LUDWIG-MAXIMILIANS-UNIVERSITÄT
TECHNISCHE UNIVERSITÄT MÜNCHEN

Institut für Informatik XII

Masterarbeit
in Bioinformatik

On Abstaining Classifiers

Caroline Friedel

Aufgabensteller:	Prof. Dr. Stefan Kramer
Betreuer:	Ulrich Rückert
Abgabedatum:	8. April 2005

Ich versichere, dass ich diese Masterarbeit selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

8. April 2005 ______________________
Caroline Friedel

Doubt is not a pleasant condition, but certainty is absurd.

Voltaire (1694-1778)

Abstract

Contrary to standard non-abstaining classifiers, abstaining classifiers have the choice to label an instance with any of the given class labels or to refrain from giving a classification in order to improve predictive performance. Our interest in abstaining classifiers is motivated by applications for which reliable predictions can only be obtained for a fraction of instances such as, for example, chemical risk assessment which involves the prediction of toxic side-effects.

The goal of this thesis was to define an appropriate method to choose between classification and abstention which does not rely on any specific characteristics of machine learning algorithms or applications. In this way, any non-abstaining classifier can be converted into an abstaining classifier by calculating a so-called optimal abstention window.

Abstaining classifiers have to trade off improved predictive performance against reduced coverage taking into account the costs associated with misclassifications and abstaining respectively. Depending on the specific application, abstaining will be more or less preferred for the same cost scenarios. Nevertheless, we can make statements as to which cost scenarios clearly prohibit abstention.

To accommodate lack of knowledge concerning the exact costs, three-dimensional curves are introduced illustrating the behavior of abstaining classifiers for a variety of cost scenarios. These curves moreover can be used to compare models derived by different machine learning algorithms as well as to combine different abstaining classifiers. Due to relationships between different abstention windows for the same classifier, they can be computed efficiently in time linear in the number of instances in the validation set and linear in their size.

The existence of such efficient algorithms makes it possible to apply the presented methods to a variety of classification problems even if they involve large datasets. In this thesis, these methods are evaluated for EST classification as well as the prediction of carcinogenicity and mutagenicity of chemical compounds. For each of these applications classification accuracy can be improved decisively with the help of abstaining classifiers.

Additionally, abstaining is analyzed in the framework of voting ensembles and theoretical bounds for equal and unequal misclassification costs are obtained based on the PAC-Bayesian theorem. These results are moreover extended to allow different thresholds for positive and negative predictions and concur to a large extent with the empirical results.

Zusammenfassung

Im Gegensatz zu gängigen Klassifikatoren haben sich enthaltende Klassifikatoren die Wahl, ob sie einem Beispiel eine Klassifikation zuordnen oder nicht, um die Klassifikationsgenauigkeit zu verbessern. Unser Interesse an sich enthaltende Klassifikatoren wird motiviert durch Anwendungen bei welchen zuverlässige Vorhersagen nur für einen Teil der Beispiele möglich sind, wie etwa dem Beurteilen von chemischen Risiken und der Vorhersage von toxischen Nebenwirkungen.

Das Ziel dieser Arbeit war, ein geeignetes Verfahren zu definieren, um die Entscheidung zwischen Klassifikation und Enthaltung zu treffen, welches unabhängig von spezifischen Eigenschaften von Algorithmen des maschinellen Lernens bzw. bestimmten Anwendungen ist. Auf diese Weise kann jeder sich nicht enthaltende Klassifikator in einen sich enthaltenden Klassifikator konvertiert durch die Berechnung eines sogenannten optimalen Enthaltungsfensters.

Sich enthaltende Klassifikatoren müssen Verbesserungen in der Vorhersagequalität gegenüber einer geringeren Anwendbarkeit abwägen unter Berücksichtung der Kosten, die mit falschen Vorhersagen bzw. Enthaltungen verbunden sind. Abhängig von der spezifischen Anwendung werden Enthaltungen für gleiche Kostenszenarien mehr oder weniger bevorzugt. Trotz allem können wir eine Aussage darüber treffen, welche Kostenszenarien Enthaltung eindeutig unmöglich machen.

Um mangelndem Wissen über exakte Kosten zu begegnen, werden dreidimensionale Kurven eingeführt, die das Verhalten von sich enthaltenden Klassifikatoren für verschiedenste Kostenszenarien veranschaulichen. Diese Kurven können zudem dazu verwendet werden, um Modelle, die von unterschiedlichen Algorithmen des maschinellen Lernens erzeugt wurden zu vergleichen sowie um verschiedene sich enthaltende Klassifikatoren zu kombinieren. Aufgrund von Beziehungen zwischen Enthaltungsfenstern für denselben Klassifikator können sie zudem effizient in Zeit linear in der Anzahl der Beispiele in der Validierungsmenge und linear in ihrer Größe berechnet werden.

Die Existenz von solchen effizienten Algorithmen ermöglicht es erst, die vorgestellten Methoden für eine Reihen von Klassifikationsproblemen anzuwenden auch wenn diese mit großen Datenmengen verbunden sind. Im Rahmen dieser Arbeit werden diese Methoden für EST Klassifikation und die Vorhersage von Karzinogenizität oder Mutagenizität von chemischen Verbindungen ausgewertet. Für jede dieser Anwendungen kann die Vorhersagegenauigkeit mit Hilfe von sich enthaltenden Klassifikatoren deutlich verbessert werden.

Abschließend wird Enthaltung im Rahmen von sich abstimmenden Ensembles analysiert und theoretische Schranken werden für gleiche und ungleiche Kosten für falsche Klassifikationen auf Basis des PAC-Bayesian Theorems bestimmt. Diese Ergebnisse werden darüber hinaus erweitert um unterschiedliche Schwellwerte für positive und negative Vorhersagen zu ermöglichen und stimmen weitgehend mit den empirischen Resultaten überein.

Contents

Chapter 1

Introduction

The objective of the following sections is to provide an insight into the scope of this thesis as well as its motivation. We first introduce briefly the notion of supervised learning and classification and then move on to justify and formally define the idea of abstaining in classification.

1.1 Classification

Classification involves the task of determining to which element of a finite set of possible classes or categories an object belongs. The choice for a class label is based on previously seen training examples whose class is known. The generalization step beyond the observations is called supervised learning and many learning algorithms have been proposed.

Classification is not only an issue in machine learning and data mining, but is natural to human thinking. Any physician, for example, is presented daily with the task of classifying patients showing different symptoms based on his observations or further tests he may conduct. Additionally, he is able to refer to acquired knowledge about diseases as well as experiences from prior patients. Obviously, for this problem computer programs are clearly insufficient. However, there are many classification tasks for which machine learning algorithms have been used successfully in various application areas. In bioinformatics, such tasks comprise the detection of homology for low sequence similarity [3], tumor classification [11], protein fold recognition [13], the prediction of β-turns in proteins [33] and many more.

To formally define the presented concepts, some terminology has to be introduced first. Throughout this thesis, an example or object is referred to as an *instance* which is described by a set of attributes.

Definition 1.1 (Instance). *An instance x is defined by a k-tuple of the form $(x_1, \ldots, x_k) \in A_1 \times \cdots \times A_k$. The x_i, $1 \leq i \leq k$, are called attributes and A_i, $1 \leq i \leq k$, denotes the set of possible values x_i may assume.*
The instance space is denoted as $\mathcal{X} \subseteq A_1 \times \cdots \times A_k$.

Each A_i may either be a set of discrete values or be continuous. For example, the eye color of a person can be specified by a limited number of terms, whereas his or her body weight is given by continuous values. Each instance may belong to a class or category. We denote the set of possible class labels $\mathcal{Y}$ as a finite set of discrete labels such that $\mathcal{Y} = \{y_1, y_2, \ldots, y_l\}$. A classifier is then defined as follows.

Definition 1.2 (Classifier). *Given an instance space $\mathcal{X}$ and a set of possible class labels $\mathcal{Y} = \{y_1, y_2, \ldots, y_l\}$, a (non-abstaining) classifier labels each instance $x \in \mathcal{X}$ with an element from $\mathcal{Y}$.*

The task of a learning algorithm is to derive a classifier which labels correctly as many instances as possible. For this purpose, the learner is provided with a so-called *training set* which is composed of labeled instances from $\mathcal{X}$. In chapter 6 a range of machine learning algorithms such as decision trees or support vector machines are described in detail. For now it is only relevant to know that such algorithms exist and that they can be used to induce classifiers.

The performance of the resulting classifier or model can be evaluated against a set of instances from $\mathcal{X}$ called *test set*. For an accurate estimate training and test set have to be disjoint, that is no instance should be contained in both sets. Often a third set is required to tune parameters or to compute optimal thresholds, for example. This set is then called *validation set* and should not overlap with both training and test set as well. Note that we can use neither training nor test set for this purpose for very different reasons. For the validation step an accurate estimate of the performance of the classifier is required, however for most classification algorithms the induced classifier performs better on the training data than on the instance space in general. This effect is called overfitting. On the other hand, determining optimal thresholds, for example, involves an additional learning step. If we used the test set for this learning step, accurate estimates of the classifier's performance could no longer be obtained from the test set. For the methods presented later a validation set is indeed necessary.

1.2 Abstaining in Classification

Having described the concept of classification, the notion of abstaining appears to be counter-intuitive at first. After all, the objective of classification is to come up with a labeling for an instance and not to refrain from doing so. Yet, in every aspect of human life abstaining plays a central role. A physician confronted with unusual and ambiguous symptoms may refer the patient to a specialist instead of giving an unsafe diagnosis. During elections, a large fraction of eligible voters prefer not casting their vote to voting a candidate they find unacceptable. Indeed, most people are hesitant in choosing between two equally unattractive alternatives and if they are forced to do so all the same cannot reason their choice properly in most cases.

Although abstaining is a common phenomenon, it is rarely applied in machine learning, because the choice to abstain is often based on a variety of factors which are difficult to quantify. There are several problems associated when introducing abstention to machine learning. To understand these problems consider the following example.

In a far away country, the population is offered two oracles to turn to for advice. The first oracle always gives an answer to any question no matter how confident it is about the correct answer. The second one on the other hand has the possibility to shrug its shoulders – metaphorically speaking – and offer a "don't know" instead of an unsafe advice. Consequently, the question arises which of the oracles people trust in and whose counsel they tend to seek accordingly. Generally, this depends on how often the advice given by an oracle is correct. There are two possible reasons for the first oracle's answering of every question. Either the oracle is omniscient and actually knows every answer or – more likely – it is unable to admit that there exists something it does not know anything about. In the second case its advice fails in many cases and as people are unable to distinguish between advice given from sound knowledge or reckless ignorance, they start turning to the second oracle, which may not always give an answer but if it does, the answer is helpful.

As a strategy to win people back, the first oracle now decides to specify for each advice how confident it is that it will work, so that people can decide themselves if they follow this specific advice. However, this approach is also flawed. The confidence values depend strongly on the "ego" of the oracle, that is how strongly it believes in itself. Some oracles may be shy and insecure and thus do not dare to claim the correctness of their advice even if they are good, whilst others boast about their omniscience. As a consequence, people have to learn from their experience and the experience of others when to believe the oracle and when not.

To compete with the first oracle, the second one on the other hand may choose to give advice only if it feels absolutely safe in its decree and feign ignorance the rest of the time. Unfortunately, this can have the opposite effect to what is intended. Instead of rushing in masses to the second oracle, people might turn their back on it because they hardly ever actually get any advice from it at all. To prevent this from happening, the oracle has to find the right balance between the two extremes.

This example illustrates the various issues involved when extending the common classification model to handle abstention. Obviously, an abstaining classifier is superior to a classifier which labels instances with "brute force" no matter how inappropriate it may be. However, there is a trade-off between abstention frequency and prediction accuracy. Accuracy is defined as the number of instances classified correctly divided by the total number of instances classified at all. If conducted adequately, abstention improves the performance of a classifier but on the other hand reduces the number of instances it can be applied to. The importance attached to each of these aspects determines which direction an abstaining classifiers leans to. Additionally, there is a connection between classifiers which supply confidence values for their predictions and abstaining classifiers. In fact, any classifier of the first type can be converted into an abstaining classifier by a separate learning step. Accordingly, we can distinguish between two types of abstaining classifiers subject to if abstaining is an integral part of the model or involves a separate step. This is worked out in detail in the next chapter.

Most classification tasks allow abstention in some way or another. Nevertheless there exist areas for which it is forbidden. In a criminal trial, for example, the possible outcomes can always only be "guilty" or "not guilty", but never "don't know". In bioinformatics, there are several important fields of study which can benefit from abstention and in which it is already used, albeit rather informally in most cases. For example, if the function of a newly determined gene cannot be ascertained, it is not assigned some arbitrary function but instead labeled with a variation of "unknown function".

Having motivated the use of abstaining classifiers sufficiently, we can eventually proceed to define them formally. The definition of a traditional (non-abstaining) classifier thereby serves as a prototype and we introduce a new label $\perp$ to denote the choice to abstain.

Definition 1.3 (Abstaining Classifier). *Given an instance space $\mathcal{X}$ and a set of possible class labels $\mathcal{Y} = \{y_1, y_2, \ldots, y_l\}$, an abstaining classifier is defined as a classifier which labels* *an instance $x \in \mathcal{X}$ with an element from $\mathcal{Y} \cup \{\perp\}$.*

For the remainder of this thesis, we restrict ourselves to two-class problems, that is classification tasks which involve only two categories of instances. Two-class problems can be described as *concept learning* tasks. A *concept* is defined as function $c : \mathcal{X} \rightarrow \{0, 1\}$ such that for all instances $x \in \mathcal{X}$ corresponding to that concept $c(x) = 1$ and for all others $c(x) = 0$. The instances corresponding to the concept are called positive instances whereas the remaining ones are called negative. The set of possible class labels then becomes $\mathcal{Y} = \{P, N\}$. To prevent misunderstandings capital letters are used for the actual class and small ones for the prediction of a classifier.

1.3 Related Work

1.3.1 Cost-Sensitive Active Classifiers

Active classifiers differ from so-called passive classifiers by being allowed to demand values of not specified attributes before tying themselves down to a class label. The request for further attributes corresponding to tests is determined by the costs associated with those tests compared to the costs of misclassifications. Although this idea is not new and has been explored in different frameworks, the task of learning active classifiers has always been addressed by first learning the underlying concept and only afterwards finding the best active classifier. Greiner *et al.* [24] propose to consider the problems of learning and active classification jointly instead of in two separate steps. They show that learning active classifiers can be done efficiently if the learner may only ask for a constant number of additional tests, but in general is often intractable.

Their notation deviates slightly from our previous definitions. For the sake of continuity, it is modified to fit in our setting. All attributes are presumed to be binary, that is $A_i = \{0, 1\}$ for $1 \leq i \leq k$. A concept is regarded as an indicator function $c : \mathcal{X} \rightarrow \{0, 1\}$, so that an instance is positive if it belongs to the underlying concept and negative otherwise. The set of possible concepts is defined as $\mathcal{C} = \{c_i\}$ and a labeled instance is given as a pair $(x, c(x))$. Furthermore a stationary distribution $P : \mathcal{X} \rightarrow [0, 1]$ over the space of instances is assumed. Instances for both training and test set are drawn randomly according to P.

Initially, either no attributes (*empty blocking*) or a subset (*arbitrary blocking*) of the instance's attributes are revealed for free. Fur any further attribute values a price has to be payed by the classifier. Accordingly, the classifier can choose at any point to output a prediction or obtain further tests at the costs associated. This leads to a recursive procedure. The quality of an active classifier is determined by the *expected cost* of the active classifier on an instance. The value of expected cost is also determined recursively.

The class of all possible active classifiers is denoted as $\mathcal{A}^{all}$ and the set of active classifiers considered may be reduced to a particular subset $\mathcal{A} \subseteq \mathcal{A}^{all}$. The concept c, the set of possible active classifiers $\mathcal{A}$ and the distribution P then determine the optimal active classifier. This results in an optimization problem which is tractable, for example, if the number of additional tests the classifier can request is limited, but in general can be *NP-hard*. Instead of the separate optimization step, directly computing the active classifier is proposed by Greiner *et al.*, without learning the full concept or the complete distribution. They introduce an algorithm which allows to learn active classifiers in $\mathcal{A}^l$ (classifiers which ask at most for

l additional attributes) for any concept class, any distribution and any blocking process in polynomial time. Contrary to that, the problem of learning classifiers in $\mathcal{A}^{\approx l}$ (active classifiers, which ask for at most l further attributes *on average*) still is *NP-hard*.

Although active classification does not lead to any abstention – every instance is classified once no further tests are to be performed – parallels to abstaining can be drawn. The choice to abstain on an instance in most cases entails more extensive tests as well. For example, if a physician is unable to tell the source of a patient's problems from the symptoms only, additional tests are mandatory. These may be blood tests or an electrocardiogram or any other of a range of possible medical tests. The nature of these tests is not specified any further in our framework, but may be by combining active classification and abstaining.

1.3.2 Abstaining in Rule Learning

Although standard rule learning approaches have been applied successfully in practice, there are few theoretical results concerning their predictive performance. Rückert and Kramer [44] introduce a framework for learning ensembles of rule sets whose expected error can be bounded theoretically and which relies on a greedy hill-climbing approach (*stochastic local search* (SLS), [31]).

Ensembles essentially are sets of classifiers and in this case the individual classifiers are composed of several rules. The final prediction of the ensemble results from a voting among its members depending on their accuracy on the training set. Therefore, a separate probability distribution Q_i over the set of all y_i-labeled rule sets r_i is calculated for each class label $y_i \in \mathcal{Y}$. The prediction result for an instance x is given as $c_V(\bar{Q}, x) = \mathrm{argmax}_{y_i \in \mathcal{Y}}\, c(Q_i, x)$, with $\bar{Q} := (Q_1, \ldots, Q_{|\mathcal{Y}|})$ and $c(Q_i, x) := \mathbf{E}_{r_i \sim Q_i}[r_i(x)]$. In the two-class case this leads to the following decision rule:

$$c_V(\bar{Q}, x) = \begin{cases} y_1 & \text{if } c(Q_1, x) - c(Q_2, x) \geq 0 \\ y_2 & \text{if } c(Q_1, x) - c(Q_2, x) < 0. \end{cases}$$

Obviously, the value of $|c(Q_1, x) - c(Q_2, x)|$ indicates the certainty of the corresponding prediction as $c(Q_i, x)$ can be regarded as a score for class y_i. This allows a simple extension of the rule learning framework by introducing a threshold θ such that instances are only classified if $|c(Q_1, x) - c(Q_2, x)| \geq \theta$. Thus, the above equation becomes

$$c_V(\bar{Q}, x) = \begin{cases} y_1 & \text{if } c(Q_1, x) - c(Q_2, x) \geq \theta \\ \perp & \text{if } -\theta < c(Q_1, x) - c(Q_2, x) < \theta \\ y_2 & \text{if } c(Q_1, x) - c(Q_2, x) \leq -\theta. \end{cases}$$

In order to derive a theoretical bound on the classification error of ensembles of rule sets the PAC-Bayesian theorem [35] is used. This bound is improved additionally by admitting abstention. In chapter 7, we use a similar approach to determine a bound on the *expected cost* of abstaining voting classifiers for ensembles in general.

1.3.3 Cautious and Delegating Classifiers

Cautious classifiers were introduced by Ferri and Hernández-Orallo [19] similar to our definition of abstaining classifiers. A cautious classifier extends the set of original classes C by an additional class “unknown” or $\perp$, which results in new set C'. Consequently the cautious classifier is described as a function from the instance space to C'.

The authors propose various measures of performance for cautious classifiers based on the confusion matrix. For this purpose, they distinguish between a cost-insensitive context and a cost-sensitive one. In the first context, standard performance measures are extended to accommodate abstention and two additional measures – *efficacy* and *capacity* – are defined. Both of these measures are motivated as areas in two-dimensional curves which either plot *accuracy* (fraction of correctly classified instances among those actually classified) against *abstention* (fraction of instances abstained on) for *efficacy* or *error* (portion of misclassified instances) against a parameter α of a parameterized form of cautious classifiers. For a more extensive explanation we refer to [19].

Furthermore, several approaches are presented to convert probabilistic classifiers into cautious classifiers by imposing thresholds on the class probabilities which specify when to abstain. One of these approaches relies on windows whose size may vary but is the same for all classes and class biases which influence the degree of abstention for each class. By increasing the window size and consequently increasing the amount of abstention for a fixed class bias, a two-dimensional accuracy-abstention curve can be created which illustrates the behavior of the classifier for changing abstention rates.

Alternatively, costs of cautious classifiers can be calculated by multiplying the confusion matrix with a cost matrix. The cost can be plotted against abstention instead of error or accuracy. For unknown costs, the authors suggest extending so-called *receiver operating curves* in order to describe visually the behavior of a cautious classifier. This approach is explored in detail in chapter 3.

Cautious classifiers as such do not specify how to proceed with abstained instances. Ferri *et al.* [18] describe an approach which refers the abstained instances to a second classifier. This process is called delegating and only the first classifier is trained on the complete training set whereas the subsequent classifier is trained only on instances delegated to it. The next step for any instance then can either be classification, another delegation to a successor classifier or a referral back to the original model. The threshold for delegating is chosen such that at least a fixed proportion of instances is not delegated.

In our framework, we use a similar approach as Ferri and Hernández-Orallo to create abstaining classifiers from non-abstaining classifiers providing confidence values for their predictions. The abstaining classifiers are also specified by thresholds, however their performance is evaluated for the most part in terms of expected cost not accuracy or error rate. Furthermore instead of fixed window size and class biases an optimization step is used to determine the optimal thresholds between classification and abstention. To illustrate the behavior of abstaining classifiers for unknown costs three-dimensional cost curves are used which plot the expected cost of classifiers against costs for misclassifications and abstention.

1.4 Outline of the Thesis

The objective of this thesis is to show that abstaining can be of benefit in a machine learning context as well as to describe a method to construct abstaining classifiers independent of specific machine learning algorithms or applications and to choose among a set of similarly derived ones.

In chapter 2 we introduce the model of abstaining classifiers as it is considered for the rest of this thesis. Any classification model can be converted into an abstaining classifier provided that it calculates confidence values for its predictions. In fact, a large set of abstaining classifiers can be created with this approach. Which of these classifiers is optimal for a specific task is determined by the costs expected if it was applied to a randomly drawn instance. In this context we review the notion of costs in machine learning applications and the characteristics of expected cost. Based on normalization of expected cost, a sequence of increasingly strict conditions which are necessary to allow abstention is imposed upon cost matrices.

If the exact costs and class distributions are specified, computing the best abstaining classifier is straightforward. Unfortunately, for many problems costs are either known only approximately or not at all, rendering the determination of the optimal classifier impossible. To circumvent this problem, three-dimensional curves are introduced in chapter 3, which visualize the behavior of a given classifier for a variety of cost scenarios and class distributions. For this purpose, existing visualization techniques are extended which tackle the same problem for non-abstaining classifiers. These are ROC curves (see e.g. [39]) and cost curves [14]. Additionally a new type of cost curves is introduced which is easier to analyze for fixed class distributions and otherwise equivalent to the original type of cost curves.

Up to this point, only individual abstaining classifiers are considered. Chapter 4 revolves around the question of how to combine several abstaining classifiers produced by different models to obtain higher-level abstaining classifiers. Two methods are presented, one of which takes a vote among individual classifiers weighted by their expected cost. The second one utilizes a separate-and-conquer approach to obtain a sequence of classifiers to be applied one after the other.

As the usability of abstaining classifiers and cost curves is strongly determined by the computational effort necessary to derive them, we present two algorithms in chapter 5 for efficiently computing cost curves and optimal abstaining classifiers. The first one adopts a dynamic programming approach in combination with bounds on expected costs to calculate the optimal classifier for each cost scenario from a subset of possible classifiers. The second method relies on an algorithm for directly computing the optimal abstaining classifier in linear time and uses further information about optimal classifiers for related cost scenarios. In this context, several important characteristics of optimal abstaining classifiers are described which greatly reduce the running time of both algorithms.

In the next chapter abstaining classifiers are evaluated on two classification tasks which involve three separate data sets. These tasks include the prediction of carcinogenicity and mutagenicity respectively of chemical compounds based on occurrences of molecular fragments and the classification of EST sequences from mixed plant-pathogen EST pools based on codon bias. We show that the predictive accuracy can be improved by abstaining from unsafe predictions and analyze the characteristics of abstained instances. Furthermore, the different types of cost curves are used to compare different classification algorithms with regard to their performance in mutagenicity prediction and to analyze the relationship between optimal abstention rate and false positive or false negative rate as well as the dependency between abstention rate and accuracy. Last but not least, the performance of higher-level abstaining classifiers is examined.

In chapter 7, we focus on ensembles of classifiers in a framework similar to the one described above for rule learning. These ensembles are allowed to abstain depending on the agreement between the individual classifiers. Instead of bounding the expected error, the expected cost is bounded using the PAC-Bayesian theorem and formulas are derived to directly compute the optimal threshold for abstaining. Equal and unequal misclassification costs are distinguished for this purpose.

In the final chapter the results are summarized and possible starting points for further studies are presented.

Chapter 2

Abstaining in a Cost-Sensitive Context

So far we have defined abstaining classifiers only informally as classifiers which may or may not classify an instance. In this chapter, we pose and answer more detailed questions concerning the nature and characteristics of such abstaining classifiers.

2.1 Abstention Windows

Definition 1.3 states only the most basic characteristic of an abstaining classifier which is the option to abstain, but does not provide a specification as to how the classifier decides to abstain. In principle several methods are conceivable. For example, a classifier which chooses randomly between abstaining and classifying an instance would also qualify as an abstaining classifier. But as we want to improve the performance of a classifier with regard to prediction accuracy or any other performance measure by abstaining we require a more sophisticated decision process which is based on specific properties of instances. One way to achieve this is to design a machine learning algorithm specifically for this purpose, which learns an extra class or which abstains if certain tests are unsuccessful. In [22], for example, a classification system for EST sequences is presented which may abstain if no reading frame of a sequence is classified to be coding. The alternative approach consists of adding a separate step to the learning procedure which is independent of any specific machine learning algorithm and therefore yields a meta-classification scheme.

Such a meta-classification scheme rests its decision to abstain upon the confidence the underlying base classifier has in a prediction. It uses the fact that most machine learning algorithms do not only output a class prediction for an instance but also produce scores associated with the predictions. These can be class probabilities as for Naive Bayes (see page 77) or the distance from a separating hyperplane as for SVMs (see page 76). The difference between the scores for the two available classes then implies the degree of certainty of the prediction.

Definition 2.1 (Margin). *Let $s_p(x)$ the score for a positive prediction of instance $x \in \mathcal{X}$, and $s_n(x)$ the score for a negative prediction. The margin of this instance is defined as $m(x) := s_p(x) - s_n(x)$.*

An instance x is labeled positive if $m(x)$ is positive and negative otherwise. The margin of an instance cannot only be used to determine the prediction for this instance, but also the reliability of this prediction. If the absolute value of the margin is large, we can trust the prediction with higher confidence than for small values. Table 2.1 shows an example dataset with predicted class probabilities. Obviously when using class probabilities in the two-class case, the margin is strictly monotone in the probability for the positive class. Nevertheless, as not all classifiers necessarily produce probabilities, we employ the term margin to avoid confusion.

In the presented example we notice that instances x_6 and x_7 are correctly classified with high confidence, whereas the misclassified instances x_3 and x_8 have small absolute margin values. If we classify all instances in the dataset an accuracy of 60% is achieved. Yet, by restricting the instances to be classified to those x_i for which $m(x_i) \leq -0.25$ or $m(x_i) \geq 0.15$, we can increase classification accuracy to 75%. This intuitive example gives rise to the idea of an abstention window, such that instances are abstained on if they fall within this abstention window and classified otherwise. The term abstention window has already been used by Ferri and Hernández-Orallo [19] and Ferri *et al.* [18], but no formal definition has been given. We use the following definition.

Definition 2.2 (Abstention Window). *An abstention window a is defined as a pair (l, u) such that the prediction of a on an instance $x \in \mathcal{X}$ is given by*

$$\pi(a, x) = \begin{cases} p & \text{if } m(x) \geq u \\ \perp & \text{if } l < m(x) < u \\ n & \text{if } m(x) \leq l. \end{cases}$$

Learning abstaining classifiers involves two separate learning steps. First, a non-abstaining classifier is learned from a training set and then this classifier is applied to a second validation set which results in margins for the instances of the validation set. These margins are then used to calculate optimal thresholds for positive and negative classification, i.e. the optimal abstention window. Optimality for an abstention window is specified in the following sections. Note that we have to use a separate validation set instead of either training or test set to determine the optimal abstention window because this calculation involves an additional learning step. This was explained more extensively on page 2.

For a given classifier Cl (induced by a machine learning algorithm), there are a large number of possible abstention windows. In fact, the set of possible abstention windows is uncountably infinite as both the upper and lower threshold are real numbers. However, for practical purposes the number of abstention windows considered has to be limited. When

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
$s_p(x_i)$	0.6	0.3	0.55	0.2	0.35	0.9	0.05	0.4	0.7	0.65
$s_n(x_i)$	0.4	0.7	0.45	0.8	0.65	0.1	0.95	0.6	0.3	0.35
$m(x_i)$	0.2	-0.4	0.1	-0.6	-0.3	0.8	-0.9	-0.2	0.4	0.3
y_i	P	N	N	N	P	P	N	P	P	N

Table 2.1: Examples for class probabilities on a sample $S \subseteq \mathcal{X}$, with five positive and five negative instances. y_i denotes the class label of instance x_i. Based on the predicted class probabilities six instances would be classified correctly and four would be misclassified.

comparing different abstention windows for their behavior on the validation set, we observe that there are sets of abstention windows which behave in the same way on the validation set for any performance measure. In the previous example, the abstention window $(-0.25, 0.15)$ resulted in a prediction accuracy of 75% on the validation set. However, the abstention windows $(-0.21, 0.11)$, $(-0.25, 0.19)$ and $(-0.29, 0.15)$ exhibit the same prediction accuracy. Again we could produce an infinite number of abstention windows such that all of them show this property. When trying to find the best abstention window in terms of classification accuracy or any other measure, the problem arises which of these to choose. In principle, any of these windows can be chosen but the most reasonable choice appears to be the abstention window for which the thresholds lie exactly between two neighboring margin values. Consequently, we define the set of abstention windows as follows.

Definition 2.3. *Let Cl be a given classifier and $S = \{x_1, \dots, x_n\} \subseteq \mathcal{X}$ a validation set. Let $M = \big\{m(x_1), \dots, m(x_n)\big\}$ be the margins obtained by applying Cl on S and w.l.o.g. $m(x_1) \leq \cdots \leq m(x_n)$. Let $\varepsilon > 0$ be an arbitrary but constant value. Then the set of abstention windows for classifier Cl is defined as*

$$\begin{aligned}\mathcal{A}(Cl) =& \Big\{(l,u) | \exists\, 1 \leq j < n : l = m(x_1) - \varepsilon \wedge m(x_j) \neq m(x_{j+1}) \wedge u = \frac{m(x_j) + m(x_{j+1})}{2}\Big\} \\ &\cup \Big\{(l,u) | \exists\, 1 \leq j < n : m(x_j) \neq m(x_{j+1}) \wedge l = \frac{m(x_j) + m(x_{j+1})}{2} \wedge u = m(x_n) + \varepsilon\Big\} \\ &\cup \Big\{(l,u) | \exists\, 1 \leq j \leq k < n : m(x_j) \neq m(x_{j+1}) \wedge l = \frac{m(x_j) + m(x_{j+1})}{2} \\ &\qquad\qquad \wedge\, m(x_k) \neq m(x_{k+1}) \wedge u = \frac{m(x_k) + m(x_{k+1})}{2}\Big\}.\end{aligned}$$

$\mathcal{A}(Cl)$ consists of three subsets. The first subset comprises all abstention windows which have the lower threshold below the smallest margin value and a variable upper threshold. The second one contains the windows with variable lower threshold and upper threshold above the largest margin value. Abstention windows with both variable lower and upper threshold are included in the third subset. The value of ε determines the difference between the smallest margin value and lowest possible threshold and between the largest margin value and highest possible threshold and can be assigned by the user.

When only one classifier is considered as for this chapter, the abbreviation $\mathcal{A}$ is used for $\mathcal{A}(Cl)$. The question of how to compute the optimal abstention window efficiently is resolved later in chapter 5. But first we have to define when exactly an abstention window is optimal.

2.2 Abstaining Classifiers and Expected Cost

There are several ways to define the optimality of an abstention window. One approach would be to select the window with lowest error rate (i.e. lowest rate of misclassifications) or highest accuracy. However, as error rate is negatively correlated to window width, the optimal abstention window would always be the one which abstains on all instances. For this reason, a performance measure is desired which has two components, one of these rewarding decreasing misclassification probabilities and the other one penalizing increasing abstention probabilities. The weight of each component depends on the costs associated with the corresponding events.

2.2.1 Costs in Supervised Learning

There are many types of costs in supervised learning. A variety of those is listed by Turney [51]. Costs can be associated with misclassifications, with tests (i.e. attributes or measurements) or teachers (i.e. abstaining on an instance and referring it to an expert) and many, many more. They can be constant or conditional, that is depend on the individual instance. In general, the term cost is defined abstractly and independent of specific units of measurement. Such units might be monetary or itself be abstract as e.g. health or quality of life. We restrict ourselves to two types of costs: costs for correct and wrong classifications and costs for abstaining. We also assume that these costs are constant, which means invariable over time and for instances of the same class.

Costs for the different events in general differ greatly from each other and are also measured in different units as we see when returning to the introductory example. If a physician classifies a healthy person to be sick, this results in an unnecessary treatment which may or may not be damaging to the person's health. Here we have a combination of monetary costs for the treatment and costs concerning health or life quality due to stress caused by a wrong diagnosis. On the other hand, not treating a sick person has much more severe consequences depending on the disease which makes a simple two-class scenario inappropriate. Costs for abstaining, on the contrary, are determined by further tests necessary to diagnose or to exclude an illness. By comparing predictions with actual classes, costs associated with certain events can be given in the form of a matrix.

Definition 2.4 (Cost Matrix). *Costs for correct classification, misclassification and abstention are given by a cost matrix C which is defined by the following table* [†]

	Predicted Class		
True Class	p	n	$\perp$
P	$C(P, p)$	$C(P, n)$	$C(P, \perp)$
N	$C(N, p)$	$C(N, n)$	$C(N, \perp)$.

2.2.2 Expected Cost

Based on the cost matrix we can define the expected cost of an abstention window provided that we know the probabilities associated with each possible event. These probabilities have to be estimated by applying the abstention window to a validation set. The estimations are based on the number of times a positive or negative instance is classified positive or negative or abstained on. We use the following terms to denote these counts.

Definition 2.5. *Let $S = \{x_1, \ldots, x_n\} \subseteq \mathcal{X}$ be the validation set, $\{y_1, \ldots, y_n\}$ the corresponding class labels and $M = \{m(x_1), \ldots, m(x_n)\}$ be the set of margins computed on S by a given*

[†] This is based on the assumption that instances are strictly separated into two classes P and N which are completely disjoint.

classifier Cl. If $a = (l, u)$ is an abstention window, then we introduce the notation

$$\begin{aligned} TP(a) &:= \sum_{\substack{1\leq i\leq n \\ y_i=P}} \delta(m(x_i) \geq u) & TN(a) &:= \sum_{\substack{1\leq i\leq n \\ y_i=N}} \delta(m(x_i) \leq l) \\ FN(a) &:= \sum_{\substack{1\leq i\leq n \\ y_i=P}} \delta(m(x_i) \leq l) & FP(a) &:= \sum_{\substack{1\leq i\leq n \\ y_i=N}} \delta(m(x_i) \geq u) \\ UP(a) &:= \sum_{\substack{1\leq i\leq n \\ y_i=P}} \delta(l < m(x_i) < u) & UN(a) &:= \sum_{\substack{1\leq i\leq n \\ y_i=N}} \delta(l < m(x_i) < u) \end{aligned}$$

for its true positives, true negatives, false negatives, false positives, unclassified positives and unclassified negatives. $\delta(F) = 1$ if F is true and $\delta(F) = 0$ otherwise.

From these counts we obtain values for the frequencies or rates of true and false positive or negative predictions and abstention. These rates provide a good estimation of the conditional probabilities given the validation set size is large enough. For this reason, rates and (empirical) probabilities from now on are used synonymously.

Definition 2.6. *Let a be an abstention window defined as before. We introduce the following notation*

$$\begin{aligned} P(p|P) = TPR(a) &= \frac{TP(a)}{FN(a) + TP(a) + UP(a)} && \textit{true positive rate} \\ P(n|P) = FNR(a) &= \frac{FN(a)}{FN(a) + TP(a) + UP(a)} && \textit{false negative rate} \\ P(\perp|P) = PAR(a) &= \frac{UP(a)}{FN(a) + TP(a) + UP(a)} && \textit{positive abstention rate} \\ P(n|N) = TNR(a) &= \frac{TN(a)}{FP(a) + TN(a) + UN(a)} && \textit{true negative rate} \\ P(p|N) = FPR(a) &= \frac{FP(a)}{FP(a) + TN(a) + UN(a)} && \textit{false positive rate} \\ P(\perp|N) = NAR(a) &= \frac{UN(a)}{FP(a) + TN(a) + UN(a)} && \textit{negative abstention rate} \end{aligned}$$

In this definition, we distinguish between the abstention probability for negative and positive instances. The abstention rate, i.e. probability of abstaining on any instance, can only be estimated correctly from the validation set if the class distribution within the validation set corresponds to the actual class distribution observed on the complete instance space $\mathcal{X}$. The expected cost is calculated by summing up for each event the product of the cost and the probability that this event occurs. If the class distribution in the validation set represents the underlying class distribution of the instance space, the expected cost can be more easily computed by summing up the cost for each instance in the validation set and then dividing by the total number of instances.

Definition 2.7 (Expected Cost). *Let a be an abstention window and C the cost matrix. The expected cost of this abstention window is defined as follows.*

$$\begin{aligned}\mathbf{EC}(C,a) :=& P(P)\Big[P(p|P)\,C(P,\,p) + P(n|P)\,C(P,\,n) + P(\perp|P)\,C(P,\,\perp)\Big]\\ &+ P(N)\Big[P(n|N)\,C(N,\,n) + P(p|N)\,C(N,\,p) + P(\perp|N)\,C(N,\,\perp)\Big]\\ =& P(P)\Big[TPR(a)\,C(P,\,p) + FNR(a)\,C(P,\,n) + PAR(a)\,C(P,\,\perp)\Big]\\ &+ P(N)\Big[TNR(a)\,C(N,\,n) + FPR(a)\,C(N,\,p) + NAR(a)\,C(N,\,\perp)\Big]\end{aligned}$$

Alternatively, we have

$$\begin{aligned}\mathbf{EC}(C,a) := \frac{1}{n}\big[&TP(a)\,C(P,\,p) + FN(a)\,C(P,\,n) + UP(a)\,C(P,\,\perp)\\ &+ TN(a)\,C(N,\,n) + FP(a)\,C(N,\,p) + UN(a)\,C(N,\,\perp)\big].\end{aligned}$$

Abstaining is rewarded by decreasing values for false positive and false negative rate and penalized by decreasing correct classification rates and increasing abstention rates. The optimal abstention window is formally defined as the one with minimal expected cost.

Definition 2.8 (Optimal Abstention Window). *Let $\mathcal{A}$ be the set of possible abstention windows on the validation set and C the cost matrix. The optimal abstention window a_{opt} is defined as*

$$a_{opt} := \operatorname*{argmin}_{a\in\mathcal{A}} \mathbf{EC}(C,a).$$

2.2.3 Costs for Correct Classifications

So far, we have associated classification costs even with correct classifications, which of course is reasonable as we can create scenarios for which this is the case. As an example, consider a charity organization which sends out letters asking for contributions to their projects. Naturally, they want to address only those people likely to respond. The cost of not addressing a potential donor is given by the loss of a donation. Sending a letter to a donor however also costs a certain amount for the posting. Thus, a correct classification of a donor still requires money, whereas the correct classification of a non-donor in fact costs nothing. This implies that costs for correct classifications have to be taken into consideration. Yet, for our purposes as few degrees of freedom – costs to be regarded – as possible are to be desired. Here we benefit from the fact that any cost matrix having non-zero costs for correct classifications can be transformed into a matrix which does not count correct classification, but is still equivalent in every respect to the original matrix.

