Features Handling by Conformal Predictors

Meng Yang

Submitted for the degree of

Doctor of Philosophy

Department of Computer Science
Royal Holloway University of London
Egham, Surrey TW20 0EX, UK

2014

Declaration

I declare that this dissertation was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

Meng Yang

18/09/2014

Abstract

Unlike many conventional machine learning methods, conformal predictors allow to supply individual predictions with valid measurement of confidence. In this thesis we adapt conformal predictors to address three common problems related to feature handling.

First of all, we consider the problem of feature selection in the context of conformal predictors. The main idea of our method is to use confidence measures as an indicator of usefulness of different feature subsets.

The second one is the problem of how to utilize the additional information which is only available in training set. Recently, Vapnik proposed a novel learning paradigm to incorporate additional information within SVM algorithm. Inspired by Vapnik's method, we propose an approach to deal with additional information by conformal predictors.

The last problem is classification using features with missing information. Conventionally, missing information is dealt with in pre-processing step, either by ignoring it or imputing it. We suggest a method which embeds the processing of missing information within conformal predictors.

Experiments have been carried out to evaluate the proposed methods using public datasets. Results demonstrate the effectiveness of these methods for feature handling.

Acknowledgements

I would like to thank all the people who helped me in or during preparation of this thesis.

I would like to express my deep gratitude to Professor Alex Gammerman and Dr. Zhiyuan Luo, my research supervisors, for their patient guidance, enthusiastic encouragement and useful critiques of this research work. I would like to express my very great appreciation to Professor Guang Li at Department of Control Science and Engineering, Zhejiang University, for his valuable and constructive suggestions during the planning and development of this research work. I would like to thank Dr. Ilia Nuretdinov, for his advice and assistance in keeping my progree on schedule.

I would also like to extend my thanks to the technicians of the laboratory of the Computer Science Department for their help in offering me the resources in running the program.

Finally, I wish to thank my parents for their support and encouragement throughout my study.

List of Tables

List of Figures

Contents

Chapter 1

Introduction

1.1 Motivation

Machine learning is a broad field of Artificial Intelligence. It is about programming computers to automatically learn a solution to a problem based on past experience [32, 28]. Classification is the most widely used form of machine learning. Learning how to classify objects to one of a pre-specified set of categories or classes is a characteristic of intelligence that has been of keen interest to researchers in Computer Science [5]. The ability to perform classification and to be learnt to classify gives people and computer programs the power to make decisions. In recent years classification algorithms have been successfully used to solve many interesting and often difficult real-world problems [73, 23], such as text categorization, fraud detection, machine vision, bio-informatics and medical diagnosis. In general, classifications can be accomplished in two broad learning frameworks, off-line and on-line [73]. In off-line mode we have a fixed training set of data ready for learning at the beginning. In on-line mode the data comes sequentially and algorithms keep

on learning on each trial to make predictions.

The classification methods assign classes to new objects and the ratios of making correct predictions have been used to evaluate their performance. In real-world problems users not only care about making correct predictions but also are concerned about how reliable algorithms are and how confident their predictions are. In order to help users make a reliable decision in complex problems, it has become essential to compute a reliable measurement of confidence that demonstrates the belief of the algorithm in the predicted result [8]. Conformal predictor is a recently developed general learning framework for making well-calibrated predictions in on-line mode, and provides prediction with reliable measures of confidence in off-line mode [92]. In machine learning, available training experience can have a significant impact on success or failure of the learner. Theoretically, having more features should result in more discriminating power. However, practical experience with machine learning algorithms has shown that this is not always the case [45]. If the data has no statistical regularity that machine learning algorithms exploit, the learning will fail. Even if the data is suitable for machine learning, the task of learning may not be easy either.

As Bellman stated in 1961 [10], high dimensional data lead to the degradation of performance. Feature selection is one useful pre-processing method in machine learning to find the optimal feature subset that classifiers can learn easier and consume less time by eliminating irrelevant or redundant information. There is a great variety of methods proposed for feature selection [51, 54]. These methods can be categorized into three board types: filters, wrappers and embedded methods.

The earliest approaches to feature selection within machine learning were filters. Statistical methods have been used to rank or weight the relationships between features and classes based on general characteristics of data. Such selections operate independently of any learning algorithm, which means undesirable information is filtered out of the data before the learning begins. A filter provides a very easy way to calculate since it only takes a short computing time. However, it is incapable of removing redundant information because this information is likely to have similar ranking or weighting [100].

Other researchers argue that the characteristics of a particular learning algorithm should be taken into account during selection. These methods, called wrappers, use learning algorithms to estimate the accuracy of selected feature subset and select the optimal one which has the best estimated accuracy. The wrappers often give better results than filters because they are optimized for a particular learning algorithm used [72]. On the other hand, the wrappers utilize the learning machine of interest as a black box and need much more time and space to re-run learning algorithms than filters. Although it is shown by John, Kohavi, and Pfleger [47] that the optimal features obtained must be from the relevant information, the wrapper approach does not use a relevance and redundancy measurement directly.

In contrast to the filter and wrapper methods, learning and feature selection in embedded methods cannot be separated, which means feature selection is performed as part of the model construction process. This approach tends to be in between filters and wrappers in terms of computational complexity. However, the features selected cannot be directly utilized by other classifiers. In this thesis, based on the measurement of confidence we are

interested in using conformal predictors as a wrapper for feature selection to demonstrate the characteristics of a dataset. Besides, irrelevance and redundance measurement are directly used.

Conformal predictors are then applied to two common classification problems related to features handling. The first problem is classification with additional information. Generally, learning is reliable when the training examples and test examples are coming from the same distribution. In the real-world, we often have some additional information which can help us making decisions. For example, normally doctors diagnose using general testing results, if the diagnosis is still unclear based on all available results, they may send the patient for some additional tests such as blood test and MRI scan. If there is a patient who does not have ability to afford additional tests, can we utilize previous additional testing results to diagnose accurately? Recently, Vapnik proposed a new learning framework, learning with privileged information, to incorporate additional information within Support Vector Machines [91]. Vapnik defined "privileged information" as the features which are only available in training set. A new method, SVM+, has been introduced to realize the framework. Inspired by this framework, we would like to adapt conformal predictions for dealing with additional information.

The other problem is classification using features with missing information. Real-life data are frequently facing with imperfect problems: errors, incompleteness, uncertainties and vagueness [48]. There are two ways to deal with missing information, either ignoring it or imputing it. Ignoring missing value may lead us losing useful information. And imputation method costs extra time and space. Based on conformal predictor we propose a new

method which performs imputation as part of a learning framework to reduce the complexity. The method assumes that missing values only exist in the test set. The method we used here is similar to the method we used for handling additional information, as additional values in the training set can be seen as missing in test set. The difference is that missing values could happen to any features, but addition information is usually related to a fixed subset of features.

1.2 Main Contributions

Feature selection by conformal predictors

We firstly investigate on an application of conformal predictor for feature selection. The measurement of confidence can be used to demonstrate the characteristics of datesets. Furthermore, irrelevant and redundant features have been analyzed and eliminated to minimize the size of feature subset. We apply the method on Abdominal Pain dataset and other 4 datasets from UCI dataset repository. Results show that the selected feature subset works well for classifiers in both off-line and on-line modes.

Conformal predictions with additional information

The existion of additional information is a very common scenario in real-life data. However, traditional learning framework does not have abilities to utilize additional information directly. The framework of learning with additional information is inspired by "learning with privileged information" paradigm. Experimental results show that conformal predictors successfully

realize the new learning framework in both off-line and on-line mode.

Conformal predictions with missing information

This research exploits the applications of conformal predictors using features with missing values. Unlike other methods which either ignoring missing information or requiring extra resources for imputation, the new conformal predictors can directly make predictions with missing information. By calculating how strange examples with hypothetical missing information and classes are from previous examples with whole information and given classes, the new method do not need to learn on a validation set to estimate missing information.

1.3 Publications

The following publications have been produced as a result of the work presented in this thesis.

The result of Section 2.3 on feature selection for abdominal pain dataset was firstly presented at AIAI workshop in 2011 and published in the proceedings [101], and then included as a book chapter [9].

- M. Yang, I. Nouretdinov, Z. Luo, and A. Gammerman. Feature selection by conformal predictor. In L. Iliadis, I. Maglogiannis, and H. Papadopoulos, editors, Artificial Intelligence Applications and Innovations, volume 364 of IFIP Advances in Information and Communication Technology, pages 439-448. Springer Berlin Heidelberg, 2011.

- T. Bellotti, I. Nouretdinov, M. Yang, and A. Gammerman. Feature

selection. Chapter 6, in book: Conformal Prediction for Reliable Machine Learning, 1st Edition. (S. Ho, V. Balasubramanian, and V. Vovk, editors). Morgan Kaufmann, 2014.

The result of prediction with additional information was presented at AIAI 2nd workshop in 2013 and published in the proceedings [102].

- M. Yang, I. Nouretdinov, and Z. Luo. Learning by conformal predictors with additional information. In H. Papadopoulos, A. Andreou, L. Iliadis, and I. Maglogiannis, editors, Artificial Intelligence Applications and Innovations, volume 412 of IFIP 141 Advances in Information and Communication Technology, pages 394-400. Springer Berlin Heidelberg, 2013.

The following publication was the result of research collaboration with Zhejiang University on the applications of SVM algorithms for the analysis of coal properties.

- Y. Wang, M. Yang, G. Wei, R. Hu, Z. Luo, and G. Li. Improved PLS regression based on SVM classification for rapid analysis of coal properties by near-infrared reflectance spectroscopy. Sensors and Actuators B: Chemical, volume 193, pages 723–729, 2014.

1.4 Outline of the Thesis

The introductory chapter presents the motivation behind this research and summarizes the main contributions. The rest of the thesis is organized as follows.

Chapter 2 is devoted to the theory of conformal predictors. We present the necessary background of classification in machine learning in Section 2.1. Section 2.2 gives a description of conformal predictors.

Chapter 3 is concerned with the application of conformal predictors for feature selection. Section 3.1 introduce the background of feature selection. Our method of feature selection based on conformal predictors is given in section 3.2. Results and discussion are given in Section 3.3.

Chapter 4 covers the development of the new solution of realizing the learning framework with additional information. It starts with the background of additional information in Section 4.1 and the idea of learning using privileged information in Section 4.2. Based on conformal predictor we propose a new way to utilize additional information in both off-line and on-line mode in Section 4.3. Results are presented in Section 4.4.

Chapter 5 considers the development of prediction with missing information. The background on missing value and the current methods to deal with missing information is described in Section 5.1. Section 5.2 presents the new algorithm of predictions with missing information using conformal predictors. Results of different methods are compared and discussed in Section 5.3.

Finally, Chapter 6 presents conclusions and suggests possible future work.

Chapter 2

Conformal Predictors for Classification

2.1 Background

2.1.1 Classification in Machine Learning

Machine learning is a branch of Artificial Intelligence, compared with the study of systems and how to perform important tasks by learning from experience [73]. This is often feasible and cost-effective where manual programming is not [59]. As a result, machine learning is widely used in computer science and other fields [60]. Classification is one of the most mature and widely used forms [23]. In machine learning, learning is divided into two categories, supervised learning and unsupervised learning [73]. In this thesis when we use the term "classification", we refer to supervised learning. In supervised learning, classification is the problem of identifying to which of a set of categories a new object belongs, on the basis of a training set of

data containing objects whose category membership is known. Classification has been studied and researched for decades, and it has already been widely used to solve many interesting and often difficult real-world problems, such as text categorization (e.g., spam filtering), fraud detection, machine vision (e.g., face detection), bio-informatics (e.g., protein classification according to their function) and medicines [73]. Classification procedure and key issues will be discussed in the next section.

2.1.2 Classification Procedure

The process of applying classification to a real-world problem is based on four elements: learning environment, data representation, applications of classification algorithms and performance evaluation [60, 45, 92].

Learning environment

Learning from experience is as important for a computer as it is for a human being. In order for there to be something to learn, the learning environment must be governed by constant laws [92]. It is assumed that a sequence of examples is generated randomly from some fixed probability distribution, say Q, on a fixed space of possible examples, say $\mathbf{Z}$. When we say the examples are chosen randomly from Q, we mean they are independently and identically distributed (i.i.d.).

Data representation

In the typical classification task, each example $\mathbf{z}_i$ is assumed to consist of an object $\mathbf{x}_i$ and its label y_i, shown as following definition: Given a set of i.i.d.

examples, $\mathbf{Z} = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)\}, \mathbf{x}_i \in \mathbf{X}^d, y_i \in Y$, where n is the number of examples and object $\mathbf{x}_i$ is described by a fixed number, d, of measurements, or features (attributes) and a label $y_i \in \{1, \ldots, C\}$ denotes its class, with C being the number of classes. If $C = 2$, this is called binary classification; if $C > 2$, this is called multi-class classification.

