 1999,155-164,New York.

[10] Charles Elkan: The Foundations of Cost-Sensitive Learning, IJCAI 2001,973-978.

[11] Xiaoyong Chai, Ling Deng and Quiang Yang: Test Cost Sensitive Naive Bayes Classification, ICDM 2004, Stanford, CA.

[12] Turney, P.D.: Types of cost in inductive concept learning. Workshop on Cost-Sensitive Learning, ICML 2000, Stanford, CA.

[13] Duda, O.R., Hart, E.P., Stork, G.D., Pattern Classification. Wiley-Interscience, New York, 2001

[14] Norton, S.W.: Generating better decision trees. IJCAI 1989, Detroit, MI

[15] Nunez, M.: The use of background knowledge in decision tree induction. Mach Learn **6** (1991) 231-250

[16] Tan, M.: Cost-sensitive learning of classification knowledge and its applications in robotics. Mach Learn **13** (1993) 7-33

[17] Turney, P.D.: Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm. J Artif Intell Res **2** (1995) 369-409

[18] Davis, J.V., Ha, J., Rossbach, C.J., Ramadan, H.E., Witchel, E.: Cost-sensitive decision tree learning for forensic classification. ECML 2006, Berlin, Germany, 2006

[19] Yang, Q., Ling, C., Chai, X., Pan, R.: Test-cost sensitive classification on data missing values. IEEE T Knowl Data En **18** (2006) 626-638

[20] Gunduz, C.: Value of representation in pattern recognition. M.S. thesis, Bogazici University, Istanbul, Turkey, 2001

[21] Zhang, Y., Ji, Q.: Active and dynamic information fusion for multisensor systems with dynamic Bayesian networks. IEEE T Syst Man Cy B **36** (2006)

[22] Zubek, V.B., Dietterich, T.G.: Pruning improves heuristic search for cost-sensitive learning. ICML 2002, San Francisco, CA, 2002

[23] Ji, S., Carin, L.: Cost-sensitive feature acquisition and classification. Pattern Recogn **40** (2007) 1474-1485

[24] Doyle, J., Thomason, R.H.: Background to qualitative decision theory. AI Mag **20**(2) (1999) 55-68

[25] Bolt J. H., Renooij S., Gaag L. C.: Upgrading Ambiguous Signs in QPNs. UAI 2003: 73-80

[26] Pearl, J.: Probabilistic reasoning in intelligent systems: Networks of plausible inference. Morgan Kauffman, 1998, California

[27] Drudzel M. J., Henrion M., Belief Propagation in Qualitative Probabilistic Networks, In N. Piera Carrete and M. G. Singh, Qualitative Reasoning and Decision Technologies,451 460,Barcelona, 1993.

[28] von Neumann, J., Morgenstern O., Theory of Games and Economic Behavior. Princeton University Press, 1947

[29] Wellman, M.P.: Fundamental concepts of qualitative probabilstic networks. Artif Intell **44**(3) (1990) 257-303

[30] Wellman, M. P.: Graphical inference in qualitative probabilistic networks: Networks, 1990, John Wiley & Sons.

[31] Renooij, S., van der Gaag, L.C.: Decision making in qualitative influence diagrams. FLAIRS Conference 1998, Menlo Park, CA, 1998

[32] Renooij, S., van der Gaag, L.C.: From qualitative to quantitative probabilistic networks. UAI 2002, San Francisco, CA, 2002

[33] Brafman, R.I., Domshlak, C., Shimony, S.E.: Qualitative decision making in adaptive presentation of structured information. ACM T Inform Syst **22**(4) (2004) 503-539

[34] Pearl, J.: From qualitative utility to conditional ought to. UAI 1993, San Mateo, CA, 1993

[35] Dubois, D., Prade, H.: Possibility theory as a basis for qualitative decision theory. IJCAI 1995, San Francisco, CA, 1995

[36] Brafman, R.I., Tennenholtz, M.: On the foundations of qualitative decision theory. AAAI 1996, Portland, OR, 1996

[37] Lehmann, D.: Expected qualitative utility maximization. Game Econ Behav **35**(12) (2001) 54-79

[38] Dubois, D., Fargier, H., Prade, H., Perny, P.: Qualitative decision theory: from savage's axioms to nonmonotonic reasoning. J ACM **49**(4) (2002) 455-495

[39] Fargier, H., Sabbadin, R.: Qualitative decision under uncertainty: back to expected utility. Artif Intell **164** (2005) 245-280

[40] Kuipers, B., Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge. Massachusetts Institute of Technology, Cambridge, 1994

[41] Blake, C.L., Merz, C.J.: UCI Repository of machine learning databases. Available at http://www.ics.uci.edu/∼mlearn/MLRepository.html, 1998