Lemma 2.9. *Given a cost matrix C with $C(P,\,p) \neq 0$ and $C(N,\,n) \neq 0$. Let a_1 and a_2 be any two abstention windows with $\mathbf{EC}(C,a_1) < \mathbf{EC}(C,a_2)$, then there exists a cost matrix C' with $\mathbf{EC}(C',a_1) < \mathbf{EC}(C',a_2)$ and $C'(P,\,p) = 0$ and $C'(N,\,n) = 0$.*

Proof. We observe that for $i \in \{1,2\}$

$$\begin{aligned}\mathbf{EC}(C, a_i) = P(P)\Big[&\big(1 - FNR(a_i) - PAR(a_i)\big)\, C(P,\, p) \\ &\qquad + FNR(a_i)\, C(P,\, n) + PAR(a_i)\, C(P,\, \perp)\Big] \\ + P(N)\Big[&\big(1 - FPR(a_i) - NAR(a_i)\big)\, C(N,\, n) \\ &\qquad + FPR(a_i)\, C(N,\, p) + NAR(a_i)\, C(N,\, \perp)\Big] \\ = P(P)\, &FNR(a_i)\, (C(P,\, n) - C(P,\, p)) + P(P)\, PAR(a_i)\, (C(P,\, \perp) - C(P,\, p)) \\ + P(N)\, &FPR(a_i)\, (C(N,\, p) - C(N,\, n)) + P(N)\, NAR(a_i)\, (C(N,\, \perp) - C(N,\, n)) \\ + P(P)\, &C(P,\, p) + P(N)\, C(N,\, n) \qquad (2.1)\end{aligned}$$

Now set $C'(P,\, y) = C(P,\, y) - C(P,\, p)$ and $C'(N,\, y) = C(N,\, y) - C(N,\, n)$ for $y \in \{p, n, \perp\}$. Obviously, we have that $C'(P,\, p) = C'(N,\, n) = 0$ and from equation (2.1) it follows that

$$\mathbf{EC}(C, a_i) = \mathbf{EC}(C', a_i) + P(P)\, C(P,\, p) + P(N)\, C(N,\, n). \qquad (2.2)$$

From the definition of a_1 and a_2 and equation (2.2) we then get that

$$\mathbf{EC}(C', a_1) + P(P)\, C(P,\, p) + P(N)\, C(N,\, n) < \mathbf{EC}(C', a_2) + P(P)\, C(P,\, p) + P(N)\, C(N,\, n)$$

Thus, we have $\mathbf{EC}(C', a_1) < \mathbf{EC}(C', a_2)$. □

The lemma also indicates how a cost matrix can be transformed to obtain zero costs for correct classifications without changing the outcome of any comparison between abstention windows. Although the expected cost $\mathbf{EC}(C', a)$ of an abstention a for the new cost matrix differs from the expected cost for the original cost matrix $\mathbf{EC}(C, a)$, the difference between $\mathbf{EC}(C', a)$ and $\mathbf{EC}(C, a)$ is the same for every abstention window. Therefore, after having computed the optimal abstention window a_{opt} and $\mathbf{EC}(C', a_{opt})$, $\mathbf{EC}(C, a_{opt})$ can be computed easily as $\mathbf{EC}(C', a_{opt}) + P(P)\, C(P,\, p) + P(N)\, C(N,\, n)$.

2.2.4 Relationship between Costs and Class Distributions

Previously we have given two definitions for expected cost, one of which is only applicable when the validation set has been sampled based on the underlying class distribution within the instance space. The following lemmas allow us to use the alternative definition, which is more intuitive to compute, even if the distribution of classes within the validation set differs from the true distribution. We now assume that $C(P,\, p) = C(N,\, n) = 0$, which is completely legitimate because of lemma 2.9.

Lemma 2.10. *Let C be a cost matrix and $P(P)$ and $P(N)$ be the true class distribution. If we have a different class distribution given by $P'(P)$ and $P'(N)$, we can create a cost matrix C', such that for any abstention window $a \in \mathcal{A}$, we have that $\mathbf{EC}(C, a) = \mathbf{EC}(C', a)$ with $\mathbf{EC}(C, a)$ the expected cost of a for $P(P)$ and $P(N)$ and cost matrix C and $\mathbf{EC}(C', a)$ the expected cost for $P'(P)$, $P'(N)$ and C'.*

Proof. Set $C'(P, y) = \frac{P(P)}{P'(P)} C(P, y)$ for $y \in \{n, \perp\}$ and $C'(N, y) = \frac{P(N)}{P'(N)} C(N, y)$ for $y \in \{p, \perp\}$. Then we have that

$$\begin{aligned}\mathbf{EC}(C', a) =& P'(P)\Big[FNR(a)\, C'(P, n) + PAR(a)\, C'(P, \perp)\Big] \\ &+ P'(N)\Big[FPR(a)\, C'(N, p) + NAR(a)\, C'(N, \perp)\Big] \\ =& P'(P)\Big[FNR(a)\, \tfrac{P(P)}{P'(P)}\, C(P, n) + PAR(a)\, \tfrac{P(P)}{P'(P)}\, C(P, \perp)\Big] \\ &+ P'(N)\Big[FPR(a)\, \tfrac{P(N)}{P'(N)}\, C(N, p) + NAR(a)\, \tfrac{P(N)}{P'(N)}\, C(N, \perp)\Big] \\ =& P(P)\Big[FNR(a)\, C(P, n) + PAR(a)\, C(P, \perp)\Big] \\ &+ P(N)\Big[FPR(a)\, C(N, p) + NAR(a)\, C(N, \perp)\Big] = \mathbf{EC}(C, a).\end{aligned}$$

□

Therefore, by changing the cost matrix appropriately we can compute the expected cost for any class distribution different from the true class distribution, and still get the correct result. In particular, this is correct for the class distribution in the validation set S. As a consequence, the expected cost can be calculated directly from the validation set by summing up the costs over all instances and then dividing by the total number of instances.

Corollary 2.11. *Let C the true cost matrix and $P(P)$ and $P(N)$ be the true class distributions. Let $S \subseteq \mathcal{X}$. There exists a cost matrix C' such that we can compute the expected cost of any abstention window $a \in \mathcal{A}$ for C, $P(P)$ and $P(N)$ by computing the average cost on instances of S using C'.*

Proof. Let $P'(P)$ and $P'(N)$ the class frequencies in S. Lemma 2.10 implies that we can construct a new cost matrix C', such that $\mathbf{EC}(C, a) = \mathbf{EC}(C', a)$ for any abstention window a. ($\mathbf{EC}(C, a)$ is calculated using $P(P)$ and $P(N)$ and $\mathbf{EC}(C', a)$ using $P'(P)$ and $P'(N)$.) Additionally, we have that

$$\begin{aligned}\mathbf{EC}(C', a) =& P'(P)\, FNR(a)\, C'(P, n) + P'(N)\, FPR(a)\, C'(N, p) \\ &+ P'(P)\, PAR(a)\, C'(P, \perp) + P'(N)\, NAR(a)\, C'(N, \perp) \\ =& \frac{TP(a) + FN(a) + UP(a)}{n}\, \frac{FN(a)}{TP(a) + FN(a) + UP(a)}\, C'(P, n) \\ &+ \frac{TN(a) + FP(a) + UN(a)}{n}\, \frac{FP(a)}{TN(a) + FP(a) + UN(a)}\, C'(N, p) \\ &+ \frac{TP(a) + FN(a) + UP(a)}{n}\, \frac{UP(a)}{TP(a) + FN(a) + UP(a)}\, C'(P, \perp) \\ &+ \frac{TN(a) + FP(a) + UN(a)}{n}\, \frac{UN(a)}{TN(a) + FP(a) + UN(a)}\, C'(N, \perp) \\ =& \frac{1}{n}\Big[FN(a)\, C'(P, n) + FP(a)\, C'(N, p) + UP(a)\, C'(P, \perp) + UN(a)\, C'(N, \perp)\Big]\end{aligned}$$

□

Thus we can use the alternative definition of expected cost, even if the validation set does not represent the correct class distribution. This is an interesting fact which becomes useful later.

2.2.5 Normalized Expected Cost

We have previously shown that we can assume zero costs for classification. Additionally to that we make the further assumption that the costs for abstaining on a positive instance or a negative instance do not differ. This assumption is reasonable since in general we do not know the class of instances abstained on and any further treatment of these instances is independent of the class, although it may depend on the attributes of the instances. The implications of this assumptions are discussed in detail on page 106. As a consequence, the equation for expected cost of an abstention window a can be rewritten, resulting in the following equation:

$$\begin{aligned}\mathbf{EC}(C,a) = P(P)\,FNR(a)\,C(P,\,n) + P(N)\,FPR(a)\,C(N,\,p)\\ + \big[P(P)\,PAR(a)\,+P(N)\,NAR(a)\big]C(\perp).\end{aligned}$$

with $C(\perp) := C(P,\,\perp) = C(N,\,\perp)$. The alternative definition then changes to

$$\mathbf{EC}(C,a) = \frac{1}{n}\Big[FN(a)\,C(P,\,n) + FP(a)\,C(N,\,p) + (UP(a) + UN(a))C(\perp)\Big].$$

So far we have concentrated on the absolute values for expected costs. As we use them only to compare abstention windows we are not interested in the absolute values, but rather in the relationships between the costs. This means we only require to know how much more expensive an abstention window is relative to another one. In fact several cost matrices can be constructed which are all equivalent, that is any comparison between abstention windows has the same result for all of these cost matrices.

Definition 2.12. *Two cost matrices C and C' are called equivalent ($C \equiv C'$) if $\exists k \in \mathbb{R}^+$ such that for all abstention windows $a \in \mathcal{A}$ we have that*

$$\mathbf{EC}(C,a) = k \cdot \mathbf{EC}(C',a).$$

An equivalence class $\bar{C}$ is defined as the set of all cost matrices which are equivalent to C, i.e. $\bar{C} := \{C' | C' \equiv C\}$.

We can get any element of an equivalence class $\bar{C}$ by multiplying every entry of C by a constant value $k \in \mathbb{R}^+$. As we can clearly see, all cost matrices of an equivalence class show the same behavior concerning comparisons between abstention windows.

Lemma 2.13. *Let C and C' be two cost matrices with $C \equiv C'$, then for any two abstention windows a_i and $a_j \in \mathcal{A}$ it is true that*

$$\mathbf{EC}(C,a_i) < \mathbf{EC}(C,a_j) \iff \mathbf{EC}(C',a_i) < \mathbf{EC}(C',a_j).$$

Proof. As $C \equiv C'$, there exists $k > 0$ such that $\mathbf{EC}(C, a_t) = k\,\mathbf{EC}(C', a_t)$ for any abstention window $a_t \in \mathcal{A}$. Thus, we have that

$$\mathbf{EC}(C, a_i) < \mathbf{EC}(C, a_j) \iff k\,\mathbf{EC}(C', a_i) < k\,\mathbf{EC}(C', a_j) \iff \mathbf{EC}(C', a_i) < \mathbf{EC}(C', a_j).$$

□

Therefore, we can conclude that when computing the optimal abstention window given a cost matrix C we can use any cost matrix of its equivalence class $\bar{C}$ instead of C and nevertheless get the same results. In particular, we can also use the cost matrix from $\bar{C}$ for which $C(P, n) = 1$. Such a cost matrix can be obtained from C by dividing every entry of the matrix by the costs for false negative predictions $C(P, n)$. This leads to the definition of normalized expected cost.

Definition 2.14 (Normalized Expected Cost). *Let $a \in \mathcal{A}$ and C be an arbitrary cost matrix. Define $\mu := \frac{C(N,p)}{C(P,n)}$ and $\nu := \frac{C(\perp)}{C(P,n)}$. The normalized expected cost of abstention window a is defined as*

$$\begin{aligned}\mathbf{NEC}(C, a) &:= \frac{\mathbf{EC}(C, a)}{C(P,\, n)} \\ &= P(P)\, FNR(a) + P(N)\, FPR(a)\, \mu + \big[P(P)\, PAR(a) + P(N)\, NAR(a)\big]\nu\end{aligned}$$

or alternatively

$$\mathbf{NEC}(C, a) := \frac{FN(a) + FP(a)\,\mu + \big[UP(a) + UN(a)\big]\,\nu}{n}$$

We observe that the value of normalized expected cost for an abstention window differs from the value of expected cost for this window. However, because of the equivalence between the corresponding cost matrices, the optimal abstention window in terms of normalized expected cost is also the optimal abstention window in terms of expected cost. The original value of expected cost can be obtained from the normalized expected cost by a multiplication with $C(P,\, n)$. For the remainder of this thesis, the term cost scenario is used to denote an equivalence class of cost matrices which in turn is described by ratios between costs.

2.3 Restrictions to Abstention

In the preceeding sections we have shown that any classification algorithm can be employed to create an abstaining classifier by computing the optimal abstention window characterized by minimum expected cost on a validation set. However, the definition of the set of abstention windows $\mathcal{A}$ also includes windows which do not abstain at all as the corresponding lower and upper thresholds are equal. For any given cost scenario we always compute the optimal abstention window and only afterwards check if this actually results in any abstention. To avoid the costly computation step, we desire a condition which tells us a priori that for a given cost scenario abstention is too expensive. Essentially, we require a necessary (but not sufficient) condition for abstention to be effectively possible.

For this purpose, we assume that our validation set correctly represents the distribution of positive and negative classes and thus calculate the normalized expected cost by computing the average cost of instances in the validation set, since this makes the further analysis easier and more comprehensible. However, as we have seen before, we can use the alternative definition even if the distribution in the dataset differs from the correct class distribution. Consequently, these results can be extended to the original definition. We aim to find conditions of the form $\nu \leq c$ for some $c > 0$, such that we can conclude that abstaining is too expensive whenever we know that the condition is violated.

The proofs of the next lemmas all follow the same principle. First, we assume that some restriction on the values of ν is violated and then we show that for any abstention window $a_i = (l_i, u_i)$ with $l_i < u_i$ (which means that the abstention window abstains on at least one instance) we can construct a new abstention window $a_c = (l_c, u_c)$ which has a lower abstention rate than a_i and lower expected cost on the validation set S for this cost scenario. Note that if $l_i < l_c$ (or $u_i > u_c$) there exists at least one instance in S which is abstained on by a_i but classified by a_c. This is a result of the definition of $\mathcal{A}$ with respect to the validation set. The following lemma shows that the costs for abstaining cannot be higher than the maximum of the costs for false negatives (1) and false positives (μ), since in this case expected cost can always be reduced by classifying an instance no matter how. Note that we do not know if μ is greater or smaller than 1. This depends on the original values of $C(P, n)$ and $C(N, p)$.

Lemma 2.15. *Let $S = \{x_1, \ldots, x_n\}$ be the validation set. Let μ and ν be defined as in definition 2.14 and $\nu > \max\{1, \mu\}$. Given an abstention window $a_i = (l_i, u_i) \in \mathcal{A}$ with $l_i < u_i$, we can always construct a new abstention window a_c with $l_i \leq l_c \leq u_c \leq u_i$ and either $l_c > l_i$ or $u_c < u_i$ and $\mathbf{NEC}(C, a_c) < \mathbf{NEC}(C, a_i)$.*

Proof. Construct a_c with $l_i \leq l_c \leq u_c \leq u_i$ such that there exists at least one instance x_j which is abstained on by a_i but classified by a_c. (This means that either $l_i < l_c$ or $u_i > u_c$.) Let d be the number of such instances. The difference in expected cost between a_c and a_i is only determined by these instances, thus we have

$$\mathbf{NEC}(C, a_c) - \mathbf{NEC}(C, a_i) \leq \frac{d \max\{1, \mu\} - \nu\, d}{n} = \frac{d}{n} (\max\{1, \mu\} - \nu) < 0.$$

□

Therefore, we can conclude that no abstention window which abstains on at least one instance can ever be optimal if $\nu > \max\{1, \mu\}$. However, the same is true if the costs for abstaining are greater than the minimum of the costs for false negatives and false positives. The idea is that we can always reduce costs by classifying all abstained instances either positive if $\mu < 1$ or negative otherwise.

Lemma 2.16. *Let S, μ and ν be defined as before. If $\nu > \min\{1, \mu\}$ and $a_i \in \mathcal{A}$ an abstention window with $l_i < u_i$, then there always exists another abstention window a_c with $l_c = u_c$ and $\mathbf{NEC}(C, a_c) < \mathbf{NEC}(C, a_i)$.*

Proof. Construct a new abstention window a_c with $l_c = u_c = l_i$ if $\mu < 1$ and $l_c = u_c = u_i$ otherwise (see figure 2.1(a)). Let $d := |\{x_j \in S | l_i < m(x_j) < u_i\}|$ be the number of instances with margins between l_i and u_i. Note that these are the only instances for which

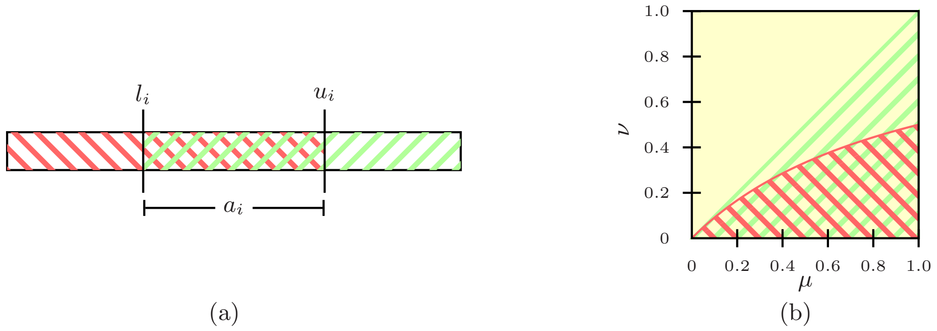


Figure 2.1: Figure (a) illustrates the relationship between abstaining and non-abstaining classifiers. The abstention window $a_i = (l_i, u_i)$ abstains on all instances in the crosshatched range and classifies the remaining instances. The neighboring non-abstaining classifiers have either threshold l_i or u_i. Figure (b) visualizes the increasing strictness of the conditions for $\mu \leq 1$. The yellow region corresponds to the condition $\nu \leq \max\{1, \mu\}$, the green hatched to $\nu \leq \min\{1, \mu\}$ and finally the red hatched to $\nu \leq \frac{\mu}{1+\mu}$.

the predictions of a_i and a_c differ. Let $d_m := |\{x_j \in S | l_i < m(x_j) < u_i \wedge y_j \neq \pi(a_c, x_j)\}|$ the number of instances among these which are misclassified by a_c. We know that $d > 0$ (from the definition of $\mathcal{A}$) and $d_m \leq d$. Thus, we have for the difference in normalized expected cost between a_c and a_i that

$$\begin{aligned}\mathbf{NEC}(C, a_c) - \mathbf{NEC}(C, a_i) &= \frac{1}{n}\big(d_m \min\{1, \mu\} - d\,\nu\big)\\ &< \frac{1}{n}\big(d_m \min\{1, \mu\} - d\,\min\{1, \mu\}\big) = \frac{1}{n}\min\{1, \mu\}(d_m - d) \leq 0\end{aligned}$$

and the newly defined abstention window has lower expected cost. □

So far, we can conclude that we have $\nu \leq \min\{1, \mu\}$ if the optimal abstention window in $\mathcal{A}$ does actually abstain on at least one instance in the validation set. Still this is not the most stringent restriction we can make. The final condition can be obtained by comparing any abstention window with its neighboring non-abstaining classifiers. See figure 2.1(a) for this. The abstention window a_i abstains on all instances which fall in the green and red crosshatched region. The neighboring non-abstaining classifiers either have l_i or u_i as thresholds for positive classification. One of them classifies the same instances as negative as a_i and the complete green hatched region as positive, whereas the other one classifies the red hatched region as negative and the remainder positive. Evidently, at least one abstention window a_i with $l_i < u_i$ must have lower expected cost than both neighboring non-abstaining classifiers for abstaining to be useful. If no such abstention window exists, we can always reduce the expected costs of an abstaining classifier by converting it to a non-abstaining classifier. From this observation the subsequent lemma follows.

Lemma 2.17. *Let S be the validation set and μ, $\nu > 0$ be defined as before. If $\nu > \frac{\mu}{1+\mu}$ and $a_i \in \mathcal{A}$ an abstention window with $l_i < u_i$, then there always exists another abstention window a_c with $l_c = u_c$ and $\mathbf{NEC}(C, a_c) < \mathbf{NEC}(C, a_i)$.*

Proof. By contradiction:
First assume that for all abstention windows a_c with $l_c = u_c$ it is the case that $\mathbf{NEC}(C, a_c) \geq$

$\mathbf{NEC}(C, a_i)$. Thus, in particular, it is true that $\mathbf{NEC}(C, a_l) \geq \mathbf{NEC}(C, a_i)$ and $\mathbf{NEC}(C, a_u) \geq \mathbf{NEC}(C, a_i)$, whereby $a_l = (l_i, l_i)$ and $a_u = (u_i, u_i)$. Let $d_y = |\{x_j \in S | l_i < m(x_j) < u_i \wedge y_j = y\}|$ for $y \in \{P, N\}$ the instances of class y for which the predictions of a_i, a_l and a_u differ. We have that both $d_N > 0$ and $d_P > 0$. If $d_N = 0$ costs could be reduced by classifying all instances positive which fall within the abstained range (as a_l does). If $d_P = 0$, costs could be reduced by classifying those instances negative (as a_u does). The difference in expected cost between a_l and a_i is

$$\mathbf{NEC}(C, a_l) - \mathbf{NEC}(C, a_i) = \frac{1}{n}\big(d_N\,\mu - (d_P + d_N)\,\nu\big) \overset{\text{by def.}}{\geq} 0 \iff d_N \geq d_P \frac{\nu}{\mu - \nu} \tag{2.3}$$

Note that for $\nu = \mu$ the above equation implies that $\mu = 0$ which contradicts the assumption that $\mu > 0$. We then get

$$\begin{aligned}\mathbf{NEC}(C, a_u) - \mathbf{NEC}(C, a_i) &= \frac{1}{n}\big(d_P - (d_P + d_N)\,\nu\big) \overset{\text{Equ. (2.3)}}{\leq} \frac{1}{n}\big(d_P - (d_P + d_P \frac{\nu}{\mu - \nu})\,\nu\big)\\ &= \frac{d_P}{n}\big(\frac{\mu - \nu - \nu\mu}{\mu - \nu}\big) < \frac{d_P}{(\mu - \nu)\,n}\big(\mu - \frac{\mu}{1 + \mu} - \mu\frac{\mu}{1 + \mu}\big)\\ &= \frac{d_P}{(\mu - \nu)\,n}\big(\frac{\mu + \mu^2 - \mu - \mu^2}{1 + \mu}\big) = 0 \end{aligned} \tag{2.4}$$

But equation (2.4) is a contradiction to the assumption. □

The presented lemmas impose ever increasingly strong restrictions on abstaining. Figure 2.1(b) visualizes this for $\mu \leq 1$. Lemma 2.15 still leaves the complete yellow shaded region, whereas Lemma 2.16 limits this to the green hatched rectangle. The last theorem finally excludes all cost scenarios but those that fall in the red hatched area. This implies that only for a small part of possible cost scenarios abstention can in fact improve expected costs. These last results can of course be extended such that they apply to any cost matrix.

Theorem 2.18 (Necessary Condition for Abstaining). *Let $S \subseteq \mathcal{X}$ be the validation set and $a_{opt} \in \mathcal{A}$ an abstention window, such that $l < u$ and $a_{opt} = \operatorname{argmin}_{a \in \mathcal{A}} \mathbf{EC}(C, a)$, then we have for the cost matrix C that*

$$C(\perp) \leq \frac{C(P,\, n)\, C(N,\, p)}{C(P,\, n) + C(N,\, p)}$$

Proof. From lemma 2.17 we know that if $\nu > \frac{\mu}{1+\mu}$, we can always construct a non-abstaining classifier from a_{opt} which has smaller expected cost than a_{opt}. Thus from the optimality of a_{opt} we can conclude that $\nu \leq \frac{\mu}{1+\mu}$. The theorem then results by inserting the original definition of ν and μ. □

The results presented in this chapter require knowledge about costs and class distributions. Unfortunately, this knowledge may be limited. In the following chapter, we examine ways to deal with this problem.

Chapter 3

Visualizing the Behavior of Abstaining Classifiers

In the last chapter we have shown that given a cost matrix and the class distribution finding the optimal abstention window is straightforward. However, there are only few applications for which cost matrices and class distributions are known for certain at all. In most cases, attaching an unequivocal value to costs and class distributions is intricate, as many factors play into the generation of costs and each of those may be rated differently by different people. Even though the exact values for costs are not important and only the ratios between costs are required, the task at hand does not become easier.

The same problems also apply to non-abstaining classifiers and have been approached several times in different ways before. Commonly visualizations are used which illustrate the behavior of classifiers for a variety of cost matrices and class distributions. In the following, two such curves for non-abstaining classifiers are presented and then extended to accommodate abstaining classifiers.

3.1 ROC Curves and Cost Curves for Non-Abstaining Classifiers

We have already given the formal definition of a non-abstaining classifier in the introduction. But since it can be considered as a special case of an abstaining classifier with zero probability for abstaining, we introduce the following notation analogously to the previous chapter.

Definition 3.1 (Threshold). *A threshold t is defined as an abstention window $a = (l, u) \in \mathcal{A}$ with the further restriction of $l = u := s$. The prediction of t on an instance $x \in \mathcal{X}$ is given by*

$$\pi(t, x) = \begin{cases} p & \textit{if } m(x) \geq s \\ n & \textit{if } m(x) < s. \end{cases}$$

Again we can compute for a threshold t the values for $TP(t)$, $FP(t)$, $TN(t)$ and $FN(t)$ as well as the corresponding probabilities of correct or wrong classifications. As a consequence, expected cost can be defined in the same way as for an abstention window. However, as t does not abstain at all, costs for abstaining are of no avail.

Definition 3.2 (Expected Cost). *Let t be a threshold, i.e. non-abstaining classifier, and C be a cost matrix defined as before. The expected cost of t is defined as*

$$\mathbf{EC}(C, t) = FNR(t) \cdot P(P) \cdot C(P, n) + FPR(t) \cdot P(N) \cdot C(N, p)$$

Furthermore we can define the set of possible thresholds for a given classifier as a subset of the set of abstention windows $\mathcal{A}(Cl)$.

Definition 3.3. *Let Cl be a given classifier and $S = \{x_1, \ldots, x_n\} \subseteq \mathcal{X}$ a validation set. Let $M = \big\{m(x_1), \ldots, m(x_n)\big\}$ be the margins obtained by applying Cl on S and $m(x_1) \leq \cdots \leq m(x_n)$. Furthermore, let $\varepsilon > 0$ be an arbitrary but constant value. The set of thresholds for classifier Cl is defined as*

$$\begin{aligned}\mathcal{T}(Cl) := \{a | a \in \mathcal{A}(Cl) \wedge l = u\} = \Big\{ t | s = m(x_1) - \varepsilon \vee s = m(x_n) + \varepsilon \Big\} \\ \cup \Big\{ t | \exists\, 1 \leq j < n : m(x_j) \neq m(x_{j+1}) \wedge s = \frac{m(x_j) + m(x_{j+1})}{2} \Big\}.\end{aligned}$$

As before, we use the abbreviation $\mathcal{T}$ for $\mathcal{T}(Cl)$ if only one classifier is considered at all.

3.1.1 Receiver Operating Characteristic (ROC)

Receiver Operating Characteristic graphs have their origin in signal detection, where they were used to visualize the trade-off between hit rate and false alarm rate [16]. Since then, they have been applied to a wide range of problems as the analysis of diagnostic systems [48], medical purposes [2] and data mining [39].

A point in a ROC curve is derived by plotting the true positive rate of a threshold t on the y-axis against the corresponding false positive rate on the x-axis. A ROC curve for a classifier Cl then results from connecting the points for all $t \in \mathcal{T}(Cl)$ or fitting a curve to them. Example ROC curves for three classifiers are given in figure 3.1(a).

With the help of ROC curves the behavior of classifiers can be studied without knowledge of class distributions and misclassification costs. In general, the closer to the upper left corner the curve is, the better is the corresponding classifier. A diagonal line on the other hand represents a completely random classifier. Additionally, ROC curves can be used to compare the performance of different classifiers based on the notion of dominance which is defined as follows.

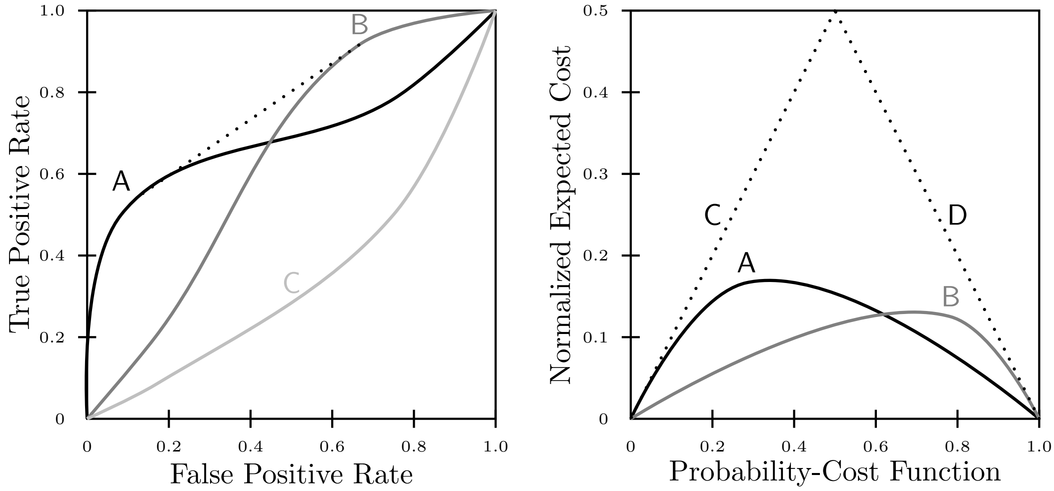
Definition 3.4. *Let P_i and P_j be two points in a ROC curve, t_i and t_j the corresponding thresholds and $\vec{p_i} = (FPR(t_i), TPR(t_i))$ and $\vec{p_j} = (FPR(t_j), TPR(t_j))$ the corresponding position vectors. We say that P_i dominates P_j ($P_i \preceq P_j$) if $FPR(t_i) \leq FPR(t_j)$ and $TPR(t_i) \geq TPR(t_j)$.*

Information about dominance relationships between two points P_i and P_j is useful when comparing the corresponding classifiers because no threshold can ever be optimal for any cost scenario if it is dominated by another threshold. This is shown by the next lemma.

Lemma 3.5. *Given two points P_i and P_j in a ROC curve and $P_i \preceq P_j$, then we have for the corresponding thresholds t_i and t_j that*

$$\mathbf{EC}(C, t_i) \leq \mathbf{EC}(C, t_j)$$

for all possible cost matrices C.



(a) ROC Curves

(b) Cost Curves

Figure 3.1: Figure (a) depicts ROC curves for three example classifiers. Classifier C is dominated by both of the remaining classifiers, whereas none of these is dominated by the other. The convex hull is indicated by a dotted line. Figure (b) shows example cost curves for two non-abstaining classifiers A and B. C represents a classifier which labels every instance as negative and D labels every instance as positive.

Proof. This becomes clear by using the definition of expected cost for a threshold. Hence, we have

$$\begin{aligned}\mathbf{EC}(C, t_i) &= (1 - TPR(t_i)) \cdot C(P,\, n) + FPR(t_i) \cdot C(N,\, p) \\ &\leq (1 - TPR(t_j)) \cdot C(P,\, n) + FPR(t_j) \cdot C(N,\, p) = \mathbf{EC}(C, t_j).\end{aligned}$$

for arbitrary values of $C(P,\, n)$ and $C(N,\, p)$. □

For this reason, a classifier Cl_1 can be considered to be better than another classifier Cl_2 if for every point P_j in the ROC graph for Cl_2 there exists a point P_i in the ROC Graph for Cl_1 with $P_i \preceq P_j$. In this case, we say that Cl_1 dominates Cl_2 ($Cl_1 \preceq Cl_2$). In figure 3.1(a) Classifier C is obviously dominated by both A and B and can consequently be excluded. Unfortunately, we neither have that A$\preceq$B nor that B$\preceq$A, which makes it difficult to decide in favor of any of these two.

If we extend this approach to more than three classifiers, we observe that the number of potentially useful classifiers can be reduced by computing the convex hull of all ROC graphs. Classifiers which do not contribute a point to the convex hull no longer have to be taken into consideration as the convex hull dominates all other points. Note that we can reach any point on the convex hull even if it is not part of any of the ROC curves. This is due to the fact that for any point P_c on the convex hull, there exist two points P_i and P_j in one of the original ROC curves, such that P_c lies on the straight line connecting these two points. Hence, we can conclude, that there exists a value $\sigma \in [0 : 1]$, such that $FPR(t_c) = \sigma\, FPR(t_i) + (1 - \sigma)FPR(t_j)$ and $TPR(t_c) = \sigma\, TPR(t_i) + (1 - \sigma)TPR(t_j)$ and the point P_c can be reached by choosing randomly between the corresponding thresholds or classifiers respectively with probabilities σ and $(1 - \sigma)$ (see also Witten and Frank [54]).

An alternative performance measure for classifiers based on ROC curves offers the area under the ROC curve (AUC) which reduces the two dimensional curve to a single value. This value can be employed similar to accuracy or error rate ([26], [5]) and be estimated by the Mann-Whitney-Wilcoxon test statistic or direct integration. The advantage of the AUC is that it can be easily used to compare classifiers and extended to more than two classes [25].

3.1.2 Cost Curves

Cost curves are an alternative way of illustrating a classifier's performance independent of actual costs and class distributions and were introduced by Drummond and Holte ([14] and [15]). For this kind of curves, normalized expected cost for a threshold t is plotted against the so-called probability-cost function on the x-axis. Note that the definition of normalized expected cost by Drummond and Holte differs from our previous definition. Normalization is performed by dividing the expected cost for a given classifier – i.e threshold – by the expected cost of the worst possible classifier. The worst possible classifier assigns all positive instances to the negative class and all negative ones to the positive class and therefore misclassifies all of them. Of course, this a rather hypothetical case since a classifier of that kind could be transformed to a perfect classifier without effort simply by switching the predictions to the respective other class. Nevertheless, this type of normalization is useful for plotting expected cost. It differs from the one presented in chapter 2 in that the value of the expected cost is normalized with respect to some default classifier instead of with respect to the cost matrix. To prevent confusion of the two types of normalization, we use the notation of Drummond and Holte here.

Definition 3.6. *Let $t \in \mathcal{T}(Cl)$ be a threshold for a given classifier Cl and C a cost matrix. Let $TPR := TPR(t)$ and $FPR := FPR(t)$. The normalized expected cost is defined as*

$$\mathbf{NE}[C] = \frac{(1 - TPR) \cdot P(P) \cdot C(P, n) + FPR \cdot P(N) \cdot C(N, p)}{P(P) \cdot C(P, n) + P(N) \cdot C(N, p)}.$$

The probability-cost function is defined such that we can express normalized expected cost as a linear equation with respect to it.

Definition 3.7. *Let C be a cost matrix, $L \in \{P, N\}$ and $\bar{l} = n$ if $L = P$ and $\bar{l} = p$ otherwise. The probability-cost function $PCF(L)$ is defined as*

$$PCF(L) = \frac{P(L) \cdot C(L, \bar{l})}{P(P) \cdot C(P, n) + P(N) \cdot C(N, p)}.$$

Definitions 3.6 and 3.7 can be used to rewrite the normalized expected cost, resulting in the following theorem.

Theorem 3.8. *Given a threshold t, a cost matrix C and TPR and FPR defined as before, we have that*

$$\mathbf{NE}[C] = (1 - TPR - FPR) \cdot PCF(P) + FPR.$$

Proof. We observe that

$$\begin{aligned} \mathbf{NE}[C] &= (1 - TPR) \cdot PCF(P) + FPR \cdot PCF(N) \\ &= (1 - TPR) \cdot PCF(P) + FPR \cdot (1 - PCF(P)). \end{aligned} \tag{3.1}$$

From equation (3.1) the theorem follows directly. □

Therefore, normalized expected cost can be plotted against $PCF(P)$ on the x-axis which is limited to the range from 0 to 1 as $PCF(P) + PCF(N) = 1$ and thus $PCF(P) \leq 1$. Increasing values for $PCF(P)$ correspond to increasing values for $P(P)$ or $C(P, n)$ relative to $P(N)$ and $C(N, p)$. Since a threshold t is represented by a straight line in the cost curve, a point in the ROC curve corresponds to a line in the cost curve and vice versa a point in the cost curve corresponds to a straight line in a ROC curve, a so-called iso-performance line (see [14] and [40]). Thus, ROC curves and cost curves are dual representations. As a classifier is represented by a set of possible thresholds, a ROC curve for a given classifier Cl can be converted into a cost curve by taking the minimum over the normalized expected cost of all points in the ROC curve for each value of $PCF(P)$ evaluated. If we define normalized expected cost as a function $f(t, PCF(P))$ for a threshold t, a cost curve for a classifier Cl is defined by $\min_{t \in \mathcal{T}(Cl)} f(t, PCF(P))$ for $0 \leq PCF(P) \leq 1$.

Figure 3.1 shows example cost curves of two classifiers A and B. Using these curves several questions can be addressed. For instance, it can be determined for which values of $PCF(P)$ a classifier outperforms both trivial classifiers which either classify all instances as negative (C) or positive (D). This is called the operating range [15]. In the given example both A and B always outperform the trivial classifiers, which is not surprising as our definition of $\mathcal{T}(CL)$ for a given classifier Cl actually includes the trivial classifiers. We therefore redefine the operating range of a classifier Cl as the range of values for $PCF(P)$ for which Cl actually has lower normalized expected cost than any of the trivial classifiers. Furthermore, we can use cost curves to compare two classifiers and determine for which values of $PCF(P)$ one classifier has lower expected cost than the other as well as the significance of this difference. We refer the reader to Drummond and Holte ([14] and [15]) for a more extensive description of the capabilities of cost curves and a comparison of ROC curves and cost curves.

3.2 Abstaining under Uncertain Cost and Class Distributions

In the previous section two different types of curves were introduced which make it possible to demonstrate the behavior of a classifier produced by any machine learning algorithm without knowledge of the exact cost matrix and the class distributions. We now try to create similar visualizations which allow the same analysis for abstaining classifiers. Note that an abstaining classifier Cl_a is in fact given by a set of abstention windows $\mathcal{A}(Cl_a)$, as before a non-abstaining classifier Cl_n was given by a set of thresholds $\mathcal{T}(Cl_n)$. Again the notion of cost scenario as an equivalence class of cost matrices is used.

We then can formulate several questions which have to be addressed:

- For which cost scenarios (and class distributions) does a given classifier outperform the trivial classifiers, i.e. have lower expected cost?
- Given two abstaining classifiers Cl_i and Cl_j
  - For which cost scenarios (and class distributions) does Cl_i outperform Cl_j?
  - Is one of the them better than the other one for all (reasonable) cost scenarios?
  - What is the difference in expected cost between the two classifiers?

- Given an abstaining classifier, which abstention window should we choose for certain cost scenarios (and class distributions)?
- For which cost scenarios (and class distributions) is abstaining helpful at all for our given purpose?