Application of classification algorithms

Although the true probability distribution Q is unknown, based on the observed data a function approximation can be constructed to explain it. The construction of the function approximation is called a learning algorithm or a classifier. The examples used for learning are called training examples, and the examples to be classified are called test examples. The classifiers have all attempted to derive procedures that would be able to [59],

1. be equal or similar to a human decision-marker's behaviour.

2. handle a wide variety of problems and to be extremely general.

3. be used in practical settings with proven success.

Nowadays, a wide variety of approaches has been developed based on Artificial Intelligence, such as Logic-based techniques and Perceptron-based techniques, and Statistics, such as Bayesian Networks and Instance-based techniques [60].

Decision trees are based on Logic-based techniques which use trees as predictive models that classify examples by sorting them based on feature values. In these tree structure, nodes represent class labels and branches represent conjunctions of features that lead to those class labels. It is easy for

people to understand decision tree models after a brief explanation. Artificial Neural Networks(ANNs) are created base on Perceptron-based techniques. In an ANN, simple artificial nodes, known as neurons, are connected together to form a network which mimics a biological neural network. ANNs have been applied to many real-world problems, but they are lack of ability to reason about their output in a way that can be effectively communicated [60]. Conversely to ANNs, a Bayesian Network is a probabilistic graphical model that provides a probability that an example belongs in each class rather than simply a classification [59]. k-Nearest Neighbor algorithm is one of the Instance-based learning algorithms. It is based on the principle that the examples within a dataset will generally exist in close proximity to other examples that have similar properties[60]. A shortcoming of the k-Nearest Neighbor is that it is sensitive to the structure of the data.

Performance evaluation

The performance of a classifier is evaluated by many evaluation methods. The classifier?s discrimination power can be evaluated by two key measurements on predictions, measurement of how correct they are (accuracy) and measurement of how probable they are (confidence) [52, 22].

An estimate of classification accuracy on new examples is a common performance evaluation criterion. It is the number of correct predictions made divided by the total number of predictions made, multiplied by 100 to turn it into a percentage. A classifier's accuracy can be calculated by at least two techniques [60]. One technique is to split training set by using part of it for training and the rest for estimation performance. The other one is known

as k-fold cross-validation [31]. The training set is divided into k equal-sized subsets which are mutually exclusive to each other and for each subset the classifier is trained on all of remaining subsets. The estimated accuracy of the classifier is the average of the accuracy of each subset. Obviously, this type of validation is more expensive computationally, but it gives the most accurate estimate of a classifier's error rate [50].

If you are predicting the label of a new object $\mathbf{x}$, how confidence are you about your prediction $\hat{y}$ which will be the same as the true label y? One way to answer the question is to use confidence measure. In statistics, the term "confidence" is often associated with the concept of confidence intervals. The estimate of confidence interval points out that the number of errors in test stage is guaranteed, which can not exceed $(100 - c)\%$, where $c \in [0, 100]$ is the confidence value. All approaches that provide guarantees on errors can be broadly identified to be motivated by two theories, Bayesian learning [76] and Probably Approximately Correct (PAC) learning [82].

2.1.3 Key Issues in Classification

Issues of data

Collecting the data set is the first step in classification. If an expert is available, the most important fields (features) can be suggested. If not, a "brute-force" method is applied. It is a general problem-solving technique that consists of systematically enumerating all possible candidates for the solution and checking whether each candidate satisfies the problem's statement. Nevertheless, a data set collected by the "brute force" method can not be directly used for classification, since it often contains noise and missing

values. Thus, data pre-processing is necessary [103].

There are many problems that are often encountered in data pre-processing [74]:

- No values have been input (missing information).
- Impossible and unlikely values have been input.
- Irrelevant features are present in the data.

Missing information is an unavoidable problem in dealing with most real world data sources. There are a number of methods have been developed to handle features with missing values [16]. In general, these methods are applied in the pre-processing stage and can be categorized into two types, ignoring missing values or imputing missing values.

Some data handling software is used to check impossible values at the point of input, so that these values can be re-entered. This kind of errors is generally straightforward, such as coming across negative prices when positive ones are expected [60]. The notion of "unlikely value" means those values that are suspicious due to their relationship to a specific probability distribution. For example, a standard normal distribution with a mean of 5, a standard deviation of 3 [74]. This kind of problem is normally solved by Variable-by-Variable data cleansing method, which is using metadata, such as cardinality, max, min,variance, and deviation, to detect a number of unlikely values [60].

Generally, features are characterized as: relevant, which denotes that the features have an influence on the output; irrelevant, which indicates that features are independent of the output; and redundant, which denotes the features are correlated with other features. Feature subset selection is the

process of identifying and removing irrelevant and redundant features [100].

In addition to the features which are missing or irrelevant, the problem of additional values is another big challenge for machine learning. In real-life problems, additional information provides an opportunity for people to make better decision. Most learning frameworks in machine learning assume that the examples in training set and test set are coming from the same distribution [94]. Thus, additional information can not be directly utilized. Recently, Vapnik proposed a novel learning paradigm to incorporate additional information within machine learning, by imitating it as a "master" to a pupil [91].

Issues of learning algorithm

There are some common problems of the applications of learning algorithms.

- Need of reliable measurement of confidence.

  Given a set of training examples, statistical learning theory produces a function mapping the objects into the labels. Then the function is applied on a new object and gives a predicted label. Many machine learning algorithms have been successfully applied for industrial applications [73, 65]. For example, we used the Support Vector Machine to improve the predictive ability on properties of the coal samples [98]. But, these learning algorithms only provide the predictions - how probably are the predictions correct? This question has not been answered as well as we might like [92]. Although the Bayesian and PAC learning approaches can estimate confidence, the values generated by these methods are often impractical, invalid or unreliable [82, 56, 83].

Bayesian learning approach makes a fundamental assumption on the probability distribution of the data. The values generated by Bayesian approach are generally correct only when the observed data are actually generated by the assumed distribution, which does not happen often in real-world scenarios. This is demonstrated by Melluish et al. [75]. In their experiments, Bayesian method was applied to the data where the underlying probability distribution was not known. Results illustrated the key role of the choice of the prior distribution to obtain valid measures of probability in Bayesian learning methods. On the other hand, the PAC approach relies only on the i.i.d assumption. However, the error bounds generated by such approach are often not very practical, as demonstrated by Proedrou [82], and by Nouretdinov et al. [78].

- The problem of overfitting.

  In machine learning, overfitting generally happens when a model is too complicated or when training examples are insufficient [27]. The learner may adjust to very specific random features instead of describing the underlying relationship. Thus, the accuracy of predictions in test set is low, while it is very high in training set.

- Curse of dimensionality.

  Curse of dimensionality is another big problem in machine learning. The term "curse of dimensionality" was coined by Bellman to refer to the fact that many algorithms that work fine when the input is low dimension become intractable when the input is high-dimensional [10]. The complexity of learning during training increases exponentially with

the number of features [27].

In this thesis, we pay attentions to the issues about irrelevant features, missing information and additional information. For providing reliable measurement of confidence, conformal predictors (CP) has been studied. We applied CP on feature selection to address the problems when data contain irrelevant or unlikely values. Then, CP has been extended and adapted for learning with additional information and missing information.

2.2 Conformal Predictors

Conformal predictor is a recently developed learning framework [92]. Classifications in CP are based on hypothesis testing. The algorithm outputs a probability of the instance being a member of each of the possible classes. Thus, it can output a region prediction with the guaranteed number of errors or a single prediction with certain confidence [8].

2.2.1 Transductive and On-line Learning

Generally, classifications can be accomplished in two broad learning frameworks, off-line and on-line [89].

Most previous theoretical work in machine learning has been in an inductive and off-line framework. In off-line learning, one uses a batch of old examples to generate a prediction rule, which is then applied to new examples. Once a rule or model has been generated from the training set it will not change. We call this induction/deduction learning framework. In inductive prediction we first draw a general rule from examples in hand; this is

inductive step. When presented with a new object, we derive a prediction from the general rule; this is the deductive step.

Compared with the off-line learning, new examples in the on-line learning come and are classified one by one; the algorithm continues to learn after the true label is provided after each prediction. For example, there is a new example $\mathbf{x}_i$. Once the classifier assigned a predictive label to it, the true label of $\mathbf{x}_i$ will be revealed, the program will be updated after each example has been seen. We could say that we infer our classification directly from the data set, it is a transductive framework, one makes predictions sequentially, basing each new prediction on all the previous examples instead of repeatedly using a rule constructed from a fixed batch of examples.

One of the advantages of conformal predictor is that it can work with a framework that is transductive and on-line, which takes a shortcut, moving from old examples directly to the prediction about the new object [92].

2.2.2 Conformal Predictions

This section describes the theory behind the conformal predictors framework, and its implementation for classification.

Theory of CP

The theory of CP was recently developed by Vovk, Shafer and Gammerman [89, 92]. It is based on the Kolmogorov complexity [70] of an i.i.d. sequence of data examples. If $l(\mathbf{Z})$ is the length of a binary string $\mathbf{Z}$ and $C(\mathbf{Z})$ is its Kolmogorov complexity (the length of the minimal description of

$\mathbf{Z}$ using a universal description language), then:

$$\delta(\mathbf{Z}) = l(\mathbf{Z}) - C(\mathbf{Z}) \tag{2.1}$$

where $\delta(\mathbf{Z})$ is called the randomness deficiency of the string $\mathbf{Z}$. Intuitively, Equation 2.1 states that lower the value of $C(\mathbf{Z})$, higher the $\delta(\mathbf{Z})$, or the lack of randomness.

The Martin-Löf test [71] for randomness provides a method to connect randomness with statistical hypothesis testing. This test can be summarized as a function $t : \mathbf{X} \longrightarrow N$ (the set of natural numbers with 0 and ∞), such that $\forall n \in N, m \in N, P \in P_n$:

$$P\{x \in \mathbf{X}^n : t(x) \geq m\} \leq 2^{-m} \tag{2.2}$$

where P_n is the set of all i.i.d. probability distributions. Equation 2.2 can also be written as:

$$P\{x \in \mathbf{X}^n : t(x) \in [m, \infty)\} \leq 2^{-m} \tag{2.3}$$

Now, if we use the transformation $t'(x) = 2^{-x}$, Equation 2.3 can in turn be written in terms of a new function $t'(z)$:

$$P\{x \in \mathbf{X}^n : t'(x) \in (0, 2^{-t(x)}]\} \leq 2^{-m} \tag{2.4}$$

Hence, a function $t' : X \longrightarrow (0, 2^{-m}]$ is a Martin-Löf test for randomness if $\forall m, n \in N$, the following holds true:

$$P\{x \in \mathbf{X}^n : t'(x) \leq 2^{-m}\} \leq 2^{-m} \tag{2.5}$$

If 2^{-m} is substituted for a constant, say r, and r is restricted to the interval $[0, 1]$, Equation 2.5 is equivalent to the definition of a p-value typically used

in statistics for hypothesis testing [8]. Given a null hypothesis H_0 and a test statistic, p-value is simply defined as the probability of obtaining a result at least as extreme as the one that was actually observed, assuming that the null hypothesis is true. In other words, the p-value is the smallest significance level of the test for which H_0 is rejected based on the observed data, i.e. the p-value provides a measure of the extent to which the observed data supports or disproves the null hypothesis [8].

Non-conformity measure and p-value

In order to apply the above theory to classification problems, Vovk et al. [92] defined a NonConformity Measure (NCM) that quantifies how different a new example is from old ones.

In conformal predictors, nonconformity measure is denoted as a real-valued function $A(B, \mathbf{z}_i)$ that means how different an example $\mathbf{z}_i$ is from the examples in a bag B. The bag is used to formalize the point that the order in which old examples appear does not make any difference [92]. A bag is composed by a collection of n elements, which is write as $\wr \mathbf{z}_1, \ldots, \mathbf{z}_n \int$. The bag $\wr \mathbf{z}_1, \ldots, \mathbf{z}_{10} \int$ is the bag we get from $\mathbf{z}_1, \ldots, \mathbf{z}_{10}$ when we ignore their order.

As we mentioned before, a NCM is used to score how different an example is from a bag of old examples. Many classifiers can be adjusted to be nonconformity measures. For example, the conformity measure underlying k-Nearest Neighbour classifier is defined as,

$$\alpha_i^y = \frac{\sum_{j=1}^{k} D_{ij}^y}{\sum_{j=1}^{k} D_{ij}^{-y}} \tag{2.6}$$

where D_i^y denotes the list of sorted distances between an example $\mathbf{x}_i$ and

other examples with the same label y. D_i^{-y} denotes the list of sorted distances between an example $\mathbf{x}_i$ and other examples with different labels. α_i denotes the value of nonconformity measure of the example $\mathbf{x}_i$. The higher α_i is, the stranger the example $\mathbf{x}_i$ is from the old examples.