In general any visualization capable of addressing these questions for non-abstaining classifiers can be extended to accommodate abstention. Unfortunately, we always have to add at least one dimension since further degrees of freedom are created. In the following, we present three types of curves for the evaluation of abstaining classifiers – an extension to ROC curves and to the original cost curves as well as a new type of cost curves.

3.2.1 ROC Curves for Abstaining Classifiers

In the original definition of ROC curves the true positive rate of a threshold is plotted against the corresponding false positive rate. When including abstention two additional dimensions are necessary: one for the positive abstention rate and one for the negative abstention rate. This results in a four-dimensional curve which is extremely impractical for human interpretation. Therefore, instead of both positive and negative abstention rate only the overall abstention rate is used.

Definition 3.9. *Given an abstention window $a \in \mathcal{A}$ and a validation set $S \subseteq \mathcal{X}$, the abstention rate $AR(a)$ is defined as*

$$AR(a) := \frac{UP(a) + UN(a)}{n}$$

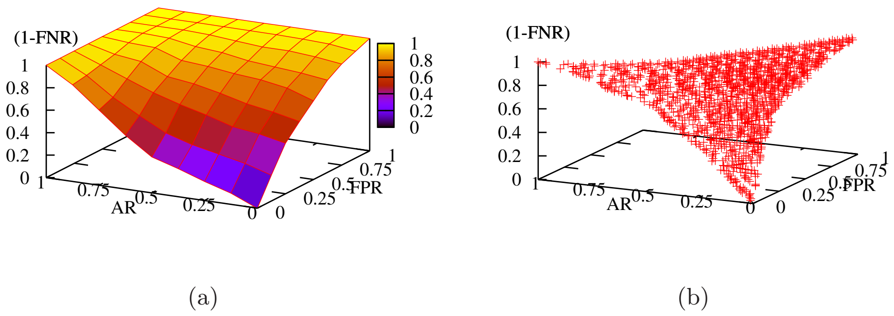
where $UP(a)$ and $UN(a)$ denote the number of positive and negative instances in S abstained on by a.

Unfortunately, the abstention rate depends on the class distribution on the validation set and may differ from the abstention rates obtained for other class distributions. However, without the assumption that the overall abstention rate does not change no matter how the class distribution is altered, no intelligible visualization could be devised.

An additional problem arises when extending the definition of ROC curves to abstention. For the original ROC curves, only one threshold is increased such that the rate of positive predictions is rising and as a result both true positive rate and false positive rate increase. However, for abstaining classifiers, two thresholds can be changed. If only the lower threshold for an abstention window changes but the upper threshold remains as it is, both true positive rate and false positive rate are not affected at all. Plotting the true positive rate against false positive and abstention rate then would not describe the behavior of the abstention window properly. This can only be achieved by using false negative rate instead of true positive rate. For the original ROC curves those two values are complementary. To remain as close as possible to the original definition, a ROC curve for abstaining classifiers then is described as follows.

Definition 3.10. *Let a be an abstention window. The corresponding point P in the ROC curve is given by the position vector*

$$\vec{p} := (FPR(a), AR(a), (1 - FNR(a))).$$



(a)

(b)

Figure 3.2: Figure (a) depicts a three dimensional ROC curve as it is expected to look like, whereas figure (b) shows an example ROC curve for a real-life application. Points with high false positive rate and abstention rate are missing completely, because high abstention rate results in low false positive rate and vice versa.

Accordingly, we can see that the value at the z-axis is affected even if only the lower threshold is changed.

The ROC curve for a classifier Cl results from plotting all abstention windows $a \in \mathcal{A}(Cl)$. Intuitively, one might expect a graph as depicted in figure 3.2(a) with (1-FNR) increasing as FPR and AR increase. However, the curve in figure 3.2(b) more accurately reflects the behavior of the ROC curve for an abstaining classifier. For small values of abstention or false positive rate we observe the expected behavior, yet points with high abstention and false positive rate are missing completely. The reason for this is that high false positive and high abstention rate exclude each other as many false positives mirror the fact that a considerable amount of instances is actually classified. Vice versa we cannot misclassify instances if we already abstain on a majority of them.

Given such a three dimensional ROC curve, we can use it the same way as the two dimensional curves to compare classification schemes based on the convex hull. Additionally, it can help to choose an appropriate abstention window without exact knowledge of class distributions and costs. We might tend to select an abstention window with moderately low abstention, false positive and false negative rate. If false negative predictions are considered more expensive than false positive predictions, false negative rate can be reduced at the expense of false positive rate or abstention rate. On the other hand, abstention rates can be kept low by increasing both false positive and false negative rates, and so on. The disadvantage of this approach is that it is a rather inexact method to choose an appropriate abstention window as without knowledge of costs the optimal abstention window can only be estimated and is difficult to find by visual inspection only. Furthermore, we are at a loss to answer most of the above mentioned questions.

3.2.2 Cost Curves for Uncertain Costs and Class Distributions

As we have seen, 3D ROC curves are insufficient for determining a suitable abstention window for unknown costs or class distributions or for comparing classifiers. To circumvent this problem we now extend the cost curves described on page 26 which enable us to illustrate the behavior of classifiers for changing costs and class distributions.

The definition of expected cost for an abstention window has already been given in the previous chapter. Again, as for ROC curves, we have to use the overall abstention rate on the validation set instead of both positive and negative abstention rate to reduce the dimensionality of the constructed curves. The expected cost for an abstention window is then normalized with respect to the expected cost of the worst classifier conceivable which has $FPR = 1$, $FNR = 1$ and $AR = 1$. In reality, no classifier ever reaches this maximum expected cost since it is not possible to misclassify all instances and at the same time abstain on all of them. Nevertheless, this normalization is essential to represent the expected cost in terms of the probability-cost function.

Definition 3.11. *Let a be an abstention window and C a cost matrix. Define $FNR := FNR(a)$, $FPR := FPR(a)$ and $AR := AR(a)$. The normalized expected cost of a is defined as*

$$\mathbf{NE}[C] = \frac{P(P) \cdot FNR \cdot C(P,\, n) + P(N) \cdot FPR \cdot C(N,\, p) + AR \cdot C(\perp)}{P(P) \cdot C(P,\, n) + P(N) \cdot C(N,\, p) + C(\perp)}.$$

The probability-cost functions $PCF(P)$, $PCF(N)$ and $PCF(\perp)$ are defined analogously to definition 3.7 with $PCF(P) + PCF(N) + PCF(\perp) = 1$.

Definition 3.12. *Let C be a cost matrix, $L \in \{P, N\}$ and $\bar{l} = n$ if $L = P$ and $\bar{l} = p$ otherwise. The probability-cost function $PCF(L)$ is defined as*

$$PCF(L) = \frac{P(L) \cdot C(L,\, \bar{l})}{P(P) \cdot C(P,\, n) + P(N) \cdot C(N,\, p) + C(\perp)}.$$

The probability-cost function $PCF(\perp)$ is defined as

$$PCF(\perp) = \frac{C(\perp)}{P(P) \cdot C(P,\, n) + P(N) \cdot C(N,\, p) + C(\perp)}.$$

By inserting definition 3.12 into the equation for normalized expected cost we receive the following result.

Theorem 3.13. *Given an abstention window a and a cost matrix C and $FNR := FNR(a)$, $FPR := FPR(a)$ and $AR := AR(a)$ we have that*

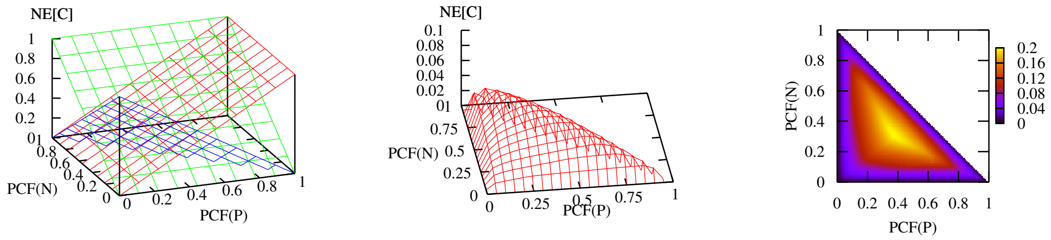
$$\mathbf{NE}[C] = (FNR - AR) \cdot PCF(P) + (FPR - AR) \cdot PCF(N) + AR.$$

Proof. From definition 3.12 we obtain

$$\begin{aligned}\mathbf{NE}[C] &= FNR \cdot PCF(P) + FPR \cdot PCF(N) + AR \cdot PCF(\perp) \\ &= FNR \cdot PCF(P) + FPR \cdot PCF(N) + AR \cdot (1 - PCF(P) - PCF(N)) \qquad (3.2)\end{aligned}$$

The theorem follows directly from the last equation. □

Based on theorem 3.13 a cost curve is created by setting the x-axis to $PCF(P)$, the y-axis to $PCF(N)$ and the z-axis to $\mathbf{NE}[C]$. As a consequence, an abstention window $a \in \mathcal{A}$ is depicted as a plane in this type of cost curves. The operating range of an abstention window is then defined as the area for which the abstention window outperforms the three trivial



(a) (b) (c)

Figure 3.3: Example cost curves for uncertain costs and class distributions. Figure (a) shows the three trivial classifiers which either label all instances as positive (green) or negative (red) or abstain on all of them (blue). Figure (b) shows a cost curve for an example classifier and (c) the same curve projected to 2D. Note that in figure (a) only individual abstention windows are depicted whereas figures (b) and (c) contain cost curves for classifiers which are represented by a set of abstention windows.

classifiers which either classify every instance as negative ($z = x$) or positive ($z = y$) or abstain on every instance ($z = -x - y + 1$). See also figure 3.3(a).

Given a classifier Cl and the corresponding set of abstention windows $\mathcal{A}(Cl)$, we can compute a cost curve by computing the minimum normalized expected cost over all possible abstention windows $a \in \mathcal{A}(Cl)$ for each combination of $PCF(P)$ and $PCF(N)$. By encoding the value of expected cost by colors, the three dimensional curve can be projected into two dimensions which makes it easier to be interpreted. The darker the color, the lower is the expected cost. Figure 3.3(b) shows such a cost curve in 3D and figure 3.3(c) the same curve projected to 2D by colors.

Before describing how these cost curves can be used to compare classifiers and to answer the questions we posed, first a second type of cost curves is introduced which presumes fixed class distributions and alters only the cost scenarios.

3.2.3 Cost Curves for Uncertain Costs and Fixed Class Distributions

In the last section we have presented an extension of standard cost curves to abstaining which can deal with both uncertain costs and class distribution. However, as we will see in the next section, these curves are difficult to analyze. We now introduce a second type of cost curves for which class distributions have to be kept fixed in some way. Now, the definition of normalized expected cost from 2.14 is used again. Remember that for $\mu = \frac{C(N,p)}{C(P,n)}$ and $\nu = \frac{C(\bot)}{C(P,n)}$ normalized expected cost is defined as

$$\mathbf{NEC}(C,a) = P(P)\,FNR(a)\, + P(N)\,FPR(a)\,\mu + \big(P(P)PAR(a) + P(N)NAR(a)\big)\,\nu$$

or alternatively

$$\mathbf{NEC}(C,a) = \frac{FN(a)\, + FP(a)\,\mu + \big(UP(a)\, + UN(a)\big)\,\nu}{n}$$

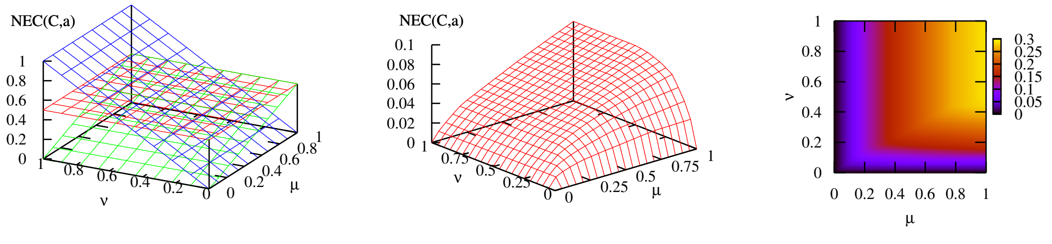
Note that the data set $S \subseteq \mathcal{X}$ on which the cost curves are computed does not have to reflect the actual distribution of classes in the instance space $\mathcal{X}$. In this case the first definition can be used with any class distribution supposed to be the true class distribution. As determining the correct class distribution is not a trivial task, the alternative definition can be used if we either know or – lacking further information – assume that the class distribution in S actually is the correct one.

Without loss of generality, we can take for granted that $\mu \leq 1$ if the positive class is always defined to be the one with highest misclassification costs. This might seem a rather stringent restriction considering the fact that the misclassification costs are unknown. Yet even though establishing the exact cost values is complicated, determining the class with highest misclassification costs for most problems is not. Suppose the task of predicting whether a specific person is suffering from a certain perilous disease. Although we do not know how much more expensive not detecting the disease is compared to treating a healthy person, it is in many cases safe to say that it is more expensive. Furthermore, we can assume that $\nu \leq 1$ on the grounds of the limitations to abstaining presented in chapter 2. The cost curve then is created by plotting normalized expected cost against values of μ and ν between 0 and 1.

Once again, an abstention window $a \in \mathcal{A}$ is represented by a plane in the cost curve. An abstention window can be compared against the trivial classifiers which are also defined by planes. If every instance is classified as negative, the resulting plane is defined by $z = P(P)$ and is parallel to the base area. On the other hand, if every instance is labeled positive we have that $z = P(N)\,x$ and if all instances are abstained on $z = y$. As before the cost curve for an abstaining classifier results by taking the minimum over the expected cost for all corresponding abstention windows for each combination of μ and ν and the cost curve can be projected to a two dimensional curve using color coding. (See figure 3.4 for examples).

3.3 Analyzing Cost Curves for Abstaining Classifiers

To avoid repetition, the behavior of both types of cost curves as well as the basic approach to analyzing them is described in one section. We distinguish between them by using the terms cost curves type I and type II. For the sake of simplification, we now assume that any



(a) (b) (c)

Figure 3.4: Example cost curves for uncertain costs but fixed class distributions. Figure (a) shows the three trivial classifiers which either label all instances as positive (green) or negative (red) or abstain on all of them (blue). Figure (b) shows a cost curve for an example classifier and (c) the same curve projected to 2D. Note that in figure (a) only individual abstention windows are depicted whereas figures (b) and (c) contain cost curves for classifiers which are represented by a set of abstention windows.

cost curve is given by a function $f(a, x, y)$ with a an abstention window and x, $y \in [0, 1]$. For the first type of cost curves x is $PCF(P)$ and y $PCF(N)$, therefore increasing values of x result from increasing values of $C(P, n)$ and/or $P(P)$, whilst increasing values of y are a consequence of increasing values of $C(N, p)$ and/or $P(N)$. As $PCF(\perp) = 1 - x - y$, the costs for abstaining are negatively correlated to x and y.

It is due to these dependencies between costs and class distributions, that the first type of cost curves are difficult to interpret. If we change the class distributions but leave costs constant, the values for $PCF(P)$ and $PCF(N)$ are changed completely. This becomes clear in the following example. Let the costs for misclassification be $C(P, n) = 6$ and $C(N, p) = 4$ and the cost for abstaining be $C(\perp) = 2$. For equal class distributions we then have $PCF(P) = \frac{3}{7}$ and $PCF(N) = \frac{2}{7}$. However if we change class probabilities only slightly to $P(P) = 0.4$ and $P(P) = 0.6$, we have that $PCF(P) = PCF(N) = \frac{6}{17}$.

The second type of cost curves is distinctively easier to interpret. Increasing values of x correspond to increasing $C(N, p)$ relative to $C(P, n)$ and increasing values for y correspond to increasing $C(\perp)$ relative to $C(P, n)$ and there exist no dependencies between x and y. Yet, it is a big disadvantage of these curves that class distributions have to be fixed, i.e. a specific class distribution has to be chosen (see also page 31). Nevertheless, we demonstrate in the next section that the second type of cost types can also be used to explore different class distributions without computing a new curve each time.

Although cost curves are continuous in theory, expected cost has to be computed for specific values of x and y in order to plot the curves. The number of values chosen for x and y determines the resolution of the curve and is denoted as Δ. The more values we choose, the better the cost curve. Unfortunately, the time required for calculating a cost curve strongly depends on the values chosen for Δ, as we see in chapter 5. We thus can define a cost curve as a $\Delta \times \Delta$ matrix:

Definition 3.14. *Let Cl_p be a classifier and $\mathcal{A}(Cl_p)$ the set of possible abstention windows over $S \subseteq \mathcal{X}$. Let Δ be the desired resolution. We define a cost curve as a matrix $K(p)$ with*

$$k_{i,j}(p) := \min_{a \in \mathcal{A}(Cl_p)} f(a, i/\Delta, j/\Delta), \quad 0 \leq i, j \leq \Delta.$$

For the first type of cost curves any entry $k_{i,j}(p)$ with $i+j > \Delta$ within the cost curve matrix is irrelevant because $PCF(P) = i/\Delta$ and $PCF(N) = j/\Delta$ and $PCF(P) + PCF(N) \leq 1$. This is not the case for the second type of cost curves. However, we may occasionally restrict the entries considered to those with $j \leq \Delta/2$ which implies that $\nu \leq \frac{1}{2}$. As we have shown before, abstaining is only possible if $\nu \leq \frac{\mu}{1+\mu} \leq \frac{1}{2}$. For this reason, any entry of the matrix with $j > \Delta/2$ does not provide any further information at all.

To analyze the difference in expected cost between two classifiers we can simply compute the difference between the corresponding cost curves. The two classifiers may have been generated by two different classification algorithms or also different settings of the same algorithm. As a consequence, we obtain a new curve or matrix.

Definition 3.15. *Given two classifiers Cl_p and Cl_q, let $K(p)$ and $K(q)$ be the corresponding cost curves. The differential cost curve $D(p, q)$ is defined as the difference between $K(p)$ and $K(q)$:*

$$d_{i,j}(p, q) := k_{i,j}(p) - k_{i,j}(q), \quad 0 \leq i, j \leq \Delta.$$

Note that for a specific classifier Cl_p $k_{i,j}(p)$ denotes the minimum expected cost of any abstention window of Cl_p for the cost scenario specified by i and j. For the first type of cost curves i and j specify values for $PCF(P)$ and $PCF(N)$, whereas for the second one they specify μ and ν. For the second type of cost curves for example, we have that

$$k_{i,j}(p) = \min_{a \in \mathcal{A}(Cl_p)} \mathbf{EC}(C, a)$$

with $C(P, n) = 1$, $C(N, p) = i/\Delta$ and $C(\perp) = j/\Delta$ since normalized expected cost is the same as expected cost for a normalized cost matrix. Thus, normalization is performed simply by computing expected cost for normalized cost matrices. Therefore the values of expected cost can be directly compared between cost curves and by computing $d_{i,j}(p, q)$ we compute the difference of expected cost between the optimal abstention window of Cl_p and the optimal abstention window of Cl_q for the cost scenario specified by i and j.

If all entries in the differential cost curve $D(p, q)$ for two classification algorithms Cl_p and Cl_q are positive, this implies that Cl_q outperforms Cl_p for all combinations of costs (and also class distributions for the first type of cost curves). A classifier Cl_s outperforms another classifier Cl_t for a specific cost scenario if there exists an abstention window of Cl_s that has lower expected cost than any abstention window of Cl_t for this scenario. On the other hand, Cl_p is superior to Cl_q if all entries of $D(p, q)$ are negative. If one of these possibilities applies, we are fortunate as we can completely discard one of the classifiers in either case.

Unfortunately, most of the time we have $d_{i,j}(p, q) > 0$ for some i and j and $d_{i,j}(p, q) < 0$ for others. In this case, the cost curves have to be studied thoroughly. Now the absolute value of the difference becomes important or additional information which allows us to restrict the possible ranges for i and j. If so we can use only a sub-matrix of the complete curve.

If we compute the differential cost curve between the set of trivial classifiers we described before and any other abstaining classifier Cl, we observe that this curve contains no negative entries at all, as the trivial classifiers are contained in the set of abstention windows for any classifier. Nevertheless, it is interesting to examine when exactly the entries are actually greater than zero.

As the actual cost values are rather arbitrary, one can alternatively examine how much better one classifier is relative to the other one. For this purpose, the definition of the differential cost matrix can be rewritten as

$$d_{i,j}(p, q) := \frac{k_{i,j}(p) - k_{i,j}(q)}{\max\{k_{i,j}(p), k_{i,j}(q)\}}, \quad 0 \leq i, j \leq \Delta. \tag{3.3}$$

The entries in the matrix still are positive for cost scenarios for which classifier Cl_q outperforms Cl_p and negative otherwise.

Instead of considering the difference between cost values one might compute the ratio $k_{i,j}(p)/k_{i,j}(q)$ between those values. Again this is possible because the normalization is the same for both cost curves. In this case, entries in the resulting matrix are greater than 1 if classifier Cl_q is better than Cl_p and smaller than 1 otherwise. Unfortunately, these ratios are more difficult to analyze in a plot as the ranges of possible values differ strongly. If Cl_q is superior to Cl_p, we observe values between 1 and ∞, whereas otherwise we observe only values between 0 and 1. Therefore, we can have both very large and very small values in the same plot and changes between cost scenarios for which Cl_q is the better choice appear to be more pronounced than for cost scenarios for which Cl_p is better even though it may not be the case.

If we have more than two classifiers, it can be tiresome to compare all differential cost curves as the number of differential cost curves is quadratic in the number of classifiers. In this case we use a different curve.

Definition 3.16. *Let $Cl_1, \dots, Cl_p$ be p classifiers and $K(1), \dots, K(p)$ the corresponding cost curves. Then we define the minimum cost curve M as*

$$m_{i,j} := \min_{1 \le s \le p} k_{i,j}(s)$$

and the index matrix I as

$$i_{i,j} := \operatorname*{argmin}_{1 \le s \le p} k_{i,j}(s).$$

The minimum cost curve is only of minor interest here since it only contains the minimum cost that can be achieved for every cost scenario, but gives no hint as to which classifier to use. For practical purposes, the index matrix is of greater importance. Any classifier which is not contained in the index matrix at all can be eliminated completely. We may even remove a classifier, which is optimal only for very few cost scenarios and differs only insignificantly from some other classifier. Here pairwise differential cost curves turn out to be helpful again.

The answers to the questions raised on page 27 can be determined easily with help of the introduced matrices. When comparing one specific classifier against the trivial classifiers or another classifier we calculate a differential cost curve. Negative and positive entries indicate the cost scenario for which either one is superior. The absolute value of the difference tells us how much better one classifier is. Unfortunately, it is difficult to determine for the first type of cost curves exactly which costs and class distributions correspond to values of i and j due to the elaborate definition of $PCF(P)$ and $PCF(N)$. Contrary to that, the second type of cost curves is easy to interpret. A value of i corresponds to false positive costs of i/Δ and a value of j to abstention costs of j/Δ. Although the same questions might be answered without the help of the differential cost curve by simply comparing the curves for the two classifiers, the use of the differential cost curve makes this task easier. We do not only obtain the exact cost scenarios for which either of the classifiers is superior – which is difficult to determine by visual inspection of the two curves only – but we can also use the color projection to 2D for easier analysis of the differences.

For a specific classifier the best abstention window to choose is exactly the one with minimum expected cost for each cost scenario (and class distribution). Alternative curves can be computed containing the optimal lower and upper thresholds. These curves visualize the shifts in the optimal abstention window for changing costs (and class distributions for the first type of cost curves). Finally, the scenarios for which abstaining is of benefit unfold when studying the curve which shows the optimal abstention rate for each cost scenario. Abstention can only be applied successfully for those cost scenarios for which the optimal abstention rate is actually greater than zero.

3.4 Comparison between both Types of Cost Curves

We have presented two types of cost curves and described how each of them can be used to compare classifiers produced by different classification algorithms. The distinction between the two types at first glance appears to be clear. The first one is to be used if both costs and class distributions are unknown, whereas the second one applies to unknown costs yet fixed class distributions. However, when using the abstention rate instead of positive as well as negative abstention rate, both types of representations are in fact equivalent.

Suppose the class distributions are originally given as $P'(P)$ and $P'(N)$ but then the focus changes to a different distribution given by $P(P)$ and $P(N)$. Lemma 2.10 allows us to use the original class distribution to calculate normalized expected cost by simply changing the cost matrix to a matrix C' with $C'(P,\, n) = \frac{P(P)}{P'(P)}$ and $C'(N,\, p) = \frac{P(N)}{P'(N)}\mu$.

Initially, the costs for abstaining are not affected because the abstention rate of the validation set is used for any class distribution. Therefore, changes in the class distribution do not affect abstention rate. However, to obtain normalized costs the complete matrix has to be divided by $C'(P,\, n)$ resulting in a second matrix C''. By this means, the value of normalized expected cost for $P(P)$ and $P(N)$ and the original matrix C can be obtained by looking up the costs for C'' in a cost curve computed for $P'(P)$ and $P'(N)$ and multiplying it by $\frac{P(P)}{P'(P)}$.

In order to make this observation easier to comprehend, an example is given. We presume that the positive class is the one with highest misclassification costs and that $C(P,\, n) = 1$, $C(P,\, n) = \mu$ and $C(\perp) = \nu$ for some constants $\mu, \nu \in [0 : 1]$. For this example, we set $\mu = 0.8$ and $\nu = 0.4$ and do not change them at all. Our aim is to examine how expected costs change with changing class distributions. Using the first type of cost curves, this task is easy. We simply compute the appropriate values for $PCF(P)$ and $PCF(N)$ and then look up the pre-computed costs for this scenario.

Alternatively, we can compute a cost curve of the second type for each class distribution. However, the additional effort then is immense. The second possibility is to use the original cost curve and only change the costs considered. Suppose a cost curve of type II for $P'(P) = 0.5$ has been calculated, but now we want to examine the expected cost for $P(P) = 0.8$. First a new cost matrix C' is derived such that $P'(P) \cdot C'(P,\, n) = P(P) \cdot 1$ and $P'(N) \cdot C'(N,\, p) = P(N) \cdot \mu$. Then, a normalized cost matrix C'' is computed from C' with $C''(P,\, n) = 1$. Accordingly, we can determine the normalized expected cost for $P(P) = 0.8$, $\mu = 0.8$ and $\nu = 0.4$ by looking up the pre-computed expected cost for $P'(P) = 0.5$, $\mu = 0.2$ and $\nu = 0.25$ and multiplying this value by $\frac{P(P)}{P'(P)} = 1.6$ in the end.

If the value for $P(P)$ in the example was decreasing instead of increasing, the costs for false positives might actually become greater than the costs for false negatives depending on the original value of $C(N,\, p)$. In this case a second cost curve has to be computed which presumes the negative class as the one with highest misclassification costs. Nevertheless, this requires only one additional curve.

Hence, when analyzing changing class distributions with the second type of cost curves two steps have to be performed for each scenario considered. First, the new cost matrix C'' has to be computed – this is analogous to the computation of $PCF(P)$ and $PCF(N)$ for the first type of cost curves – and then the value of expected cost for C' is obtained by a multiplication with $\frac{P(P)}{P'(P)}$. This second step does not have to be performed for the first type of cost curves, thus when comparing expected cost for different cost scenarios and class distributions, the second type of cost curves is not completely equivalent. However in most cases we are more interested in the optimal classifiers for certain scenarios or the optimal abstention rate or false positive and negative rate than in the exact value of expected cost. For these purposes the multiplication is not relevant because the matrices C' and C'' are equivalent. Therefore the optimal abstention window for C'' is also optimal for C' and in this case the effort for using the second type of cost curves is the same as when using the first type.

The advantage of the second type of cost curves is that they are easier to evaluate if only cost scenarios change but not class distributions. If both costs and class distributions are variable the two curves are equivalent to a large extent if the abstention rate on the validation set is assumed to correspond to the expected abstention rate on any sample from $\mathcal{X}$. A more accurate estimate of expected costs can be achieved by distinguishing between positive and negative abstention rate. However, in this case, the first type of cost curves cannot be applied at all without adding a further dimension. The second type is still applicable, yet class distributions cannot be changed anymore.

Chapter 4

Combining Abstaining Classifiers

In the previous chapters the emphasis has always been on one model which is transformed into an abstaining classifier by choosing one of its possible abstention windows. Such a classifier abstains on a range of instances and thus is able to give more confident predictions for those instances which are actually classified. However, for different classifiers the optimal abstention window for a given cost scenario might cover a different selection of instances. This situation is illustrated in figure 4.1(a). The instance space $\mathcal{X}$ is depicted by the blue circle. The green hatched area represents those instances on which the first abstention window a_1 abstains on, whereas the red hatched area represents the ones which the second abstention window a_2 does not classify. In the depicted case the two abstention windows do overlap, but they do not have to in general. In the following chapter, we focus on the problem of how to combine two (or more) abstention windows in such a way that we achieve high confidence predictions for a wider range of instances than for any of the original abstention windows.

Unfortunately, this cannot be solved as easily as classifying instances when any of the two abstention windows would classify and abstaining only when both windows would vote to do so, as the predictions of the two abstention windows may contradict each other. In the above example, a_1 might classify an instance as positive which a_2 classifies negative. Alternatively, one of the abstention windows might abstain on an instance which the other one misclassifies. In this case classifying the instance is detrimental and abstaining the better choice. Thus, the essential problem we are faced with when combining two (or more) abstaining classifiers consists of how to resolve contradictory predictions appropriately. We present two different approaches which either combine the predictions of abstention windows by weighting them according to expected cost or prevent contradictory predictions altogether by applying one window after the other.

Both approaches to the combination of abstention windows result in meta-classification schemes, which are independent of the actual algorithms used to produce the base classifiers. This is not surprising since abstention windows itself were introduced as a form of meta-classification. However, they are computed from one model only, whereas several models are involved when combining abstaining classifiers.

4.1 Approaches to Combining Classifiers

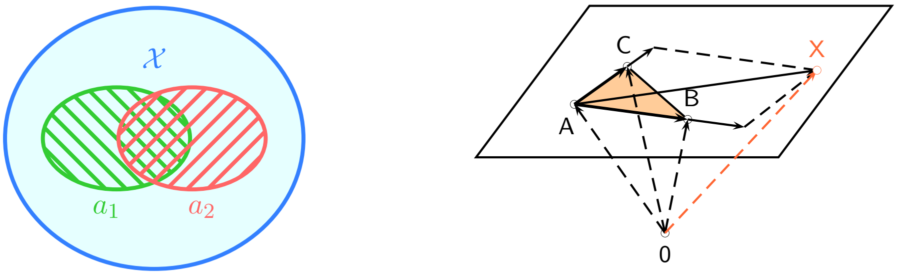
There are several approaches to combining base-level models. Such are bagging [6], boosting [21] and stacking [55] (and meta-decision trees (MDTs, [49]) as a special case of stacking). The general idea behind all of these methods is that a set of diverse base level classifiers is used to create a higher level classifier. However, the way the base level models are combined differs greatly between the methods.

4.1.1 Bagging

For bagging multiple models are derived using the same classification algorithm by taking bootstrap samples of the original training data and using each sample to train one of the base classifiers. Bootstrap samples are created from the training set by drawing with replacement. Each of the samples has the same size as the original set, but some instances of the training set are missing or represented more than once. As the training sets differ between each other, each of the resulting classifiers behaves slightly different on the test data. The final prediction result follows from a vote among the multiple models. Bagging is most effective for unstable classification algorithms, for which small perturbations within the learning set cause distinctive changes in the model constructed.

4.1.2 Boosting

While for bagging the base classification models can be computed in parallel, for boosting they have to be computed one after another, as in later iterations the classification algorithm is encouraged to produce models which perform good on training instances misclassified by the previous models. This is achieved by storing weights for the instances of the learning set. At the beginning the weights are uniformly distributed but with each iteration the weights of misclassified instances are increased. For the final prediction the individual classification models are weighted based on their performance on the training data. Boosting originated from a specific theoretical framework of computational learning theory, the so-called PAC



(a) (b)

Figure 4.1: Figure (a) illustrates how abstention windows may cover different instances within the instance space $\mathcal{X}$ and thus be used complementary. The presented abstention windows a_1 and a_2 do overlap, but in many cases they are disjoint. Figure (b) describes a plane in a three-dimensional space through three non-collinear points A, B and C. If A, B and C correspond to specific abstention windows, we can reach any point lying on the rectangle between these points (orange) by choosing between them with appropriate probabilities.

(probably approximately correct) model of machine learning. (An introduction can be found in Mitchell [37].)

4.1.3 Stacking

Both bagging and boosting use only one classification algorithm to produce the base classifiers. For stacking on the other hand several algorithms can be used to produce the base level (level-0) models. The prediction of the models then can be combined by a higher level (level-1) model, which has been trained on the predictions of the base level classifiers. The corresponding training set is derived in the same way as for the calculation of optimal abstention windows by applying the base level classifiers on a separate validation set. Essentially, the level-1 classifier is trained to decide for each instance how much weight to give to each base model and how to combine the predictions and any machine learning algorithm can be used to train the higher level model. For example, meta-decision trees – as the name implies – use a modification of standard decision tree learning.

4.2 Combining in ROC Space

As we have seen before in the original two-dimensional ROC curve, any point on a straight line between two points in the ROC graph can be reached by choosing between the corresponding models with appropriate probabilities. A similar approach can be used to reach any point on a plane defined by three points in the three-dimensional ROC space. As these points represent abstention windows, this method effectively combines these windows.

Lemma 4.1. *Given three abstention windows a_1, a_2, a_3 and the corresponding points in the ROC curve P_i, $1 \leq i \leq 3$ with $\vec{p_i} = (FPR(a_i), AR(a_i), (1 - FNR(a_i)))$. We can reach any point P_m on the rectangle defined by a plane through P_1, P_2 and P_3 by choosing any of the abstention windows a_i with probability ρ_i such that $\rho_1 + \rho_2 + \rho_3 = 1$.*

Proof. If we choose the abstention windows according to the probabilities ρ_i we observe that $\vec{p_m} = \rho_1 \cdot \vec{p_1} + \rho_2 \cdot \vec{p_2} + \rho_3 \cdot \vec{p_3}$. We have to prove now that

i) for any values for the ρ_i such that $\rho_1 + \rho_2 + \rho_3 = 1$, P_m lies on the rectangle defined by P_1, P_2 and P_3.

ii) for any point on this rectangle there exist such ρ_i.

A plane through three non-collinear points A, B, C with corresponding position vectors $\vec{a}$, $\vec{b}$ and $\vec{c}$ is given by the following equation (see also figure 4.1(b)).

$$\vec{x} = \vec{a} + \sigma(\vec{b} - \vec{a}) + \tau(\vec{c} - \vec{a}) \quad (-\infty < \sigma, \tau < \infty) \tag{4.1}$$

Thus, for P_1, P_2 and P_3 we have the rectangle between these points defined by

$$\vec{x} = \vec{p_1} + \sigma(\vec{p_2} - \vec{p_1}) + \tau(\vec{p_3} - \vec{p_1}) \quad (\sigma, \tau \in [0:1] \wedge \sigma + \tau \leq 1) \tag{4.2}$$

Obviously we can see that P_m lies on that rectangle by setting $\sigma = \rho_2$ and $\tau = \rho_3$. On the other hand we know that for any point X on that rectangle, there exist σ and τ such that for the corresponding position vector $\vec{x}$ equation (4.2) holds. Now the ρ_i can be found easily by setting $\rho_1 = 1 - \sigma - \tau$, $\rho_2 = \sigma$ and $\rho_3 = \tau$. □

The lemma suggests a simple way to combine abstaining classifiers in the three-dimensional ROC curve by computing the convex hull. The higher level classifier is created by choosing randomly among three abstention windows on the convex hull with corresponding probabilities ρ_i. This way no contradictions occur as always only one single classifier delivers the final prediction. Unfortunately, there are two problems associated with this approach. First three-dimensional ROC curves are difficult to analyze and secondly – and this is the major drawback – none of the combined classifiers can ever surpass all base classifiers with regard to expected cost as the following lemma shows.

Lemma 4.2. *Let a_1, a_2 and a_3 be three abstention windows and $\vec{p_i}$, $1 \leq i \leq 3$ the corresponding vectors in the ROC curve. Let a_m be a combined abstention window with $\vec{p_m} = \rho_1 \cdot \vec{p_1} + \rho_2 \cdot \vec{p_2} + \rho_3 \cdot \vec{p_3}$ and $\rho_1 + \rho_2 + \rho_3 = 1$. Then for any cost matrix C, we have that $\mathbf{EC}(C, a_m) \geq \min_{1 \leq i \leq 3} \mathbf{EC}(C, a_i)$.*

Proof. From the definition of $\vec{p_m}$ it follows that

$$\begin{aligned}
\mathbf{EC}(C, a_m) &= \rho_1 \cdot \mathbf{EC}(C, a_1) + \rho_2 \cdot \mathbf{EC}(C, a_2) + \rho_3 \cdot \mathbf{EC}(C, a_3) \\
&\geq \rho_1 \cdot \min_{1 \leq i \leq 3} \mathbf{EC}(C, a_i) + \rho_2 \cdot \min_{1 \leq i \leq 3} \mathbf{EC}(C, a_i) + \rho_3 \min_{1 \leq i \leq 3} \mathbf{EC}(C, a_i) \\
&= \min_{1 \leq i \leq 3} \mathbf{EC}(C, a_i).
\end{aligned}$$

□

Based on this lemma one might conclude that combining abstention windows is inappropriate to improve expected cost. However, it only shows that the naive method of choosing randomly among classifiers according to a given probability distribution is unsuitable. Therefore, more sophisticated methods for combining the predictions of different abstention windows are necessary.

4.3 Weighted Voting

Of course, any of the previously described meta-classification schemes could be used to combine several models into one abstaining classifier in a straightforward way. We only have to compute a classification model using one of these methods, apply it to a validation set and eventually calculate the optimal abstention window based on the margins of the validation instances.

An alternative idea, which is pursued further now, is to use the estimations for expected cost and the optimal abstention windows computed to create voting classifiers. The prediction of a base level classifier is provided by an abstention window and can be either positive, negative or the choice to abstain. The weight that is given to each vote depends on the expected cost for the corresponding abstention window. Abstention windows receive more weight if they are expected to have low cost and vice versa. To evaluate the performance of this higher level abstaining classifier, the model has to be applied to a separate test set $T \subseteq \mathcal{X}$ as the estimation of expected cost on the validation set would be highly optimistic.

4.3.1 Weighting

So far we have only presented the general idea of how to combine the abstention windows, but not explained in detail in which way weighting and voting among the base level classifiers is to be performed.

We presume a fixed cost scenario which is specified by a cost matrix C. If the costs are unknown cost curves can be used to derive higher level abstaining classifiers for a variety of costs. In the first step, the optimal abstention window for this cost scenario is determined for each classifier and based on the expected costs for this window weights are calculated. Note that the expected cost of each abstention window is estimated from the validation set $S \subseteq \mathcal{X}$ but the predictions of the combined classifier are derived for a separate test set T which is disjoint from both training and validation set. To distinguish between the two sets, the function for expected cost is extended to a third parameter for the set on which the expected costs are computed. Thus, $\mathbf{EC}(C, a, P)$ denotes the expected cost of abstention window a on the set $P \subseteq \mathcal{X}$ given cost matrix C. We use the following notation to describe the optimal abstention window for each classifier.

Definition 4.3. *Let Cl_1, Cl_2, …, Cl_t be the used base level classifiers. The optimal abstention window for each base level model Cl_i is denoted by*

$$a_{opt}(Cl_i) := \text{argmin}_{a \in \mathcal{A}(Cl_i)} \mathbf{EC}(C, a, S).$$

The corresponding lower and upper thresholds are denoted by $l_{opt}(Cl_i)$ and $u_{opt}(Cl_i)$.

Each instance in T is described by t attributes, each of which gives the margin of the specified instance for one of the t base level classifiers. Based on the margin, the prediction of each classifier, that is its optimal abstention window is calculated. For practical reasons, the predictions are now given as numbers with 1 denoting a positive prediction, -1 a negative one and 0 the choice to abstain.

Definition 4.4. *Let $(m_1(x), \ldots, m_t(x))$ be the predicted margins of the t classifiers for an instance $x \in T$. Then the prediction of classifier Cl_i on this instance is given by a function π with*

$$\pi(Cl_i, x) := \begin{cases} 1 & \textit{if } m_i(x) \geq u_{opt}(Cl_i) \\ 0 & \textit{if } l_{opt}(Cl_i) < m_i(x) < u_{opt}(Cl_i) \\ -1 & \textit{if } m_i(x) \leq l_{opt}(Cl_i) \end{cases}$$

We can use these definitions to present several concepts of weighting and combining the predictions of the base classifiers. An intuitive way of combining different classification models is to choose the one with minimum expected cost for each scenario. Thus the final prediction for an instance x is given by $\pi(Cl_i, x)$ if $i = \text{argmin}_{1 \leq q \leq t} \mathbf{EC}(C, a_{opt}(Cl_q), S)$. No weighting is involved at this stage, nevertheless this provides an useful baseline classifier to compare against. Any method combining abstaining classifiers has to outperform the baseline classifiers at least for some cost scenarios to be of relevance.

There are several alternative ways of weighting. Since the weight of a classifier is supposed to increase with decreasing cost, weighting by inverse expected cost is appropriate and the following weight is attached to each classifier Cl_i:

$$w(Cl_i) := \frac{1}{\mathbf{EC}(C, a_{opt}(Cl_i), S)}. \tag{4.3}$$

This results in very high weights for models which have small values of expected cost (i.e. close to zero), whereas classifiers with high values for expected cost are given almost no weight at all. Unfortunately, for very small values of expected cost the resulting weights can become very large.

To avoid such problems we can use an alternative weighting scheme. In this case, the weight of a classifier is given by the sum over the expected costs for the remaining classifiers divided by the sum over all values for expected costs. It is obvious that the weight of a classifier is large if it performs distinctively better than the remaining classifiers and small otherwise. Furthermore, dividing by the total expected cost becomes unnecessary, because it is just a constant normalizing term which has no effect on the final outcome. Hence, the weight of a classifier can be given by

$$w(Cl_i) := \sum_{\substack{1 \leq q \leq t \\ q \neq i}} \mathbf{EC}(C, a_{opt}(Cl_q), S). \tag{4.4}$$

In chapter 6 both methods of weighting are compared and shown to be approximately equivalent.

4.3.2 Voting

Having defined two weighting schemes, we can proceed to explain how final predictions for an instance are determined. As predictions are given by either -1, 0 or 1, the final prediction result on a given instance can be derived by summing up the predictions of each classifier multiplied by the weight given to the classifier. We call this the direct sum method, which requires only a function $\phi(x)$ to be calculated for an instance x with

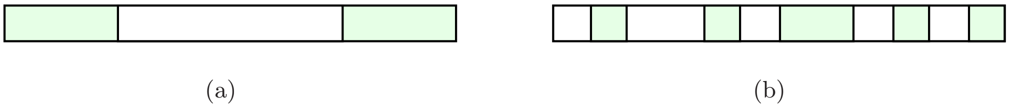
$$\phi(x) = \sum_{1 \leq q \leq t} w(Cl_q) \cdot \pi(Cl_q, x). \tag{4.5}$$

Thus the class prediction $\pi(x)$ for the instance is determined by the sign of $\phi(x)$:

$$\pi(x) = \begin{cases} 1 & \text{if } \phi(x) > 0 \\ 0 & \text{if } \phi(x) = 0 \\ -1 & \text{if } \phi(x) < 0 \end{cases} \tag{4.6}$$

Unfortunately this type of voting is biased against abstaining to a large extent as abstaining is only exercised if either all base classifiers vote for abstaining or the sum of weights for one class exactly equals the sum of weights for the other class, which is rather unlikely. If the amount of abstaining has to be reduced as far as possible, this is the appropriate choice. An alternative method consists of counting the weights for each possible prediction and eventually choosing the one with highest weight. This is called the majority vote method. For this approach the votes for each label are calculated as

$$\phi(x, y) = \sum_{\substack{1 \leq q \leq t, \\ \pi(Cl_q, x) = y}} w(Cl_q). \tag{4.7}$$



(a)

(b)

Figure 4.2: Ordering of instances imposed by different classifier. Instances with successive margin values for one classifier (a) can be scattered widely for another one (b). In figure (a) the instances classified by an abstention window are colored green. Figure (b) shows the same instances dispersed widely over the range of margins of the second classifier.

Thus the final prediction matrix P for an instance is given by

$$\pi(x) = \operatorname*{argmax}_{y \in \{-1,0,1\}} \phi(x,y). \tag{4.8}$$

The expected cost of the final higher level classifiers can be estimated by applying the prediction rules to the test data and computing the false negative rate, false positive rate and positive and negative abstention rate from the counts of each event. Expected cost then is defined exactly as for the validation set.

4.4 The Separate-and-Conquer Approach

In the previous approaches all optimal abstention windows for the set of classifiers are employed simultaneously to determine the final classification. Alternatively, a sequence of abstention windows can be learned which are to be applied one after the other. Let this sequence be given by abstention windows $(a_1, \ldots, a_q)$. Abstention window a_i is applied to those instances which the previous abstention windows $a_1, \ldots, a_{i-1}$ leave unclassified.

This approach is founded on the idea that after applying abstention windows $a_1, \ldots, a_{i-1}$ only instances remain to be classified for which the previous abstention windows were incapable of giving a confident prediction. As the order imposed on the instances by the margin values differs greatly between classifiers, neighboring instances for one classifier may be dispersed widely for another classifier. This situation is illustrated in 4.2. Thus, removing a sequence of instances determined by an abstention window from one classifier may result only in the removal of isolated, scattered instances for another classifier. These instances might have been exactly those for which the second classifier could not give accurate predictions. Having removed them, the second classifier now might be able to resolve the remaining instances successfully.

To learn such a sequence of abstention windows a separate-and-conquer approach is pursued. Separate-and-conquer is a technique commonly applied to rule learning (see also page 77). At each steps several instances are removed from the data set which are already covered in some way and only the remaining instances are used for the next steps. For our purpose, the best abstention window over all instances in the validation set is computed first and afterwards those instances are removed from the validation set which are classified by this window. Only the instances abstained on remain in the set. This procedure is repeated until no instances are left in the validation set or no further changes occur. The complete method is described in algorithm 4.1.

Algorithm 4.1 Separate-and-Conquer algorithm for combining abstention windows. Let S be the validation set and each $x \in S$ described by a vector $(m_1(x), \ldots, m_t(x))$ giving the margins of the t classifiers on that instance. A procedure computeOptimalWindow(S, C, l) is presumed which computes the optimal abstention window for classifier Cl_l given the current set of instances S and the cost matrix C. The final abstention windows are stored in a set W and are to be executed in the exact order in which they have been determined.

```
procedure SeparateAndConquer($S$, $C$)
  $W \leftarrow \emptyset$
  $S' \leftarrow \emptyset$
  while $S \neq \emptyset$ and $S' \neq S$ do
    $S' \leftarrow S$
    $a_{opt} \leftarrow$ computeOptimalWindow($S$, $C$, 1)
    $cl \leftarrow 1$
    for $l \leftarrow 2$ to $t$ do
      $a_{tmp} \leftarrow$ computeOptimalWindow($S$, $C$, $l$)
      if $\mathbf{EC}(C, a_{tmp}, S) < \mathbf{EC}(C, a_{opt}, S)$ then
        $a_{opt} \leftarrow a_{tmp}$
        $cl \leftarrow l$
      end if
    end for
    $W \leftarrow W \cup \{(a_{opt}, cl)\}$
    $S \leftarrow \{x \in S | l_{opt} < m_{cl}(x) < u_{opt}\}$
  end while
  return $W$
end procedure
```

This procedure bears resemblance to the delegating classifier approach presented by Ferri *et al.* [18] since by abstaining an abstention window essentially delegates the classification to the subsequent window. This is not surprising, as delegating classifiers themselves can be regarded as a variation of the separate-and-conquer approach. The major difference of our method to the delegating approach is that an additional validation set is used to calculate the optimal thresholds for delegation and that these thresholds are determined by optimization instead of a simple frequency criterion. Furthermore, all the base classifiers are trained on the same training set and therefore the separate-and-conquer procedure is deferred to the next (higher) level which involves learning the sequence of optimal abstention windows.

Several changes may be imposed on the basic method presented. The sequence of abstention windows obtained by the algorithm tends to abstain less often than the individual optimal abstention windows for this cost scenario due to the design of the separate-and-conquer algorithm which continues learning abstention windows until either all instances in the validation set are classified or no changes occur. However, for low abstention costs, it is often more favorable in terms of expected cost to have higher abstention rate instead of higher (mis)classification rate. Therefore, better results with higher abstention rates can be achieved by decreasing the abstention costs slightly during learning compared to the actual costs. However, there appears to be no clear rule for how much the abstention costs have to be reduced. This depends on the application and in some cases also on the original abstention costs.

Another disadvantage of the original method is that the first abstention window in the sequence in general classifies the majority of instances leaving only a fraction of instances processed further. This counteracts against the desired effect that only instances are supposed to be classified in the first steps which can be done so with high confidence. To circumvent this problem, abstention costs can be chosen at the beginning which are decisively lower than the original costs. With each iteration they are increased until they have reached the level of the original abstention costs. The false positive costs on the contrary are not changed at all throughout the whole time. Consequently, the number of instances classified increases with each step. The first abstention windows chosen are able to classify a fraction of instances with great certainty and the following ones may be able to perform better after those instances have been removed.

4.5 Conclusion

In this chapter we have introduced the idea of combining abstention windows to obtain better predictive performance. For this purpose, two methods were presented. The first one takes a vote among the optimal abstention windows for different classifiers. To account for performance differences between classifiers, the votes are weighted depending on the expected cost of each classifier. The second method learns a sequence of abstention windows which are to be applied to an instance one after the other until a classification has been derived or the last abstention window applied. The performance of these methods will be evaluated in chapter 6 with the help of cost curves.

Of course, the presented methods by far do not represent a complete list of how several abstention windows can be combined to produce higher level classifiers and a variety of other methods may be devised. The aim of this chapter however was not to provide such a conclusive list but to introduce the notion of combining abstaining classifiers and to exemplify some methods to achieve this as well as to illustrate some of the problems which have to be faced.

Chapter 5

Computation of Cost Curves

In the previous chapters, the benefits of abstaining have been motivated and cost curves for abstaining classifiers were introduced. However, for abstaining to be applicable to large-scale analysis, efficient algorithms for computing optimal abstention windows and cost curves are required. In the following we will restrict ourselves to cost curves for unknown costs and presume fixed class distributions to make the presented results and proofs easier to understand. Nevertheless, the algorithms can also be extended to the first type of cost curves as well. Furthermore we assume that the validation set used to calculate optimal abstention windows correctly reflects the underlying class distribution of the classification problem. This allows us to use the alternative definition of normalized expected cost of an abstention window $a \in \mathcal{A}$:

$$\mathbf{NEC}(C, a) = \frac{FN(a) + FP(a) \cdot \mu + (UP(a) + UN(a))\nu}{n}$$

Here $\mu = \frac{C(N,p)}{C(P,n)}$ and $\nu = \frac{C(\perp)}{C(P,n)}$ with $\mu, \nu \in [0:1]$ and n is the number of instances in the validation set S. The second type of cost curves is created by setting the x-axis to μ and the y-axis to ν and plotting the expected cost of the optimal abstention window for each cost scenario.

Although a cost curve is continuous in theory, in order to plot it we have to calculate the optimal abstention window and its expected cost for specific values of μ and ν. Therefore, we introduced a value Δ on page 33 which specifies the number of values evaluated for μ and ν, respectively. Accordingly, a cost curve was defined as a $\Delta \times \Delta$ matrix K such that $k_{i,j}$ is the expected cost of the optimal abstention window for false positive costs $\mu = i/\Delta$ and abstention costs $\nu = j/\Delta$. Increasing values of i therefore correspond to increasing false positive costs and increasing values of j to increasing abstention costs.

We now present two algorithms for efficiently computing cost curves. Both of them presume that the output of the classifier has already been computed on the validation set $S = \{x_1, \ldots, x_n\}$ which is used to learn optimal abstention windows and that the set of margins $M = \{m(x_1), \ldots, m(x_n)\}$ has been obtained. To facilitate computation the margins are sorted and only a vector of sorted margin values is used. As several instances may have the same margin, only distinct margin values are stored and two additional vectors are calculated which contain the number of positive and negative instances for each margin value.

Definition 5.1. *Let $S = \{x_1, \ldots, x_n\}$ be the validation set and $\{y_1, \ldots, y_n\}$ the corresponding class labels. With $\vec{m} = (m_1, \ldots, m_k)$ we denote the vector of distinct margins such that*

$m_1 < \cdots < m_k$ *and* $\forall\, 1 \leq i \leq k\, \exists x_j \in S : m_i = m(x_j)$ *and* $\forall x_j \in S\, \exists 1 \leq i \leq k : m(x_j) = m_i$. *Furthermore the vectors* $\vec{p} = (p_1, \ldots, p_k)$ *and* $\vec{n} = (n_1, \ldots, n_k)$ *are defined as*

$$p_i := |\{x_j \in S | m(x_j) = m_i \text{ and } y_j = P\}|$$

and

$$n_i := |\{x_j \in S | m(x_j) = m_i \text{ and } y_j = N\}|.$$

We have that $p_i + n_i > 0\, \forall\, 1 \leq i \leq k$.

Note that for any algorithm with running time $O(g(n))$ for some function $g(n)$ which takes the sorted vector of margins as input an offset for sorting the margins has to be included in the actual running time. As a consequence the final running time of the algorithm then is $O(g(n) + n \log n)$ which still is $O(g(n))$ if $g(n) = \Omega(n \log n)$.

The naive approach to computing the cost curve would consist of calculating the cost of every abstention window and choosing the one with minimum cost for every cost scenario. Obviously, the number of abstention windows in $\mathcal{A}$ is quadratic in the number of distinct margins k in the validation set since the number of combinations of lower and upper threshold is quadratic in k. Therefore, the running time of this algorithm is $O(\Delta^2 k^2)$. Although the validation set size is in most cases relatively small, the multiplication by Δ^2 leads to a tremendous increase in running time even for small values of k and makes this algorithm unsuitable for practical purposes

In this chapter two algorithms are presented for computing cost curves. The first one begins with determining a relevant subset of abstention windows and continues by finding the abstention window in this subset with minimum cost for selected combinations of μ and ν. The second algorithm avoids memorizing abstention windows by directly computing the window with minimum expected cost for each cost scenario. Both of these algorithms rely on algorithms for the calculation of optimal abstention windows for specific cost scenarios. The first one uses a variation of the naive, quadratic algorithm for calculating the optimal abstention window, whereas the second one employs a linear algorithm as well as additional dependencies between optimal abstention windows for different cost scenarios to additionally improve the effective running time.

The first algorithm for calculating a cost curve is presented in section 5.1. Following this, the running time for calculating an optimal abstention window for a specific cost scenario is improved in section 5.2 from quadratic running time to running time in $O(k \log k)$ to linear running time. Finally, we use this linear algorithm in section 5.3 to create an algorithm for computing a cost curve which operates in time linear in the number of instances in the validation set. Readers solely interested in the linear algorithm may skip section 5.1 and the first part of section 5.2 and directly go to page 63. However, to fully understand the presented concepts definitions and lemmas introduced in the preceeding sections are necessary.

5.1 The 3CSAW Algorithm

The 3CSAW (**C**omputing **C**ost **C**urves from a **S**ubset of **A**bstention **W**indows) algorithm consists of two steps. First a subset of all possible abstention windows $\widehat{\mathcal{A}} \subseteq \mathcal{A}$ is derived such that no abstention window $a \in \mathcal{A} \setminus \widehat{\mathcal{A}}$ can ever be optimal for any cost scenario. Afterwards the optimal abstention window for each cost scenario is computed from this subset by using a combination of bounds on expected cost and dynamic programming.

To properly describe this algorithm, first the necessary notation has to be introduced. As only μ and ν are variable in the definition of normalized expected cost above, a new function is used to describe normalized expected cost. As the division by n does not change the outcome of any comparisons based on this cost function, it is omitted here.

Definition 5.2. *Given an abstention window $a \in \mathcal{A}$, the function $cost(a, \mu, \nu)$ denotes the cost of this abstention window on the validation set S:*

$$cost(a, \mu, \nu) := FN(a) + FP(a)\,\mu + (UP(a) + UN(a))\,\nu = \mathbf{NEC}(C, a)\,n$$

with $C(P,\, n) = 1$, $C(N,\, p) = \mu$ and $C(\perp) = \nu$.

As we do not distinguish between abstaining on positive or negative instances, the counts for each of those events are gathered in one value. Note, that we do not consider the frequencies of abstaining or misclassification events, but the actual number of their occurrences.

Definition 5.3. *Given an abstention window a, $A(a)$ is defined as the number of instances a abstains on if applied to S. Hence,*

$$A(a) := UP(a) + UN(a)$$

with $UP(a)$ the number of positive instances in the validation set a abstains on and $UN(a)$ the number of negative instances.

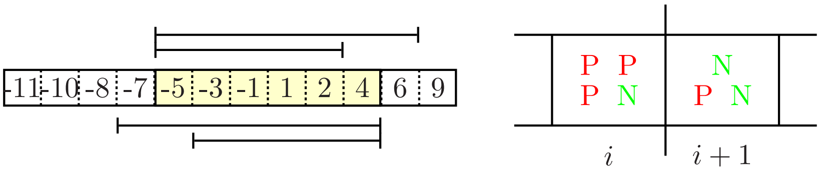
In chapter 2 abstention windows were defined such that both the lower and the upper threshold lies exactly between two adjacent margin values. For simplification a function is introduced which determines the value of a threshold given the index of the margin value closest to the threshold from below. If the threshold is to lie below the lowest margin value m_1 or above the largest margin value m_k, a user specified value ε gives the difference of the threshold to m_1 or m_k, respectively.

Definition 5.4. *Let $\varepsilon > 0$ be an arbitrary but constant value. The function $v : \{0, \dots, k\} \to \mathbb{R}$ is defined as*

$$v(i) = \begin{cases} \frac{m_i + m_{i+1}}{2} & \textit{if } 1 \leq i < k \\ m_1 - \varepsilon & \textit{if } i = 0 \\ m_k + \varepsilon & \textit{if } i = k. \end{cases}$$

Hence, the function $v(i)$ calculates the threshold which lies between two margin values m_i and m_{i+1}. Note that after sorting the margin values, the actual values of the margins are no longer relevant for the calculation of the cost curve except for determining the threshold values. The only necessary information is the number of positive and negative instances corresponding to each margin value as well as the index of the margin values closest to the thresholds from below. An abstention window a then is described by two indices i and j such that the value of the lower threshold is $v(i)$ and the value of the upper threshold $v(j)$. The expected cost of this abstention window is determined as

$$cost(a, \mu, \nu) = \underbrace{\sum_{1 \leq s \leq i} p_s}_{FN(a)} + \mu \underbrace{\sum_{j < s \leq k} n_s}_{FP(a)} + \nu \underbrace{\sum_{i < s \leq j} (n_s + p_s)}_{A(a)}.$$



(a) (b)

Figure 5.1: Figure (a) shows the successors for an abstention window on $\vec{m} = (-11, -10, -8, -7, -5, -3, -1, 1, 2, 4, 6, 9)$. The original abstention window is shown in yellow. The successor windows are depicted by lines below or above the margin vector. Figure (b) shows for a threshold $v(i)$ the two adjacent margin values m_i and m_{i+1}. In this case $p_i = 3$, $n_i = 1$, $p_{i+1} = 1$ and $n_{i+1} = 2$. As both $n_i > 0$ and $p_{i+1} > 0$, an abstention window with lower or upper threshold $v(i)$ cannot be excluded beforehand.

We also define a successor function on an abstention window a which calculates all abstention windows a' that are created by increasing or decreasing the lower or upper threshold of a by only one step. The set of these windows is denoted as the successors of a (see figure 5.1(a)).

Definition 5.5. *Let $a = (v(i), v(j))$ be an abstention window in $\mathcal{A}$. The successor function $succ : \mathcal{A} \rightarrow \mathcal{A}^+$ is defined as*

$$succ(a) = \bigcup \begin{cases} \{(v(i-1), v(j))\} & \text{if } 1 \leq i \\ \{(v(i), v(j+1))\} & \text{if } j < k \\ \{(v(i+1), v(j)), (v(i), v(j-1))\} & \text{if } i < j \\ \{(v(i+1), v(j+1))\} & \text{if } i = j \wedge j < k \\ \{(v(i-1), v(j-1))\} & \text{if } i = j \wedge 1 \leq i. \end{cases}$$

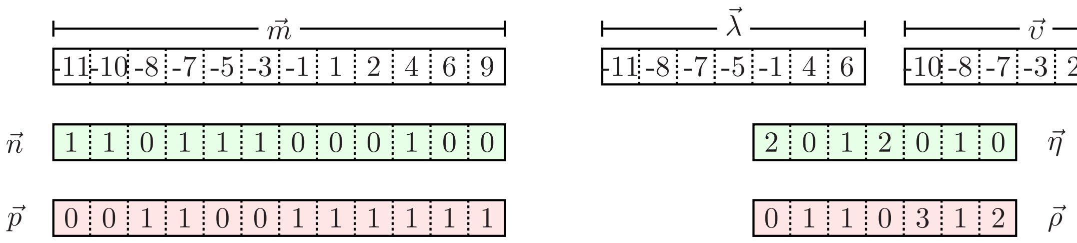
Although the number of abstention windows in $\mathcal{A}$ is quadratic in the number of distinct margins, only those abstention windows are eligible for minimum cost for which no successor has lower expected cost. The following lemma provides a characteristic for this type of abstention windows.

Lemma 5.6. *Let μ and ν be the costs for false positives and abstention respectively, with $0 < \mu, \nu \leq 1$. Let $a_{opt} = (v(i), v(j))$ be the optimal abstention window for this cost scenario, i.e. $a_{opt} := \operatorname{argmin}_{a \in \mathcal{A}} cost(a, \mu, \nu)$. Then we have that $n_i > 0$ and $p_{i+1} > 0$ if $i > 0$ as well as $n_j > 0$ and $p_{j+1} > 0$ if $j < k$.*

Proof. For $i = j$, the lemma follows directly from the optimality of a_{opt} as for both $n_i = 0$ or $p_{i+1} = 0$, we could improve expected cost by decreasing or increasing the threshold.
Thus we will now assume that $i < j$ and prove that both n_i and p_{i+1} have to be greater than zero for $i > 0$. (see also figure 5.1(b)). The proof for $n_j > 0$ and $p_{j+1} > 0$ follows analogously. From $i < j$ and the definition of $\mathcal{A}$, we know that $A(a_{opt}) = |\{x \in S | m_{i+1} \leq m(x) \leq m_j\}| > 0$. Thus we can conclude from theorem 2.18 that

$$\nu \leq \frac{\mu}{1+\mu} < \mu \leq 1. \tag{5.1}$$

Assume now that $n_i = 0$. As a consequence, we have that $p_i > 0$ (Def. 5.1). Now let $a_c = (v(i-1), v(j))$ be the successor of a_{opt} which results from decreasing the lower threshold.



(a)

(b)

Figure 5.2: This figure illustrates the preprocessing step for the calculation of optimal abstention windows. Figure (a) shows example values for $\vec{m}$, $\vec{n}$ and $\vec{p}$ and figure (b) the $\vec{\lambda}$, $\vec{\upsilon}$, $\vec{\eta}$ and $\vec{\rho}$ which are calculated from the original vectors in the preprocessing step.

This successor exists since $i > 0$. As the predictions of a_c and a_{opt} differ only for those instances with margin m_i, the difference in expected cost between a_c and a_{opt} is

$$\begin{aligned} cost(a_c, \mu, \nu) - cost(a_{opt}, \mu, \nu) =& FN(a_c) - FN(a_{opt}) + (FP(a_c) - FP(a_{opt}))\,\mu \\ &+ (A(a_c) - A(a_{opt}))\,\nu \\ =& (p_i + n_i)\nu - p_i = p_i(\nu - 1) \overset{\text{Equ. (5.1)}}{<} 0, \end{aligned}$$

which is a contradiction to the optimality of a_{opt}.
If $p_{i+1} = 0$, we have $n_{i+1} > 0$. Let now $a_c = (v(i+1), v(j))$ be the successor of a_{opt} to be considered. This successor exists because $i < j$. The difference in expected cost between a_c and a_{opt} then is

$$cost(a_c, \mu, \nu) - cost(a_{opt}, \mu, \nu) = p_{i+1} - (n_{i+1} + p_{i+1})\nu = -n_{i+1}\,\nu < 0.$$

This again is a contradiction. □

From this lemma we can conclude that only abstention windows have to be considered which exhibit the described characteristic. This means that if the lower threshold lies between two margins m_i and m_{i+1}, at least one negative instance must have margin m_i and at least one positive instance must have margin m_{i+1}. The same applies to the upper threshold. As a consequence, if we have a sequence of margins $m_i, \ldots, m_j$ which either corresponds only to positive instances – i.e. $n_q = 0\,\forall\, i \leq q \leq j$ – or negative instances – i.e. $p_q = 0\,\forall\, i \leq q \leq j$ –, none of the thresholds $v(q)$, $i \leq q < j$ is relevant as lower or upper threshold for an abstention window. For this reason, successive margin values which correspond to instances of the same class can be collected in a preprocessing step such that each sequence of instances of the same class is represented by a constant number of values. Now two vectors are required to store the margin values as a sequence of margin values is described by its smallest and largest value. Two additional vectors give the number of positive and negative instances for each sequence of margin values. Note that one of those numbers has to be zero except if the smallest and largest value are equal, i.e. there are both negative and positive instances having the same margin value. See figure 5.2 for an example.

Definition 5.7. *Given the vectors* $\vec{m}$*,* $\vec{p}$ *and* $\vec{n}$ *the vectors* $\vec{\lambda} = (\lambda_1, \ldots, \lambda_t)$*,* $\vec{\upsilon} = (\upsilon_1, \ldots, \upsilon_t)$*,* $\vec{\rho} = (\rho_1, \ldots, \rho_t)$ *and* $\vec{\eta} = (\eta_1, \ldots, \eta_t)$ *are defined such that* $\lambda_1 < \lambda_2 < \cdots < \lambda_t$*,* $\upsilon_1 < \upsilon_2 < \cdots < \upsilon_t$ *and* $\forall\, 1 \leq i < t\, \exists 1 \leq r \leq s \leq k : \lambda_i = m_r \wedge \upsilon_i = m_s$*. Additionally,*

$$\rho_i := |\{x_j \in S | \lambda_i \leq m(x_j) \leq \upsilon_i \wedge y_j = P\}|$$

and

$$\eta_i := |\{x_j \in S | \lambda_i \leq m(x_j) \leq \upsilon_i \wedge y_j = N\}|.$$

Furthermore we require that $\nexists 1 \leq i < t$ *such that* $\rho_i = 0$ *and* $\rho_{i+1} = 0$ *or* $\eta_i = 0$ *and* $\eta_{i+1} = 0$*. If* $\rho_i > 0$ *and* $\eta_i > 0$ *for* $1 \leq i \leq t$ *then we have* $\lambda_i = \upsilon_i$*.*

These vectors can be computed in time $O(k)$ as described in algorithm 5.1. For this purpose the margin vector $(m_1, \ldots, m_k)$ is passed over step by step. A new entry in the preprocessed vector is created whenever different margin values which correspond to both positive and negative instances would have to be collected in one entry. For each entry in $\vec{m}$ the counts of positive and negative instances and the smallest and largest margin values are updated in the preprocessed vectors. The length of the preprocessed vectors t in general is much smaller than the number of distinct margin values k. This is due to the fact that the margin of an instance reflects both the prediction for an instance as well as the confidence associated with it. Therefore, for negative margin values we will have long sequences of

Algorithm 5.1 Preprocessing step for the computation of abstention windows. The algorithm calculates the vectors $\vec{\lambda}$, $\vec{\upsilon}$, $\vec{\rho}$ and $\vec{\eta}$ as introduced in definition 5.7 and takes the margin vector as input, as well as the vectors containing the counts of positive or negative instances for each margin value.

```
procedure preProcess($\vec{m}$, $\vec{p}$, $\vec{n}$)
    $\lambda_1, \upsilon_1 \leftarrow m_1$,
    $\rho_1 \leftarrow p_1$
    $\eta_1 \leftarrow n_1$
    $t \leftarrow 1$
    for $j \leftarrow 2$ to $k$ do
        if $(p_j \neq 0 \vee \rho_t \neq 0) \wedge (n_j \neq 0 \vee \eta_t \neq 0)$ then
            ▷ Create a new entry in the vector
            $t \leftarrow t + 1$
            $\lambda_t \leftarrow m_j$
            $\rho_t \leftarrow 0$
            $\eta_t \leftarrow 0$
        end if
        ▷ Update counts and largest margin value of the sequence
        $\rho_t \leftarrow \rho_t + p_j$
        $\eta_t \leftarrow \eta_t + n_j$
        $\upsilon_t \leftarrow m_j$
    end for
    return $\vec{\lambda}$, $\vec{\upsilon}$, $\vec{\rho}$, $\vec{\eta}$
end procedure
```

negative instances interrupted occasionally by positive instances and vice versa for positive margin values long sequences of positive instances interrupted by negative ones.

Analogously to the function υ from definition 5.4, a function ψ is introduced which operates on the preprocessed vectors instead of the original margin values.

Definition 5.8. *Given the vectors $\vec{\lambda}$, $\vec{\upsilon}$, $\vec{\rho}$ and $\vec{\eta}$ the function $\psi : \{0, \ldots, t\} \to \mathbb{R}$ is defined as*

$$\psi(i) = \begin{cases} \frac{\upsilon_i + \lambda_{i+1}}{2} & \textit{if } 1 \leq i < t \\ \lambda_1 - \varepsilon & \textit{if } i = 0 \\ \upsilon_t + \varepsilon & \textit{if } i = t. \end{cases}$$

The preprocessed margins represent the starting-point for the computation of potential abstention windows. The pseudocode for this method is given in algorithm 5.2. Essentially, any combination of lower and upper threshold on the preprocessed vector is considered. The lower threshold is increased step by step within an outer loop, whereas the upper threshold is

Algorithm 5.2 The complete algorithm for the computation of abstention windows. It computes a subset of abstention windows $\widehat{\mathcal{A}} \subseteq \mathcal{A}$, such that none of the abstention windows in $\mathcal{A} \setminus \widehat{\mathcal{A}}$ can be optimal for any cost scenario and takes as input the number of instances in the validation set n and $\vec{\lambda}$, $\vec{\upsilon}$, $\vec{\rho}$ and $\vec{\eta}$ (see definition 5.7).

```
procedure computeWindows($n$, $\vec{\lambda}$, $\vec{\upsilon}$, $\vec{\rho}$, $\vec{\eta}$)
    $FN, TN \leftarrow 0$
    $\widehat{\mathcal{A}} \leftarrow \emptyset$
    $l \leftarrow 0$
    while $l \leq t$ do
        $FP, TP \leftarrow 0$
        $u \leftarrow t$
        while $u \geq l$ do
            $a \leftarrow (\psi(l), \psi(u))$
            $FN(a) \leftarrow FN$, $FP(a) \leftarrow FP$, $A(a) \leftarrow n - FP - TP - FN - TN$
            $\widehat{\mathcal{A}} \leftarrow \widehat{\mathcal{A}} \cup \{a\}$
            repeat
                $FP \leftarrow FP + \eta_u$
                $TP \leftarrow TP + \rho_u$
                $u \leftarrow u - 1$
            until $u = l \vee (\eta_u > 0 \wedge \rho_{u+1} > 0)$
        end while
        repeat
            $FN \leftarrow FN + \rho_{l+1}$
            $TN \leftarrow TN + \eta_{l+1}$
            $l \leftarrow l + 1$
        until $l = t \vee (\eta_l > 0 \wedge \rho_{l+1} > 0)$
    end while
    return $\widehat{\mathcal{A}}$
end procedure
```

Algorithm 5.3 Naive algorithm for the computation of the cost curve. Let $\widehat{\mathcal{A}}$ be the set of abstention windows computed before and Δ the number of values between 0 and 1 that are to be evaluated for μ and ν respectively. The cost curve is stored in a matrix K.

```
procedure computeCostCurve($\widehat{\mathcal{A}}$, $\Delta$)
    for $i \leftarrow 0$ to $\Delta$ do
        for $j \leftarrow 0$ to $\Delta$ do
            $k_{i,j} \leftarrow \min_{a \in \widehat{\mathcal{A}}} cost(a, i/\Delta, j/\Delta)$
        end for
    end for
    return $K$
end procedure
```

decreased step by step within an inner loop until it meets the lower threshold. At each step the counts for false and true positives or negatives are updated. Abstention windows are only stored for later use if they exhibit the characteristic described in lemma 5.6. This means that among the instances closest to both lower and upper threshold from below is at least one negative instance and among the ones closest from above at least one positive instance. By using the preprocessed vectors, we have already excluded a large number of windows from $\widehat{\mathcal{A}}$ which can never fulfill the criterion from lemma 5.6, because the threshold would be within a sequence of instances of the same class. The abstention windows omitted additionally in this algorithm have thresholds separating a sequence of only positive instances below the threshold from a sequence of only negative instances above the threshold. Any of these windows could be improved by changing the thresholds, therefore they cannot be optimal for any cost scenario and are irrelevant for our purpose.

The running time of this algorithm is linear in the number of abstention windows considered and accordingly quadratic in the length of the preprocessed vectors.

Theorem 5.9. *The subset of potential abstention windows can be computed in $O(t^2)$, where t is the length of $\vec{\lambda}$, $\vec{v}$, $\vec{\rho}$ and $\vec{\eta}$, respectively.*

Proof. From the pseudocode of the algorithm it is obvious, that the running time is determined by the number of combinations of l and u considered, as for each of those combinations only a constant number of operations is performed. As we have that $0 \leq l \leq u \leq t$ there are only $O(t^2)$ such combinations, thus the running time is quadratic in t. □

We know that $t \leq k \leq n$, with k the number of instances with different margin values and n the total number of instances. Thus in the worst case the running time of the algorithm is quadratic in the number of instances. Nevertheless, in most cases t will be much smaller than both n and k, as we have seen before.

From the subset of abstention windows $\widehat{\mathcal{A}}$ the cost curve can be calculated. The naive implementation finds the optimal abstention window for every combination of μ and ν by calculating the expected cost for each abstention window in $\widehat{\mathcal{A}}$ and then choosing the one with minimal cost (see algorithm 5.3). The running time of this naive implementation is $O(\Delta^2|\widehat{\mathcal{A}}|)$ which is $O(\Delta^2 t^2)$ in the worst case.

We now propose an improved algorithm which uses a combination of dynamic programming and bounds on expected cost to reduce the effort for finding the optimal abstention windows. For this purpose the abstention windows are divided into disjoint subsets of approximately equal size. In each subset the value for the false negatives is constant. Furthermore, we assume that the abstention windows are sorted in ascending order by false positives and consequently in descending order by the number of abstained instances.

This division as well as the sorting can easily be performed while calculating the abstention windows in algorithm 5.2 without further computational effort since for a given value of l the false negatives are constant. Furthermore the false positives increase and the abstained instances decrease while u is decremented. For every subset we can compute a lower bound on the cost of each of its abstention windows by using the value for the false negatives of the subset and the lowest number of false positives and abstained instances of any abstention window of this subset. The minimum values can also be determined beforehand, so that the bound for a subset of $\widehat{\mathcal{A}}$ can be determined in constant time during the computation of the curve. One after the other, the subsets are evaluated for the current cost scenario by comparing the lower bound of the subset with the best value of expected cost so far. If the lower bound on the expected cost for a subset exceeds this minimum cost, no abstention window of this subset is evaluated, since any of them would lead to an increase in cost. If the bound is below this value of expected cost, we calculate the expected cost of each window in this subset.

To get a good initial guess for the minimal cost and therefore exclude many subsets of $\widehat{\mathcal{A}}$ at each step, the optimal abstention windows of the two cost scenarios most similar are evaluated for the current scenario and the best of these is chosen. The most similar cost scenarios are those for which one of the two cost types is the same as in the current scenario, and only the other one is smaller by $\frac{1}{\Delta}$. This means that when calculating the optimal abstention window for specific values of $\mu = i/\Delta$ and $\nu = j/\Delta$, we consider the optimal abstention windows for $\mu = i/\Delta$ and $\nu = (j-1)/\Delta$ and $\mu = (i-1)/\Delta$ and $\nu = j/\Delta$, respectively. The idea behind this approach is that an abstention window which is optimal for similar cost scenarios will at least be close to the minimum cost for the current cost scenario. At this point we use dynamic programming to avoid having to compute the optimal abstention window for the similar cost scenarios all over again and store the indices of the optimal abstention windows for each cost scenario together with the actual value for expected cost.

In the worst case no subset can be omitted and the running time is still $O(\Delta^2|\widehat{\mathcal{A}}|)$, while in the best case all of them are omitted and the running time is $O(\Delta^2\frac{|\widehat{\mathcal{A}}|}{s})$ with s the average size of the subsets. The choice of the parameter s has a large influence on the running time. The larger s the larger are the subsets and thus more abstention windows are skipped if a subset is omitted. On the other hand the smaller s, the tighter is the bound for each subset and thus more subsets can actually be excluded.

This method improves the effective running time decisively compared to the naive implementation. However, even further refinements are possible as increasing the costs for abstention without changing the costs for false positives has no effect on the expected cost of abstention windows which do not abstain at all. Thus, the optimal abstention window will not change any more as soon as it no longer abstains. This is stated formally in the following lemmas.

Lemma 5.10. *Let μ, ν_1, $\nu_2 \geq 0$ with $\nu_1 < \nu_2$ and $\mathcal{A}$ be the set of all abstention windows. Let $a_1 := \operatorname{argmin}_{a\in\mathcal{A}} cost(a, \mu, \nu_1)$. For any abstention window a_2 with $cost(a_2, \mu, \nu_2) < cost(a_1, \mu, \nu_2)$, we have that $A(a_2) < A(a_1)$.*

Proof. By contradiction:
Assume there exists an abstention window a_2 with $cost(a_2, \mu, \nu_2) < cost(a_1, \mu, \nu_2)$ and $A(a_2) \geq A(a_1)$. Thus we get

$$\begin{aligned}
&cost(a_2, \mu, \nu_1) - cost(a_1, \mu, \nu_1) \\
&= FN(a_2) - FN(a_1) + \mu(FP(a_2) - FP(a_1)) + \nu_1(A(a_2) - A(a_1)) \\
&\leq FN(a_2) - FN(a_1) + \mu(FP(a_2) - FP(a_1)) + \nu_2(A(a_2) - A(a_1)) \quad (5.2) \\
&= cost(a_2, \mu, \nu_2) - cost(a_1, \mu, \nu_2) < 0 \quad (5.3)
\end{aligned}$$

Equation (5.2) results from the fact that $A(a_2) - A(a_1) \geq 0$ and that $\nu_1 < \nu_2$ and equation (5.3) is a contradiction to the optimality of a_1 for μ and ν_1. □

This lemma implies that when increasing the abstention costs, the number of abstained instances either decreases or the expected cost of the optimal abstention window does not change. Furthermore it can be concluded that as soon as the optimal abstention window for a certain cost scenario does not abstain any more, it is also optimal for all cost scenarios with the same false positive costs but higher abstention costs.