Given a new test object, $\mathbf{x}_{n+1}$, a null hypothesis is assumed that $\mathbf{x}_{n+1}$ belongs to the class, y. The non-conformity measures of all the examples are re-computed assuming the null hypothesis is true. A p-value function is defined as:

$$p(y) = \frac{\#\{i = 1, ..., n+1 : \alpha_i^y \geq \alpha_{n+1}^y\}}{n+1} \tag{2.7}$$

This process is repeated with the null hypothesis supporting each of the labels, thus providing a transductive inferential procedure of classification [8]. P-values satisfy the Martin-Löf test definition in Equation 2.5, thus p-values inherit the following validity property :

$$\Pr\{p(y) \leq \varepsilon\} \leq \varepsilon$$

where ε is a given significance level, $0 \leq \varepsilon \leq 1$, such that 1- ε is a desirable confidence level.

The p-values measure the proportion of examples that are less conforming than the new example with the hypothetic label. Obviously, $0 < p(y) \leq 1$. The lower p-value is, the more nonconforming the example is, in relation to the entire set. Conformal predictors could output two types of prediction, single prediction and region prediction [92].

In single prediction, the label which has the highest p-value is assigned to $\mathbf{x}_{n+1}$. If p_j and p_k are the two highest p-values obtained, then p_j is called the credibility of the decision, and $1 - p_k$ is the confidence of the classifier in the decision. With the measurement of confidence, we could say how confident

the prediction is; and the measurement of credibility indicates how suitable the training data are for classifying the example.

Region predictions are presented as sets Γ^{ε} for a specified confidence level $1-\varepsilon$, which contain all the class labels with their p-values greater than $1-\varepsilon$. In on-line mode the confidence threshold $(1-\varepsilon)$ directly translates to the frequency of errors ε, which means the number of errors is controlled by the given confidence level [89].

The approach is summarized in Algorithm 1.

Algorithm 1 Conformal Predictors for Classification

Require: training example sequence $\{\mathbf{z}_1 = (\mathbf{x}_1, y_1), \mathbf{z}_2 = (\mathbf{x}_2, y_2), ..., \mathbf{z}_n = (\mathbf{x}_n, y_n)\}$

Require: new example $\mathbf{x}_{n+1}$

Require: nonconformity measure A

Require: significance level ε

for $y \in Y$ **do**

$\mathbf{z}_{n+1} = (\mathbf{x}_{n+1}, y)$

for $i \in 1, \ldots, n$ **do**

$\alpha_i = A(\wr \mathbf{z}_1, \ldots, \mathbf{z}_{i-1}, \mathbf{z}_{i+1}, \ldots, \mathbf{z}_n \int, \mathbf{z}_i)$

end for

$\alpha_{n+1} = A(\wr \mathbf{z}_1, \ldots, \mathbf{z}_n \int, \mathbf{z}_{n+1})$

$p(y) = \frac{\#\{i=1,\ldots,n+1 : \alpha_i \geq \alpha_{n+1}\}}{n+1}$

end for

return region prediction $\Gamma^{\varepsilon}_{n+1} = \{y : p(y) > \varepsilon\}$

return single prediction $\hat{y}_{n+1} = \arg\max_y \{p(y)\}$

return confidence $confidence(\hat{y}_{n+1}) = 1 - \max_{y \neq \hat{y}_{n+1}} \{p(y)\}$

return credibility $credibility(\hat{y}_{n+1}) = \max_y \{p(y)\}$

2.2.3 Performance Evaluation

Single prediction is also called as simple prediction in other algorithms. The measurement of accuracy and corresponding confidence is used to evaluate the discrimination power.

In region prediction, efficiency is used to show the goodness of predictions. If the predictive set does not contain the true label, it makes an error. If the

size of the set is 0 or 1, this prediction is certain, otherwise it is uncertain and we have multiple prediction. The size of predictions indicates the efficiency of classification. The smaller the set we have, the more efficient prediction will be. Naturally, the higher the confidence level $1 - \varepsilon$ is, the more multiple predictions will appear. For on-line learning, it has been proved that the region prediction is well-calibrated, in the sense that the error rate converges to less than or equal to ε, if we assume the underlying probability distribution is exchangeable [92].

2.2.4 Characteristics of CP

Validity

Validity is one of the advantages of conformal predictors. The conformal predictors are always valid in on-line setting that the frequency of prediction errors does not exceed a pre-specified significance level, ε, at every confidence level $1 - \varepsilon$ [92].

Flexibility

The framework is extensible to any kind of machine learning algorithms for classification, as long as a suitable non-conformity measure is defined. Thus, if a particular algorithm is suitable for an application, this framework can be applied on top of the algorithm to obtain conformal prediction regions as the output [89].

Computational overhead

The framework has some limitations too [92]. A major limiting factor of the framework (in its transductive form) is the computational inefficiency of the framework. Since the framework is based on transductive inference, the NCM has to be recomputed for all the data instances when a new example enters the system. This is a huge computational overhead. This resulted in the design of the Inductive Conformal Predictors framework [92, 80, 79], where the training set is divided into training and calibration portions. The calibration portion is used to compute the p-values when a new example is observed, thus significantly reducing the required computations. However, this approach trades off computational overhead for a loss in predictive efficiency, and hence has to be implemented after careful empirical evaluation [92].

Chapter 3

Feature Selection by Conformal Predictors

Feature selection remains an open area of research within machine learning community. Since the first review of these methods [12], there have been many developments. Based on conformal prediction, we propose a new feature selection method to select optimal features for efficient region predictions [9]. We start from background of feature selection.

3.1 Background

3.1.1 Feature Selection

The learning of the classifier is inherently determined by the features. In theory, more features should provide more discriminating power, but this is not always the case. The nature of high dimensionality of data can cause the problem of "curse of dimensionality" [10]. In practice, with a limited amount

of training examples, excessive number of features will not only significantly slow down the learning process, but also cause poor classification performance as irrelevant or redundant features may confuse the learning algorithm.

Recent research has shown that it is common to machine learning algorithm to be affected by irrelevant or redundant information [45]. The simple nearest neighbour algorithm is sensitive to irrelevant attributes, and its sample complexity (the number of training examples needed to achieve a given accuracy level) grows exponentially with the number of irrelevant attributes [68, 69, 3]. The naive Bayes classifier can be adversely affected by redundant attributes due to its assumption that attributes are independent given the class [67]. Sample complexity for decision tree algorithms can grow exponentially on some concepts (such as parity) as well. Decision tree algorithms such as C4.5 [84, 85] can sometimes over-fit the training data, resulting in large trees. In many cases, removing irrelevant and redundant information can result in C4.5 producing smaller trees [54].

According to these, it is very important for classification to reduce dimensionality and eliminate irrelevant or redundant features. Generally, there are two approaches: feature extraction and feature selection.

Feature extraction transfers or projects original features to lower dimensional spaces, and the obtained features here are generated from the original features.

Feature selection is used to reduce the dimension of objects from $\mathbf{X}^d$ to $\mathbf{X}^m$, with $m < d$. This process can be denoted as following function:

$$fs(\mathbf{X}^d) = \mathbf{X}^m, m < d, \mathbf{X}^m \subset \mathbf{X}^d$$

where, fs is a feature selection method.

By eliminating irrelevant and redundant features, feature selection can help user for better understanding of data, resulting in cheaper collection of a reduced set of features. Compared with feature selection, feature extraction has no saving in data acquisition costs and loses data interpretability due to transformation or projection.

Feature selection is the process of selecting feature subset from a given feature space with the intention of meeting one or more of the following goals [29]:

- Choose the feature subset that maximises the performance of the learning algorithm.
- Minimise the size of the feature subset without reducing the performance of a learning problem significantly.
- Reduce the requirement for storage and computational time to classify data.

Feature selection algorithms (with a few notable exceptions) perform a search through the space of feature subsets, and, as a consequence, must address four basic issues affecting the nature of the search [61]:

1. Starting point. One must select a point in the feature subset space from which to begin the search. For example, one might begin with no feature and successively add attributes. In this case, the search is said to proceed forward through the search space and the approach is called forward selection. Conversely, the search can begin with all features and successively remove them. In this case, the search proceeds backward and the approach is called backward elimination. The search also can begin somewhere in the

middle and move outwards from this point.

2. Search organisation. Search organisation is used to search feature space to generate a candidate feature subset. An exhaustive search of the feature subspace is impractical. With N initial features there exist 2^N possible subsets. This is huge number even for medium-sized N. Heuristic search strategies are more feasible than exhaustive ones and can give good results, although they do not guarantee finding the optimal subset [45]. In each iteration of this searching procedure, all remaining features yet to be selected (rejected) are considered for selection (rejection). There are many variations to this simple process, but generation of subsets is basically incremental (either increasing or decreasing). The order of the search space is $O(N^2)$ or less. These procedures are very simple to implement and very fast in producing results, because the search space is only quadratic in terms of the number of features [29].

3. Evaluation strategy. How feature subsets are evaluated is the single biggest differentiating factor among feature selection algorithms. Algorithms that perform feature selection as the preprocessing step prior to learning can generally be placed into three categories, the filters, the wrappers and embedded methods [55].

4. Stopping criterion. One must decide on some criterion for stopping the search. Both search organisation and evaluation function can influence the choice for a stopping criterion. Stopping criteria based on a search organisation include whether a predefined number of features are selected, and whether a predefined number of iterations reached. Stopping criteria based on an evaluation function can be: whether addition (or deletion) of any

feature does not produce a better subset; and whether an optimal subset according to some evaluation function is obtained.

When features are coming, search organisation generates subsets of features for evaluation. The generation procedure can start from any start point. Then, an evaluation function measures the goodness of a feature subset, and this value is compared with the previous best. If it is found to be better, it replaces the previous best feature subset. The loop continues until some stopping criterion is satisfied. Without a suitable stopping criterion the feature selection process may run exhaustively or forever through the space of feature subsets. The feature selection process halts by outputting a selected subset of features for the classification algorithm.

3.1.2 Existing Feature Selection Methods

Since feature selection has been studied by the statistics and machine learning communities for many years, many methods have been developed for feature selection.

Based on the general characteristics of the data filter methods use individual search or heuristic search to evaluate the merit of feature subsets. The feature evaluation function is independent on learning algorithms. In general, filters are categorized into two approaches, feature ranking and feature weighting, according to the different ways of evaluating features.

Many feature selection algorithms use feature ranking as a principal or auxiliary selection mechanism because of its simplicity, scalability, and good empirical success [39]. Feature ranking methods always use some statistical approaches as evaluation functions to rank individual features. For example,

when the label is binary, a typical feature evaluation function used is Fisher discriminant ratio (FDR) [77], shown as the following Equation 3.1.

$$FDR(\mathbf{X}_k) = \frac{\left(\mu_{(\mathbf{X}_k,1)} - \mu_{(\mathbf{X}_k,2)}\right)^2}{\sigma^2_{(\mathbf{X}_k,1)} + \sigma^2_{(\mathbf{X}_k,2)}} \tag{3.1}$$

where $\mathbf{X}_k$ is the vector of the features, $k = 1, \ldots, d$. μ and σ are the mean and the standard deviation of example $\mathbf{X}_k$, respectively. Given a training set, the main idea of FDR is to find a line in the original feature space that can separate examples as much as possible. In other words, the bigger the square of the difference between the mean of two kinds of examples is and at the same time the smaller the within-class scatters are, the better the expected line is. Top ranking features are selected as the feature subset $\mathbf{X}^m$ such that $FDR(\mathbf{X}^m)$ achieves its maximum, $\mathbf{X}^m \subset \mathbf{X}^d$.

If the label takes more than two values and the features are categorical, feature evaluation function can be constructed based on χ^2 statistic [87]. The main idea of χ^2 statistic is using χ^2 to determine whether the relative frequencies of the classes in adjacent intervals are similar enough to justify merging. The formula for computing the χ^2 value for two adjacent intervals is

$$\chi^2 = \sum_{i}^{2} \sum_{j}^{C} \frac{(A_{ij} - E_{ij})^2}{E_{ij}} \tag{3.2}$$

where C is the number of classes, A_{ij} is the number of examples in the i-th interval with class j, and E_{ij} is the expected frequency of $A_{ij} = R_i * C_j / N$. R_i is the number of examples in the i-th interval, C_j is the number of examples of class j in the two intervals, N is the total number of examples in the two intervals,. The extent of the merging process is controlled by automatically defined χ^2 threshold. The threshold is determined through attempting to maintain the comprehensibility of the original data.

The other way to select features by filter method is to assign a relevance weight to each feature. Each feature's weight reflects its ability to distinguish among the class values, such as RELIEF method [57]. RELIEF use heuristic research to rank features by their weights. Features which exceed the user-specified threshold will be selected. The method only operates on binary-classes. The algorithm works by randomly sampling examples from the training data. For each example sampled, the nearest example of the same class (nearest hit) and opposite class (nearest miss) is found. A feature's weight is assigned or updated according to how well it distinguishes the sampled example from its nearest hit and nearest miss. The weight calculating formula used by RELIEF is shown in Equation 3.3:

$$W_{\mathbf{X}} = W_{\mathbf{X}} - \frac{\text{diff}(\mathbf{X}, \mathbf{R}, \mathbf{H})^2}{n} + \frac{\text{diff}(\mathbf{X}, \mathbf{R}, \mathbf{M})^2}{n}, \tag{3.3}$$

where W_X is the weight for the feature X, R is a randomly sampled example, H is the nearest hit, M is the nearest miss, and n is the number of randomly sampled examples. The function *diff* calculates the difference between two examples for a given feature, *diff* $\in [0, 1]$. It is clear that a feature's weight will be high if it differentiates between examples from different classes and has the same value for examples of the same class.