Corollary 5.11. *Let μ, ν_1, $\nu_2 \geq 0$ with $\nu_1 < \nu_2$. If $a_1 = \operatorname{argmin}_{a\in\mathcal{A}} cost(a, \mu, \nu_1)$ and $A(a_1) = 0$, we have that $a_1 = \operatorname{argmin}_{a\in\mathcal{A}} cost(a, \mu, \nu_2)$.*

Proof. By contradiction:
Assume there exists an abstention window $a_c \in \mathcal{A}$ with $cost(a_c, \mu, \nu_2) < cost(a_1, \mu, \nu_2)$. Lemma 5.10 then implies that $A(a_c) < A(a_1) = 0$. As the number of instances abstained on can never be negative, this is a contradiction. □

Corollary 5.11 suggests a refinement of the presented algorithm. As soon as the optimal abstention window for any ν_1 and μ_1 does no longer abstain on any instance, we can use this window for any cost scenario ν_2 and μ_2 with $\nu_2 > \nu_1$ and $\mu_2 = \mu_1$. Therefore, only constant time is required for these cost scenarios. Algorithm then 5.4 describes the complete method for computing the cost curve from $\widehat{\mathcal{A}}$.

The complete 3CSAW procedure is composed of algorithms 5.1, 5.2 and 5.4. It is based mostly on the fact that a large number of abstention windows can be discarded before computing the cost curve as they cannot be optimal for any cost scenario. However, a large number of abstention windows are retained which will never be optimal, but cannot be excluded without knowing the exact costs. The next step is to further improve the running time by directly computing the optimal abstention window for each cost scenario. As this is only possible if the optimal abstention window for a given cost scenario can be computed efficiently, we first focus on this problem. Note that any useful algorithm for this problem has to have a running time in $o(k^2)$ and thereby be better than the naive approach which computes the expected cost of every abstention window.

5.2 Computing the Optimal Abstention Window

For this section the preprocessing step is avoided and the vector of distinct margins $(m_1, \ldots, m_k)$ is used again, since the preprocessing step already requires linear time and the best algorithm

Algorithm 5.4 Improved Algorithm for computing the cost curves using dynamic programming and lower bounds on costs. Δ is defined as in algorithm 5.3. We are given a set W of subsets from $\widehat{\mathcal{A}}$ such that $\cup_{w \in W} w = \widehat{\mathcal{A}}$ and $w \cup w' = \emptyset \, \forall w \neq w' \in W$. The cost values are stored in a matrix K. Furthermore we assume that we can access any abstention window as $a_{r,s}$, where r denotes the subset in which the abstention window is contained and s the index of the abstention window within the subset. The index of the subset containing the optimal abstention window is stored in a matrix G and the index of the optimal abstention window within its subset in a matrix T. Furthermore we have a function $\mathit{bound}(w)$ which computes the lower bound for a subset w of $\widehat{\mathcal{A}}$.

```
**procedure** computeCostCurve($W$, $\Delta$)
  **for** $i \leftarrow 0$ to $\Delta$ **do**
    **for** $j \leftarrow 0$ to $\Delta$ **do**
      $\triangleright$ Test if the optimal abstention window for $\nu = (j-1)/\Delta$ does abstain at all
      **if** $j > 0 \wedge A(a_{g_{i,j-1},t_{i,j-1}}) = 0$ **then**
        $k_{i,j} \leftarrow k_{i,j-1}$
        $g_{i,j} \leftarrow g_{i,j-1}$, $t_{i,j} \leftarrow t_{i,j-1}$
      **else**
        $\triangleright$ Initial guess for minimum expected cost
        **if** $i = 0 \wedge j = 0$ **then**
          $k_{i,j} \leftarrow \infty$, $g_{i,j} \leftarrow -1$, $t_{i,j} \leftarrow -1$
        **else if** $i = 0$ **then**
          $k_{i,j} \leftarrow \mathit{cost}(a_{g_{i,j-1},t_{i,j-1}}, i/\Delta, j/\Delta)$
          $g_{i,j} \leftarrow g_{i,j-1}$, $t_{i,j} \leftarrow t_{i,j-1}$
        **else if** $j = 0$ **then**
          $k_{i,j} \leftarrow \mathit{cost}(a_{g_{i-1,j},t_{i-1,j}}, i/\Delta, j/\Delta)$
          $g_{i,j} \leftarrow g_{i-1,j}$, $t_{i,j} \leftarrow t_{i-1,j}$
        **else**
          $k_{i,j} \leftarrow \min \begin{cases} \mathit{cost}(a_{g_{i,j-1},t_{i,j-1}}, i/\Delta, j/\Delta) \\ \mathit{cost}(a_{g_{i-1,j},t_{i-1,j}}, i/\Delta, j/\Delta) \end{cases}$
          Update $g_{i,j}$ and $t_{i,j}$ such that $\mathit{cost}(a_{g_{i,j},t_{i,j}}, i/\Delta, j/\Delta) = k_{i,j}$
        **end if**
        $\triangleright$ Compute actual minimum cost using bounds
        **for all** $w \in W$ **do**
          **if** $\mathit{bound}(w) < k_{i,j}$ **then**
            $k_{i,j} \leftarrow \min \begin{cases} k_{i,j} \\ \min_{a \in w} \mathit{cost}(a, i/\Delta, j/\Delta) \end{cases}$
            Update $g_{i,j}$ and $t_{i,j}$ such that $\mathit{cost}(a_{g_{i,j},t_{i,j}}, i/\Delta, j/\Delta) = k_{i,j}$
          **end if**
        **end for**
      **end if**
    **end for**
  **end for**
  **return** $K$
**end procedure**
```

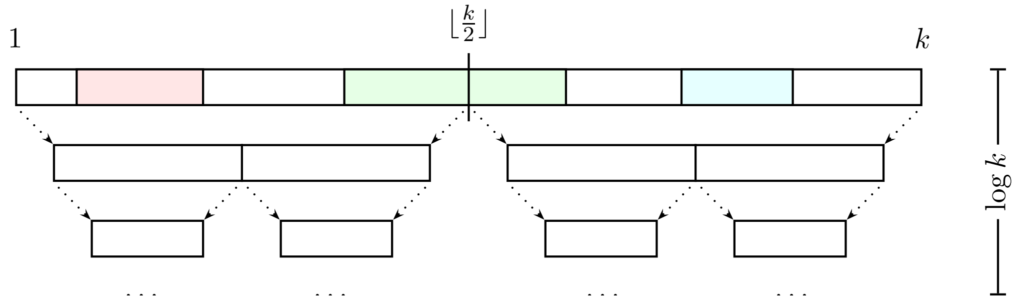


Figure 5.3: Computation of the optimal abstention window with divide-and-conquer. The vector of values is divided into two sub-vectors. The optimal abstention window for the left vector (red) and the optimal abstention window for the right vector (blue) are computed recursively. Finally the optimal window is computed which crosses the divide (green), and then the best one of those three is taken.

we present for finding the optimal abstention window is linear as well. To begin with a divide-and-conquer algorithm is introduced and later on its fundamental idea is extended to develop the linear algorithm.

5.2.1 The Divide-and-Conquer Algorithm

A divide-and-conquer approach typically partitions the problem at hand into several smaller subproblems which have the form of the original problem [10]. The subproblems are then solved recursively and the solution of the original problem is derived by combining the solutions of each subproblem. Thus, three essential steps can be distinguished: the divide step, the conquer step and finally the combine step.

In our case the divide step consists of splitting the margin vector into two sub-vectors $(m_1, \ldots, m_p)$ and $(m_{p+1}, \ldots, m_k)$ with $p = \lfloor \frac{k}{2} \rfloor$. Now the optimal abstention windows for the left and right vector are determined recursively. In the combine step the best abstention window $a = (v(i), v(j))$ is computed with $i \leq p$ and $j \geq p$ (see figure 5.3). The final solution is created by choosing the window with minimal expected cost from the three candidates.

The combine step has a major influence on the running time achieved. If every combination of values for the lower and upper threshold had to be considered in this step, the running time of this algorithm would be even worse than the one of the naive algorithm, since the combine step alone would take time $O(k^2)$. What makes this algorithm more efficient than the naive one is the fact that the optimal lower and upper threshold in the combine step can be determined independently of each other.

Lemma 5.12. *Let $(m_1, \ldots, m_k)$ be the predicted margins, μ and ν defined as before and $1 \leq p \leq k$. Let $\mathcal{L} := \{a = (v(i), v(j)) | a \in \mathcal{A} \wedge i \leq p \wedge j = p\}$ be the set of abstention windows with lower threshold equal to or smaller than $v(p)$ and upper threshold $v(p)$, $\mathcal{U} := \{a = (v(i), v(j)) | a \in \mathcal{A} \wedge i = p \wedge j \geq p\}$ the set of abstention windows with lower threshold $v(p)$ and upper threshold equal to or larger than $v(p)$ and $\mathcal{G} := \{a = (v(i), v(j)) | a \in \mathcal{A} \wedge i \leq p \wedge j \geq p\}$ the set of windows with lower threshold below $v(p)$ and upper threshold above $v(p)$. Let $a_l := \operatorname{argmin}_{a \in \mathcal{L}} cost(a, \mu, \nu)$ and $a_u := \operatorname{argmin}_{a \in \mathcal{U}} cost(a, \mu, \nu)$. Then we have for the abstention window $a_g = (l_l, u_u)$ that $a_g = \operatorname{argmin}_{a \in \mathcal{G}} cost(a, \mu, \nu)$.*

Proof. By contradiction:
Assume there exists an abstention window $a_c \in \mathcal{G}$ such that $cost(a_c, \mu, \nu) < cost(a_g, \mu, \nu)$. Now we define two abstention windows $a_e = (l_c, v(p))$ and $a_f = (v(p), u_c)$. (See figure 5.4) Thus we observe

$$\begin{aligned} & cost(a_e, \mu, \nu) - cost(a_l, \mu, \nu) \\ & = FN(a_e) - FN(a_l) + \mu(FP(a_e) - FP(a_l)) + \nu(A(a_e) - A(a_l)) \\ & = FN(a_c) - FN(a_g) + \mu\, O + \nu(A(a_e) - A(a_l)) \\ & = FN(a_c) - FN(a_g) + \nu(A(a_e) - A(a_l)) \end{aligned} \tag{5.4}$$

Analogously we have

$$cost(a_f, \mu, \nu_1) - cost(a_u, \mu, \nu_1) = \mu(FP(a_c) - FP(a_g)) + \nu(A(a_f) - A(a_u)) \tag{5.5}$$

By adding up equations (5.4) and (5.5) we get

$$\begin{aligned} & cost(a_e, \mu, \nu) - cost(a_l, \mu, \nu) + cost(a_f, \mu, \nu) - cost(a_u, \mu, \nu) \\ & = FN(a_c) - FN(a_g) + \nu(A(a_e) - A(a_l)) + \mu(FP(a_c) - FP(a_g)) + \nu(A(a_f) - A(a_u)) \\ & = FN(a_c) - FN(a_g) + \mu(FP(a_c) - FP(a_g)) + \nu(A(a_c) - A(a_g)) \\ & = cost(a_c, \mu, \nu) - cost(a_g, \mu, \nu) \overset{(*)}{<} 0 \end{aligned} \tag{5.6}$$

$(*)$ follows from the definition of a_c. But equation (5.6) implies that either $cost(a_e, \mu, \nu) - cost(a_l, \mu, \nu) < 0$ or $cost(a_f, \mu, \nu) - cost(a_u, \mu, \nu) < 0$ which is a contradiction to the definition of a_l and a_u. □

From this lemma we can conclude that in the combine step we can first compute the optimal lower threshold and afterwards the optimal upper threshold and then combine these results to yield the best abstention window crossing the split point. This is a direct consequence of our definition of the cost function. The cost of any abstention window $a = (l, u)$ with $l \leq v(p) \leq u$ can be determined by first summing up the costs of the abstention windows $a_l = (l, v(p))$ and $a_u = (v(p), u)$ and then subtracting the expected cost of the abstention window $a_p = (v(p), v(p))$. As the cost for a_p is constant, the overall cost can be minimized by separately minimizing the costs for a_l and a_u. Both the optimal lower and upper threshold can be computed in linear time. Therefore the divide-and-conquer approach improves the

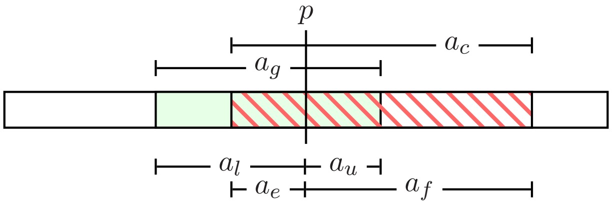


Figure 5.4: This figure illustrates the proof for lemma 5.12. $a_l = \text{argmin}_{a \in \mathcal{L}} cost(a, \mu, \nu)$ and $a_u = \text{argmin}_{a \in \mathcal{U}} cost(a, \mu, \nu)$. a_g is the abstention window we get by combining the lower threshold from a_l and the upper threshold from a_u. The lemma shows that this is the optimal abstention window with $l \leq v(p) \leq u$.

Algorithm 5.5 Divide-and-conquer algorithm for computing the abstention window with minimal cost given values for μ and ν.

```
procedure computeOptimalWindow($\mu$, $\nu$)
    return computeMinWindow(1, $k$, $\mu$, $\nu$)
end procedure
```

Algorithm 5.6 Recursive Algorithm for computing the optimal abstention window for $(m_i, \ldots, m_j)$. $(m_1, \ldots, m_k)$, $(p_1, \ldots, p_k)$ and $(n_1, \ldots, n_k)$ are stored in global variables.

```
 1: procedure computeMinWindow($i$, $j$, $\mu$, $\nu$)
 2:     ▷ divide step
 3:     $q \leftarrow \lfloor \frac{i+j}{2} \rfloor$
 4:     ▷ conquer step
 5:     $a_1 \leftarrow$ computeMinWindow($i$, $q$, $\mu$, $\nu$)
 6:     $a_2 \leftarrow$ computeMinWindow($q+1$, $j$, $\mu$, $\nu$)
 7:     ▷ combine step
 8:     $FP(a_1) \leftarrow FP(a_1) + \sum_{q+1 \leq s \leq j} n_s$
 9:     $FN(a_2) \leftarrow FN(a_2) + \sum_{i \leq s \leq q} p_s$
10:     $a_g \leftarrow$ extendWindow($i$, $j$, $q$, $\mu$, $\nu$)
11:     $a_{opt} \leftarrow \text{argmin}_{a \in \{a_1, a_2, a_g\}} cost(a, \mu, \nu)$
12:     return $a_{opt}$
13: end procedure
```

asymptotic running time for the computation of an optimal abstention window compared to the naive algorithm. The complete procedure is described by algorithms 5.5, 5.6 and 5.7.

Algorithm 5.6 calculates the optimal abstention window on the vector $(m_i, \ldots, m_j)$. First the vector is divided into two sub-vectors $(m_i, \ldots, m_q)$ and $(m_{q+1}, \ldots, m_j)$ with $q = \lfloor \frac{i+j}{2} \rfloor$ and then the optimal abstention windows for $(m_i, \ldots, m_q)$ and $(m_{q+1}, \ldots, m_j)$ respectively are determined recursively. These are called a_1 and a_2. As the counts of the false positives and false negatives for both a_1 and a_2 have only been determined from the sub-vector each window was calculated on, these counts have to be updated for the whole vector $(m_i, \ldots, m_j)$ (lines 8 and 9). Eventually, the optimal abstention window a_g is determined which crosses the split point. This is described in algorithm 5.7. The optimal lower threshold is determined first by decreasing the threshold step by step until it is smaller than m_i. At each step the counts for false negatives and abstained instances are updated as these are the only values that change and the current abstention window is compared against the best window so far. The same steps are performed for the upper threshold with the exception that the threshold is increased step by step until it is larger than m_j and that the counts actualized are the false positives and abstained instances.

Theorem 5.13. *Let μ and ν be defined as before and k the number of distinct margins. The optimal abstention window for μ and ν can be calculated in time $O(k \log k)$.*

Proof. From lemma 5.12 it follows that the combine step can be computed in time $O(k)$ as the optimal lower and upper threshold of a_g can be determined one after the other and the number of possible values for the thresholds is linear in k. For each threshold only a constant

Algorithm 5.7 Algorithm for computing the optimal abstention window a_g given values for μ and ν, two indices i and j and a cut index q, such that $v(i-1) \leq l_g \leq v(q)$ and $v(q) \leq u_g \leq v(j)$. $\vec{p}$ and $\vec{n}$ are stored in global variables.

```
procedure extendWindow($i$, $j$, $q$, $\mu$, $\nu$)
    ▷ compute false negatives and false positives for threshold $v(q)$
    $FN_q \leftarrow \sum_{i \leq r \leq q} p_r$, $FP_q \leftarrow \sum_{q+1 \leq r \leq j} n_r$
    ▷ compute optimal lower threshold for the abstention window
    $FN \leftarrow FN_q$, $FP \leftarrow FP_q$, $A \leftarrow 0$
    $a_l \leftarrow (v(q), v(q))$, $FN(a_l) \leftarrow FN$, $FP(a_l) \leftarrow FP$, $A(a_l) \leftarrow A$
    for $r \leftarrow q$ to $i$ do
        $FN \leftarrow FN - p_r$
        $A \leftarrow A + p_r + n_r$
        $a_{tmp} \leftarrow (v(r-1), v(q))$
        $FN(a_{tmp}) \leftarrow FN$, $FP(a_{tmp}) \leftarrow FP$, $A(a_{tmp}) \leftarrow A$
        if $cost(a_{tmp}, \mu, \nu) < cost(a_l, \mu, \nu)$ then
            $a_l \leftarrow a_{tmp}$
        end if
    end for
    ▷ compute upper threshold for the abstention window
    $FN \leftarrow FN_q$, $FP \leftarrow FP_q$, $A \leftarrow 0$
    $a_u \leftarrow (v(q), v(q))$, $FN(a_u) \leftarrow FN$, $FP(a_u) \leftarrow FP$, $A(a_u) \leftarrow A$
    for $r \leftarrow q+1$ to $j$ do
        $FP \leftarrow FP - n_r$
        $A \leftarrow A + p_r + n_r$
        $a_{tmp} \leftarrow (v(q), v(r))$
        $FN(a_{tmp}) \leftarrow FN$, $FP(a_{tmp}) \leftarrow FP$, $A(a_{tmp}) \leftarrow A$
        if $cost(a_{tmp}, \mu, \nu) < cost(a_u, \mu, \nu)$ then
            $a_u \leftarrow a_{tmp}$
        end if
    end for
    $a_g \leftarrow (l_l, u_u)$
    $FN(a_g) \leftarrow FN(a_l)$, $FP(a_g) \leftarrow FP(a_u)$, $A(a_g) \leftarrow A(a_l) + A(a_u)$
    return $a_g$
end procedure
```

number of operations is performed. The counts of the false positives of a_1 and false negatives of a_2 can also be updated in linear time. Thus we have for the running time $T(k)$:

$$T(k) = 2T(\tfrac{k}{2}) + O(k) \overset{(*)}{=} O(k \log k).$$

$(*)$ is a consequence of the well known master theorem [10]. □

5.2.2 The Linear Algorithm

Previously we have presented an algorithm which runs in $O(k \log k)$. If our only goal was to determine the optimal abstention window for a given cost scenario, we could finish here, as

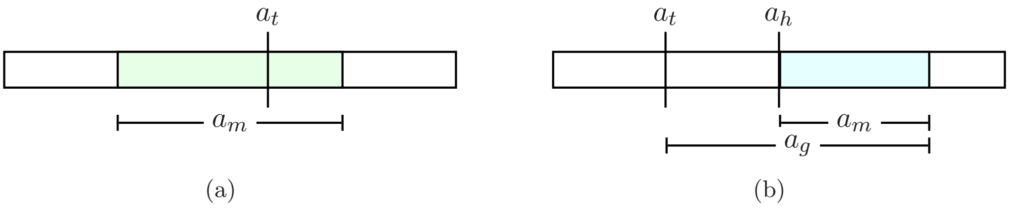


Figure 5.5: Figure (a) visualizes the relationship between the optimal abstention window a_m and the optimal threshold a_t between positive and negative prediction. a_m is located around a_t, such that $l_m \leq l_t$ and $u_m \geq u_t$. Note that $l_t = u_t$. Figure (b) illustrates the proof to lemma 5.14. a_t again denotes the optimal threshold. The assumption is that the optimal abstention window a_m is not located around a_t. The proof essentially states that in this case the extension of a_m to a_g leads to a reduction of expected cost, which is a contradiction to the choice of a_m.

sorting the n instances by margin already takes time $O(n \log n)$. Thus the complete algorithm will always need $O(n \log n)$ time. However, as we later use the algorithm to compute the optimal abstention window for several cost scenarios and sorting has to be performed only once, further improvements are useful and necessary.

Lemma 5.12 does not only prove the correctness of the divide-and-conquer approach, it also provides a way to design a linear algorithm. The lemma's essential statement is that if a position q within the optimal abstention window a_m is known such that $l_m \leq v(q)$ and $u_m \geq v(q)$, we can compute the optimal abstention window in linear time. Thus the only prerequisite still required is a method to determine such a position efficiently. The following lemma shows that the optimal abstention window for cost scenarios with $\nu \leq \frac{\mu}{1+\mu}$ is always located around the threshold of the optimal non-abstaining classifier (see figure 5.5(a)). This threshold can be determined efficiently. If $\nu > \frac{\mu}{1+\mu}$, finding the optimal threshold suffices as abstention is too expensive anyway (see lemma 2.17).

Lemma 5.14. *Let $\mu > 0$ and $\nu \leq \frac{\mu}{1+\mu}$. Define $\mathcal{T} := \{a | a \in \mathcal{A} \wedge l = u\}$ and let $a_t := \operatorname{argmin}_{a \in \mathcal{T}} cost(a, \mu, \nu)$ and $a_m := \operatorname{argmin}_{a \in \mathcal{A}} cost(a, \mu, \nu)$. Then we have $l_m \leq l_t = u_t \leq u_m$.*

Proof. By contradiction: Assume that $l_m > l_t$ or $u_m < u_t$.
We only show that $l_m > l_t$ leads to a contradiction to the optimality of a_m. The assumption $u_m < u_t$ can be lead to a contradiction in the same way.
Let $d_y := |\{x_j \in S | l_t < m(x_j) < l_m \wedge y_j = y\}|$ be the number of instances of class y whose margins lie between l_t and l_m and $d := d_P + d_N$.
Now define two windows $a_g = (l_t, u_m)$ and $a_h = (l_m, l_m)$. (see figure 5.5(b)). Thus we have

$$\begin{aligned} & cost(a_t, \mu, \nu) - cost(a_h, \mu, \nu) \\ & = FN(a_t) - FN(a_h) + \mu(FP(a_t) - FP(a_h)) + \nu\, 0 \\ & = -d_P + \mu\, d_N \overset{(*)}{<} 0 \end{aligned} \tag{5.7}$$

(*) holds as a_t is the optimal threshold between positive and negative classification, thus a_h must have expected cost greater or equal to a_t. However, if a_t and a_h have equal expected cost, we could choose a_h as optimal threshold so that the theorem holds.

From equation (5.7) we know that

$$d_P > \mu\, d_N \iff d_P > \mu\,(d - d_P) \iff d_P > \frac{\mu\, d}{1+\mu} \tag{5.8}$$

Now have a look at the difference in cost between a_m and a_g:

$$\begin{aligned}
&cost(a_m, \mu, \nu) - cost(a_g, \mu, \nu) \\
&= FN(a_m) - FN(a_g) + \mu(FP(a_m) - FP(a_g)) + \nu(A(a_m) - A(a_g)) \\
&= FN(a_m) - FN(a_g) + \nu(A(a_m) - A(a_g)) \\
&= d_P - \nu(d_N + d_P) \overset{\text{Equ. (5.8)}}{>} \frac{\mu\, d}{1+\mu} - \nu\, d = d\Big(\frac{\mu}{1+\mu} - \nu\Big) \geq 0
\end{aligned} \tag{5.9}$$

But equation (5.9) is a contradiction to the choice of a_m as the abstention window with minimum expected cost. □

Based on lemma 5.14 we can formulate a linear algorithm for computing the optimal abstention window for any cost scenario. The algorithm consists of two parts (see algorithm 5.8). First the optimal threshold between positive and negative prediction is determined and afterwards – if abstention costs are low enough – the optimal abstention window located around this threshold. The correctness of this algorithm follows from lemmas 5.12 and 5.14.

Theorem 5.15. *Let $\mu, \nu \in [0:1]$ be defined as before and k the number of distinct margins. The abstention window with minimal cost can be computed in time $O(k)$.*

Proof. Lines 6-15 of the pseudocode describe the calculation of the optimal threshold. The for-loop is iterated k times and each iteration requires only constant time, therefore this step requires time in $O(k)$. Extending the threshold to the optimal abstention window can be done in $O(k)$ as we have seen before. As a consequence, the final running time is $T(k) = O(k)$. □

5.3 Computation of Cost Curves in Linear Time

In the previous section we have introduced an algorithm which determines the optimal abstention window for a given cost scenario in linear time. We can now use this algorithm to compute the complete cost curve. A naive implementation would apply this algorithm for each cost scenario – of which there are $O(\Delta^2)$ – separately, resulting in a running time of $O(\Delta^2\, k)$. In practice we can indeed do better by employing the relationships between optimal abstention windows for different cost scenarios. Furthermore, the preprocessing step introduced in algorithm 5.1 can be used again to exclude abstention windows beforehand which can never be optimal under any cost scenario. Therefore, the running time of the naive algorithm is reduced to $O(\Delta^2 t)$.

In the 3CSAW algorithm we have used corollary 5.11 which states that if the optimal abstention window a_m for some cost scenario has equal lower and upper threshold, it is also optimal for any cost scenario with the same costs for false positives and higher abstention costs. Additionally to that we can impose further restrictions on the lower and upper thresholds based on the optimal abstention windows for lower abstention costs. If the value for μ is constant the optimal abstention window for any abstention costs is always contained within the optimal abstention windows for lower abstention costs (see figure 5.6(a)).

Lemma 5.16. *Let $\mu, \nu_1, \nu_2 > 0$ and $\nu_1 < \nu_2$. Now let $a_m := \operatorname{argmin}_{a \in \mathcal{A}} cost(a, \mu, \nu_1)$ and $a_n := \operatorname{argmin}_{a \in \mathcal{A}} cost(a, \mu, \nu_2)$. Then it follows that $l_m \leq l_n$ and $u_n \leq u_m$.*

Proof. By contradiction: Assume we have for a_n that $l_n < l_m$ or $u_n > u_m$.
First let $l_n < l_m$ and $d_y := |\{x_j | l_n < m(x_j) < l_m \wedge y_j = y\}|$ be the number of instances of class y whose margins lie between l_n and l_m. Define a new abstention window $a_h := (l_n, u_m)$ (see figure 5.6(b)). As the optimal threshold between positive and negative classification a_t is the same for both cost scenarios, we know that $l_n \leq l_t \leq u_m$. The definition of a_m then implies

$$cost(a_m, \mu, \nu_1) - cost(a_h, \mu, \nu_1) = d_P - \nu_1(d_P + d_N) < 0 \tag{5.10}$$

(If $cost(a_m, \mu, \nu_1) - cost(a_h, \mu, \nu_1) = 0$, we could use a_h instead of a_m and a_n would fulfill the lemma.) Now let $a_g := (l_m, u_n)$. The difference in cost between a_n and a_g is

$$cost(a_n, \mu, \nu_2) - cost(a_g, \mu, \nu_2) = -d_P + \nu_2(d_P + d_N) > -d_P + \nu_1(d_P + d_N) \overset{\text{Equ. (5.10)}}{>} 0 \tag{5.11}$$

But equation (5.11) is a contradiction to the definition of a_n.
For $u_n > u_m$, we can derive a contradiction in the same way. □

Now let μ, ν_1 and ν_2 be defined as in the previous lemma and a_m the optimal abstention window for μ and ν_1 and a_t the optimal threshold between positive and negative classification

Algorithm 5.8 Linear algorithm for computing the abstention window with minimal cost given values for μ and ν. The vectors $(m_1, \ldots, m_k)$, $(p_1, \ldots, p_k)$ and $(n_1, \ldots, n_k)$ are stored in global variables.

```
procedure computeOptWindow($\mu$, $\nu$)
    $FN \leftarrow 0$, $FP \leftarrow \sum_{1 \leq q \leq k} n_r$
    $q \leftarrow 0$
    $a_m \leftarrow (v(q), v(q))$
    $FN(a_m) \leftarrow FN$, $FP(a_m) \leftarrow FP$, $A(a_m) \leftarrow 0$
    for $r \leftarrow 1$ to $k$ do
        $FN \leftarrow FN + p_r$
        $FP \leftarrow FP - n_r$
        $a_{tmp} \leftarrow (v(r), (v(r))$
        $FN(a_{tmp}) \leftarrow FN$, $FP(a_{tmp}) \leftarrow FP$, $A(a_{tmp}) \leftarrow 0$
        if $cost(a_{tmp}, \mu, \nu) < cost(a_m, \mu, \nu)$ then
            $a_m \leftarrow a_{tmp}$
            $q \leftarrow r$
        end if
    end for
    if $\nu \leq \frac{\mu}{1+\mu}$ then
        $a_m \leftarrow$ extendWindow(1, $k$, $q$, $\mu$, $\nu$)
    end if
    return $a_m$
end procedure
```

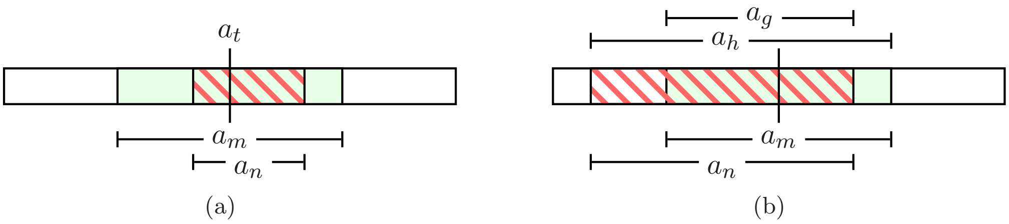


Figure 5.6: Figure (a) shows the relationship between optimal abstention windows for the same false positive costs μ but different abstention costs ν_1 (window a_m) and ν_2 (window a_n) with $\nu_1 < \nu_2$. a_t denotes the optimal threshold between positive and negative classification for both scenarios. Figure (b) illustrates the proof to lemma 5.16. a_m is the optimal abstention window for μ and ν_1 and a_n the one for μ and ν_2. The assumption is that $l_n < l_m$. We do not know if $u_m \leq u_n$ or $u_m > u_n$, but we only require that both $u_m \geq l_n$ and $u_n \geq l_m$. This holds as we have the same optimal threshold between positive and negative classification for both cost scenarios.

for μ. Then lemma 5.16 allows us to limit the number of abstention windows considered for costs μ and ν_2 to those abstention windows $a_i \in \mathcal{A}$ which lie within a_m and are located around a_t, i.e. $l_m \leq l_i \leq l_t$ and $u_m \geq u_i \geq u_t$.

Furthermore, we observe that with increasing costs for false positives the optimal threshold between positive and negative classification is never moved into the negative direction. As false positives are penalized more strongly, positive predictions on the whole are increasingly avoided. The next lemma formalizes this observation.

Lemma 5.17. *Let $\mu_1, \mu_2, \nu \in (0:1]$ and $\mu_1 < \mu_2$. If $a_s = \operatorname{argmin}_{a \in \mathcal{T}} cost(a, \mu_1, \nu)$ and $a_t = \operatorname{argmin}_{a \in \mathcal{T}} cost(a, \mu_2, \nu)$, then we have that $l_s \leq l_t$.*

Proof. By contradiction: Assume that $l_s > l_t$. Note that $u_s = l_s$ and $u_t = l_t$.
Define d_y as $|\{x_j | l_t < m(x_j) < l_s \wedge y_j = y\}|$ for $y \in \{P, N\}$. As $l_s > l_t$ we must have $d_P + d_N > 0$. Furthermore we presume that $cost(a_s, \mu_1, \nu) < cost(a_t, \mu_1, \nu)$ and $cost(a_t, \mu_2, \nu) < cost(a_s, \mu_2, \nu)$. Otherwise one of the thresholds would be optimal for both scenarios. Thus we have

$$cost(a_s, \mu_1, \nu) - cost(a_t, \mu_1, \nu) = d_P - \mu_1 d_N < 0 \tag{5.12}$$

and

$$cost(a_t, \mu_2, \nu) - cost(a_s, \mu_2, \nu) = \mu_2 d_N - d_P < 0 \tag{5.13}$$

From equation (5.12) we can conclude that $d_N > 0$. By summing over equation (5.12) and (5.13) we yield

$$\mu_2 d_N < \mu_1 d_N \iff \mu_2 < \mu_1$$

which is a contradiction to the choice of μ_1 and μ_2. □

This last lemma as well as the ones before allows several improvements from the naive algorithm by storing intermediate results. The algorithm then consists of two parts. First the optimal thresholds between positive and negative classification are computed for each $\mu = i/\Delta$, $0 \leq i \leq \Delta$ as described in algorithm 5.9. For this purpose the abstention costs are set to 1 but any other value could be used because none of the evaluated abstention windows abstains and therefore abstention costs are irrelevant. Two vectors $\vec{\tau}$ and $\vec{q}$ are used to store the results. τ_i denotes the optimal abstention window for $\mu = i/\Delta$ with equal lower and

Algorithm 5.9 Algorithm for computing the optimal thresholds for $\mu = i/\Delta$, $0 \leq i \leq \Delta$. The output is stored in two vectors $\vec{q} = (q_1, \ldots, q_\Delta)$ and $\vec{\tau} = (\tau_1, \ldots, \tau_\Delta)$, such that $\tau_i = \text{argmin}_{a \in \mathcal{T}}\, cost(a, i/\Delta, 1)$ and the value of the threshold of τ_i is $\psi(q_i)$. The vectors $\vec{\lambda}$, $\vec{v}$, $\vec{\rho}$, $\vec{\eta}$ are defined as in definition 5.7 and stored as global variables.

```
procedure computeThresholds($\Delta$)
  for $i \leftarrow 0$ to $\Delta$ do
    if $i = 0$ then
      $q_i \leftarrow 0$
      $\tau_i \leftarrow (\psi(q_i), \psi(q_i))$
      $FN(\tau_i) \leftarrow 0$, $FP(\tau_i) \leftarrow \sum_{1 \leq r \leq t} \eta_r$
    else
      $q_i \leftarrow q_{i-1}$
      $\tau_i \leftarrow \tau_{i-1}$
      $FN \leftarrow FN(\tau_{i-1}), FP \leftarrow FP(\tau_{i-1})$
      for $r \leftarrow q_{i-1} + 1$ to $t$ do
        $FN \leftarrow FN + \rho_r$
        $FP \leftarrow FP - \eta_r$
        $a_{tmp} \leftarrow (\psi(r), \psi(r))$
        $FN(a_{tmp}) \leftarrow FN$, $FP(a_{tmp}) \leftarrow FP$
        ▷ No abstention, thus third argument without effect
        if $cost(a_{tmp}, i/\Delta, 1) < cost(\tau_i, i/\Delta, 1)$ then
          $\tau_i \leftarrow a_{tmp}$
          $q_i \leftarrow r$
        end if
      end for
    end if
  end for
  return ($\vec{q}$, $\vec{\tau}$).
end procedure
```

upper threshold and q_i the index position such that the value of the optimal threshold is $\psi(q_i)$. For computing the optimal value for q_i only indices greater than or equal to q_{i-1} are considered at all because for increasing false positive costs the threshold is never moved into the negative direction (lemma 5.17). Note that increasing indices correspond to increasing margin values. In the worst case, we have to evaluate $O(t)$ possible thresholds for each i. For each threshold only constant time is required, therefore the worst case running time of algorithm 5.9 is in $O(\Delta\, t)$.

Subsequently, the complete cost curve is computed based on the results for $\vec{\tau}$ and $\vec{q}$. The pseudocode for this procedure is given in algorithm 5.10. For each combination of i and j the same steps are performed. Remember that this corresponds to false positive costs $\mu = i/\Delta$ and abstention costs $\nu = j/\Delta$. If the costs for abstention exceed $\frac{\mu}{1+\mu}$ (i.e. $j \leq \frac{\Delta\, i}{\Delta + i}$) no extra work is necessary. Otherwise, first the optimal lower threshold is determined and afterwards the optimal upper threshold. As in the linear algorithm for the calculation of one optimal abstention window, the lower threshold at the beginning is assigned the value

Algorithm 5.10 The complete algorithm for computing the cost curve. The cost curve is stored in a matrix K, the vectors $\vec{q}$ and $\vec{\tau}$ are used as defined in algorithm 5.9 and the vectors $\vec{\lambda}$, $\vec{\upsilon}$, $\vec{\rho}$, $\vec{\eta}$ are stored as global variables.

```
procedure computeCostCurve($\Delta$)
    $(\vec{q}, \vec{\tau}) \leftarrow$ computeThresholds($\Delta$)
    for $i \leftarrow 0$ to $\Delta$ do
        $k_{i,0} \leftarrow 0$, $l \leftarrow 0$, $u \leftarrow t$
        for $j \leftarrow 1$ to $\Delta$ do
            ▷ Compute the lower threshold of the optimal abstention window
            $a_l \leftarrow \tau_i$, $s \leftarrow q_i$
            $FN \leftarrow FN(\tau_i)$, $FP \leftarrow FP(\tau_i)$, $A \leftarrow 0$
            if $j \leq \frac{\Delta\, i}{\Delta + i}$ then
                for $r \leftarrow q_i$ to $l + 1$ do
                    $FN \leftarrow FN - \rho_r$
                    $A \leftarrow A + \rho_r + \eta_r$
                    $a_{tmp} \leftarrow (\psi(r-1), \psi(q_i))$
                    $FN(a_{tmp}) \leftarrow FN$, $FP(a_{tmp}) \leftarrow FP$, $A(a_{tmp}) \leftarrow A$
                    if $cost(a_{tmp}, i/\Delta, j/\Delta) < cost(a_l, i/\Delta, j/\Delta)$ then
                        $a_l \leftarrow a_{tmp}$, $s \leftarrow r - 1$
                    end if
                end for
            end if
            $l \leftarrow s$
            ▷ Compute the upper threshold of the optimal abstention window
            $a_u \leftarrow \tau_i$, $s \leftarrow q_i$
            $FN \leftarrow FN(\tau_i)$, $FP \leftarrow FP(\tau_i)$, $A \leftarrow 0$
            if $j \leq \frac{\Delta\, i}{\Delta + i}$ then
                for $r \leftarrow q_i + 1$ to $u$ do
                    $FP \leftarrow FP - \eta_r$
                    $A \leftarrow A + \rho_r + \eta_r$
                    $a_{tmp} \leftarrow (\psi(q_i), \psi(r))$
                    $FN(a_{tmp}) \leftarrow FN$, $FP(a_{tmp}) \leftarrow FP$, $A(a_{tmp}) \leftarrow A$
                    if $cost(a_{tmp}, i/\Delta, j/\Delta) < cost(a_u, i/\Delta, j/\Delta)$ then
                        $a_u \leftarrow a_{tmp}$, $s \leftarrow r$
                    end if
                end for
            end if
            $u \leftarrow s$
            $k_{i,j} \leftarrow cost(a_l, i/\Delta, j/\Delta) + cost(a_u, i/\Delta, j/\Delta) - cost(\tau_i, i/\Delta, j/\Delta)$
        end for
    end for
    return $K$.
end procedure
```

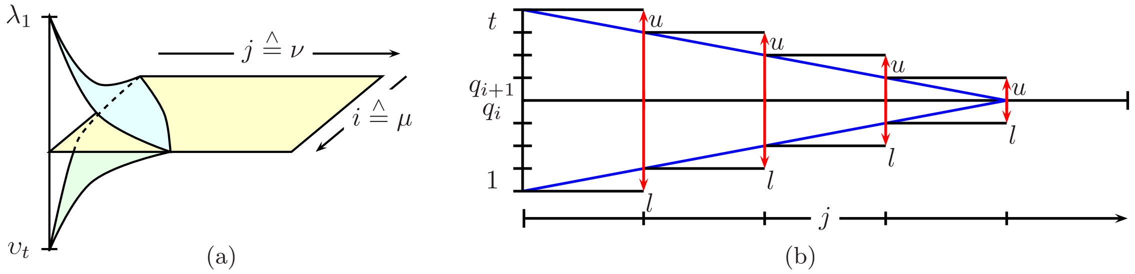


Figure 5.7: Figure (a) shows schematically the lower an upper thresholds of the optima abstention window in relationship to i and j, i.e. the false positive costs μ and abstention costs ν. The yellow plane corresponds to the optimal threshold between positive and negative classification. The upper threshold is depicted in blue as soon as it increases, and the lower threshold in green as soon as it decreases. Figure (b) illustrates the relationship of optimal abstention windows for fixed values of i and increasing values of j. The indices of the preprocessed vectors range from 1 to t and the optimal threshold between positive and negative classification lies between υ_{q_i} and $\lambda_{q_{i+1}}$. The blue lines indicate the change in the optimal lower and upper threshold. The lower threshold decreases and the upper threshold increases with j until they meet each other. For each value of j only the red range has to be evaluated for lower and upper threshold as this corresponds to the optimal abstention window for $j-1$. l and u are used to store the indices of the optimal threshold values for $j-1$.

of the optimal threshold between positive and negative classification. Following this, the threshold is decreased step by step and the counts of false negatives and abstained instances are updated. However in this case, we do not continue until the threshold is below the smallest margin value but use the information we have about the optimal lower threshold for i and $j-1$. In lemma 5.16 we have shown that the optimal abstention window for the current cost scenario i and j is contained in the optimal abstention window for i and $j-1$. Therefore the last threshold we have to evaluate for this cost scenarios is $\psi(l)$ with $\psi(l)$ the optimal lower threshold for i and $j-1$. The same applies to the calculation of the upper threshold. The threshold is increased step by step until we have reached the optimal upper threshold for i and $j-1$. Two variables l and u are used to store the indices of the optimal thresholds for i and $j-1$ such that the optimal abstention window for i and $j-1$ is $(\psi(l), \psi(u))$. When $j=0$, l is initialized with 0 and u with t. The relationship between the optimal abstention windows for different values of i and j is illustrated in figure 5.7.