As mentioned in [61] "The rationale for wrappers is to select the feature subset which should provide a better estimate of accuracy predicted by an induction method than a separate measure that has an entirely different inductive bias". Most of the variations in its applications are due to the following: the selection of feature subset search organisation; the ways to assess the prediction performance of a induction algorithm and halt it; and which

induction algorithms to use. A typical wrapper method can use different kinds of classifiers for evaluation; hence no representative method is chosen for demonstration of the categories under evaluation in this section.

John, Kohavi and Pfleger [47] were the first to consider the wrapper [4] as a general framework for feature selection in machine learning. They also presented formal definitions for two degrees of feature relevance, and claimed that the wrapper is able to discover relevant features. They gave the assumptions of two degrees of feature relevance: a feature $\mathbf{X}_k$ is said to be strongly relevant to the labels if the probability distribution of the classes, given the full feature set, changes when $\mathbf{X}_k$ is removed; a feature $\mathbf{X}_k$ is weakly relevant if it is not strongly relevant and the probability distribution of the classes, given some subset $\mathbf{X}^m$ (containing $\mathbf{X}_k$), $\mathbf{X}^m \subset \mathbf{X}^d$, changes when $\mathbf{X}_k$ is removed. The rest of the features which are not strongly or weakly relevant are said to be irrelevant. To test these assumptions, experiments were firstly applied on three artificial and three natural datasets using ID3 and C4.5 as induction algorithm [84, 85]. Both forward selection and backward elimination search were used. Accuracy was estimated by using 25-fold cross validation on training data and a disjoint test set was used for reporting final accuracies. Results showed that feature selection did not significantly change ID3 or C4.5's classification performance. The main effect of feature selection was to reduce the size of the trees. Then, Caruanna and Freitag [19], Vafaie [94] and Cherkauer [24] used greedy search and genetic search to improve the feature subset performance of ID3 and C4.5, respectively.

Langley and Sage [67] noted that the performance with redundant features can be improved by removing such features. They used the naive Bayes

classifier as induction algorithm. Because the naive Bayes classifier assumes that probability distribution for features are independent of each other for each class. A forward heuristic search strategy is employed to select features. The rationale for a forward search is that it should immediately detect dependencies when harmful redundant features are added [46]. The selection will stop adding features when none of alternatives improves classification accuracy. Experimental results showed overall improvement and increased learning rate on three out of six real world datasets, with no change on the remaining three [67]. The Correlation-Based Feature Selection (CFS) method is another method selecting feature subsets based on the degree of redundancy among features [45]. The method evaluates the worth of a subset of features by considering the individual ability of each feature along with the degree of redundancy between them, which aims to find the subsets of features that are individually highly correlated with the labels but have low inter-correlation [58].

An embedded feature selection method is a machine learning algorithm that returns a model using a limited number of features. Many algorithms can be turned into embedded methods for feature selection. Assume that we have a classifier $f(\mathbf{x}) = y$. First, we parameterize the function by $\mathbf{w}$ and add in the σ weights where $\sigma \in \{0,1\}^m$. $\sigma_i = 1$ represents the use of feature i or $\sigma_i = 0$ rejection of feature i. For each training example, $\mathbf{x} = (\sigma_1 x_1, ..., \sigma_m x_m)$, $f(\mathbf{x}) = f(\sigma_1 x_1, ..., \sigma_m x_m)$. For example, a linear classifier could be denoted as:

$$(\mathbf{w}, \mathbf{x}) + b = \sum_{i=1}^{m} (w_i, \sigma_i x_i) + b = \sum_{\sigma \neq 0} w_i x_i + b$$

where m is the number of features and b is a constant offset. The aim is to

find $\mathbf{w}$ and σ that minimize the generalisation error:

$$min_{\sigma,\mathbf{w}} R(\mathbf{w}, \sigma)$$

where

$$R(\mathbf{w}, \sigma) = \int L(f(\mathbf{w}, (\sigma_1 x_1, ..., \sigma_m x_m)), y) df(\mathbf{x}).$$

where L is a loss function. The evaluation criterion is based on the generation error. And selection stops when no more changes of the generation error or user defined stopping criterion is achieved.

To meet the goal of feature selection, methods should be checked by their ability of removing irrelevant and redundant features and computational complexity. Wrappers utilize the learning machine of interest as a black box to score feature subsets according to their predictive power, see as Figure 3.1. Filters select subsets of features independently of the chosen predictor, see as Figure 3.2; Embedded methods perform feature selection in the process of training and are usually specific to given learning machines, see as Figure 3.3.

Among these three methods, filters rank features according to their importance in differentiating examples of different classes and can be very fast with the cost of $O(N)$ (where N is the number of features). However, it is incapable of removing redundant features because redundant features likely have similar rankings or weightings. A Fast Correlation-Based Filter (FCBF) solution is proposed to address this problem [99]. In the method the evaluation of individual features is based on the correlation between a feature an the label. The approach is reported to be very useful when a dataset contains a very large number of features [64].

The wrapper method offers a simple and powerful way to address the problem of feature selection. It often gives better prediction results than

filters because it is optimized for the particular learning algorithm used [72]. Although the wrapper approach does not use a relevance and redundancy measure directly, it is shown by Kohavi and Sommerfield that the optimal feature subset obtained must be from the relevant feature subset [58]. But, wrappers are often criticized because they require massive amounts of computation [54]. An exhaustive search would definitely find the optimal solution; however, a search on 2^N possible feature subsets (where N is the number of features) is computationally impractical. Narendra and Fukunaga introduced the branch and bound algorithm, which finds the optimal feature subset if the criterion function used is monotonic. However, although the branch and bound algorithm makes problems more tractable than an exhaustive search, it becomes impractical for feature selection problems involving more than 30 features [46]. Although heuristic search strategies seem to be particularly computationally advantageous, generally result in an $O(N^2)$ worst case search.

Compared with filters and wrappers, embedded methods offer the same advantages as wrapper methods concerning the interaction between the feature selection and the classification, and present a better computational complexity since the selection of features is directly included in the classifier constructing during the training process. But, the selected features may not perform well in other classifiers [64].

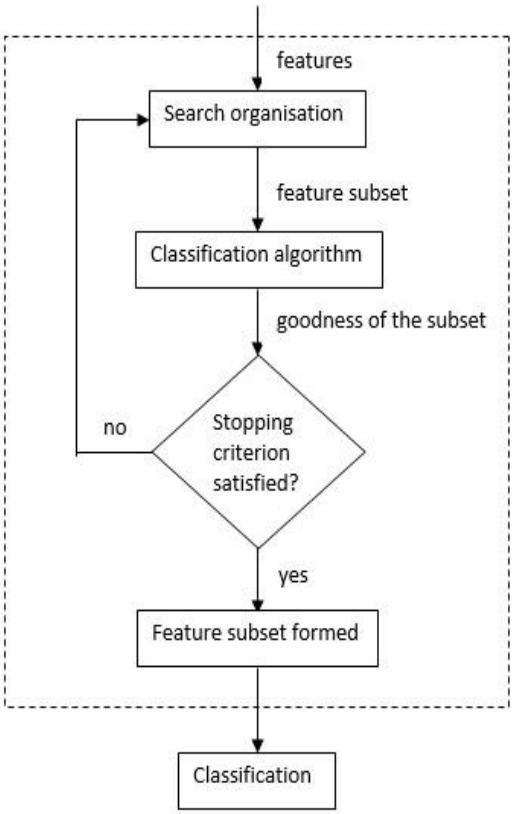


Figure 3.1: The feature wrapper approach

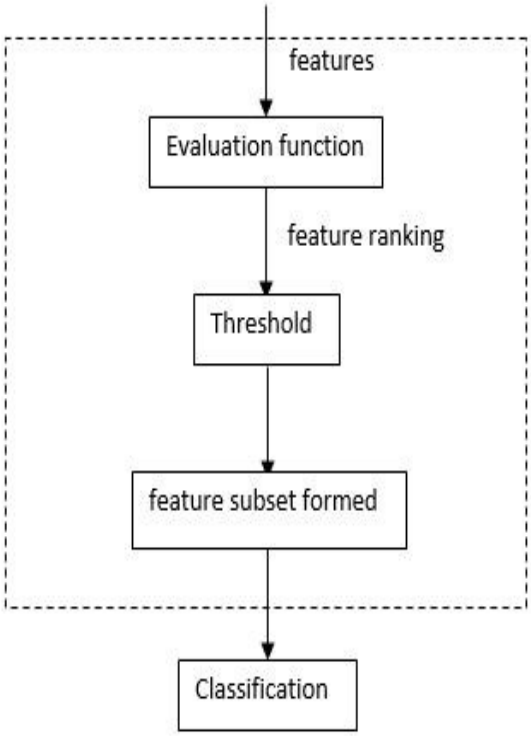


Figure 3.2: The feature filter approach

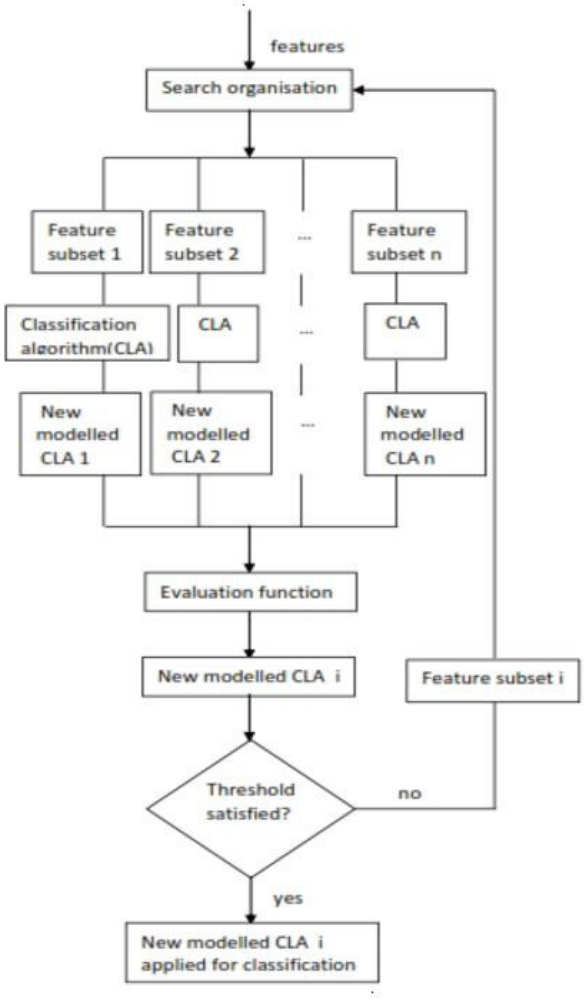


Figure 3.3: The embedded approach

3.2 Feature Selection Based on CP

Previous feature selection methods work to maximize accuracies of classifications. In conformal predictors, accuracy is guaranteed for a given significance level and efficiency of region prediction is the main evaluation criterion. So far there is only one published feature selection method designed for confor-

mal predictor, named as Strangeness Minimization Feature Selection [15].

3.2.1 Strangeness Minimisation Feature Selection (SMFS)

SMFS method is specifically designed for conformal predictors. The intuition for the approach is that reducing overall strangeness implies an increase in conformity amongst the examples in the sequence [15]. It means that the feature subset which minimises overall strangeness is the most relevant to maximising conformity between training examples.

The goal of SMFS is defined in a way similar to a wrapper feature selection method as a search to minimize the strangeness value across all possible feature subsets. To reduce the computational complexity in practical implementations, the method restricts the nonconformity measurement as linear one, so that the feature subset can be found based on the value of individual features. This is the way how filter methods find the optimal feature subset. According to the paper [15], the optimal subset can also be found by the forward heuristic search method based on the performance of feature subset.