The algorithm effectively calculates two abstention windows a_l and a_u for each i and j with $a_l = \text{argmin}_{a_r \in \mathcal{A} \wedge u_r = \psi(q_i)} cost(a_r, i/\Delta, j/\Delta)$ and $a_u = \text{argmin}_{a_r \in \mathcal{A} \wedge l_r = \psi(q_i)} cost(a_r, i/\Delta, j/\Delta)$. The optimal abstention window a_m for this cost scenario then is defined by the lower threshold of a_l and the upper threshold of a_u. Obviously, we have that $FN(a_m) = FN(a_l)$, $FP(a_m) = FP(a_u)$ and $A(a_m) = A(a_l) + A(a_u)$. The expected cost of a_m then can be calculated as

$$cost(a_m, i/\Delta, j/\Delta) = cost(a_l, i/\Delta, j/\Delta) + cost(a_u, i/\Delta, j/\Delta) - cost(\tau_i, i/\Delta, j/\Delta).$$

Determining a_l and a_u requires at most time $O(t)$ for given i and j because we only evaluate $O(t)$ lower or upper thresholds and evaluating a threshold can be done in constant time. Consequently, the asymptotic running time of this algorithm is still $O(\Delta^2 t)$ as for the naive implementation. Nevertheless, the practical running time has been greatly reduced because for most cost scenarios only a fraction of possible thresholds has to be evaluated.

We have now presented two algorithms for efficiently computing a cost curve from the results for the validation set. Both of these algorithms rely on characteristics of optimal abstention windows as well as relationships between optimal windows for different cost scenarios. What eventually made it possible to go beyond explicitly calculating all abstention windows and comparing their costs, was the observation that the optimal abstention window is always located around the optimal threshold between positive and negative classification.

Chapter 6

Evaluation

The benefits of abstaining in general as well as the presented cost curves for abstaining classifiers and the methods for combining abstaining classifiers were evaluated on two biological classification tasks which involve the prediction of origin for EST sequences and prediction of mutagenicity or carcinogenicity of chemical compounds. Furthermore, we analyzed the characteristics of instances abstained on and the behavior of optimal false negative and positive rate as well as abstention rate in relationship to each other and the dependency between optimal abstention rate and classification accuracy.

For this purpose, at least two sets of instances were required for each classification task. A training set was necessary to calculate a classification model for distinguishing the classes and a validation set to calculate optimal abstention windows and cost curves. When evaluating the performance of the methods a test set was needed as well. Unfortunately, the number of labeled instances available for each task were in general small. As a consequence, the available data sets were not split in two (or three) separate sets but tenfold cross-validation was used.

For cross-validation the data sets are split into ten approximately equally large subsets. Alternately, one of these subsets is used as test or validation set and the remaining nine subsets as training set. Each instance is used exactly once for the validation set or the test set, therefore one unequivocal prediction is obtained for it. Although the predictions are provided by different models, the results are treated as if only one model was applied to one

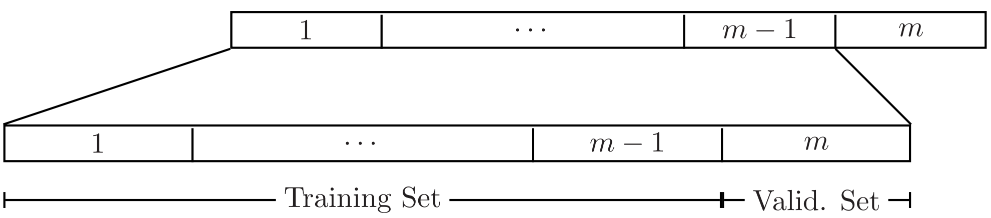


Figure 6.1: Nested loops of m-fold cross-validation. The original data set is split into m approximately equally large subsets. At each iteration 1 subsets is used as training set and the remaining $m-1$ subsets are referred to an internal cross-validation. The remaining instances are again split into m subsets. Alternately one subset is used as validation set and the remaining instances are used for training classifiers.

separate validation set. This is reasonable as the training set size is only slightly smaller than the size of the original set and hence the different models are assumed to agree to a large extent. If three separate sets are required, an external cross-validation is performed to obtain a test set and within each fold an internal cross-validation provides the information required for validation (see figure 6.1).

6.1 Classification Tasks

6.1.1 Separation of mixed plant-pathogen EST collections

The first classification task involves the prediction of origin for sequences from mixed plant-pathogen EST pools (see also [22]), that is if they correspond to plant or fungal genes. Such EST pools are derived by extracting EST sequences from infected plants and are helpful to determine genes involved in plant defense or pathogen virulence. Due to the high-throughput nature of these experiments, biological methods for determining the origin of a sequence become infeasible, thus fast and reliable computational methods are necessary.

Homology search within genome databases in many cases fails due to biased taxa representation within genome databases and sequence homology between plant and pathogen genes. In [22] a method is presented which relies on machine learning methods only – in this case support vector machines – to distinguish between plant and pathogen EST sequences based on differences in codon bias between the two organisms. Codon bias denotes the fact that not all nucleotide triplets coding for the same amino acid are used in equal proportions. Some codons may be preferred above others and there is decisive variation between species as to which codons are preferred and the frequencies with which they occur [47].

The EST dataset used for training and evaluating this method contained 3217 unigene sequences of diverse lengths from barley (*Hordeum vulgare*, 1315 sequences) and blumeria (*Blumeria graminis*, 1902 sequences) for which the coding frame had been determined previously using the Sputnik EST analysis pipeline [43]. A minimum sequence length threshold of 100 base pairs was imposed and unigene sequences were used to avoid redundancy, so that each gene from the plant or fungal organism was represented by at most one EST sequence in the data set. On account of the small size of the data set a majority of genes of each organism were not represented in it at all. As the method does not rely on sequence homology but on the underlying codon composition of the genes, this does not constitute a problem.

In order to derive attributes for the sequences, codon occurrences were computed starting at the begin of the sequence up to and including the first stop codon. As some codons may be missing in an EST sequence which in most cases represents only a part of a gene sequence, pseudocounts were included when computing the codon frequency. Accordingly, the frequency of a codon c was defined as

$$F(c) = \frac{n_c + 1}{\sum_{c' \in Codons} n_{c'} + 64}.$$

where n_i denotes the number of occurrences of codon i in this sequence. Consequently, an instance of this dataset has exactly 64 attributes giving the frequencies of the 64 coding triplets. For our purposes we restricted ourselves to the task of predicting the origin of a sequence provided that the coding frame is known. This is of course a simplification of the problem as the coding frame is in general unknown for a newly sequenced EST sequence. Nevertheless, the coding frame of a sequence can also be predicted with high confidence using machine learning techniques [22].

6.1.2 Predictive Toxicology

As a consequence of the amount of chemicals employed in every area of human activity, the evaluation of toxic side-effects such as carcinogenicity or mutagenicity of chemicals has become a major issue. However, in spite of efforts on the side of the US National Toxicology Program (NTP, http://ntp-server.niehs.nih.gov/) for example, which effects standardized bioassay tests exposing rodents to various chemicals in order to identify compounds potentially carcinogenic in humans, only a small fraction of chemicals has actually been tested. As in the case of EST origin prediction, this is due to the time-consuming and expensive nature of such experiments.

In order to reduce costs, the need for reliable models for toxicity predictions based only on molecular properties and chemical structures has arisen. The major phases involved in developing such models comprise the generation of appropriate descriptors of the chemicals and afterwards the construction of models based on those descriptors. This task was addressed in the Predictive Toxicology Challenge (PTC) 2000-2001 [29] for rodent carcinogenicity results from the US National Toxicology Program. For our purpose, we chose only the training set from this challenge due to structural dissimilarities between training and test set [50], as well as a second data set derived from the carcinogenic potency database (CPDB, http://potency.berkeley.edu/cpdb.html, [23]) which offers mutagenicity results based on *Salmonella*/microsome assays [1]. This second dataset was used by Helma *et al.* [30] to analyze the benefits of molecular fragments as descriptors compared to molecular properties as well as to compare different machine learning algorithms.

For both datasets we distinguished only between the positive (carcinogenic/mutagenic) and the negative (non-carcinogenic/non-mutagenic) class. The NTP dataset contained results obtained for experiments in male and female rat and mice which could be of any of the following categories: CE (Clear Evidence of Carcinogenic Activity), SE (Some Evidence), EE (Equivocal Evidence), NE (No Evidence) and IS (Inadequate Study). For earlier experiments the description of the result might also be P (Positive), E (Equivocal) or N (Negative). A compound for which the result was P, CE or SE in any of the four experiments was declared positive. If all the experiments resulted in EE, IS or E the compound was excluded. Otherwise it was declared to be negative. The final carcinogenicity dataset was comprised of 408 instances, 179 of which were negative and 229 positive. The mutagenicity dataset contained 684 instances, 341 of which were positive and 343 were negative.

The instances were given as SMILES strings [53] which had been tested for validity and if necessary corrected as described by Helma *et al.*[28]. Chemical compounds were described by frequently occurring molecular fragments. This approach has been shown to produce satisfactory classification accuracy and be superior to simple molecular properties both by Kramer *et al.* [34] and Helma *et al.* [30]. The fragments were calculated with FreeTreeMiner [45], a program for mining frequent free trees – i.e. un-rooted trees – in graph data. Previous approaches used only paths but the extension to free trees appears to be at least equivalent. For the carcinogenicity dataset frequent free trees were computed which occurred in at least

10% of the structures and for the second set the frequency threshold was set to 5% due to its larger size. However, those thresholds were chosen rather arbitrarily. The results of Helma *et al.* [30] imply that by decreasing the frequency threshold and thus including more fragments, classification accuracy can be increased by 1-2%, but as a consequence the computational effort also increases tremendously.

6.2 Machine Learning Algorithms

Five machine learning algorithms were used to derive models for abstaining classifiers, those being support vector machines (SVM, [8]), decision trees (C4.5, [42]), rule learning (PART, [20]), Naive Bayes [37] and Random Forests [7]. For support vector machines the LIBSVM implementations [9] were chosen, whereas for the remaining algorithms the implementations of the WEKA workbench [54] were employed.

6.2.1 Support Vector Machines

Support Vector Machines (SVM) serve for classifying data based on linear decision rules (see also [52] and [4]). Given a training set $(x_1, y_1), \ldots, (x_n, y_n)$ such that x_i represent the attributes of instance i and $y_i \in \{-1, +1\}$ the corresponding class, SVM aim to find a hyperplane separating the training instances by their classes and maximizing the distance to the closest examples. The classification of a new instance then depends on which side of the hyperplane it is located.

As in most cases it is impossible to separate samples by a linear function in the original space, training instances may be transformed into a higher dimensional space by a function ϕ, such that a linear maximum-margin hyperplane in this higher dimensional space can be found. For solving this problem it is sufficient to give the dot product of two instances in this space. $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$ is called a kernel function which can be, for example, linear, polynomial, sigmoid or a radial basis function (RBF).

Support vector machines generally are not prone to overfitting and they can be computed efficiently as there exist several fast algorithms for finding the optimal hyperplane ([32], [38]). However, as they can describe intricate decision boundaries, the resulting classifiers are difficult to comprehend when non-linear kernels are used.

6.2.2 Decision Trees – C4.5

A decision tree is – as the name suggests – a tree describing sequences of tests. Each internal node prescribes a test on an attribute and has one successor node for each possible attribute value [37]. A class label is associated with each leaf node such that the classification of an instance can be derived by following the path from the root to a leaf. Decision trees are generally most suitable if instances are described by a fixed set of attributes which on their part can take on only a small number of possible values and if class labels are discrete-valued and the training data may contain errors or missing attribute values.

C4.5 is a greedy algorithm for constructing decision trees using a divide-and-conquer approach. At each step the training set is split into several subsets according to the values for a certain attribute. The best split attribute is chosen based on the expected reduction in entropy achieved if instances are sorted according to the attribute. The procedure is repeated recursively for each subset until all instances in the subset are in the same class or no more attributes remain to be tested. After building the complete tree a pruning step may be applied removing nodes at the lower levels of the tree to avoid overfitting.

6.2.3 PART

PART differs from other rule learning algorithms which first learn a set of rules and afterwards improve it in an optimization step by learning one rule at a time and refraining from global optimization. The algorithm is based on the generation of partial decision trees and combines the two major paradigms of rule learning which are the construction of rules from decision trees and the separate-and-conquer approach. In the latter one, by turns the best rule is extracted from the data set and the instances covered by the rule are removed from the set. A similar approach is used in chapter 4 to combine abstention windows. PART achieves a predictive accuracy comparable to other state-of-the-art rule learning algorithms on standard datasets while operating efficiently due to the avoidance of post-pruning.

6.2.4 Naive Bayes

The Naive Bayes algorithm relies on the Bayes theorem which makes it possible to calculate the most probable hypothesis within a hypothesis space H, given the data D as well as prior knowledge of the probabilities of hypothesis in H. In general, however, we are more interested in determining the most probable classification of an instance, not the most probable hypothesis. Given a set of class labels V, the Bayes optimal classification is therefore provided by

$$\operatorname*{argmax}_{v_j \in V} \sum_{h_i \in H} P(v_j|h_i)P(h_i|D). \tag{6.1}$$

As the Bayes optimal classifier requires calculating the posterior probability of every hypothesis in H, it is in most cases too expensive to apply. Alternatively, the label for an instance may be chosen solely depending on its attribute values. The optimal label for an instance with attribute values $(a_1, \dots, a_n)$ then is given by

$$\operatorname*{argmax}_{v_j \in V} P(v_j|a_1, \dots, a_n) = \operatorname*{argmax}_{v_j \in V} P(a_1, \dots, a_n|v_j)P(v_j). \tag{6.2}$$

Unfortunately, estimating the probabilities $P(a_1, \dots, a_n|v_j)$ from the training data is impossible but for very large training sets. To circumvent this problem, the Naive Bayes classifier assumes conditional independence for attribute values given the class label. As a consequence, the previous equation simplifies to

$$\operatorname*{argmax}_{v_j \in V} P(v_j) \prod_i P(a_i|v_j). \tag{6.3}$$

The learning step of the Naive Bayes algorithm consists of estimating the values for the $P(v_j)$ and $P(a_i|v_j)$. The classification of an instance is determined based on the estimated probabilities by using equation (6.3).

6.2.5 Random Forests

The Random Forests algorithm represents a variation of the bagging approach mentioned before and grows several decision trees. Each tree is grown on a slightly different training set, which is constructed using bootstrap sampling, i.e. sampling with replacement. The way each tree is grown differs from the C4.5 method such that at each node a constant number of attributes is chosen randomly and only the best test among the selected attributes is evaluated. Furthermore, no pruning is applied. The classification for an instance by the forest of trees is derived by calculating the prediction of each tree and then taking a vote among the trees.

6.3 Preliminary Analysis

6.3.1 Classification Performance

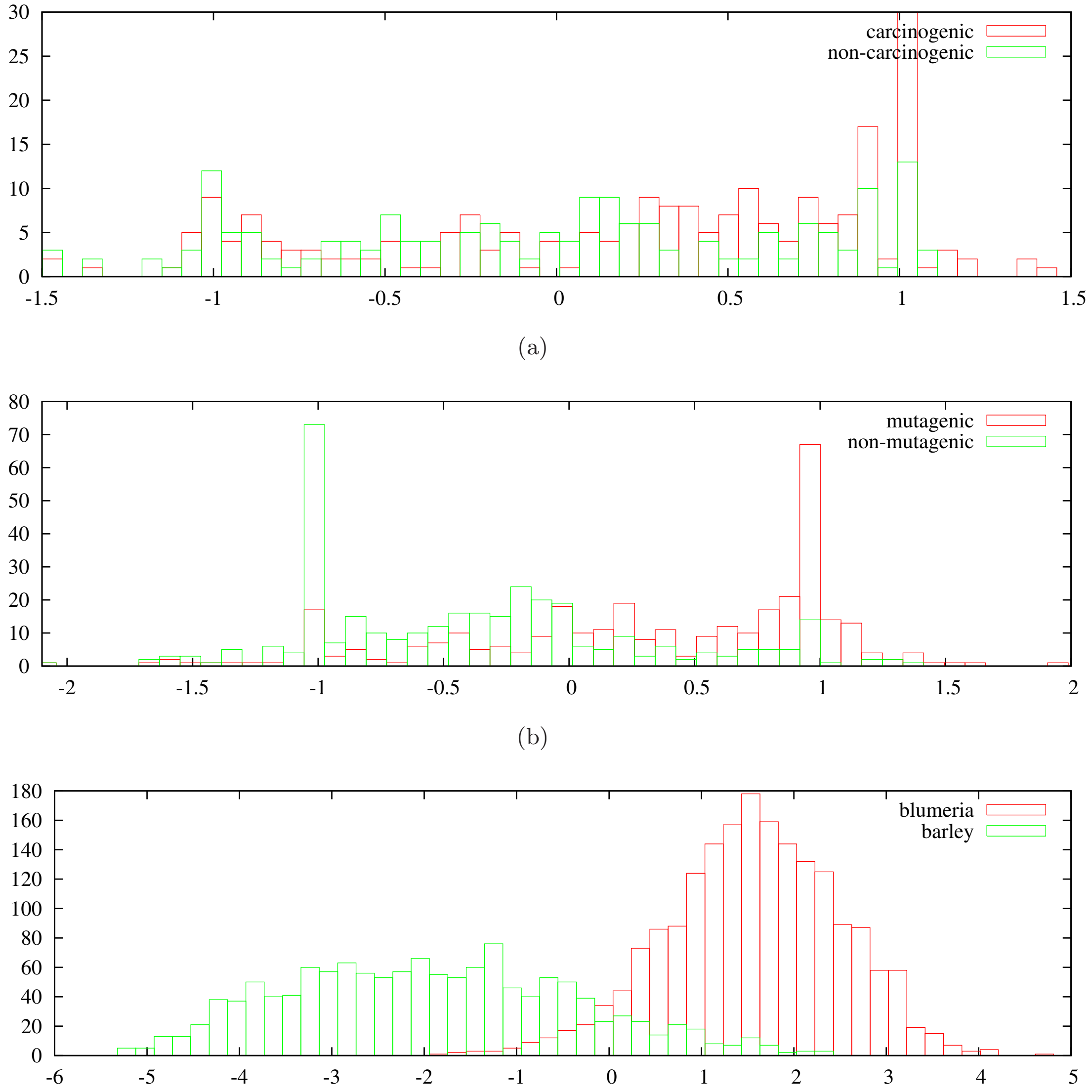
The three data sets described were chosen for two reasons. First they represent interesting biological classification tasks important for agricultural disease control on the one hand and for the prevention of chemical hazards on the other hand. Secondly, the classification accuracy, i.e. the percentage of correct predictions, which could be obtained by using any of the described machine learning algorithms, differed greatly between these data sets. Table 6.1(a) on page 80 contains the expected classification accuracy as estimated by tenfold cross-validation for all five algorithms as well as a baseline classifier (ZeroR) which always predicts the majority class. For simplification in the following passages the classification algorithms and the classifiers produced by them are used synonymously. J4.8 denotes the WEKA implementation of C4.5. For the support vector machines a RBF kernel was chosen and all classification algorithms were used with default settings.

The table shows that for carcinogenicity prediction the baseline classifier could hardly be improved upon, whereas for mutagenicity prediction accuracies between 69% and 77% were achieved. This is consistent with the results described by Kramer *et al.* [34] and Helma *et al.* [30]. For EST classification the range of prediction accuracies spread from around 82% for J4.8 up to almost 93% for SVM. For both mutagenicity prediction as well as the prediction of EST origin the baseline classifier was clearly surpassed by any of the classification algorithms.

6.3.2 Distribution of Margin Values

The differences between the data sets become more obvious when analyzing the distribution of margins between the positive and negative class for each data set. For the EST origin prediction the blumeria class was proclaimed as the positive class. However, this distinction is rather arbitrary. In fact, this may be one of the rare cases that both types of misclassifications actually result in equal or only slightly different costs.

Figure 6.2 shows the histograms of the margin values computed by the support vector machine classifiers which were among the top classifiers for all data sets. For carcinogenicity prediction (6.2(a)) the positive and negative instances were hardly separated at all and negative and positive margin values occurred regularly for both classes. For mutagenicity prediction (6.2(b)) the separation appeared to be more pronounced, however for small absolute values of the margin the classes still were mixed to a large extent. Only for the EST



(a)

(b)

(c)

Figure 6.2: Distribution of margin values for instances from the carcinogenicity (a), mutagenicity (b) and EST (c) dataset. To compute those margins, support vector machines were used. For the EST data set the blumeria class was chosen as positive class.

dataset (6.2(c)) a largely unequivocal separation of the classes was achieved.

These observations have an important implication for the following results. Both carcinogenicity prediction and EST classification can be expected to benefit only to a small degree from abstention for very different reasons. In the first case, as both positive and negative instances are scattered widely over the range of the margin values, probably no abstention window achieves much higher accuracy than the non-abstaining classifier. In the second case, the overlap between instances from blumeria and barley is only small. Although results can be improved by abstaining from instances in this overlap, an additional extension of abstention

Data set	Accuracy (in %)					
	SVM	Random Forests	Naive Bayes	PART	J4.8	ZeroR
Carcinogenicity data	56.9	55.8	55.1	54.7	56.1	56.1
Mutagenicity data	75.3	72.8	69.4	75.4	76.8	50.1
EST data	92.9	88.5	87.1	84.9	82.5	59.1

(a) Classification accuracies achieved for carcinogenicity and mutagenicity prediction as well as EST classification by several machine learning algorithms. For the support vector machines a RBF kernel was chosen and all classification algorithms were used with default settings.

Algorithm	Carcinogenicity Data		Mutagenicity Data		EST Data	
	Accuracy	Abst. Rate	Accuracy	Abst. Rate	Accuracy	Abst. Rate
SVM	62.3	56.4	81.3	67.3	96.3	11.3
Random Forests	60.8	68.1	84.1	53.2	93.6	20.4
Naive Bayes	57.6	62.9	84.5	54.6	93.5	24.3
PART	54.5	62.3	79.7	65.4	87.4	9.5
J4.8	59.0	64.7	82.8	60.8	84.1	20.5

(b) Accuracy and abstention rates achieved by the optimal abstention windows of different classifiers on each task. Misclassification costs were assumed to be equal and the abstention costs were set to $\frac{1}{5}$ of the misclassification costs for mutagenicity and EST origin prediction and to $\frac{2}{5}$ for carcinogenicity prediction.

Table 6.1: Classification accuracy achieved without and with abstention. Accuracy is defined as the percentage of correctly classified instances of all classified instances and was estimated using tenfold cross-validation. Note that an internal cross-validation was used to calculate the optimal abstention windows for table (b). J4.8 denotes the WEKA implementation of C4.5.

windows beyond this overlap only results in abstaining of instances which would be classified correctly otherwise and therefore increases costs.

6.3.3 Optimal Abstention Windows

As we have seen in chapter 2, we can easily calculate the optimal abstention window provided that the costs and class distribution are known. To illustrate the benefits of abstention compared to classifying all instances, equal misclassification costs were assumed for the following analysis although for both carcinogenicity and mutagenicity prediction this clearly does not hold true. Nevertheless, this assumption was useful as it allowed us to compare classification accuracy obtained with the help of abstention to accuracy obtained on all instances.

For this test both a validation and a test set was required, as the estimates of prediction accuracy of the optimal abstention window based on the validation set would have been highly optimistic. Therefore, an external cross-validation provided the test information and the internal cross-validation the information for validation as described before. In table 6.1(b) the estimated prediction accuracies are given alongside with the abstention rates necessary to achieve these results. Abstention costs of $\frac{1}{5}$ of the misclassification costs were assumed for both EST and mutagenicity data, however for carcinogenicity prediction abstention costs were raised to $\frac{2}{5}$, as for lower values the abstention rates yielded were close to 100%.

For each of the three datasets significant improvements could be observed. The most notable improvement was achieved by the Naive Bayes classifier with around 6 percentage points for EST classification and almost 15 percentage points for mutagenicity prediction. Unfortunately, these improvements were associated with high abstention rates, especially for mutagenicity and carcinogenicity prediction where up to two thirds of the instances were abstained on. In absolute terms this means that between 364 and 448 instances were abstained on for mutagenicity prediction and between 230 and 278 for carcinogenicity prediction. As accuracies reached for EST origin prediction were already high without abstention, the required abstention rates were much smaller with values between 9.5% and 24.3% which corresponds to 305 to 783 abstained instances due to the larger size of this dataset.

By decreasing the abstention costs additionally, classification accuracy could be improved even further. However there were limits to what could be achieved. For instance, the accuracy on EST prediction could be pushed up to 99.9% for the support vector machines with abstention costs of 10^{-3} leading to an abstention rate of 70%, but these results did not change anymore even for costs as low as 10^{-20}. This is exactly the observation that was expected. Abstention even at this point is more expensive than the few misclassifications because so many correct classified instances are abstained on. On the other hand, when reducing the abstention rates, accuracies decrease again. This illustrates strongly the trade-off between these two quantities. We have to pay for improved predictive accuracy by reduced coverage and what we can achieve therefore depends mostly on how much we are willing to pay.

6.3.4 Characteristics of Abstained Instances

As there are several possible explanations for instances to be abstained on, we examined if there are any common characteristics of abstained instances. For example, instances may either belong to a separate class not observed in the training set or alternatively show properties of both classes. Additionally, the inductive bias of the algorithm may prevent deriving an appropriate hypothesis for all instances. The inductive bias of a machine learning algorithm is the set of assumptions which allows it to generalize beyond the training data.

The first step was to analyze how many instances in the validation set were abstained on by all of the classifiers involved. If these numbers were decisively higher then expected at random, the obvious conclusion would be that the instances all classifiers agreed on to abstain did in fact exhibit some special properties. For all of the sets the same cost scenarios were considered as in the previous test. In most cases the number of instances all classifiers consented on to abstain were higher than expected at random. However, they were not high enough to suggest a great concurrence between the classifiers as to which instances are supposed to be abstained on.

As these results were inconclusive, an additional test was performed for which a new class was introduced composed of instances abstained on. The instances of this class were derived by calculating the optimal abstention windows using an internal cross-validation as before and then applying this windows to the test set of the external cross-validation. The cost scenarios were chosen such that about 30% of the instances were abstained on and all three classes were present in approximately equal proportions in the new data sets. Instances abstained on were

Classifier 1	Classifier 2				
	SVM	Random Forests	Naive Bayes	PART	J4.8
SVM	55.4	58.9	51.5	58.8	58.2
Random Forests	58.2	61.7	52.9	59.8	61.7
Naive Bayes	77.8	78.9	74.3	80.4	78.0
PART	59.8	59.3	56.7	61.3	59.9
J4.8	67.4	66.5	59.6	66.1	67.4

Table 6.2: The table shows the classification accuracy (in %) achieved when introducing abstained instances as a separate class for mutagenicity prediction. Classifier 1 denotes the classifier whose optimal abstention window was used to define the instances of the third class. Classifier 2 was used to derive models based on the modified datasets.

given the new class label, whereas the remaining instances kept their original class. These modified datasets were used to train new models using all five of the classification algorithms. The performance of these models was estimated by tenfold cross-validation. Of course, these estimates were expected to be highly optimistic as the information from the same data set was used to establish the third class.

Nevertheless, certain conclusions can be drawn from these findings as the results for mutagenicity prediction show (see table 6.2). Here we observed that for all modified datasets but one the classification accuracies were decisively lower than the previous results on the original dataset which implies that the abstained instances to a large extent do not represent a separate class or exhibit special properties. However, for the dataset modified with the help of the Naive Bayes classifier the results were comparable to previous results even under consideration of overfitting effects.

When analyzing an unpruned decision tree calculated for this data set an interesting observation could be made. Almost two thirds of the instances abstained on were associated with one leaf of the tree. The rule obtained by following the path from the root to this leaf tested the occurrence of a number of molecular fragments. If none of these fragments were found in an instance, it was assigned to the abstention class. A similar observation was made on the rules calculated by PART, yet in this case different fragments were concerned. In decision trees for the remaining modified datasets on the contrary, the instances of the additional class were distributed over many leaves and no bias towards one individual leaf was observed.

These results imply that instances abstained on by the Naive Bayes classifier were characterized by a lack of certain molecular fragments important for an appropriate classification. Obviously, abstaining is the most sensible decision for such instances.

6.4 Analysis of Cost Curves

For the previous section the benefits of abstaining were illustrated using fixed cost scenarios. However, the actual cost scenarios may differ from the assumed cost scenarios decisively for each of the three datasets involved. Both in mutagenicity and carcinogenicity costs associated with false negative predictions are distinctively higher than the costs for false positive predictions. Furthermore, the exact values for abstaining are unclear. For EST prediction, both types of misclassification costs probably are approximately equal but the abstention costs still remain difficult to set. In all those cases, determining the actual class distributions is problematic. Thus, to further analyze and compare classifiers cost curves as presented previously were calculated.

We use the mutagenicity dataset to illustrate the capability of cost curves for uncertain costs (and class distributions). The mutagenicity dataset is well-suited for this purpose, as reasonably high classification accuracies could be obtained contrary to carcinogenicity prediction, but not as high values as for EST classification, where abstaining was only of minor use. Furthermore, no single classifier among those calculated was optimal for all cost scenarios, whereas for EST classification on the other hand support vector machines outperformed all other classifiers.

Mutagenicity prediction constitutes a perfect example for the problems in assigning exact costs and class distributions. Although we can easily establish the class with highest misclassification costs, we are at a loss to determine exactly how much more expensive false negative predictions are compared to false positive predictions as there are many factors which play into establishing the costs. Moreover, it is unclear which percentage of chemical compounds is mutagenic since corresponding results exist only for a minority of compounds due to high costs for tests and ambiguous outcomes of experiments.

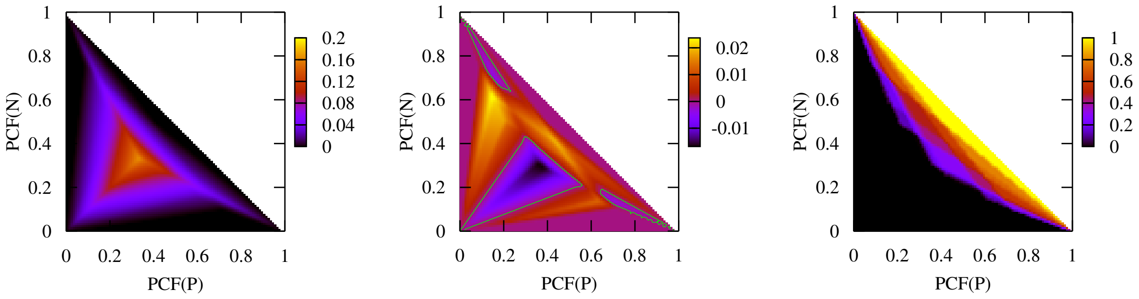
The benefits of abstaining in this case are obvious. We should not rely completely on a computational model for risk assessment given a limited and noisy dataset. However, if we can identify a subset of instances for which we are able to give predictions of high confidence, some experimental (wet lab) tests may be avoided or prioritized differently. As we have seen before, abstaining is only possible if the costs for it are rather low compared to costs for misclassifications. If we defined abstention costs by the additional experiments required, abstention clearly would be too expensive. However, we can reason that without abstaining these experiments would have to be conducted anyway since evidently no prediction could be relied upon.

To calculate cost curves only a validation set was required and as a consequence only one cross-validation was performed to obtain the following curves. If the actual performance of the abstaining classifiers was to be examined as well, nested cross-validations would have to be used again. A corresponding analysis is described later on. The aim of the following section is to show how to use cost curves to obtain information about optimal abstention windows and costs if the exact cost matrix cannot be established.

6.4.1 Type I Cost Curves

As we do not know the correct class distribution between mutagenic and non-mutagenic compounds, the first type of cost curves is the intuitive choice. The starting step in the analysis was a comparison against the trivial classifiers which either label all instances as positive or negative or abstain completely. As a given classifier comprises abstention windows which do exactly that, its cost curve can never be worse than the cost curve for the trivial classifiers. Nevertheless, the difference between the cost curves is of interest as it indicates how much better a classifier is with respect to the trivial classifiers.

When calculating the differential cost curve relative to the trivial classifiers for all classification algorithms, we could observe a certain behavior for all of them. If either $PCF(P)$ or $PCF(N)$ was low or both were high, none of the classifiers outperformed the trivial classi-



(a)

(b)

(c)

Figure 6.3: Differential cost curves and optimal abstention rate for cost curves which allow changing the cost scenarios as well as changing class distributions (Type I). Figure (a) depicts the differential cost curve between the trivial classifiers and Random Forests and figure (b) the difference in cost curves between J4.8 and Random Forests. The cost scenarios for which no difference between the two classifiers was observed are marked by the green line. Figure (c) then shows the optimal abstention rate for the Random Forests classifier.

fiers. As explained before, these situations either correspond to low costs for false negatives or false positives or low costs for abstaining. In any such case, the best advice is always to resort to the trivial classifiers. The absolute value of the difference increased with $PCF(P)$ or $PCF(N)$ up to the point at which abstaining became cheap enough to improve expected cost. For an example see figure 6.3(a).

The differential cost curves to the trivial classifiers suggested a ranking among classifiers. The differences appeared to be smallest for Naive Bayes, thus it was assigned the last position in our ranking. The next positions were in increasing order of performance SVMs, PART and J4.8. The best of these classifiers appeared to be Random Forests, as the areas in which it outperformed the trivial classifiers were most extended and also the absolute value of the difference was slightly higher. However these differences were very subtle and probably not statistically significant. The computation of the cost curves would have to be repeated several times to determine the statistical significance of the differences. Interestingly enough, this ranking differed from the ranking induced by the estimated accuracies.

The next step was to compare the pairwise differential cost curves. To avoid having to consider all of them, the ranking induced by the differential cost curves to the trivial classifiers was used and only Random Forests was compared to SVMs, PART, J4.8 and Naive Bayes. The differential cost curve for Naive Bayes and Random Forests was positive in all entries, therefore Naive Bayes was discarded because we could always do better with Random Forests. For the other algorithms the results were more ambiguous and the corresponding differential cost curves contained both positive and negative entries, as can bee seen in figure 6.3(b), for example.

Instead of comparing the differential cost curves for each pair of classifiers, we instead computed the minimum cost curve and the corresponding index matrix. As expected, Naive Bayes did not occur at all in the index matrix. Moreover, the index matrix proved to be additionally useful as it showed that PART minimized the cost for only very few points in the cost space, therefore it was deemed to be reasonable to exclude this one as well. We were now left with three classifiers, i.e. SVMs, J4.8 and Random Forests. Yet, for those cost scenarios where SVM performed best, the difference in expected cost to J4.8 or Random Forests was very small and most likely insignificant, so it was eliminated as well.

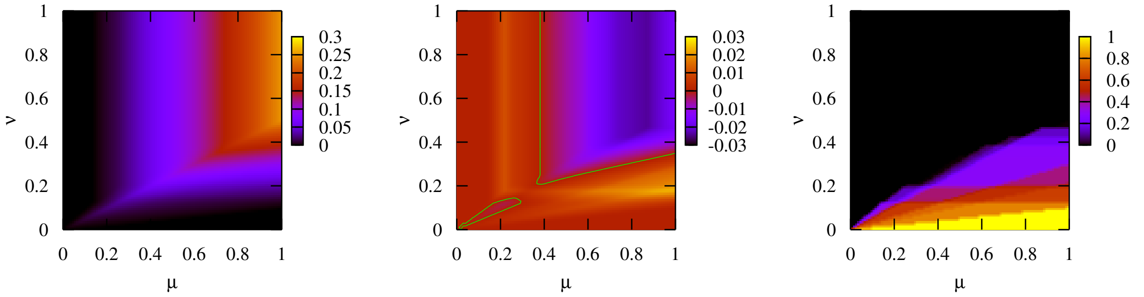
The classifiers that now remained were exactly those two which already topped the list based on classification accuracy. However, we now have a good indication for which cost and class distributions we might take either of the two. For instance, let be $P(P) = \frac{1}{3}$, $C(P, n) = 9$, $C(N, p) = 4.5$ and $C(\perp) = 4$. This implies $PCF(P) = PCF(N) = 0.3$ and figure 6.3(b) suggests using J4.8 for this cost scenario. On the other hand, for $C(P, n) = 6$, $C(N, p) = 7.5$ and $C(\perp) = 3$, Random Forests would be the better choice.

So far, we only discussed which classifier to choose, but did not take into consideration if this actually involved abstention. Figure 6.3(c) illustrates the abstention rate associated with the optimal abstention window for Random Forests. Quite clearly for most cost scenarios, abstention was not involved at all due to high abstention costs. Right enough, for the first cost scenario suggested no abstention was applied. However, in the second case the optimal abstention window did abstain on around 20% of the instances and abstention could improve the expected cost for this cost scenario.

6.4.2 Type II Cost Curves

In order to examine the second type of cost curves, fixed class distributions were required. Lacking further information about the actual ratio of mutagenic to non-mutagenic chemicals in the "chemical universe", i.e. instance space, the distribution of the data set was used. Of course, in reality mutagenic chemicals are supposed to be distinctively less common than non-mutagenic ones and any other fixed distribution could have been used for our tests.

The analysis was performed in the same way as before. First the differential cost curves between the trivial classifiers and all five algorithms were computed (see figure 6.4(a) for an example). This allowed to propose a ranking quite similar to before, the only difference being that PART and J4.8 switched their places within the ranking.



(a) (b) (c)

Figure 6.4: Differential cost curves and optimal abstention rate for cost curves which change only the cost scenarios involved (Type II). Figure (a) presents the differential cost curve between the trivial classifiers and the Random Forests classifier and figure (b) the difference in cost curves between J4.8 and Random Forests. The green line shows the scenarios for which the difference is zero. The last figure illustrates the optimal abstention rate for the Random Forests classifier.

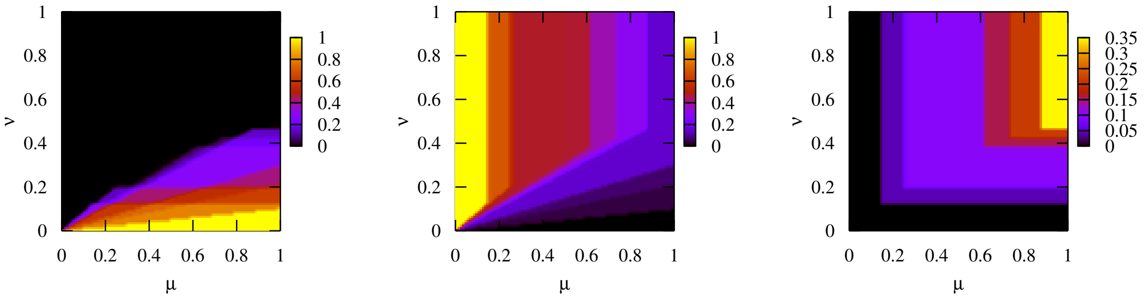
Again Random Forests was on top of the list and was compared against all other classifiers. These comparisons yielded results similar to the above ones. Naive Bayes was outperformed by Random Forests for all possible scenarios, whereas for the other three models the picture was ambiguous. For low costs of false positive (i.e. low μ) or low abstention costs (i.e. low ν), Random Forests had lowest expected cost, but with increasing ν and μ, the other classifiers performed better. Analyzing the index matrix again lead to the conclusion that Naive Bayes and PART could be ignored. SVMs were only of use for very low levels of false positives costs. For all other cost scenarios either J4.8 or Random Forests were the best choice. In figure 6.4(b) the decision boundaries for using either of these can be seen.

Contrary to before, the cost curves are quite easy to analyze. Suppose we have $\nu > 0.2$ and $\mu < 0.3$, then Random Forest should be chosen, while for $\nu > 0.4$ and $\mu > 0.6$ J4.8 is the best choice. By examining the corresponding abstention rates we observe that Random Forests are superior to J4.8 for scenarios which either have false positive costs below 0.4 or abstention costs low enough to enable abstention. This suggests than Random Forest could make better use of abstention on the given data. Of course, on any other dataset the situation might be reversed.

6.4.3 Optimal Abstention Rate and False Positive and Negative Rate

After illustrating how cost curves can be used to compare classifiers, we examined how abstention rate, false negative rate and false positive rate of the optimal abstention windows change with costs (and class distributions), first using the cost curves of the second type since they are easier to analyze. Figure 6.5 shows the abstention rates, false positive rates and false negative rates corresponding to the optimal abstention windows in the cost curve for Random Forests. Similar results can be obtained for all classification algorithms and each of the presented classification tasks.

It is evident that abstaining was only a valid choice if the costs for abstaining were distinctively smaller than the costs for false negatives and for false positives as well, which confirms the results from chapter 2. Furthermore, we observed a negative correlation between



(a) (b) (c)

Figure 6.5: This figure illustrates the relationship between optimal abstention rate (a), false positive rate (b) and false negative rate (c). The curves were calculated using the Random Forests classifier on the mutagenicity data set and tenfold cross-validation.

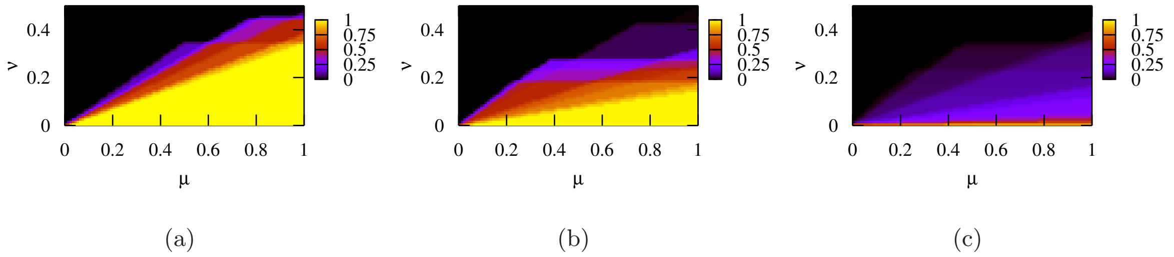


Figure 6.6: Optimal abstention rates for the three classification tasks for varying cost scenarios. The curves show the abstention rate of the optimal abstention window instead of its expected cost for the carcinogenicity data (a), the mutagenicity data (b) and the EST data (c). Support vector machines were used in each case.

abstention rate and false positive rate. With increasing abstention costs ν yet constant false positive costs μ, more and more of the abstained instances were labeled positive thus leading to an increase in false positive rate. If costs for abstention reached the point beyond which abstention is too expensive, false positive rate did not change as long as μ remained constant. False negative rate on the other hand appeared to be positively correlated to both μ and ν. This means that the more expensive abstaining or false positive classifications became, the more instances were labeled negative, perhaps wrongly so.

Similar results could be achieved with the first type of costs curves (see 6.3(c)). Abstaining was only put into action for small abstention costs, i.e. high values for $PCF(N)$ and $PCF(P)$, which at the same time lead to a reduction in false positive rate and false negative rate. False negative rate was high for small values of $PCF(P)$ and decreased while $PCF(P)$ increased or $PCF(N)$ decreased. This is in accordance with a scenario where either the probability for the positive class or the cost for misclassifying it is small. Similar observations were made for the false positive rate. This makes it clear that the choice between classification and abstention is not only influenced by costs associated with certain events but also by class distributions. Even if misclassification costs are much higher than abstention costs but one class is very rare, abstention is still more expensive than always predicting the majority class. However, rareness in most cases leads to high misclassification costs for a class.

6.4.4 Optimal Abstention Rate and Classification Accuracy

In the previous tests only the cost curves for one dataset were discussed. The same evaluations were of course performed on the EST and carcinogenicity dataset as well. To avoid unnecessary repetitions this is not specified here any further. Additionally, the two datasets were used to explore the relationship between optimal abstention rate and classification accuracy achieved without abstaining. The predictive accuracies observed on the mutagenicity data did not vary sufficiently to allow statements about this relationship on a larger scale. Contrary to that, the accuracies obtained for the three datasets varied greatly, therefore the optimal abstention rates were compared between classification tasks. The optimal abstention rates for support vector machines are depicted in figure 6.6 using the second type of cost curves. Again, similar results could be obtained for all classification algorithms and also for cost curves of type I.

These figures suggest a strong dependency between optimal abstention rates and classification accuracies achieved. For carcinogenicity data where the estimated classification accuracy for all models hardly exceeded the baseline accuracy, abstention could decrease expected cost even for comparatively high abstention rates for which abstention on the mutagenicity data or the EST data already was too expensive. A similar effect can be observed for mutagenicity prediction compared to EST classification. For the last task only a small fraction of instances was abstained on even for very low abstention costs. Therefore, abstention rates decreased at the same time as classification accuracy increased.

These observations can be easily explained. Abstention is always the “last resort” when classification is too expensive. This can be either due to high misclassification costs or high misclassification rates. If classification already can be performed with high confidence, abstention is only necessary for the most ambiguous instances.

6.5 Performance of Combined Classifiers

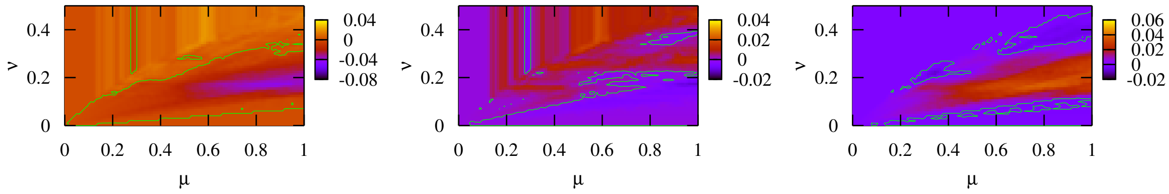
The mutagenicity data set was used a second time to analyze the performance of combined abstaining classifiers within cost curves of the second type for the same reasons as before. In the last section, cost curves were shown to be of help for determining which classifiers to choose in which cost scenario, yet the performance of the optimal abstention windows was not evaluated on an additional test set. Such an evaluation is performed in the next section additionally to examining the combining approaches. Therefore nested loops of cross-validation were again necessary.

6.5.1 Baseline Classifier

The methods for combining abstaining classifiers were compared against a baseline classifier which always chooses the abstention window with minimum expected cost of any of the base classifiers for each cost scenario. Although this method is optimal on the validation set it may not be optimal on the test set and any of the base level classifiers can surpass the baseline classifier in expected cost for certain cost scenarios. Hence, the cost curve derived by applying the baseline classifier to the test set was compared against similar curves for each of the base level classifiers.

The comparison showed that the baseline classifier outperformed the support vector machines for most cost scenarios but for small costs for abstention or false positives. When comparing the Random Forest model to the baseline classifier, the results were more ambiguous and for some of the cost scenarios the first classifier was better, for others the second one prevailed. The results were clearer for the remaining classifiers. In general, they were surpassed by the baseline classifier for those cost scenarios which allowed abstention. Yet, if no instances were abstained on each of the classifiers outperformed the baseline classifier for most values of false positive costs.

At first view, these results are confusing. As the baseline classifier effectively chooses only one of the classifiers, it should have equal expected cost to one of the other classifiers for every cost scenario. However, variation in the predictions occur due to the cross-validation. Within each fold a different classifier may be optimal for each cost scenario and therefore the baseline classifier does not correspond to one single classifier for each scenario. If only one



(a) (b) (c)

Figure 6.7: (a) Difference in expected cost between the baseline classifier and the cost curve derived by the direct sum method. (b) Difference in expected cost between the baseline classifier and the cost curve derived by majority vote. (c) Difference between the cost curves derived by direct sum and majority vote.

fold is examined the above described observation can indeed be made.

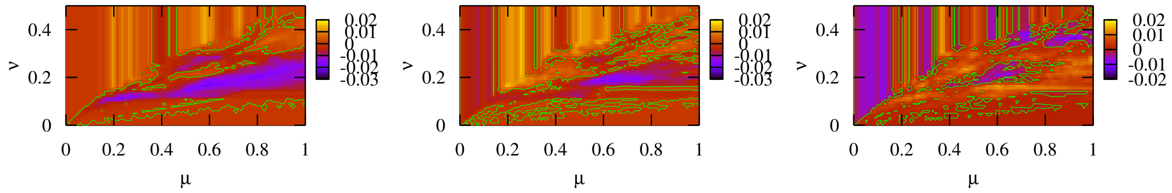
6.5.2 Weighted Voting

Having compared the baseline classifier against each base model, we proceeded with analyzing the voting methods and compared them against each other as well as against the baseline classifier which also presents a way to combine abstaining classifiers. Two voting methods as well as two weighting schemes were introduced before. Interestingly, the different weighting methods differed only insignificantly given the type of voting was the same in our analysis. This implies that the results of the votes do not depend on which of the two weighting schemes is used. Therefore we can conclude that the two weighting schemes are equivalent and can focus on comparing the voting methods using only one weighting scheme. In this case the second one was used.

We have presented two voting methods which are denoted as the direct sum and the majority vote method. Figure 6.7(a) shows the differential cost curve of the first method compared to the baseline classifier and figure 6.7(b) the same for the majority vote. Finally, figure 6.7(c) presents the differences between the direct sum and majority vote. The cost curves were restricted to the region with $\nu \leq \frac{1}{2}$. The results for greater values of ν do not change as no abstention is performed and consequently are of no interest.

The figures imply several conclusions. Both voting methods outperformed the baseline classifier for abstention costs that make abstaining too expensive. But in regions for which abstention still is possible the situation is less clear. The direct sum method was only in very few cases superior to the baseline classifier. Majority vote appeared to be more successful, but still for low abstention costs relative to the costs for false positives it performed worse than the baseline classifier.

When comparing the two methods directly we observed that for expensive abstention costs, both methods were equivalent. For moderately high abstention costs the majority vote method was by far better, whereas for very small ones the situation was reverse. These results can be explained by the different behavior of these methods towards abstention. The direct sum method has a clear bias against abstention, hence it did not abstain in most cases if the abstention windows combined were relatively small, as they were for high abstention costs. When costs decreased, abstention windows broadened and the majority vote method



(a) (b) (c)

Figure 6.8: Figure (a) shows the difference in expected cost between the baseline classifier and the original separate-and-conquer approach, figure (b) the difference between the baseline classifier and the second modification of the separate-and-conquer approach and figure (c) the difference between the original method and the second modification of the separate-and-conquer approach.

abstained regularly and classified even less instances than any of the abstention windows did, quite contrary to the direct sum method.

This tendency becomes obvious in the abstention rates for the two methods. For the majority vote method they were far higher then the corresponding abstention rates for the direct sum method. An even more interesting observation could be made by comparing the abstention rates for the majority vote method to the abstention rates of the baseline classifier. For most cost scenarios allowing abstention, the majority vote method in fact had higher abstention rates than the baseline classifier. Many of these corresponded to regions in which the baseline classifier outperformed the majority method. As a consequence, we can infer that a simple weighting and voting scheme is sufficient to combine non-abstaining classifiers but not to do the same for abstaining classifiers.

6.5.3 The Separate-and-Conquer Method

As an alternative to the voting method a separate-and-conquer approach has been described which avoids voting by using a sequence of abstention windows one after the other. Two modifications were introduced which differ in the abstention costs considered. The first modification uses abstention costs slightly smaller than the actual costs (in this case 90% of the original cost) while the second one increases abstention costs with each iteration starting from zero abstention costs until the actual costs are reached.

Again these three methods were compared against the baseline classifier (see figure 6.8). The original method outperformed the baseline classifier for low abstention costs on the one hand and for values of μ and ν which did not allow abstention on the other hand. Unfortunately, the baseline classifier still appears to be superior for large regions for which abstaining was performed. Again this may be due to the bias against abstention of the combining method. As expected, the first modification changed the behavior of the classifier only insignificantly. The second one, however, improved the expected costs for scenarios in which abstention took place such that the baseline classifier was superior in fewer cases. On the other hand, for high abstention costs the expected costs seemed to be increased. Therefore, the difference in performance between the original and modified version is not quite clear as figure 6.8(c) shows and no unequivocal winner could be determined.

6.6 Conclusion

The application of abstention to carcinogenicity and mutagenicity prediction as well as EST classification showed that the predictive performance can be improved considerably by restricting classifications to a subset of all instances. However, good results are in many cases associated with high abstention rates which makes abstention only useful if the costs for not classifying instances are deemed low enough. Additionally to that, we could show that in some cases abstained instances indeed exhibit specific properties which make it reasonable to refrain from classification.

If costs are uncertain or the benefits of abstaining unclear, cost curves can be computed to compare classification algorithms for specific tasks as well as to determine the cost scenarios which favor abstention. If we discover that for a specific application costs have to be smaller than a certain fraction of the false positive costs and we know that the costs are actually higher than that, we can clearly eliminate the option to abstain. On the other hand, if we are willing to accept the reduction in coverage involved, we can improve our results decisively.

An interesting observation could be made on the relationship between abstention rate and false positive and negative rate. Each of these rates can be reduced at the expense of the other two depending on the costs associated. Classification accuracy and abstention rate interact in a similar way and optimal abstention rates for any cost scenario are much smaller if high classification accuracy can be achieved even without abstention.

Finally, we analyzed the methods to combine several abstention windows and showed that in this way expected costs can be reduced at least for some cost scenarios. However, to get reliable results the analysis would have to be repeated several times based on different cross-validation splits as in some cases the observations were inconclusive.

Chapter 7

Theoretical Bounds for Abstaining Ensembles

In this chapter a theoretical analysis of abstaining for ensembles of classifiers is presented. The objective is to bound the expected cost and give a formula for the best abstention rate, so that the optimal thresholds for abstaining for a given cost scenario can be determined in constant time. First we introduce the setting we are working on and the PAC-Bayesian theorem which can be used to bound the expected error of ensembles using the empirical error on the training set.

7.1 The Learning Setting

We assume an instance space $\mathcal{X}$, a space of possible class labels $\mathcal{Y}$ and a fixed, yet unknown distribution D over labeled instances $(x, y) \in \mathcal{X} \times \mathcal{Y}$. Again the number of classes is restricted to two, thus we can define the set of labels as $\mathcal{Y} = \{-1, 1\}$ [†]. An instance $x \in \mathcal{X}$ is defined by a k-tuple $(x_1, .., x_k)$, where each x_j is taken from a domain A_j. A domain A_j is defined as a finite set of possible values for x_j. The training algorithm is presented with a training set $S \subseteq \mathcal{X} \times \mathcal{Y}$, which consists of m labeled instances drawn according to the distribution D and outputs a concept c which assigns class labels to each instance. For any instance (x, y) drawn according to D, $c(x) = 1$ if c classifies this instance as positive and $c(x) = -1$ otherwise. $\mathcal{C}$ denotes the class of all possible concepts and is assumed to be finite. Naturally, the objective of any training algorithm is to generate a concept c with low error probability $\mathbf{Pr}_{(x,y)\sim D}[c(x) \neq y]$ or low expected loss:

Definition 7.1 (Loss). *Given a labeled instance $(x, y) \in \mathcal{X} \times \mathcal{Y}$ and a classifier $c \in \mathcal{C}$, we define the loss of c on (x, y) as*

$$l(c, x, y) := I(c(x) \neq y)$$

where $I(F)$ is 1 if F is true and 0 otherwise.

Thus the loss is 0 if the prediction of c on (x, y) is correct and 1 otherwise. This is the standard zero-one loss function. Based on this notion of loss we can define the expected and empirical loss of a concept c.

[†] Here we deviate from the notation introduced in chapter 1 and 2 for practical reasons.

Definition 7.2 (Expected and empirical loss). *Let c be a classifier in $\mathcal{C}$. The expected loss of c is defined as*

$$l(c) := \mathop{\mathbf{E}}_{(x,y)\sim D}[l(c,x,y)].$$

The empirical loss of c on S is defined as the fraction of instances in S which c misclassifies:

$$\hat{l}(c,S) := \frac{1}{m}\sum_{(x,y)\in S} l(c,x,y).$$

Additionally, if we have a probability measure Q on classifiers from $\mathcal{C}$, we use $l(Q)$ for $\mathbf{E}_{c\sim Q}[l(c)]$ and correspondingly $\hat{l}(Q,S)$ for $\mathbf{E}_{c\sim Q}[\hat{l}(c,S)]$. An ensemble of classifiers is effectively described by a probability measure Q. The task of a training algorithm for ensembles is to find a posterior distribution Q which minimizes the expected loss given a prior distribution P over $\mathcal{C}$. The prior distribution P is provided by the user based on potential information about the target distribution D. If no such information is given, an uniform prior can be chosen. The prior and posterior distribution can be compared using the Kullback-Leibler divergence which is also called relative entropy.

Definition 7.3 (Relative Entropy). *Let Q and P be a probability distribution over $\mathcal{C}$. Then the relative entropy of Q with respect to P is defined as*

$$D(Q \parallel P) := \sum_{c\in\mathcal{C}} \left(Q(c)\ln\tfrac{Q(c)}{P(c)}\right)$$

The smaller $D(Q \parallel P)$, the more similar is the posterior distribution to the prior distribution. The relative entropy, although not being symmetric, satisfies several important mathematical properties as e.g. that it is always nonnegative and that it is only zero if $Q(c) = P(c)\,\forall c \in \mathcal{C}$.

7.2 PAC Bayesian Bound for Voting Ensembles

We have now introduced all necessary terms to present the PAC-Bayesian theorem. In the following we use the notation $\forall^\delta S\ \phi(S)$ to denote that $\phi(S)$ holds for all but a fraction δ of possible samples S. Formally, this means that

$$\forall^\delta S\ \phi(S) \iff \mathop{\mathbf{Pr}}_{S\sim D}[\phi(S)] \geq 1-\delta.$$

Theorem 7.4 (PAC-Bayesian (McAllester, [35])). *Let P be a prior distribution over $\mathcal{C}$ and $\delta > 0$. Then we have that*

$$\forall^\delta S \quad \forall Q \quad l(Q) \leq B(Q,P,m,\delta)$$

where Q ranges over all distributions on $\mathcal{C}$ and

$$B(Q,P,m,\delta) := \hat{l}(Q,S) + \sqrt{\frac{D(Q \parallel P) + \ln\frac{1}{\delta} + \ln m + 2}{2m-1}}.$$

The PAC-Bayesian theorem bounds the loss expected if drawing a concept c from $\mathcal{C}$ according to Q at random, depending on the empirical loss on the training data as well as the divergence between prior and posterior distribution, δ and the training set size m. The larger the training set size the smaller is the difference between expected and empirical loss.

Instead of drawing concepts at random, we can construct a voting ensemble such that the weight of each concept is given by the posterior distribution.

Definition 7.5. *Let Q be a distribution over $\mathcal{C}$. Then we define the score of Q on x as $c(Q,x) := \mathbf{E}_{c\sim Q}[c(x)]$ and the voting classifier of Q as*

$$c_V(Q,x) := \begin{cases} 1 & \text{if } c(Q,x) > 0 \\ -1 & \text{otherwise} \end{cases}$$

The expected error of the voting classifier is defined as $l_V(Q) := \mathbf{E}_{(x,y)\sim D}[I(c_V(Q,x) \neq y)]$.

Obviously, we have that $c(Q,x) \in [-1:1]\,\forall\, x$ and $y\, c_V(Q,x) \geq 0$ if instance x is classified correctly and $y\, c_V(Q,x) \leq 0$ otherwise. The following theorem bounds the expected error for any posterior distribution Q analogously to the theorem for the rule learning setting from Rückert and Kramer [44]. We use the abbreviations $\underset{D}{\mathbf{Pr}}[F]$ to denote $\mathbf{Pr}_{(x,y)\sim D}[F]$ and $\underset{Q}{\mathbf{E}}[F]$ to denote $\mathbf{E}_{c\sim Q}[F]$.

Theorem 7.6 (Rückert and Kramer, [44]). *Let P be the prior distribution over $\mathcal{C}$, Q the posterior distribution and $\delta > 0$. Then we have that*

$$\forall^\delta\, S \quad \forall Q \quad l_V(Q) \leq 2B(Q,P,m,\delta)$$

Proof. First we see that

$$l_V(Q) = \underset{D}{\mathbf{Pr}}[y\, c_V(Q,x) \leq 0]. \tag{7.1}$$

Furthermore,

$$1 - 2l(Q) = 1 - 2\,\underset{D}{\mathbf{E}}\big[\underset{Q}{\mathbf{E}}[\mathrm{I}(c(x) \neq y)]\big] = 1 - 2\,\underset{D}{\mathbf{E}}\big[\underset{Q}{\mathbf{E}}[\frac{1}{4}(c(x)-y)^2]\big] \tag{7.2}$$

$$= 1 - \frac{1}{2}\underset{D}{\mathbf{E}}\big[\underset{Q}{\mathbf{E}}[c(x)^2 - 2c(x)y + y^2]\big] = 1 - \frac{1}{2}\Big(1 - 2\,\underset{D}{\mathbf{E}}\big[\underset{Q}{\mathbf{E}}[c(x)y]\big] + 1\Big) \tag{7.3}$$

$$= \underset{D}{\mathbf{E}}[y\, c(Q,x)] \tag{7.4}$$

Equation (7.2) is a consequence of $\mathrm{I}(a \neq b) = \frac{1}{4}(a-b)^2$ for $a,b \in \{-1,+1\}$, (7.3) results from the fact that $a^2 = 1$ for $a \in \{-1,+1\}$. Thus by applying theorem 7.4, we get

$$\forall^\delta\, S \quad \forall Q : \underset{D}{\mathbf{E}}[y\, c(Q,x)] = 1 - 2l(Q) \geq 1 - 2B(Q,P,m,\delta) \tag{7.5}$$

Now we define a random variable $M := 1 - y\, c(Q,x)$. As $\forall x,y,Q : y\, c(Q,x) \in [-1,1]$, we have that $M \geq 0$, which allows us to use Markov's inequality:

$$\forall \varepsilon > 0 : \underset{D}{\mathbf{Pr}}\big[M \geq \varepsilon\, \underset{D}{\mathbf{E}}[M]\big] \leq \frac{1}{\varepsilon} \tag{7.6}$$

By substituting the definition of M, we observe that

$$\forall \varepsilon > 0 : \Pr_D \left[y\, c(Q,x) \le 1 - 1\varepsilon + \varepsilon \mathop{\mathbf{E}}_D [y\, c(Q,x)] \right] \le \frac{1}{\varepsilon}$$

and because of equation (7.5)

$$\forall \varepsilon > 0 \quad \forall^{\delta} S \quad \forall Q : \Pr_D \left[y\, c(Q,x) \le 1 - 2\varepsilon B(Q,P,m,\delta) \right] \le \frac{1}{\varepsilon} \tag{7.7}$$

The theorem then follows from equation (7.1) by setting

$$\varepsilon = \frac{1}{2\, B(Q,P,m,\delta)}.$$

□

7.3 Bounding the Expected Cost of Abstaining Classifiers

Based on the notion of a voting classifier, we can define an abstaining voting classifier c_V^{θ}, which abstains on all instances for which the absolute value of the score is below a given threshold θ. Note that we assume the same threshold θ for both positive and negative classification at this point. If an instance is abstained on, it is given label 0.

Definition 7.7. *The abstaining voting classifier c_V^{θ} is defined as*

$$c_V^{\theta}(Q,x) := \begin{cases} 1 & \text{if } c(Q,x) \ge \theta \\ 0 & \text{if } -\theta < c(Q,x) < \theta \\ -1 & \text{if } c(Q,x) \le -\theta. \end{cases}$$

Analogously to theorem 7.6, the expected loss of the abstaining voting classifier can be bounded by the PAC-Bayesian theorem (see also [44]). However, as our intention is to provide a formula for the optimal abstention threshold θ, expected loss is insufficient, since it completely ignores abstention costs. Thus, instead of bounding the expected error, the goal now is to bound expected cost. First, we concentrate on the case of equal misclassification costs and extend this to unequal misclassification costs in the later course.

7.3.1 Equal Misclassification Costs

We now assume *equal misclassification costs* – i.e. $C(P,n) = C(N,p) = 1$ – and that costs for abstention are always smaller than the misclassification costs. Thus we can formulate expected cost as a function.

Definition 7.8. *Let $\nu \in [0:1]$. The function $cost(Q,x,y)$ is defined as*

$$cost(Q,x,y) := \begin{cases} 1 & \text{if } y\, c(Q,x) \le -\theta \\ \nu & \text{if } -\theta < y\, c(Q,x) < \theta \\ 0 & \text{if } y\, c(Q,x) \ge \theta \end{cases}$$

Additionally, we define a random variable $\mathcal{L} := cost(Q,x,y)$. Then the expected cost of the abstaining voting classifier γ_V^{θ} is defined as

$$\gamma_V^{\theta} := \mathop{\mathbf{E}}_D[\mathcal{L}] = 1 \Pr_D[y\, c(Q,x) \le -\theta] + \nu \Pr_D[-\theta < y\, c(Q,x) < \theta].$$

The following theorem bounds the expected cost of the abstaining voting classifier using the PAC-Bayesian theorem.

Theorem 7.9. *Let P and Q be defined as before, let $\delta > 0$, $\nu \in [0:1]$ and $\theta \in [0:1)$. We then have*

$$\forall^{\delta} S \quad \forall Q \quad \gamma_V^{\theta} \leq (1-\nu)\frac{2B(Q,P,m,\frac{\delta}{2})}{1+\theta} + \nu\frac{2B(Q,P,m,\frac{\delta}{2})}{1-\theta}$$

Proof. First observe that

$$\begin{aligned}
\gamma_V^{\theta} &= \mathop{\mathbf{Pr}}_{D}[y\,c(Q,x) \leq -\theta] + \nu \cdot \mathop{\mathbf{Pr}}_{D}[-\theta < y\,c(Q,x) < \theta] \\
&= \mathop{\mathbf{Pr}}_{D}[y\,c(Q,x) \leq -\theta] + \nu \cdot \Big(1 - \mathop{\mathbf{Pr}}_{D}[y\,c(Q,x) \leq -\theta] - \mathop{\mathbf{Pr}}_{D}[y\,c(Q,x) \geq \theta]\Big) \\
&= (1-\nu)\mathop{\mathbf{Pr}}_{D}[y\,c(Q,x) \leq -\theta] + \nu\big(1 - 1 + \mathop{\mathbf{Pr}}_{D}[y\,c(Q,x) < \theta]\big) \\
&\leq (1-\nu) \cdot \mathop{\mathbf{Pr}}_{D}[y\,c(Q,x) \leq -\theta] + \nu \cdot \mathop{\mathbf{Pr}}_{D}[y\,c(Q,x) \leq \theta] \qquad (7.8)
\end{aligned}$$

By setting

$$\varepsilon = \frac{1+\theta}{2B(Q,P,m,\delta)} \quad \text{and} \quad \varepsilon = \frac{1-\theta}{2B(Q,P,m,\delta)}$$

respectively, in equation (7.7) analogously to the proof of theorem 7.6 we get

$$\forall^{2\delta} S \quad \forall Q \quad \gamma_V^{\theta} \leq (1-\nu)\frac{2B(Q,P,m,\delta)}{1+\theta} + \nu\frac{2B(Q,P,m,\delta)}{1-\theta}.$$

This is, of course, equivalent to the statement of this theorem. (Substitute δ by $\frac{\delta}{2}$). □

The presented bound effectively consists of two parts, which are weighted according to misclassification and abstention costs. This becomes clear by defining a function f with

$$f(x) = \frac{1}{1+x}. \qquad (7.9)$$

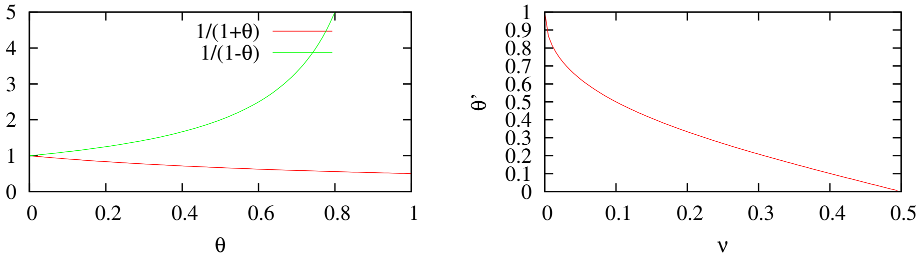
Then we can rewrite the bound from theorem 7.9 as

$$\forall^{\delta} S \quad \forall Q \quad \gamma_V^{\theta} \leq 2B(Q,P,m,\tfrac{\delta}{2})\Big((1-\nu)\cdot\underbrace{f(\theta)}_{(1)} + \nu\cdot\underbrace{f(-\theta)}_{(2)}\Big) \qquad (7.10)$$

(1) decreases as θ increases and thus rewards higher rates of abstention, whereas (2) increases with θ and thus penalizes abstention (see also figure 7.1(a)). However, the growth of (2) is much stronger than the growth of (1) and as a consequence abstention is penalized immensely except for very low abstention costs. By differentiating equation (7.10) we can determine θ', i.e. the optimal value for θ:

$$\theta' = \begin{cases} \frac{1-\sqrt{4\nu(1-\nu)}}{1-2\nu} & \text{if } \nu < \frac{1}{2} \\ 0 & \text{otherwise} \end{cases} \qquad (7.11)$$

Using equation (7.11) we can now easily compute the threshold for abstention given a certain ν. For instance if $\nu = \frac{1}{4}$, we observe that $\theta' \approx 0.27$. Figure 7.1(b) shows the optimal value of θ for all $\nu \in [0:\frac{1}{2})$.



(a) (b)

Figure 7.1: Figure (a) illustrates the behavior of the functions $\frac{1}{1+\theta}$ (red line) and $\frac{1}{1-\theta}$ (green line). Obviously the second function grows much stronger than the first one declines and therefore strongly penalizes abstention. Figure (b) depicts the optimal value for θ for values of ν between 0 and $\frac{1}{2}$.

Note that the presented bound does not depend on any specific characteristic of the PAC-Bayesian bound. In fact, we could use any bound on $l(Q)$ to get different bounds on the expected cost of the abstaining voting classifier. Nevertheless this would have no effect on the optimal abstention rate, as the value for the bound on $l(Q)$ is only a constant coefficient. Furthermore, other ways could be pursued to bound γ_V^θ not using Markov's inequality or $l(Q)$.

7.3.2 Unequal Misclassification Costs

The above results allow us to compute the optimal abstention threshold only when misclassification costs are equal. Unfortunately, misclassification costs differ more often than they do not. Thus we also have to bound the expected cost for *unequal misclassification costs*. We still make the assumption that the same threshold θ is used for positive and negative classification. For this case analyzing the value of $y\,c(Q,x)$ is insufficient as it only allows us to differ between correct and wrong classification, but not between the types of misclassification. We introduce an additional random variable, which makes it possible to do exactly that.

Definition 7.10. *Let (x,y) be drawn according to D. We define a random variable $\mathcal{Z}$ with*

$$\mathcal{Z} := y - c(Q,x).$$

We have that $\mathcal{Z} \in [-2,2]$ and $\mathcal{Z} \geq 0$ for positive instances and $\mathcal{Z} \leq 0$ for negative ones.

$\mathcal{Z}$ now allows us to distinguish between the misclassification of a positive instance and the misclassification of a negative instance. In fact, it even allows to discern abstaining on a positive instance from abstaining on a negative one as well as correct classification on a positive instance or a negative one. To make this clear we look at the values of $\mathcal{Z}$ for different values of y and $c(Q,x)$. We know that $c(Q,x) \in [-1:1]$. For a negative instance $\mathcal{Z} = -1 - c(Q,x)$, which is always less or equal to zero. If the instance is misclassified we have that $c(Q,x) \geq \theta$ and thus $\mathcal{Z} \leq -1-\theta$. Correspondingly, we get that $\mathcal{Z} \geq -1+\theta$ for a correct classification of a negative instance and $-1-\theta < \mathcal{Z} < -1+\theta$ if a negative instance

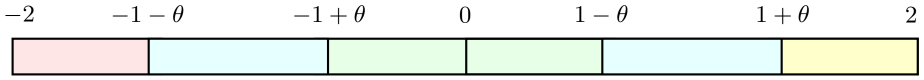


Figure 7.2: The figure shows the interesting ranges for $\mathcal{Z}$. The green parts are instances which are classified correctly. In blue we see the range for abstention. Red are negative instances which are classified incorrectly and yellow are misclassified positive instances.

is abstained on. Similar results are obtained for positive instances. Figure 7.2 shows exactly the ranges of values $\mathcal{Z}$ assumes for different events.

Again we define a function giving the cost associated with each event and presume that false negative classifications have highest misclassification costs. Thus we use a normalized cost matrix as presented in chapter 2 with $C(P, n) = 1$ and $C(N, p) = \mu$ for $\mu \in [0:1]$ as well as $C(\perp) = \nu$ for $\nu \in [0:1]$. Furthermore we impose the restriction that $\nu \leq \mu$.

Definition 7.11. *Let ν, $\mu \in [0:1]$. The function $cost(Q, x, y)$ is defined as*

$$cost(Q,x,y) := \begin{cases} 1 & \text{if } \mathcal{Z} \geq 1+\theta \\ \mu & \text{if } \mathcal{Z} \leq -1-\theta \\ \nu & \text{if } -1-\theta < \mathcal{Z} < -1+\theta \vee 1-\theta < \mathcal{Z} < 1+\theta \\ 0 & \text{otherwise} \end{cases}$$

$\mathcal{L}$ is defined as before. Then the expected cost of the abstaining voting classifier γ_V^θ is defined as

$$\gamma_V^\theta := \mathop{\mathbf{E}}_D[\mathcal{L}] = 1 \mathop{\mathbf{Pr}}_D[\mathcal{Z} \geq 1+\theta] + \mu \mathop{\mathbf{Pr}}_D[\mathcal{Z} \leq -1-\theta] + \nu \mathop{\mathbf{Pr}}_D[-1-\theta < \mathcal{Z} < -1+\theta \vee 1-\theta < \mathcal{Z} < 1+\theta].$$

Again we use the PAC-Bayesian theorem to bound the expected cost of the abstaining voting classifier with unequal misclassification costs, which results in the following theorem.

Theorem 7.12. *Let P and Q be defined as before. Let $\delta > 0$, ν, $\mu \in [0:1]$ and $\theta \in [0:1]$. Then we have*

$$\forall^\delta S \quad \forall Q \quad \gamma_V^\theta \leq (1+\mu-2\nu) \cdot \frac{2+2B(Q,P,m,\frac{\delta}{4})}{3+\theta} + 2\nu \cdot \frac{2+2B(Q,P,m,\frac{\delta}{4})}{3-\theta}$$

Proof. Observe that,

$$\begin{aligned}
\gamma_V^\theta =& \mathop{\mathbf{Pr}}_D[-\mathcal{Z} \le -1-\theta] + \mu \mathop{\mathbf{Pr}}_D[\mathcal{Z} \le -1-\theta] \\
&+ \nu\big(\mathop{\mathbf{Pr}}_D[-1-\theta < \mathcal{Z} < -1+\theta] + \mathop{\mathbf{Pr}}_D[1-\theta < \mathcal{Z} < 1+\theta]\big) \\
=& \mathop{\mathbf{Pr}}_D[-\mathcal{Z} \le -1-\theta] + \mu \mathop{\mathbf{Pr}}_D[\mathcal{Z} \le -1-\theta] \\
&+ \nu\big(\mathop{\mathbf{Pr}}_D[\mathcal{Z} < -1+\theta] - \mathop{\mathbf{Pr}}_D[\mathcal{Z} \le -1-\theta]\big) \\
&+ \nu\big(\mathop{\mathbf{Pr}}_D[-\mathcal{Z} < -1+\theta] - \mathop{\mathbf{Pr}}_D[-\mathcal{Z} \le -1-\theta]\big) \\
\le& (1-\nu) \mathop{\mathbf{Pr}}_D[-\mathcal{Z} \le -1-\theta] + (\mu-\nu) \mathop{\mathbf{Pr}}_D[\mathcal{Z} \le -1-\theta] \\
&+ \nu\big(\mathop{\mathbf{Pr}}_D[\mathcal{Z} \le -1+\theta] + \mathop{\mathbf{Pr}}_D[-\mathcal{Z} \le -1+\theta]\big) \qquad (7.12)
\end{aligned}$$

We now have to bound the expected value of $\mathcal{Z}$. Obviously we have that $\mathcal{Z} = y(1 - y\,c(Q,x)) \ge -1 + y\,c(Q,x)$. As a consequence we can observe that

$$\mathop{\mathbf{E}}_D[\mathcal{Z}] \ge \mathop{\mathbf{E}}_D[-1 + y\,c(Q,x)] = -1 + \mathop{\mathbf{E}}_D[y\,c(Q,x)] \overset{\text{Equ. (7.4)}}{=} -1 + 1 - 2l(Q)$$

Thus we have that

$$\forall^\delta S \quad \forall Q \quad \mathop{\mathbf{E}}_D[\mathcal{Z}] \ge -2B(Q,P,m,\delta) \qquad (7.13)$$

As $\mathcal{Z} \ge -2$ $\forall x, y, Q$, we can define a new random variable $M := 2 - \mathcal{Z}$ with $M \ge 0$ $\forall x, y, Q$. Again we can apply Markov's inequality:

$$\forall \varepsilon > 0 : \mathop{\mathbf{Pr}}_D\left[2 - \mathcal{Z} \ge \varepsilon \mathop{\mathbf{E}}_D[2-\mathcal{Z}]\right] \le \frac{1}{\varepsilon}$$

This is equivalent to

$$\forall \varepsilon > 0 : \mathop{\mathbf{Pr}}_D\left[\mathcal{Z} \le 2 - 2\varepsilon + \varepsilon \mathop{\mathbf{E}}_D[\mathcal{Z}]\right] \le \frac{1}{\varepsilon}$$

Because of equation (7.13) we have

$$\forall \varepsilon > 0 \quad \forall^\delta S \quad \forall Q : \mathop{\mathbf{Pr}}_D\Big[\mathcal{Z} \le 2 - \varepsilon\big(2 + 2B(Q,P,m,\delta)\big)\Big] \le \frac{1}{\varepsilon} \qquad (7.14)$$

and analogously

$$\forall \varepsilon > 0 \quad \forall^\delta S \quad \forall Q : \mathop{\mathbf{Pr}}_D\Big[-\mathcal{Z} \le 2 - \varepsilon\big(2 + 2B(Q,P,m,\delta)\big)\Big] \le \frac{1}{\varepsilon}. \qquad (7.15)$$

The theorem then follows from (7.14) and (7.15) by setting

$$\varepsilon = \frac{3+\theta}{2 + 2\,B(Q,P,m,\frac{\delta}{4})} \qquad \text{and} \qquad \varepsilon = \frac{3-\theta}{2 + 2B(Q,P,m,\frac{\delta}{4})}$$

respectively. □

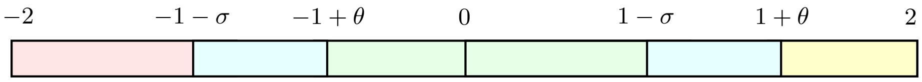


Figure 7.3: The figure shows the interesting ranges for $\mathcal{Z}$ if different threshold are used for positive and negative prediction. The green parts are instances which are classified correctly. In blue we see the range for abstention. Red are negative instances which are classified incorrectly and yellow are misclassified positive instances.