In CP, strangeness examples are computed as α-nonconformity values for each example. BY using linear nonconformity measures, it can compute strangeness values for each feature. The measurement values are denoted as β-nonconformity values. The goal of SMFS can be reformulated to minimize the sum of β-nonconformity values across subsets of features to size t with feature space F.

$$S_0 = \arg\min_{S \in G} \sum_{j \in S} \beta_j \tag{3.4}$$

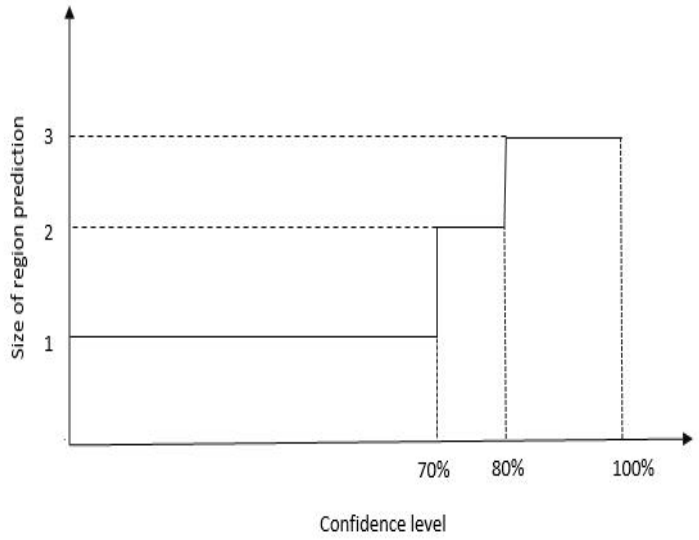
where G is a set of features with size of t, $G = \{R : R \subseteq F, |R| = t\}$ and

$S \subseteq F$. The method has two ways to stop. The first one is to pre-set the size of feature subset. The other is to test the performance of the feature subset in training set and select the optimal one.

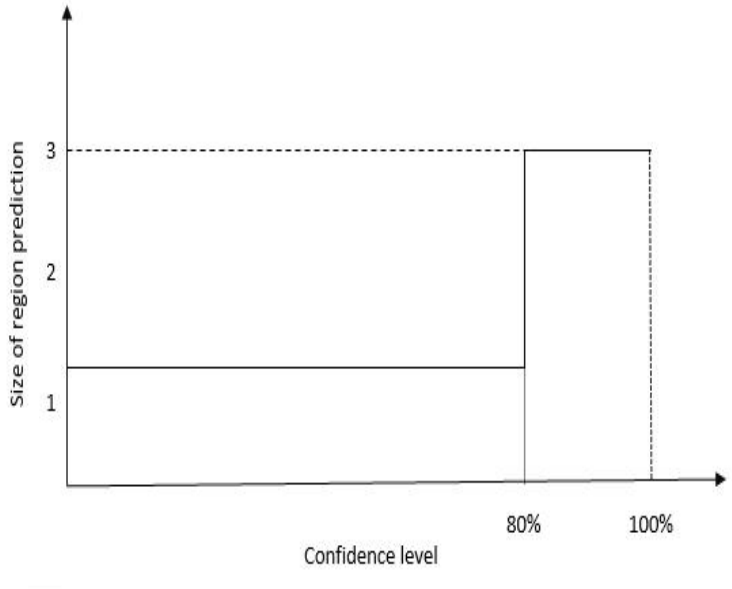
3.2.2 Average Confidence Minimization (ACM) Method

Based on conformal predictors we introduce a wrapper method and hope to select the features which have more discriminational power for efficient region predictions. The method is using the measurement of average confidence to select the features and named as average confidence minimization method [101].

In online machine learning, if an example has higher confidence, the region prediction will be more efficient at the same significance level. For example, assume that we have three labels, a, b and c, their p-values, calculated by two different feature subsets s_1 and s_2, are $p(a) = 0.4, p(b) = 0.3, p(c) = 0.2$ and $p(a) = 0.4, p(b) = 0.2, p(c) = 0.2$, respectively. The size of region prediction at different confidence levels is shown in Figure 3.4. It can be found that the subset s_2 which classifies the example with higher confidence and provides the classifier more power for efficient region prediction. Since prediction accuracy is guaranteed and efficiency is related to confidence, the average confidence of all examples is used as evaluation criterion. The feature subset which provides the highest average confidence will be selected.



(a) Feature subset s_1



(b) Feature subset s_2

Figure 3.4: Region predictions using two feature subsets

As shown in Algorithm 2, ACM method starts from empty feature subset. Heuristic search, as described in Section 2.3.1, is used to generate candidate feature subsets for evaluation. In each iteration of this searching procedure, all remaining features yet to be selected are considered for selection. Then, the feature subset which leads to the highest average confidence of single predictions is kept. When the average confidence is reaching the highest value or remaining unchanged, the selection stops.

Algorithm 2 Average Confidence Maximisation

Require: training set $\{\mathbf{z}_1 = (\mathbf{x}_1, y_1), \ldots, \mathbf{z}_n = (\mathbf{x}_n, y_n)\}$, $\mathbf{x} \in \mathbf{X}$, $y \in \mathbf{Y}$, $D = \{1, \ldots, d\}$ where $d = \dim(X)$

Require: nonconformity measure A, $S = \emptyset$

for $f \in 1, \ldots, d$ **do**

$\mathbf{C} = \mathbf{D} \backslash \mathbf{S}$

for $c \in \{1, \ldots, |\mathbf{C}|\}$ **do**

$\mathbf{S}_2 = \mathbf{S} \bigcup \mathbf{C}_c$ (c_{th} element of C)

for $i = 1 : n$ **do**

for $y \in Y$ **do**

$\mathbf{z}_i = (\mathbf{x}_i, y), \mathbf{x} \in \mathbf{S}_2$

for $j = 1 : n$ **do**

$\alpha_j = A(\wr \mathbf{z}_1, \ldots, \mathbf{z}_{j-1}, \mathbf{z}_{j+1}, \ldots, \mathbf{z}_n \int, \mathbf{z}_j)$

end for

$p(y) = \frac{\#\{j=1,\ldots,n : \alpha_j \geq \alpha_n\}}{n}$

end for

$confidence(i) = 1 - \max_{y \neq \hat{y}_i} p(y)$

end for

$Con_c = \text{mean } (confidence)$

end for

if $Con_{\mathbf{S}} < \max\{Con_c\}$ **then**

$\mathbf{S} = \mathbf{S} \bigcup \mathbf{C}_{c:\max\{Con_c\}}$

else

$Break$

end if

end for

return S

3.2.3 Issues in ACM

The ACM method is firstly applied on Abdominal Pain dataset. It is not as good as we expected because the features selected by ACM method are significantly different from those features provided by some medical experts [101]. Furthermore, it is hard to find out whether the average confidence is reaching the highest point or not.

- Evaluation strategy.

  Like other wrapper methods, ACM utilizes CP learning framework as a black box to score feature subsets according to their predictive power. It does not directly use the characteristics of data as filter methods. This is why the orders of feature subset selected by ACM are quite different from the ones selected by SMFS method and by the medical experts in previous experiments. So, if we consider the relationship measure between features and classes along with the evaluation of classifier, will it find the optimal feature subset? Furthermore, could the method be used generally?

- Stopping criterion.

  In ACM, the selection process will automatically stop when the average confidence of feature subset achieves the highest value. But our experimental results show that the selected feature subset may contain some redundant features. So, what stopping criterion can be used to avoid such case happening?

3.2.4 Improved ACM

In conformal predictors, for the object $\mathbf{x}_i$ of an example $\mathbf{z}_i$, nonconformity score α_i shows how conformity $\mathbf{x}_i$ with a hypothesis label y is from the old examples. p-values compare α_i to the other nonconformity scores of the old examples. If it is small, $\mathbf{z}_i$ is very nonconforming. If it is large, $\mathbf{z}_i$ is very conforming. Confidence of an example $\mathbf{z}_i$ is calculated by Equation 3.5, shows how confident $\mathbf{x}_i$ with a hypothesis label y is to be conforming.

$$confidence_i = 1 - \max_{y_i \neq \hat{y}_i} p(y) \tag{3.5}$$

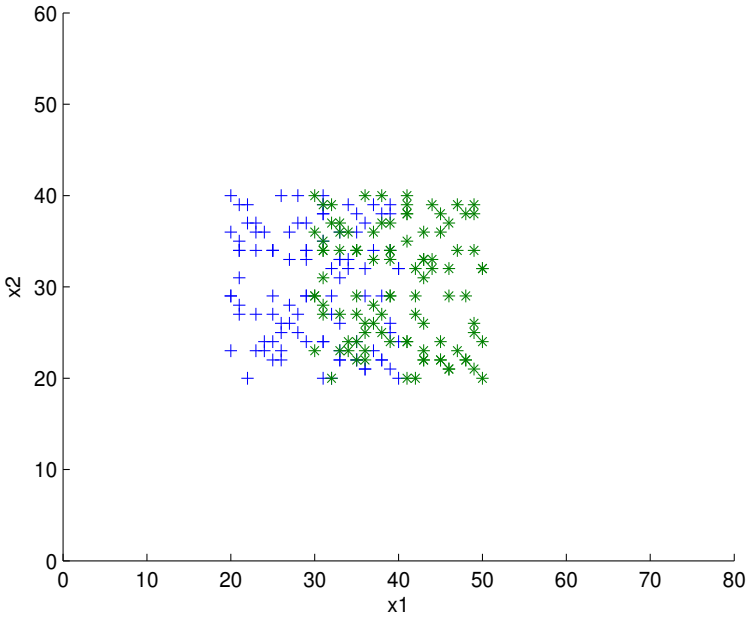
In IACM, the label for each example is known. $\{p(y) : y \neq y_i\}$ shows how conformity $\mathbf{x}_i$ with the false labels. The confidence $confidence_i$ of the example $\mathbf{z}_i$ is calculated by Equation 3.6 and shows how confident $\mathbf{z}_i$ is to be conforming. Thus, the average confidence $confidence_y$ demonstrates the discrimination power of the objects for classifying labels.

$$confidence_i = 1 - \max_{y \neq y_i} p(y) \tag{3.6}$$

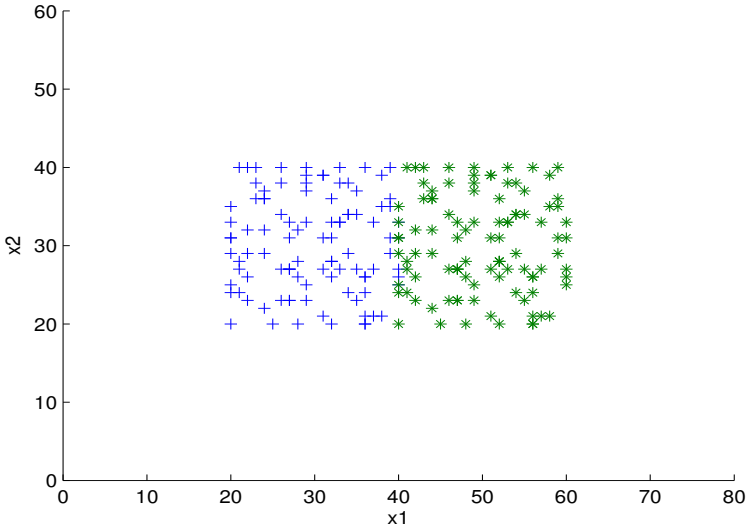
$$confidence_y = mean(\sum_{i=1}^{n} confidence_i) \tag{3.7}$$

For example, in CP underlying KNN, NC and SVM algorithms, the nonconformity measure of an example is calculated by the comparison of distances between the class labels. If the examples in different classes are far away from each other, the classification of these examples will be accurate and confident. If the examples in different classes are close to each other, the classification of these examples will be indeterminate. The average confidence in later case must be lower than the one in former case. This example can be

illustrated using simulated data. In Figure 3.5, the examples with label (+) in Figure 3.5(b) is clearly further away from the examples of the other class (*) than that in Figure 3.5(a). So we expect it to have a relatively higher confidence value, $confidence_{+,*}$. We used Nearest Centroid method (NC) and Support Vector Machine (SVM) as underlying algorithms for conformal predictors in on-line mode, respectively. Results are shown in Table 3.1 when ε is 0.05. It can be found that examples in Figure 3.5(b), which have higher $confidence_{+,*}$ and are more separable than ones in Figure 3.5(a), performance of predictions for examples in Figure 3.5(b) are better.



(a) closer classes



(b) further classes

Figure 3.5: Simulated data for illustrating

Table 3.1: Prediction results of the examples in Figure 3.5

NCM	the examples in Figure3.5(a)			the examples in Figure3.5(b)		
	accuracy	uncertain predictions	$confidence_{+,*}$	accuracy	uncertain predictions	$confidence_{+,*}$
NC	0.74	0.87	76%	0.95	0.12	96%
SVM	0.68	0.93	58%	0.91	0.72	89%

In our previous works [101], it is assumed that if the average confidence reaches the highest point, the selection stops and the selected feature subset is the optimal one. However, our experimental results showed this is not the case, when the average confidence is nearly to reach the highest point, the feature subset may start to include some correlated features. Thus the analysis of irrelevance and redundancy [17] is introduced in our method. The selection processing will stop when redundant feature is starting being included. Now we discuss what are relevance and redundance and how to use them to analyze features.

Feature relevance and feature redundancy

The notions of relevance and redundance have been formally investigated in the philosophy literature [49, 18, 38]. Conditional independence among variables finds its applications in Bayesian belief network [81] and Ovarian Cancer diagnosis [90, 43], where irrelevance is identified as independence, and redundancy as conditional independence.