As for the case of equal misclassification costs, the new bound consists of two components, one of which penalizes abstention massively, whereas the other one rewards it. Consequently, when differentiating the bound to derive a formula for optimal abstention rate, we observe that abstention is only performed for small values of ν. Thus we have that

$$\theta' = \begin{cases} \min\left\{\frac{3(1+\mu)-3\sqrt{8\nu(1+\mu-2\nu)}}{1+\mu-4\nu}, 1\right\} & \text{if } \nu < \frac{1+\mu}{4} \\ 0 & \text{otherwise} \end{cases} \tag{7.16}$$

If $\mu = 1$ this results in the same restriction to abstention as before.

7.3.3 Different Thresholds for Abstention

The definition of $\mathcal{Z}$ makes it possible to introduce a new abstaining classifier, which has different thresholds σ and θ for abstaining for positive or negative values of $c(Q,x)$. Both σ and θ are between 0 and 1.

Definition 7.13. *The abstaining voting classifier $c_V^{\theta,\sigma}$ is defined as*

$$c_V^{\theta,\sigma}(Q,x) := \begin{cases} 1 & \textit{if } c(Q,x) \geq \sigma \\ 0 & \textit{if } -\theta < c(Q,x) < \sigma \\ -1 & \textit{if } c(Q,x) \leq -\theta. \end{cases}$$

$\mathcal{Z}$ is still defined as before, but the ranges for false negatives, false positives, abstained instances and correctly classified instances have changed. See figure 7.3 for the new ranges.

We define the expected cost $\gamma_V^{\theta,\sigma}$ analogously to definition 7.11. Thus we get the following theorem bounding the expected cost for the abstaining voting classifier $c_V^{\theta,\sigma}(Q,x)$.

Theorem 7.14. *Let P and Q be defined as before, let $\delta > 0$, μ, $\nu \in [0:1]$ and $\theta, \sigma \in [0:1]$, then we have*

$$\begin{aligned} \forall^\delta S \quad \forall Q \quad \gamma_V^{\theta,\sigma} \leq & (1-\nu) \cdot \frac{2+2B(Q,P,m,\frac{\delta}{4})}{3+\theta} + (\mu-\nu) \cdot \frac{2+2B(Q,P,m,\frac{\sigma}{4})}{3+\sigma} \\ & + \nu \cdot \frac{2+2B(Q,P,m,\frac{\delta}{4})}{3-\theta} + \nu \cdot \frac{2+2B(Q,P,m,\frac{\delta}{4})}{3-\sigma} \end{aligned}$$

Proof. Similar to before we have

$$\begin{aligned}\gamma_V^{\theta,\sigma} &= \underset{D}{\mathbf{Pr}}[\mathcal{Z} \geq 1+\theta] + \mu \underset{D}{\mathbf{Pr}}[\mathcal{Z} \leq -1-\sigma] \\ &\quad + \nu \underset{D}{\mathbf{Pr}}[-1-\sigma < \mathcal{Z} < -1+\theta \vee 1-\sigma < \mathcal{Z} < 1+\theta] \\ &\leq (1-\nu) \underset{D}{\mathbf{Pr}}[-\mathcal{Z} \leq -1-\theta] + (\mu-\nu) \underset{D}{\mathbf{Pr}}[\mathcal{Z} \leq -1-\sigma] \\ &\quad + \nu\big(\underset{D}{\mathbf{Pr}}[\mathcal{Z} \leq -1+\theta] + \underset{D}{\mathbf{Pr}}[-\mathcal{Z} \leq -1+\sigma]\big) \qquad (7.17)\end{aligned}$$

The theorem then follows from (7.14) and (7.15) by setting

$$\varepsilon = \frac{3+\theta}{2+2\,B(Q,P,m,\frac{\delta}{4})} \qquad \text{and} \qquad \varepsilon = \frac{3-\theta}{2+2B(Q,P,m,\frac{\delta}{4})}$$

as well as

$$\varepsilon = \frac{3+\sigma}{2+2\,B(Q,P,m,\frac{\delta}{4})} \qquad \text{and} \qquad \varepsilon = \frac{3-\sigma}{2+2B(Q,P,m,\frac{\delta}{4})}.$$

□

Note that theorem 7.12 is only a special case of the above theorem and results by setting $\sigma = \theta$. By differentiating the bound we then get for θ' and σ':

$$\theta' = \begin{cases} \min\left\{\frac{3-3\sqrt{4\nu(1-\nu)}}{1-2\nu}, 1\right\} & \text{if } \nu < \frac{1}{2} \\ 0 & \text{otherwise} \end{cases} \qquad (7.18)$$

and

$$\sigma' = \begin{cases} \min\left\{\frac{3\mu-3\sqrt{4\nu(\mu-\nu)}}{\mu-2\nu}, 1\right\} & \text{if } \nu < \frac{\mu}{2} \\ 0 & \text{otherwise} \end{cases} \qquad (7.19)$$

Again abstention is only performed for small values of ν.

7.4 Discussion

In this last section, we discuss the conclusions that can be drawn from the theoretical bounds presented and compare the bounds for equal and unequal misclassification costs. As mentioned before, theorem 7.12 is only a special case of theorem 7.14. This is a encouraging result since the voting abstaining classifier c_V^{θ} is also only a special case of the voting abstaining classifier $c_V^{\theta,\sigma}$.

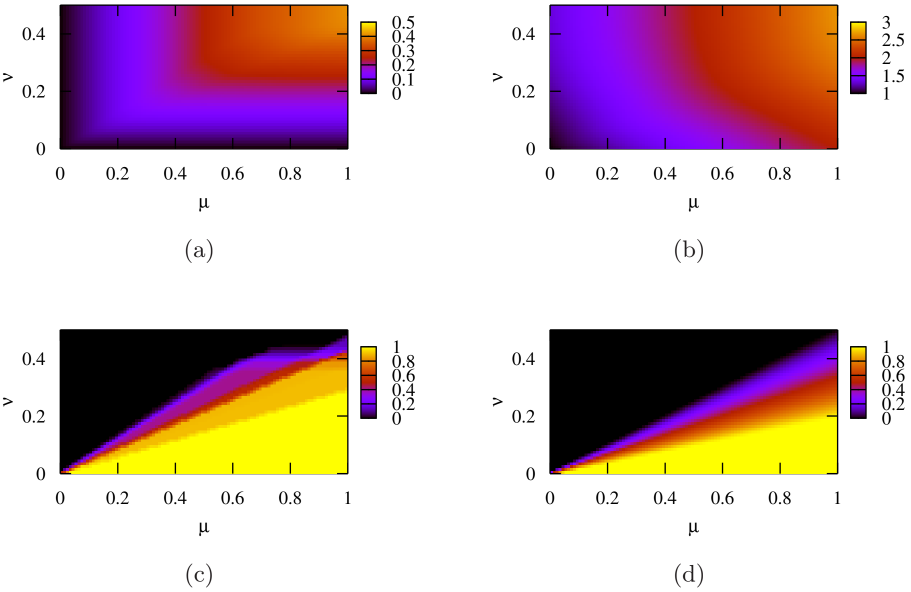
At best theorem 7.9 should also be a special case of theorem 7.12. Unfortunately, this is not the case. In fact the optimal abstention threshold as given by equation (7.16) when setting $\mu = 1$ to get equal misclassification costs, is exactly three times as high as the optimal abstention threshold given by equation (7.11). This is a consequence of the fact, that theorem 7.12 (and 7.14 as well) provides rather loose bounds as we observe when looking at a completely random dataset. For equal misclassification costs the expected cost for an abstention window a which does not abstain at all is $\mathbf{EC}(C, a) = 1 \cdot \frac{1}{2} + 1 \cdot \frac{1}{2} = 1$. From theorem 7.12, however, we get that

$$\forall^\delta S \quad \forall Q \quad \gamma_V^\theta \leq 2 \cdot \frac{2 + 2B(Q, P, m, \frac{\delta}{4})}{3} = \frac{4}{3} + k \quad \text{for some } k \geq 0.$$

Although the bound for the expected cost is not as tight as we would prefer, we can still use it to compute the optimal thresholds for positive and negative classifications. Here computing expected cost has an advantage over computing the expected error as costs are always defined arbitrarily and only their relations to each other are of interest. Hence, we are not interested in the exact value for expected cost, but in its behavior. This means that given two abstention window a_1 and a_2 with corresponding thresholds θ_1 (and σ_1 in case) and θ_2 (and σ_2) the bound on the expected cost of a_1 should be greater than the bound on the expected cost of a_2 if and only if the expected cost of a_1 is greater that the expected cost of a_2. If this is not possible, the bound should at least provide a good estimate of the optimal abstention window. This becomes clear by revisiting the random dataset example. It depends on the costs for abstaining, i.e. ν, if the non-abstaining classifier is actually the best possible. If ν is small, a threshold greater than zero will have lower values for the bound and thus be chosen.

It is remarkable that the optimal values for θ (and σ) for the presented bounds do not depend on classification error, i.e. the bound on the classification error from the PAC-Bayesian theorem. Consequently, the optimal abstention threshold which can be regarded as optimal abstention rate is invariable even if classification error is improved and when comparing two ensembles characterized by distributions Q and Q', it suffices to compare the bounds on the expected error $B(Q, P, m, \delta)$ and $B(Q', P, m, \delta)$. This is a consequence of the fact that only the expected value of $\mathcal{L}$ is bounded using the PAC-Bayesian theorem but not its variance which might differ between ensembles. This characteristic of the presented bounds is also their major drawback, since in real life applications, as we have seen before, classification accuracy does indeed have a severe effect on the optimal abstention rate and is not necessarily a meaningful indicator for the performance of an algorithm under different cost scenarios.

One conclusion we can draw from equations (7.16) and (7.18) is that abstaining in general does not make sense if $\nu \geq \frac{1+\mu}{4}$ or $\nu \geq \frac{\mu}{2}$ respectively. In chapter 2 we have concluded that $\nu \leq \frac{\mu}{1+\mu}$ is a necessary condition for abstention. Obviously it is true that $\frac{1+\mu}{4} \geq \frac{\mu}{1+\mu}$, so this provides no further restriction. On the other hand $\frac{\mu}{2} \leq \frac{\mu}{1+\mu}$ and therefore this condition imposes a stronger limitation on the cost scenarios for which abstention is possible. This condition may not be self-evident and also not true for all possible datasets. Nevertheless, it can be made plausible by the following example. Assume we have that $\nu \geq \frac{\mu}{2}$ and are given an instance in a dataset of size m which we can either abstain on (abstention window a_1) or classify positive (abstention window a_2). If we abstain on this instance we have the expected cost $\mathbf{EC}(C, a_1) = \frac{\nu}{m}$, whereas if we classify this instance positive we have that $\mathbf{EC}(C, a_2) = \frac{P(N)\mu}{m}$. For $P(N) = P(P)$ we observe that $\mathbf{EC}(C, a_2) = \frac{P(N)\mu}{m} = \frac{\mu}{2m} \leq \frac{2\nu}{2m} = \mathbf{EC}(C, a_1)$. Thus in this situation abstaining does not make sense. Although in real datasets abstention may still be applied if $\nu \geq \frac{\mu}{2}$, we can observe a certain correlation between optimal threshold values and optimal abstention rate for certain datasets. Figures 7.4(a) and 7.4(b) show the cost curves computed for the carcinogenicity dataset and from theorem 7.14 and figures 7.4(c)



(a)

(b)

(c)

(d)

Figure 7.4: Figure (a) shows the cost curve for the carcinogenicity data for $\nu < \frac{1}{2}$ and figure (c) the corresponding optimal abstention rates. Analogously, figure (b) contains the cost curve derived by calculating the bound for the optimal thresholds provided that $B(Q, P, m, \delta) = 0.3$. Figure (d) then shows the optimal values for σ given by equation (7.19).

and 7.4(d) show the corresponding abstention rates/thresholds.

A consequence of allowing different thresholds for positive and negative predictions is that the optimal threshold for positive classifications does only depend on the relationship between μ and ν, i.e. the costs for false positives and abstained instances, but not on the costs for false negatives, while the optimal threshold for negative classifications only depends on the costs for false negatives and abstained instances. This is due to the fact that θ and σ can never be negative. Thus we cannot have the situation that the threshold for positive classification lies below 0 or the threshold for negative classifications lies above 0.

Indeed, we can make a similar observation for abstention windows calculated on real-life data. The optimal abstention window a_{opt} is always located around the optimal threshold a_t between positive and negative prediction. Therefore we have that $l_{opt} \leq l_t$ and $u_{opt} \geq l_t$ and the lower threshold is only determined by the ratio between false negative and abstention costs and the upper threshold only by the ratio between false positive and abstention costs. However, the optimal threshold a_t is determined by the ratio between false negative and false positive costs.

If we would allow that θ, $\sigma \in [-1 : 1]$, theorem 7.14 would still hold, however the constraint $-\theta \leq \sigma$ would have to be imposed when determining the optimal thresholds. On the contrary, if we have the same threshold for positive and negative classifications, the optimal threshold is determined by both misclassification costs.

Chapter 8

Conclusion

8.1 Summary

Abstaining classifiers differ from common classifiers in that they are allowed to refrain from a classification if it appears to be doubtful. In principle, several ways are conceivable to create abstaining classifiers. In this thesis, we have presented a method by which any classification model supplying prediction scores can be used to derive a set of abstaining classifiers described by so-called abstention windows. An abstention window is defined by a pair of thresholds and deemed to be optimal if it has minimal expected cost among all possible windows for the same model.

Expected cost depends on the cost matrix which attaches costs to certain events as well as the probabilities of these events. We showed that any cost matrix can be transformed such that correct classifications are associated with zero costs and explored additionally the relationship between costs and class distributions. Furthermore, we were able to obtain a necessary – but not sufficient – condition for abstention to be possible which greatly limits the cost scenarios for which abstention may be applied.

As costs are often uncertain and class distributions not fixed, two types of cost curves were introduced which make it possible to examine the behavior of classification models for different cost scenarios or class distributions. Under the assumption that the abstention rate on the validation corresponds to the abstention rate expected on any sample from $\mathcal{X}$, both curves are indeed equivalent, yet the second type is distinctively easier to analyze.

Since the predictions of abstention windows may overlap and complement each other, they can be combined using different approaches. The first method presented takes a vote among optimal abstention windows produced by different models weighted by their expected cost. For the second one a sequence of abstention windows is calculated which is to be applied one after the other. The learning procedure iteratively computes the optimal abstention window and then removes the instances from the validation set which are covered, that is classified, by this window.

Cost curves as well as optimal abstention windows can be calculated efficiently and two algorithms were presented for this purpose. Both of them rely on several characteristics of optimal abstention windows which allow excluding abstention windows during computation without explicitly considering them or calculating their expected cost. The major factor which made it possible to derive linear algorithms for the computation of both optimal abstention windows and cost curves was the dependency between optimal abstaining and non-abstaining classifiers. This dependency leads to the fact that the optimal abstention window is always located around the optimal threshold between positive and negative prediction.

We evaluated the performance of abstaining classifiers as well as the usability of cost curves on two classification tasks. Here, abstaining could be shown to improve predictive accuracy decisively at the expense of coverage. For mutagenicity prediction, abstained instances were analyzed particularly and it could be shown that for one model the choice to abstain could be attributed to specific characteristics of instances. Additionally, cost curves were used to compare several classification algorithms concerning their behavior on the mutagenicity dataset and dependencies between abstention rate and false negative and false positive rate on the one hand and accuracy on the other hand were examined. Furthermore, the analysis of combined abstaining classifiers suggested that the predictions of different abstention windows can be assembled successfully to obtain higher level abstaining classifiers.

In the last chapter, we presented bounds on the expected error of abstaining voting ensembles for equal as well as unequal abstention costs. These bounds can be used to directly determine the optimal threshold for abstention for any cost scenario in constant time. Although the results derived yield rather loose bounds, they are nevertheless useful to analyze the behavior of the optimal thresholds for different cost scenarios.

8.2 Outlook

The results presented in this thesis raise a number of questions which can be starting points for further research but go beyond the scope of this thesis.

8.2.1 Extension to Multi-Class Problems

Throughout this thesis only two-class problems have been considered, that is problems featuring only two types of classes. In reality though, many classification tasks involve more than two categories. Naturally, abstention is also possible in these cases, but the presented methods have to be extended by reducing the multi-class problems to binary problems. There are several ways to achieve this reduction such as, for example, learning a classifier for each class against the remaining classes or for each pair of classes (pairwise coupling, [27]) or by using error correcting output codes [12].

However, when increasing the number of classes, the complexity of the problem increases as well. For more than two classes, abstention cannot only be performed by choosing none of the classes, but also by choosing more than one. In this case, some classes are definitely excluded, but the remaining ones all appear to be possible and for want of information the classifier refuses to name a specific one of these. As the number of subsets is exponential in the number of classes, the problem of choosing the best subset may actually be intractable in general.

8.2.2 Abstention Costs

For our purposes we presumed that it is equally expensive to abstain on a positive instance and a negative one. This assumption is reasonable if after being abstained on all instances are submitted to the same procedure without regard to the class. However, applications are conceivable for which this is not the case. When classifying EST sequences by their codon frequencies for example, it can make a difference if a sequence from blumeria is abstained on or one from barley, in particular if the next step for an abstained instance is a BLAST search. As plant genomes are clearly overrepresented in public databases compared to fungi genomes, it is much more likely to find a close homolog for a plant gene than for a fungus gene.

Therefore, one might consider a delegating classifier approach, i.e. an approach where instances abstained on are delegated to a second classifier, as an example against equal abstention costs. If the second classifier performs worse on negative than on positive instances for example, the costs for abstaining on a negative instance are higher than the costs for abstaining on a positive one. However, this is only valid if the two classifiers are completely independent of each other. If the second classifier is trained on delegated instances only as described by Ferri *et al.* [18], increasing the abstention costs for the negative class has a converse effect since it will lead to less abstention on this class. This consequently leads to an even poorer performance of the second classifier on negative instances as it has seen even less instances of this class during training. As a consequence, abstention costs for the negative class would have to be increased additionally and even less negative instances would be abstained on and so on until none of the negative instances of the validation set would actually be abstained on.

Additionally to the class, abstention and misclassification costs may depend on the specific instance. For some instances it may be beneficial to classify them even if the probability of misclassification is high because the costs for further tests or experiments would be tremendous or vice versa abstain even if the probability of misclassification is low because the correct class can be determined easily in a different way.

Therefore unequal abstention costs and conditional costs are the major points which will need looking into in the future. The principle idea of abstaining based on abstention windows can be easily extended to both unequal abstention costs and conditional costs. However, the optimal abstention window in this case probably has to be determined by the brute force approach which explicitly calculates the expected cost of every window. Unfortunately, cost curves can only be applied for equal abstention costs because distinguishing between the two classes in abstaining would increase dimensionality and therefore make the curves unsuitable for human interpretation.

8.2.3 Higher-Level Abstaining Classifiers

The subject of combining several abstention windows has been broached relatively shortly and the presented methods still leave many possibilities to connect the predictions of the individual models. Rule learning approaches with the inclusion of negations are conceivable as well as graph theoretical solutions or extensions of ensemble methods. For example, bagging could be extended to abstaining by learning optimal abstention windows from bootstrap samples and voting among them.

8.2.4 Theoretical Bounds

As mentioned before, the presented results on expected cost for unequal misclassification costs provide rather lose bounds. Additionally, we observe that the bound does not depend on the classification error either for equal or unequal misclassification costs. In reality though, there exists a strong interaction between expected cost or optimal abstention rate and accuracy. These points might be addressed by using other ways to bound the probabilities for misclassifications and abstaining than Markov's inequality and expected error on the training set.

8.2.5 Active Classification and Abstaining

In the introduction we mentioned active classifiers which may ask for the values of additional attributes before classification. As such, active classification does not involve abstention. Nevertheless, one might tackle the problem of learning active classifiers in a framework that integrates misclassification and attribute costs using the concept of abstention. Alternatively, the notion of abstention used in this thesis could be extended such that an abstaining classifier can suggest which of a range of tests is to be performed in the case of abstention. For this purpose, the methods for learning active classifiers might prove to be useful.

8.3 Conclusion

The central question of this thesis was if and how abstention can improve the reliability of predictions. Our results suggest that indeed improvements can be achieved on the premises that the costs for not classifying an instance are low enough. In general, there is no universally valid rule what "low enough" actually means in each case. This depends on the application and the predictive performance achieved without abstaining and can be explored with the help of cost curves. Contrary to that, we can make statements about cost scenarios which clearly prohibit abstention in any case.

As optimal abstention windows can be determined efficiently, the obvious solution for any classification task is to apply the presented methods for deriving abstaining classifiers and cost curves to the problem and to examine for which cost scenarios abstention is possible and if any of these scenarios does correspond to the correct one. Based on these observations, we can either exclude abstention completely because we know that abstention costs for the specific tasks are not as low as required or use it as a suitable method to improve predictive performance if they are.

Appendix A

Table of Definitions

		Page
$\mathcal{X}$	Instance space. An instance is defined as a k-tuple $(x_1, \ldots, x_k) \in A_1 \times \cdots \times A_k$ with A_i, $1 \leq i \leq k$, the set of possible values x_i may assume.	1
$\mathcal{Y}$	Set of class labels, in our case $\mathcal{Y} = \{P, N\}$.	2
Classifier	A function $Cl : \mathcal{X} \rightarrow \mathcal{Y}$.	2
Abstaining Classifier	A function $Cl : \mathcal{X} \rightarrow \mathcal{Y} \cup \{\perp\}$.	2
$m(x)$	Margin of instance $x \in \mathcal{X}$.	9
$a = (l, u)$	Abstention window a with lower threshold l and upper threshold u.	10
$\pi(a, x)$	Prediction of abstention window a on instance x.	10
$\mathcal{A}(Cl)$	Set of possible abstention windows for classifier Cl. Shortcut $\mathcal{A}$ if it is obvious from the context which classifier is meant.	11
$\mathcal{T}(Cl)$	Set of possible thresholds for classifier Cl, i.e. $\mathcal{T}(Cl) = \{a \mid a \in \mathcal{A}(Cl) \wedge l = u\}$. Shortcut $\mathcal{T}$ if it is obvious from the context which classifier is meant.	24
C	Cost matrix. $C(Y, y)$ is the expected cost of labeling an instance of class Y with label y. $Y \in \{P, N\}$ and $y \in \{p, n, \perp\}$. If $C(P, \perp) = C(N, \perp)$ we use the term $C(\perp)$ to denote the abstention costs.	12
$TP(a)$	True positives: Number of positive instances in the validation set classified correctly by abstention window a.	12
$FN(a)$	False negatives: Number of positive instances in the validation set misclassified by abstention window a.	12
$UP(a)$	Unclassified positives: Number of positive instances in the validation set abstained on by abstention window a.	12
$TN(a)$	True negatives: Number of negative instances in the validation set classified correctly by abstention window a.	12
$FP(a)$	False positives: Number of negative instances in the validation set misclassified by abstention window a.	12

		Page
$UN(a)$	Unclassified negatives: Number of negative instances in the validation set abstained on by abstention window a.	12
$A(a)$	Number of instances in the validation set abstained on by abstention window a.	51
$TPR(a)$	True positive rate of abstention window a.	13
$FNR(a)$	False negative rate of abstention window a.	13
$PAR(a)$	Positive abstention rate of abstention window a.	13
$TNR(a)$	True negative rate of abstention window a.	13
$FPR(a)$	False positive rate of abstention window a.	13
$NAR(a)$	Negative abstention rate of abstention window a.	13
$AR(a)$	Overall abstention rate of abstention window a.	28
$\mathbf{EC}(C, a)$	Expected cost of abstention window a on the validation set given cost matrix C.	13
$\mathbf{EC}(C, a, P)$	Expected cost of abstention window a on the set $P \subseteq \mathcal{X}$ given cost matrix C.	43
$\bar{C}$	Equivalence class of cost matrices. Two cost matrices C and C' are equivalent ($C \equiv C'$) if $\exists k > 0\, \forall a \in \mathcal{A}\ \mathbf{EC}(C, a) = k\, \mathbf{EC}(C', a)$.	17
$\mathbf{NEC}(C, a)$	Normalized expected cost of abstention window a given cost matrix C.	18
μ	False positive costs relative to false negative costs, i.e. $\mu = \frac{C(N,p)}{C(P,n)}$.	18
ν	Abstention costs relative to false negative costs, i.e. $\nu = \frac{C(\perp)}{C(P,n)}$.	18
Δ	Resolution of a cost curve, i.e. number of values evaluated for x and y.	33
$PCF(L)$	Probability-cost function for $L \in \{P, N, \perp\}$.	30
$K(p)$	Cost curve for a classifier Cl_p represented by a $\Delta \times \Delta$ matrix.	33
$D(p, q)$	Differential cost curve. Difference between cost curves of two classifiers Cl_p and Cl_q.	33
M	Minimum cost curve. Given a set of classifiers $Cl_1, \ldots, Cl_p$ with cost curves $K(1), \ldots, K(p)$: $m_{i,j} := \min_{1 \leq s \leq p} k_{i,j}(s)$.	35
I	Index matrix. Given a set of classifiers $Cl_1, \ldots, Cl_p$ with cost curves $K(1), \ldots, K(p)$: $i_{i,j} := \text{argmin}_{1 \leq s \leq p} k_{i,j}(s)$.	35
$\vec{m} = (m_1, \ldots, m_k)$	Vector of distinct margin values in the validation set S, i.e. $\forall 1 \leq i \leq k\, \exists x \in S : m_i = m(x)$, $\forall x \in S\, \exists 1 \leq i \leq k : m(x) = m_i$ and $m_1 < \cdots < m_k$.	49
$\vec{p} = (p_1, \ldots, p_k)$	p_i denotes the number of positive instances having margin m_i.	49
$\vec{n} = (n_1, \ldots, n_k)$	n_i denotes the number of negative instances having margin m_i.	49
$\widehat{\mathcal{A}}$	Subset of the set of abstention windows $\mathcal{A}$ such that no abstention window $a \in \mathcal{A} \setminus \widehat{\mathcal{A}}$ can be optimal for any cost scenario.	50

		Page
$cost(a, \mu, \nu)$	Cost of abstention window a on the validation set for false positive costs μ and abstention costs ν. $cost(a, \mu, \nu) = \mathbf{NEC}(C, a)\, n$ with n the size of the validation set and $C(P, n) = 1$, $C(N, p) = \mu$ and $C(\perp) = \nu$.	51
$v(i)$	The value of a threshold between margin values m_i and m_{i+1}.	51
$succ(a)$	Set of abstention windows which can be obtained by increasing or decreasing the lower or upper threshold of a by one step only.	52
$\vec{\lambda} = (\lambda_1, \ldots, \lambda_t)$	Result of the preprocessing step. λ_i is the smallest margin for a sequence of instances of the same class.	53
$\vec{\upsilon} = (\upsilon_1, \ldots, \upsilon_t)$	Result of the preprocessing step. υ_i is the largest margin for a sequence of instances of the same class.	53
$\vec{\rho} = (\rho_1, \ldots, \rho_t)$	ρ_i denotes the number of positive instances x with $\lambda_i \leq m(x) \leq \upsilon_i$.	53
$\vec{\eta} = (\eta_1, \ldots, \eta_t)$	η_i denotes the number of negative instances x with $\lambda_i \leq m(x) \leq \upsilon_i$.	53
$\psi(i)$	The value of a threshold between margin values υ_i and λ_{i+1}.	55
D	Distribution D over labeled instances $(x, y) \in \mathcal{X} \times \mathcal{Y}$.	93
$c(x)$	Concept. $c(x) = 1$ if c classifies this instance as positive and $c(x) = -1$ otherwise.	93
$\mathcal{C}$	Set of possible concepts $c(x)$.	93
$l(c, x, y)$	Loss of concept c on labeled instance (x, y).	93
$l(c)$	Expected loss of concept c on the instance space.	94
$\hat{l}(c, S)$	Empirical loss of concept c on sample S.	94
$l(Q)$	Expected loss of ensemble Q.	94
$\hat{l}(Q, S)$	Empirical loss of ensemble Q on sample S.	94
$D(Q \parallel P)$	Relative Entropy or Kullback-Leibler divergence.	94
$c(Q, x)$	Score of ensemble Q on instance x.	95
$c_V(Q, x)$	Voting classifier.	95
$l_V(Q)$	Expected error of the voting classifier.	95
c_V^θ	Abstaining voting classifier with threshold θ.	96
$cost(Q, x, y)$	Cost of applying ensemble Q on labeled instance (x, y).	96
$\mathcal{L}$	Random variable over the cost of instances.	96
γ_V^θ	Expected cost of the abstaining voting classifier c_V^θ.	96
$\mathcal{Z}$	Random variable used to distinguish false negative from false positive predictions. $\mathcal{Z} := y - c(Q, x)$.	98
$c_V^{\theta,\sigma}$	Abstaining voting classifier having different thresholds for positive (σ) and negative (θ) predictions.	101
$\gamma_V^{\theta,\sigma}$	Expected cost of the abstaining voting classifier $c_V^{\theta,\sigma}$.	101

Bibliography

[1] Ames, B. N., Durston, W. E., Yamasaki, E. and Lee, F. D. (1973) Carcinogens are mutagens: A simple test system combining liver homogenates for activation and bacteria for detection. *Proc. Natl. Acad. Sci.*, 70, 2281-2285.

[2] Baker, S.G. (2003) The central role of receiver operating characteristic (ROC) curves in evaluating tests for the early detection of cancer. *J. Natl. Cancer Inst.*, 95(7), 511-5.

[3] Ben-Hur, A. and Brutlag, D. (2003) Remote homology detection: a motif based approach. *Bioinformatics*, 19, i26-i33.

[4] Boser, B.E., Guyon, I.M. and Vapnik, V. (1992) A Training Algorithm for Optimal Margin Classifiers. *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, 144 - 152.

[5] Bradley, A.P.(1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30, 1145-1159.

[6] Breiman, L. (1996) Bagging Predictors. *Machine Learning*, 24, 123 - 140.

[7] Breiman, L. (2001) Random Forests. *Machine Learning*, 45, 5-32 .

[8] Burges, C.J.C. (1998) A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, 2, 121-167.

[9] Chang, C-C. and Lin, C-J. (2001) LIBSVM: a library for support vector machines. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[10] Cormen, T.H., Leiserson C.H., Rivest, R.L. and Stein, C. (2001) *Introduction to Algorithms.* Second Edition. MIT Press, Cambridge, MA.

[11] Dettling, M. and Bühlmann, P. (2003) Boosting for tumor classification with gene expression data. *Bioinformatics*, 19, 1061-1069.

[12] Dietterich, T.G. and Bakiri, G. (1995) Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2, 263-286.

[13] Ding, C.H.Q. and Dubchak, I. (2001) Multi-class protein fold recognition using support vector machines and neural networks. *Bioinformatics*, 17, 349-358.

[14] Drummond, C. and Holte, R.C. (2000). Explicitly representing expected cost: An alternative to ROC representation. *Proc. of the 6th International Conf. on Knowledge Discovery and Data Mining*, 198-207.

[15] Drummond, C. and Holte, R.C. (2004). What ROC Curves Can't do (and Cost Curves Can). *Proceedings of the ROC Analysis in Artificial Intelligence, 1st International Workshop*, 19-26.

[16] Egan, J.P. (1975). Signal Detection Theory and ROC Analysis. *Series in Cognition and Perception*, Academic Press, New York.

[17] Elkan, C. (2000). Cost-sensitive learning and decision-making when costs are unknown. *Workshop on Cost-Sensitive Learning at the Seventeenth International Conference on Machine Learning.*

[18] Ferri, C., Flach P., and Hernández-Orallo, J. (2004). Delegating Classifiers. *Proc. of the 21st International Conf. on Machine Learning.*

[19] Ferri, C. and Hernández-Orallo, J. (2004). Cautious Classifiers. *Proceedings of the ROC Analysis in Artificial Intelligence, 1st International Workshop*, 27-36.

[20] Frank, E. and Witten, I. H. (1998) Generating accurate rule sets without global optimization. *Proceedings of the 15th International Conference on Machine Learning*, 144 - 151.

[21] Freund, Y. and Schapire, R.E.(1999). A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence*, 14(5), 771-780.

[22] Friedel, C.C., Jahn, K.H.V., Sommer, S., Rudd, S., Mewes, H.W. and Tetko, I.V. Dez. 7, 2004. Support vector machines for separation of mixed plant-pathogen EST collections based on codon usage. *Bioinformatics* doi:10.1093/bioinformatics/bti200.

[23] Gold, L. S. and Zeiger, E. (1997) Handbook of Carcinogenic Potency and Genotoxicity Databases. CRC Press, Boca Raton.

[24] Greiner, R., Grove, A.J. and Roth, D. (2002) Learning cost-sensitive active classifiers. *Artificial Intelligence*, 139, 137-174.

[25] Hand, D.J. and Till, R.J. (2001). A simple generalisation of the area under the ROC curve for multiple class classification problems. *Machine Learning*, 45, 171-186.

[26] Hanley, J.A. and McNeil, B.J. (1982) The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 143, 29-36.

[27] Hastie, T. and Tibshirani, R. (1998) Classification by pairwise coupling. *The Annals of Statistics*, 26, 451-471.

[28] Helma, C., Gottmann, E. and Kramer, S. (2000) Knowledge discovery and data mining in toxicology. *Stat. Methods Med. Res.*, 9, 329-358.

[29] Helma, C., King, R.D., Kramer, S. and Srinivasan, A. (2001). The predictive toxicology challenge 2000-2001. *Bioinformatics*, 17, 107-108. [http://www.informatik.uni-freiburg.de/∼ml/ptc/]

[30] Helma, C., Cramer, T., Kramer, S. and De Raedt, L. (2004) Data Mining and Machine Learning Techniques for the Identification of Mutagenicity Inducing Substructures and Structure Activity Relationships of Noncongeneric Compounds. *J. Chem. Inf. Comput. Sci.*, 44 (4), 1402 -1411.

[31] Hoos, H. H. (1998) Stochastic local search - methods, models, applications. Doctoral Dissertation, TU Darmstadt.

[32] Joachims, T. (1999) Making large-Scale SVM Learning Practical. In *Advances in Kernel Methods - Support Vector Learning*, Schölkopf, B., Burges, C., Smola, A., eds., MIT Press, Cambridge, MA.

[33] Kim, S. (2004) Protein β-turn prediction using nearest-neighbor method. *Bioinformatics*, 20, 40-44.

[34] Kramer, S., Frank, E. and Helma, C. (2002) Fragment Generation and Support Vector Machines for Inducing SARs. *SAR QSAR Environ. Res.*, 13, 509-523.

[35] McAllester, D. A. (1999). PAC-Bayesian model averaging. *COLT: Proceedings of the Workshop on computational Learning Theory*, 164 - 170.

[36] McAllester, D. A. (2001) PAC-Bayesian Stochastic Model Selection. *Machine Learning*, 51(1), 5-21.

[37] Mitchell, T.M. (1997). *Machine Learning*, McGraw-Hill, New York.

[38] Platt, J. (1998). Fast Training of Support Vector Machines using Sequential Minimal Optimization. *Advances in Kernel Methods - Support Vector Learning*, Schölkopf, B., Burges, C., Smola, A., eds., 185-208, MIT Press.

[39] Provost, F.J. and Fawcett, T. (1997). Analysis and vizualization of classifier performance: Comparison under imprecise class and cost distributions. *Proc. of the 3^{rd} International Conf. on Knowledge Discovery and Data Mining*, 43-48. AAAI Press.

[40] Provost, F.J. and Fawcett, T. (1998). Robust classification systems for imprecise environments. *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, 706-713.

[41] Provost, F.J., Fawcett, T. and Kohavi, R. (1998). The case against accuracy estimation for comparing induction algorithms. *Proceedings of the Fifteenth International Conference on Machine Learning*, 445-453.

[42] Quinlan, J. R. (1993) C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo, CA.

[43] Rudd, S., Mewes, H.W. and Mayer, K.F. (2003) Sputnik: a database platform for comparative plant genomics. *Nucleic Acids Res.*, 31, 128-32.

[44] Rückert, U. and Kramer, S. (2004). Towards Tight Bounds for Rule Learning. *Proc. of the 21st International Conf. on Machine Learning.*

[45] Rückert, U. and Kramer, S. (2004). Frequent Free Tree Discovery in Graph Data. *Proceedings of the 2004 ACM symposium on Applied computing*, 564 - 570.

[46] Schapire, R.E. (1999). A brief introduction to boosting. *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, 1401-1406.

[47] Sharp, P.M., Cowe, E., Higgins, D.G., Shields, D.C., Wolfe, K.H., Wright, F. (1988) Codon usage patterns in Escherichia coli, Bacillus subtilis, Saccharomyces cerevisiae, Schizosaccharomyces pombe, Drosophila melanogaster and Homo sapiens; a review of the considerable within-species diversity. *Nucleic Acids Res.*, 16, 8207-11.

[48] Swets, J.A. (1988) Measuring the accuracy of diagnostic systems. *Science*, 240, 1285-1293

[49] Todorovski, L. and Džeroski, S. (2000). Combining Multiple Models with Meta Decision Trees. *Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery*, 54-64.

[50] Toivonen, H., Srinivasan, A., King, R.D., Kramer S. and Helma, C.(2003) Statistical evaluation of the Predictive Toxicology Challenge 2000-2001. *Bioinformatics*, 19, 1183-1193.

[51] Turney, P. (2000). Types of Cost in Inductive Concept Learning. *Workshop on Cost-Sensitive Learning at the Seventeenth International Conference on Machine Learning*, Stanford University, California, 15-21.

[52] Vapnik, V. (1995) *The Nature of Statistical Learning Theory.* Springer-Verlag, New York.

[53] Weininger, D. (1988). SMILES, a chemical language and information system. 1. Introduction and encoding rules.*J. Chem. Inf. Comput. Sci.*, 28, 31-36.

[54] Witten, I.H. and Frank, E. (1999). *Data Mining: Practical machine learning tools with Java implementations.* Morgan Kaufmann, San Francisco.

[55] Wolpert, D.H. (1992). Stacked generalization. *Neural Networks*, 5, 241 - 259.