Let's consider a sequence of training examples:

$$(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n), \mathbf{x}_i \in \mathbf{X}^d, y_i \in \mathbf{Y}.$$

Let $\mathbf{X}^d$ be the full set of features, F be a set of selected features, $F \subset \mathbf{X}^d$ and $S = \mathbf{X}^d - F$. Relevant feature subset indicates that assignment of classes is depend on features, which can be formalized as follows [6, 100].

Notation 1. *A feature subset F is relevant feature subset to class y if y is dependent on F*

$$F \not\perp Y$$

Notions of feature redundancy are normally defined in terms of feature correlation. It is widely accepted that two features are redundant to each other if their values are completely correlated [100]. In reality, it may not be straightforward to determine feature redundancy when a feature is correlated (perhaps partially) with a set of features. We now formally define feature redundancy [6].

Notation 2. *Let S be the current set of features, a feature f is defined as redundancy feature if*

$$\{f \perp Y\}|S$$

.

In this work, we use two types of statistical tests to realize these two notions which reject the following two null hypotheses [6]:

1. The null hypothesis: the assignment of labels is independent of feature subset.

2. The null hypothesis: each of selected features does not contain any information useful for improving the predictive ability of the feature subset.

The test statistics are p-values. There are two p-values, the p-value for null hypothesis 1 is main p-value and the p-value for null hypothesis 2 is conditional p-value [43]. We calculate all these p-values using the Monte-Carlo method [1].

The main p-value is based on the null hypothesis that the assignment of labels is independent of the selected feature subset S, shown as Algorithm 3. E_0 is set as the number of errors occurring in classification with feature subset. We use k-Nearest Neighbour for classification. Different k have been used and the smallest number of errors is selected as E_0. The number of errors occurring in classification for a large number N (we used $N = 100$) of times on the same dataset with randomly permuted labels is denoted as E'. We then calculate the number Q of times, and note that the statistic is as good as or better than the statistic E_0 computed from the true labels. The p-value is then estimated as $(Q+1)/(N+1)$. The main p-value could be used to check the predictive ability of feature subset. If the main p-value does not exceed 0.05 (significance level), the feature subset is relevant to classification.

Algorithm 3 Main p-value calculation

Require: $N = 100$, number of trials

Require: $E_0 :=$ the smallest number of errors caused by different K.

Require: $Q := 0$

for $j := 1, ..., N$ **do**

Randomly assign labels to samples.

Recalculate $E' :=$ the number of errors occurring in classification on the dataset with randomly permuted labels.

if $E_0 \geq E'$ **then**

$Q := Q + 1$

end if

end for

$(Q + 1)/(N + 1)$ as the main p-value.

Suppose that the main p-value is significant. In this case, we wish to use conditional p-values to demonstrate the contributions of new coming features, shown as Algorithm 4. The difference from the computation of the main p-value is the following. At each step of the loop we permute randomly the m_{th} vector of feature subset S, $|S| = d$ and $m \leq d$. The labels and remaining vectors are left intact. If the conditional p-value of a particular feature in feature subset does not exceed 0.05 (significance level), the feature contains useful information for classification.

Algorithm 4 Conditional p-value calculation

Require: $N = 100$, number of trials

Require: S, a feature subset

Require: m, $m \leq |S|$

Require: $E_0 :=$ the smallest number of errors caused by different K.

Require: $Q := 0$

for $j := 1, ..., N$ **do**

Permute randomly m_{th} vector of S.

Recalculate $E' :=$ the number of errors occurring in classification on the dataset with randomly m_{th} vector.

if $E_0 \geq E'$ **then**

$Q := Q + 1$

end if

end for

$(Q + 1)/(N + 1)$ as the conditional p-value.

With change of evaluation criterion and stopping criterion, the feature selection algorithm based on conformal predictors is summarized as Algo-

rithm 5.

3.3 Results and Discussion

The algorithm is firstly applied on Abdominal Pain dataset, shown in Appendix A.1. The dataset consists of a training set of 4387 patient records and a test set of 2000 patient records, with 9 categories of diseases and 33 types of symptoms [42]. For each category of diseases, there is a list of features which said to be the most associated with corresponding diseases suggested by medical experts. Both SMFS method and ACM method are applied on this dataset as well. Features selected by IACM, ACM, SMFS were compared to the features provided by medical experts. The discrimination power of these features were also compared.

To show that IACM method is suitable for other different data types, It is applied on other 4 datasets from UCI dataset repository [14]. In these experiments, we used four different feature selection methods to make comparisons, which are FCBF, SMFS, CFS and IACM. Note that FCBF and SMFS are two filter methods while CFS and IACM are two wrapper methods. Selected features are used for classification. Classifications were implemented in both off-line mode and on-line mode.

Algorithm 5 IACM

Require: training set $\{\mathbf{z}_1 = (\mathbf{x}_1, y_1), \ldots, \mathbf{z}_n = (\mathbf{x}_n, y_n)\}$, $\mathbf{x} \in \mathbf{X}$, $y \in \mathbf{Y}$, $D = \{1, \ldots, d\}$ where $d = \dim(X)$

Require: nonconformity measure A, $S = \emptyset$, main p-value calculator MPC and conditional p-value calculator CPC

for $f \in 1, \ldots, d$ **do**

$\mathbf{C} = \mathbf{D} \backslash \mathbf{S}$

for $c \in \{1, \ldots, |\mathbf{C}|\}$ **do**

$\mathbf{S}_2 = \mathbf{S} \bigcup \mathbf{C}_c$

for $i = 1 : n$ **do**

for $y \in Y$ **do**

$\mathbf{z}_i = (\mathbf{x}_i, y), \mathbf{x} \in \mathbf{S}_2$

for $j = 1 : n$ **do**

$\alpha_j = A(\wr \mathbf{z}_1, \ldots, \mathbf{z}_{j-1}, \mathbf{z}_{j+1}, \ldots, \mathbf{z}_n \int, \mathbf{z}_j)$

end for

$p(y) = \frac{\#\{j=1,\ldots,n : \alpha_j \geq \alpha_n\}}{n}$

end for

$confidence(i) = 1 - \max_{y \neq y_i} p(y)$

end for

$Con_c = \text{mean } (confidence)$

end for

if $MPC(\mathbf{S}_{2\max\{Con_c\}}) < 0.05$ and $CPC(\mathbf{C}_{c\max\{Con_C\}}, \mathbf{S}_2) < 0.05$ **then**

$\mathbf{S} = \mathbf{S} \bigcup \mathbf{C}_{c:\max\{Con_c\}}$

else

$Break$

end if

end for

return S

3.3.1 Results of the Applications for Abdominal Pain Diagnosis

Data description

The abdominal dataset composed by 33 symptoms, shown in Appendix A.1. Each of symptoms contains different numbers of values to show different kinds or degrees of the symptom. For example, there are four types of pain, which are steady, intermittent, colicky and sharp. There could have 135 features in total when unfold these 33 symptoms. The features is boolean, which means the patient has this value of one symptom or not, separately. The provided lists of relevant features were obtained by the purely statistical method and were then discussed with medical experts. 7 of 9 diseases have the list of relevant features, APP, DIV, PPU, CHO, INO, RCO and DYS.

Comparison of selected features

Tables 3.2, 3.3, 3.4, 3.5, 3.6, 3.7, and 3.8 list the features selected for 7 diseases, separately by SMFS, ACM and IACM using NC as underlying algorithm. Features in left columns are all 135 features. Features in right three columns are optimal ones selected by different methods. Features in "Experts" column are approved by medical experts to be relevant for corresponding disease diagnosis. Feature subset significance method [13] is used to demonstrate how significant the selected features are in relation to expert knowledge. The method uses the comparison between two feature subsets as a measure of significance, and the smaller the significance, the less likely the overlapped features in the two feature subsets are derived just by chance.

A detailed introduction of the method is given in Appendix B. In this ex-

periment, we respectively compare the feature subsets selected by the three methods with expert's features. As it can be seen, none of the three feature selection methods select the same set of features as experts. Compared with expert's feature, features selected by ACM has the largest statistical significance values; features selected by SMFS have the least significance values for 5 out of 7 groups; features selected by IACM have the least significance values for the rest 2 groups. Furthermore, features selected by SMFS for APP, PPU, CHO and INO contain correlate features. For example, SMFS select both "rebound absent" and "rebound present" as relevant features to APP, but these two features are correlated that the symptom only has two possible values, either absent or present.

Table 3.2: Features selected for APP

Features	Experts	SMFS	ACM	IACM
Site of tenderness: right lower quadrant	√	√	√	√
Site of present pain: right lower quadrant	√	√	√	√
Pain-site onset: right lower quadrant	√			
Rebound present	√	√	√	√
Rebound absent		√		
Guarding present	√	√	√	√
Guarding absent		√		
Rectal examination: tender right side	√	√	√	√
Progress of pain: getting worse	√			
Duration of pain: 24-48 hours	√			
Age: 10-19 years	√			
Duration of pain: 12-24 hours	√			√
Pain-site onset: central		√	√	√
Aggravating factors: coughing			√	√
Colour: flushed		√	√	√
Severity of pain			√	
...				
Statistical significance		8×10^{-5}	8×10^{-5}	2×10^{-6}

Table 3.3: Features selected for DIV

Features	Experts	SMFS	ACM	IACM
Tenderness in left lower quadrant	√	√		√
Duration of pain over 48 hours	√			
Age 60-69	√		√	
Age 70 and over	√	√		
Bowel habit: mucus			√	√
Bowel habit: blood		√	√	
Pain-site onset: lower half		√		
Rectal examination: mass felt			√	
...				
Statistical significance		4×10^{-3}	1×10^{-1}	5×10^{-2}

Table 3.4: Features selected for PPU

Features	Experts	SMFS	ACM	IACM
Rigidity present	√	√		√
Rigidity absent		√		
Bowel sounds decreased	√	√	√	
Site of tenderness: general	√	√		√
Duration of pain: under 12 hours	√			
abdominal movement poor	√			
Pain-site onset: right loin			√	
Relieving factors			√	
...				
Statistical significance		9×10^{-5}	1×10^{-1}	1×10^{-3}

Table 3.5: Features selected for CHO

Features	Experts	SMFS	ACM	IACM
Murphy's test positive	√	√	√	√
Murphy's test negative		√		
Tenderness right upper quadrant	√	√	√	√
Colour: jaundiced	√			√
Sex: male or female	√			
Pain-site present: right upper quadrant		√	√	
Colour: cyanosed				√
Type of pain: steady			√	
Age: 70 and over			√	
Pain-site present: upper half			√	
...				
Statistical significance		3×10^{-3}	9×10^{-3}	4×10^{-5}

Table 3.6: Features selected for INO

Features	Experts	SMFS	ACM	IACM
Abdominal distension	√	√	√	√
Abdominal distension absent		√		
Pain colicky	√			
Vomiting present	√			
Abdominal mass present	√			√
Previous surgery	√	√		
No previous surgery		√		
pain-site present: central	√		√	√
Bowel sounds increased	√	√		
Age: 70 and over	√	√		
Site of tenderness: central			√	√
Site of tenderness: general				√
Abdominal movement		√	√	√
Bowel sounds normal		√		
Pain-site present: upper half			√	
Age: 50-59			√	
Pain-site present: general			√	
Colour: cyanosed			√	
...				
Statistical significance		3×10^{-4}	7×10^{-2}	2×10^{-3}

Table 3.7: Features selected for RHO

Features	Experts	SMFS	ACM	IACM
Pain onset in right loin	√	√	√	√
Pain onset in left loin	√	√		√
Micturation: haematuria	√	√	√	√
Aggravating factors: nil	√			
Pain-site present: left loin		√	√	√
Pain-site present: right loin		√	√	
Pain-site onset: left half			√	
Vomiting present			√	
Site of tenderness: right loin				√
Bowel habit: mucus				√
...				
Statistical significance		9×10^{-5}	1×10^{-3}	1×10^{-4}

Table 3.8: Features selected for DYS

Features	Experts	SMFS	ACM	IACM
Site of tenderness: epigastric	√	√	√	√
Pain-site present: epigastric	√	√	√	√
Pain-site present: upper half	√		√	
History of dyspepsia	√	√		√
No history of dyspepsia		√		
Pain-site onset		√	√	√
Type of pain: steady			√	
Pain-site present: right upper quadrant			√	√
Vomiting present				√
...				
Statistical significance		9×10^{-5}	2×10^{-4}	2×10^{-4}

Comparison of prediction performance

Classifications here are implemented in on-line mode. Conformal predictor underlying NC algorithm is used for classification. The experiment is repeated 10 times and the mean and the standard deviation are showed in tables. Performances of region predictions with features selected by different methods are compared, shown as Table 3.9. Since the PPU and DIV diseases have too small number of examples, about 100 examples out of 6387, we apply the method only on the rest 5 diseases. In most cases, features selected by IACM provide the most efficient region predictions and the lowest standard deviation. As it can be seen in Table 3.10, features selected by IACM also demonstrate the strongest discrimination power for single predictions, as the

predictions have the least error rates and the highest confidence values,

Table 3.9: Region predictions of Abdominal Pain dataset

Disease	Features	$\varepsilon = 0.05$		$\varepsilon = 0.10$		$\varepsilon = 0.20$	
		error rate	uncertain	error rate	uncertain	error rate	uncertain
APP	Experts	0.04±0.01	0.51±0.08	0.08±0.01	0.32±0.07	0.18±0.02	0.11±0.05
	SMFS	0.04±0.01	0.47±0.09	0.08±0.02	0.29±0.10	0.17±0.03	0.13±0.08
	ACM	0.04±0.01	**0.44±0.07**	0.09±0.02	0.28±0.05	0.16±0.02	0.11±0.04
	IACM	0.03±0.02	0.45±0.06	0.07±0.03	**0.27±0.06**	0.17±0.03	**0.07±0.03**
CHO	Experts	0.02±0.01	0.35±0.14	0.07±0.02	0.15±0.07	0.10±0.02	0.10±0.08
	SMFS	0.03±0.01	**0.27±0.09**	0.08±0.02	0.11±0.03	0.16±0.02	0.08±0.03
	ACM	0.03±0.03	0.31±0.07	0.07±0.03	0.13±0.03	0.15±0.04	0.08±0.01
	IACM	0.04±0.02	0.29±0.01	0.08±0.02	**0.10±0.04**	0.15±0.03	**0.05±0.04**
INO	Experts	0.04±0.01	0.41±0.11	0.07±0.02	0.27±0.11	0.16±0.02	0.15±0.07
	SMFS	0.03±0.01	0.41±0.10	0.07±0.02	0.27±0.14	0.15±0.04	0.19±0.13
	ACM	0.03±0.02	0.43±0.20	0.07±0.03	0.19±0.17	0.16±0.04	0.14±0.16
	IACM	0.04±0.02	**0.29±0.06**	0.07±0.01	**0.17±0.06**	0.15±0.02	**0.10±0.06**
RHO	Experts	0.02±0.01	0.48±0.06	0.03±0.02	0.44±0.08	0.03±0.02	0.42±0.08
	SMFS	0.03±0.02	0.40±0.27	0.07±0.02	0.18±0.13	0.10±0.03	0.15±0.13
	ACM	0.03±0.02	0.40±0.23	0.07±0.02	0.23±0.16	0.10±0.03	0.20±0.17
	IACM	0.03±0.02	**0.40±0.16**	0.07±0.02	**0.12±0.07**	0.12±0.02	**0.07±0.07**
DYS	Experts	0.03±0.01	0.40±0.06	0.08±0.01	0.22±0.06	0.14±0.02	0.11±0.05
	SMFS	0.04±0.01	**0.36±0.08**	0.08±0.02	0.19±0.05	0.14±0.04	0.08±0.05
	ACM	0.03±0.01	0.37±0.08	0.08±0.01	0.19±0.07	0.17±0.02	0.08±0.04
	IACM	0.04±0.01	0.38±0.03	0.07±0.03	**0.18±0.09**	0.15±0.03	**0.06±0.02**

Table 3.10: Single predictions of Abdominal Pain dataset

Disease	Experts		SMFS		ACM		IACM	
	error rate	confidence	error rate	confidence	error rate	confidence	error rate	confidence
APP	0.20±0.03	89%±2%	0.19±0.04	88%±4%	**0.17±0.01**	89%±4%	0.17±0.03	**91%±1%**
CHO	0.10±0.07	89%±4%	**0.09±0.02**	89%±3%	0.09±0.07	90%±6%	**0.09±0.02**	**92%±4%**
INO	0.13±0.04	84%±7%	0.15±0.07	85%±8%	0.14±0.02	85%±5%	**0.11±0.04**	**89%±6%**
RHO	0.25±0.08	76%±3%	0.11±0.09	82%±12%	0.11±0.06	82%±14%	**0.09±0.07**	**90%±6%**
DYS	0.15±0.04	90%±3%	0.15±0.04	91%±2%	0.13±0.02	90%±4%	**0.12±0.03**	**92%±2%**

3.3.2 Results of the Applications on Other Datasets

Data description

The 4 datasets are all from UCI dataset repository to provide a wide variety of application areas, sizes, combinations of feature types, shown in Appendix A.2. There are no experts' features for these datasets. Comparison of prediction performance has been made.

Comparison of predictive performance

Feature selection is performed using NC as we did for Abdominal Pain dataset. To avoid using the same classifier for prediction, 1-Nearest Neighbour is used to provide estimated accuracies for comparison in off-line mode and conformal predictors underlying 1-Nearest Neighbour is used to make region predictions and performance of uncertain predictions are compared. We use the Nearest Neighbour algorithm because it is sensitive to irrelevant and redundant features [67]. 10-fold cross-validation are applied to each dataset for finding the feature subset in training set and making classification in test set. The same division of data into folds is used for each algorithm to ensure fair comparison. FCBF and CFS are reported to be sensitive to irrelevant and redundant features [100, 46].

Results of classification in off-line mode are shown as Table 3.11. Average accuracies and standard deviation of single predictions have been calculated. The features selected by all the methods improve classifier's predictive power. Features selected by IACM have the best performance of predictions. Results of classification in on-line mode are shown as Table 3.12. The uncertain predictions under different significance levels have been compared. Selected

features make the valid region predictions as the percentages of errors never exceed the pre-set significance level. SMFS and IACM are two feature selection method which are designed for conformal predictor. Results of SMFS method are better or at least no worse than FCBF and CFS methods. The region predictions with features selected by IACM are more efficient than the ones with features selected by SMFS. As expected features selected by IACM achieve the most efficient region predictions.

Table 3.11: Performance of Nearest Neighbour algorithms for classification

	All features	Filter		Wrapper	
Dataset	error rate	FCBF	SMFS	CFS	IACM
		error rate	error rate	error rate	error rate
Breast cancer	0.11±0.01	0.11±0.01	0.11±0.13	0.11±0.01	**0.09±0.01**
Heart disease	0.26±0.04	0.19±0.04	0.22±0.07	0.22±0.02	**0.13±0.03**
Dermatology	0.14±0.02	0.14±0.02	**0.08±0.01**	0.08±0.04	0.08±0.02
LSVT	0.08±0.06	**0.08±0.03**	0.10±0.01	0.08±0.05	**0.08±0.03**

3.3.3 Discussion

Classification algorithms work to imitate the behave of people about making decisions. The results of predictions are expected to be close to experts' opinions. As we have discussed before, the filter methods select features by the characteristics of their relevance with class labels, while the wrappers select features based on their performance of predictions for a specific classifier. In real life problems, doctors select features in the filter way because they consider symptoms to be important if the features are directly relevant to diagnostics. This is the reason why features selected by ACM are quite different from the features provided by doctors. Features selected by SMFS are

Table 3.12: Performance of CP for region predictions

Dataset	ε	all features		filters				wrappers			
		error rate	uncertain	FCBF		SMFS		CFS		IACM	
				error rate	uncertain	error rate	uncertain	error rate	uncertain	error rate	uncertain
LSVT	0.1	0±0.10	**0.83±0.13**	0.08±0.16	0.92±0.03	0±0.15	0.92±0.02	0.05±0.10	0.94±0.01	0±0.03	0.83±0.14
	0.2	0.17±0.02	0.50±0.25	0.08±0.10	0.83±0.20	0.08±0.09	0.75±0.28	0.08±0.06	0.83±0.08	0.11±0.08	**0.39±0.15**
Breast	0.1	0.03±0.04	0.36±0.07	0.04±0.02	0.38±0.10	0.06±0.01	0.31±0.08	0.04±0.04	0.35±0.07	0.04±0.03	**0.29±0.06**
cancer	0.2	0.09±0.03	0.14±0.05	0.11±0.05	0.17±0.09	0.15±0.05	0.13±0.06	0.14±0.03	0.13±0.05	0.10±0.05	**0.09±0.06**
Heart	0.1	0.05±0.03	0.70±0.16	0.07±0.02	0.72±0.26	0.09±0.01	0.73±0.24	0.06±0.03	0.75±0.19	0.09±0.03	**0.39±0.21**
disease	0.2	0.13±0.02	0.44±0.20	0.18±0.02	0.43±0.38	0.15±0.06	0.47±0.16	0.12±0.05	0.48±0.25	0.13±0.03	**0.15±0.19**
Dermatology	0.1	0.07±0.02	0.88±0.14	0.06±0.03	0.98±0.01	0.05±0.04	0.98±0.12	0.03±0.05	0.83±0.12	0.06±0.03	**0.76±0.16**
	0.2	0.11±0.04	0.59±0.19	0.12±0.07	0.76±0.21	0.11±0.10	0.60±0.37	0.08±0.10	0.57±0.28	0.12±0.05	**0.48±0.16**

much similar to doctors' selections, but they are included in correlated features, because correlated features in filter have same rankings. In the IACM method, the average confidence of true label is used instead of the average confidence of the predicted label, because the average confidence of true label represents how far away the classes are to each other. In some sense, we combine wrapper method with filter method as IACM considers the performance of predictions along with the characteristics of features. This is why features selected by IACM were more similar to experts' than the features selected by ACM and almost as good as SMFS's. Wrappers work based on the feedback of prediction performance of a particular algorithm while experts and filters are independent from classifiers. Thus, features selected by IACM had stronger predictive power.

We also chose other feature selection methods such as FCBF and CFS and other various datasets to make comparison. The two selection methods, one filter (FCBF) and one wrapper (CFS), are said to be sensitive to relevance and redundance features. As it can be seen from the experimental results, IACM method worked well during these experiments. In off-line classification, accuracies of predictions with features selected by IACM were no less than the others. In on-line classification, features selected by IACM always provided the most efficient region predictions, which achieved our expected goal.

Chapter 4

Conformal Predictors with Additional Information

Recently, Vapnik introduced a new method called "Learning using privileged information" (LUPI) which provides a learning paradigm under privileged (or additional) information [91]. It described the SVM+ technique to process this information in off-line mode. Inspired by the method, we would like to deal with additional information with conformal predictors in both off-line mode and on-line mode.

4.1 Background

4.1.1 Additional Information

Various data types lead to new types of problems and require the development of new types of techniques for analysis [44]. Learning with additional information is a very common scenario. For example, doctors normally make

diagnosis for a patient using general testing results. If the diagnosis is still unclear based on all of the results, they may send the patient for some additional testing such as blood test and MRI scan. In classroom, students could learn well with the help of teacher's comments. And for the following study he could also use this experience even if there are no teacher's comments any more. Both extra testing and teacher's comments are additional information to original data and help the learners improving their learning abilities. These kinds of additional information have a common characteristic that there is no corresponding additional information available for the new coming examples. The above description of learning with additional information can be summarized as Table 4.1 for classification in machine learning.

Table 4.1: Classification with additional information

Data Set	Content		
	Usual information	Additional information	Label
Training examples	Known	Known	Known
Test examples	Known	Unknown	To predict

4.1.2 Current Status of Learning with Additional Information

It is said that incorporation of additional information within machine learning algorithms became important, since it could help the classifier improve their classification performance [11, 26]. Traditionally such fusion of information was the domain of semi-supervised learning, where techniques such as co-training were employed in order to fuse separate data to exploit knowledge

encoded within unlabelled data [33]. Most researches in supervised learning, both theoretical and empirical, assume that the model is trained and tested using data drawn from the same distribution. They can not utilize additional information directly which is not available for testing examples.

Recently a new learning paradigm was proposed by Vapnik for supervised learning [91], called "Learning Using Privileged Informaiton" (LUPI), which aims to improve predictive performance of learning algorithms and reduce the amount of the required training data.

4.2 Learning Using Privileged Information

In these works, Vapnik gave additional information a specific description, "privileged information". Unlike traditional paradigm, the data in LUPI is said to be privileged as it is available only during training and not during testing [91]. Privileged information denotes the existence of an additional set of data that provides a higher level information, akin to information provided by a 'master' to a pupil, about a specific problem. Vapnik has shown that privileged information helped to improve the learning performance [95, 93, 91].

4.2.1 Privileged Information

To understand the problem that is to be solved in this work, a description of privileged information should be given [91]. Vapnik named traditional data (usual information) "technical data", as in most cases such data originated from a technical process, such as a pixel space in the case of a digit recognition

task or amino-acid space in the case of classification of proteins. To help us understand what privileged information is, it is useful to discuss examples where such information can become useful. Vapnik suggested three example types of privileged information [91]:

- Advanced technical model.

  The privileged information can be seen as a high level technical model of a particular problem to be solved. For example, in the field of biological information, amino-acid sequences are usually used to classify the proteins. The 3D-structure is a technical model developed by scientists. When information contained amino-acid sequences along with the 3D-structures, learning performance was improved.

- Future events.

  For a given set of current measurement, a future event or development can be provided as the privileged information. For example if the task in hand is the prediction of a particular treatment of a patient in a year's time, given his/her current symptoms, a doctor can provide information about the development of symptoms in three, six and nine month's time.

- Holistic description.

  The last example type of privileged information relates to holistic descriptions [86] of specific problems. For example, the goal is to classify biopsy images into two categories, cancer and non-cancer. The classification is usually based on the individual pixels of each image. However, along with the picture the doctor has a report about the images

by pathologists in a high-level holistic language.

The above three example types of privileged information are only a very small selection of the possible set of additional information that could be obtained from a number of problem domains. Vapnik stated that almost any machine learning problem contains some form of privileged information, which is currently not exploited in the learning process [93].

4.2.2 Learning Procedure

Data representation

Examples in training set are composed by usual object $\mathbf{x}$ from usual information, additional object $\mathbf{x}^*$ from additional information and label y, where examples in test set have $\mathbf{x}$ only. This can be summarized as:

Given a set of i.i.d. training examples,

$$Z = \{(\mathbf{x}_1, \mathbf{x}_1^*, y_1), ..., (\mathbf{x}_n, \mathbf{x}_n^*, y_n)\}, \mathbf{x}_i \in \mathbf{X}^d, \mathbf{x}_i^* \in \mathbf{X}^{*m}$$

where $\mathbf{x}_i$ represents usual information and $\mathbf{x}_i^*$ represents additional information. $\mathbf{x}$ is a d-dimensional feature vector and $\mathbf{x}^*$ is a m-dimensional feature vector. Thus, a classifier needs to be trained on training examples and applied on test examples, $\{\mathbf{x}_{n+1}, ..., \mathbf{x}_{n+l}\}$, with the different dimensional feature space.

SVM+ algorithm

To realize this advanced type of learning, Vapnik developed the SVM+ algorithm [91], which is based on SVM algorithm. Let $f(\mathbf{x}) = sign(\mathbf{w} \cdot \mathbf{x} + b)$

be a decision function and $\phi(\mathbf{x}^*) = \mathbf{w}^* \cdot \mathbf{x}^* + d$ be a correction function. The optimization problem of SVM+ is defined as:

$$\begin{gathered} \min_{\mathbf{w},b,\mathbf{w}^*,d} \frac{1}{2}||\mathbf{w}||_2^2 + \frac{\gamma}{2}||\mathbf{w}^*||_2^2 + C\sum_{i=1}^{n}(\mathbf{w}^* \cdot \mathbf{x}_i^* + d) \\ \forall 1 \leq i \leq n, y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - (\mathbf{w}^* \cdot \mathbf{x}_i^* + d) \\ \forall 1 \leq i \leq n, \mathbf{w}^* \cdot \mathbf{x}_i^* + d \geq 0 \end{gathered} \tag{4.1}$$

The objective function of SVM+ contains two hyperparameters, $C \geq 0$ and $\gamma \geq 0$. The term $\frac{\gamma}{2}||\mathbf{w}^*||_2^2$ in Equation 4.1 is intended to restrict the capacity of the function space containing ϕ. Then, we can construct the Lagrangian :

$$\begin{gathered} L(\mathbf{w}, b, \mathbf{w}^*, d, \alpha, \beta) = \frac{1}{2}||\mathbf{w}||_2^2 + \frac{\gamma}{2}||\mathbf{w}^*||_2^2 + C\sum_{i=1}^{n}[(\mathbf{w}^* \cdot \mathbf{x}_i^*) + d] \\ -\sum_{i=1}^{n}\alpha_i[y_i[(\mathbf{w} \cdot \mathbf{x}_i) + b] - 1 + [(\mathbf{w}^* \cdot \mathbf{x}_i^*) + d]] - \sum_{i=1}^{n}\beta[(\mathbf{w}^* \cdot \mathbf{x}_i^*) + d] \end{gathered} \tag{4.2}$$

to minimize it with respect of $\mathbf{w}$, b, $\mathbf{w}^*$, d and maximize with respect to Lagrange multipliers $\alpha \geq 0$, $\beta \geq 0$.

The solution of this problem is defined by the decision function

$$f(\mathbf{x}) = (\mathbf{w} \cdot \mathbf{x}) + b = \sum_{i=1}^{n} y_i \alpha_i K(\mathbf{x}_i, \mathbf{x}) + b \tag{4.3}$$

and the corresponding correcting function

$$\begin{gathered} \phi(\mathbf{x}^*) = (\mathbf{w}^* \cdot \mathbf{x}^*) + d \\ = \frac{1}{\gamma}\sum_{i=1}^{n}(\alpha_i + \beta_i - C)K^*(\mathbf{x}_i^*, \mathbf{x}^*) + d \end{gathered} \tag{4.4}$$

where $K(\mathbf{x}_i, \mathbf{x})$ and $K^*(\mathbf{x}_i^*, \mathbf{x}^*)$ are kernels in $\mathbf{X}$ and $\mathbf{X}^*$ spaces that define inner products in $\mathbf{Z}$ and $\mathbf{Z}^*$ spaces and α, β are the solution of the following

optimization problem: maximize the function:

$$\begin{aligned} R(\alpha,\beta) = \textstyle\sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i\alpha_j y_i y_j K(\mathbf{x}_i,\mathbf{x}_j) \\ -\frac{1}{2\gamma}\textstyle\sum_{i,j=1}^{n}(\alpha_i+\beta_i-C)(\alpha_j+\beta_j-C)K^*(\mathbf{x}_i^*,\mathbf{x}_j^*) \end{aligned} \tag{4.5}$$

subject to three types of contraints

$$\sum_{i=1}^{n}(\alpha_i+\beta_i-C)=0$$

$$\sum_{i=1}^{n} y_i\alpha_i = 0$$

$$\alpha_i \geq 0, \beta_i \geq 0$$

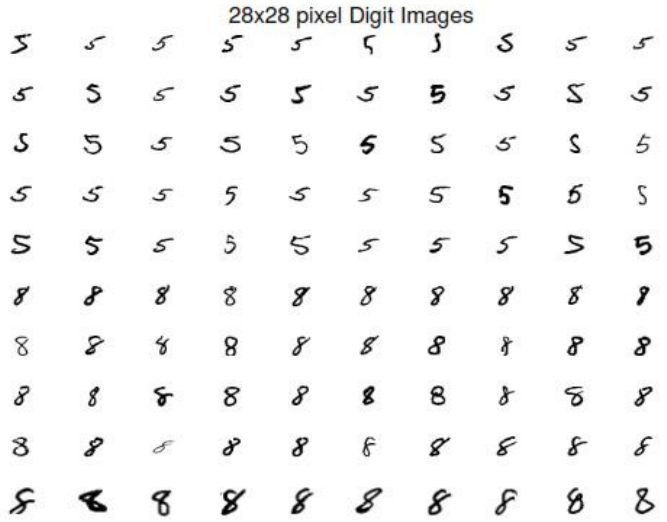
Experimental results using the above algorithm show the new paradigm has superiority in terms of performance over the original SVM method. The task of digit recognition is one of the most frequent machine learning tasks. Numerous research papers and results have been published on this topic, particularly within the supervised learning community [63]. The MNIST database of handwritten digits is an eminent source of such research [62]. Vapnik et al. used a subset of the MNIST dataset comprising 100 digits from two different groups. The two groups are 50 variations of the digit 5 and 50 variations of the digit 8. Each digit is originally a 28×28 pixel gray-scale image, shown in Figure 4.1(a). Thus an image can be thought of a 784-dimensional feature vector with values in the range of 0 to 255. To make the task of classification slightly more difficult, they created a second dataset, based on scaled down versions of the original images. A set of 100 gray-scale images at a resolution of 10×10 pixels has been created, shown in Figure 4.1(b) [33]. This dataset can again be thought of a 100-dimensional

dataset with a range of 0 to 255. Subset of the MNIST digit dataset comprising of 100 digits at two different resolutions, providing different levels of information. The lower the resolution, the more information is potentially lost. The dataset comprises two classes of digits, the numbers 5 and 8. There are 50 examples for each class.

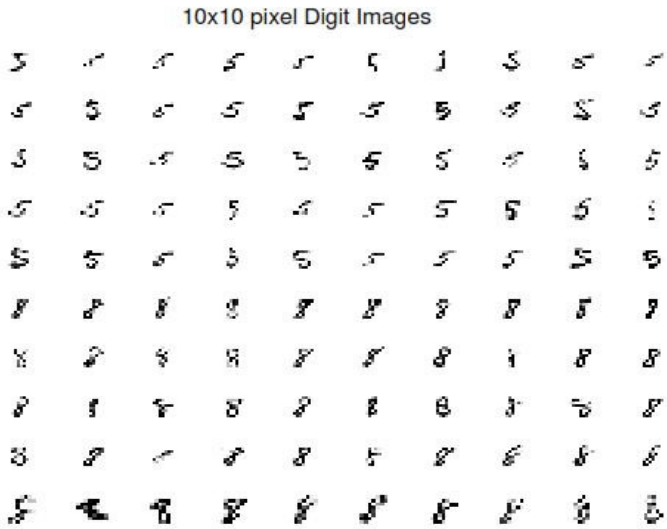
Vapnik et al. showed that a poetic description [91] of a set of images of hand-written digits provides useful information for learning. A set of poetic descriptions is obtained with the help of a language expert. By poetic description we mean a description of what the expert saw and interpreted using his/her own words in the form of a poem. To make these additional sets of data usable in computation, the text has been analysed and a set of keywords that occur across the dataset have been extracted. Finally a 21-dimensional vector has been created for each digit encoding the poetic description of each digit [91].

We repeat the experiment and the results are presented in Table 4.2. The kernel to be used is Radial Basis Function(RBF) Kernel, $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma||\mathbf{x}_i - \mathbf{x}_j||^2)$, $\gamma \geq 0$. By using cross-validation, we could find the best parameter C and γ. Vapnik et al. [91] did not provide what kernel and the corresponding parameters they used in the experiments. Given this, the results we obtained are quiet similar to [91].

An important strength of the SVM+ algorithm is the ability of the system to reject privileged information in situations when similarity measures in the correcting space are not appropriate, thus privileged information is only used when it is deemed beneficial. But it has some drawbacks. One drawback of the system is the increase in computational requirements due to the necessity



(a) 28×28 pixel digit dataset



(b) 10×10 pixel digit dataset

Figure 4.1: Subset of the MNIST digit dataset comprising of 100 digits at two different resolutions, providing different levels of information

Table 4.2: Error rate of SVM and SVM+ on the digit recognition task

Training data size	γ	C	Error rate by SVM	Error rate by SVM+
40	2^{-13}	2^{-10}	0.17	0.14
50	2^{10}	2^{-5}	0.13	0.12
60	2^{11}	2^{-1}	0.13	0.12
70	2^{-12}	2^{4}	0.13	0.11
80	2^{11}	2^{-7}	0.11	0.10
90	2^{8}	2^{-14}	0.11	0.08

of tuning more parameters than in the original SVM setting [93].

4.3 Conformal Predictors with Additional Information

In this section, we propose a novel method that incorporates additional information within conformal predictors. Like other traditional machine learning algorithms, original CP framework works on the dataset where training examples and test examples are from the same distribution. As CP is a transductive method while making predictions based on hypothesis testing, it can be successfully adapted to learning with additional information.

4.3.1 On-line Learning and Off-line Learning

In Chapter 2 we presented conformal predictors and applied them to on-line mode. The theory can also be extend to relaxation of the on-line protocol that makes it close to off-line mode [92]. This is important because we

can not always rely on a teacher to reveal correct labels immediately in most practical problems. For example, if we deal with hand-written digits recognizing problem, we can not rely on a teacher to tell us the correct interpretation of each hand-written digit.

4.3.2 Conformal Predictions with Additional Information

When we use CP method, we predict that a new object will have a label that makes it similar to the old examples in some specific way. When learning with additional information, we assume the additional information are categorized and finite and we can make prediction based on how strange a new object with a hypothetical label and hypothetical additional values are compared to old examples.

$$\alpha_i = A(\wr(\mathbf{x}_1, \mathbf{x}_1^*, y_1), ..., (\mathbf{x}_{i-1}, \mathbf{x}_{i-1}^*, y_{i-1}), (\mathbf{x}_{i+1}, \mathbf{x}_{i+1}^*, y_{i+1}), ..., (\mathbf{x}_{n+1}, \mathbf{x}^*, y)\wr, (\mathbf{x}_i, \mathbf{x}_i^*, y_i))$$

$$\alpha_{n+1} = A(\wr(\mathbf{x}_1, \mathbf{x}_1^*, y_1), \ldots, (\mathbf{x}_n, \mathbf{x}_n^*, y_n), (\mathbf{x}_{n+1}, \mathbf{x}^*, y)\wr, (\mathbf{x}_{n+1}, \mathbf{x}^*, y)), y \in Y, \mathbf{x}_i^* \in \mathbf{X}^{*m}$$

Then,

$$p(y, \mathbf{x}^*) = \frac{\#\{i = 1, ..., n+1 : \alpha_i \geq \alpha_{n+1}\}}{n+1}, y \in Y, \mathbf{x}^* \in \mathbf{X}^{*m}$$

Like p-values in conformal predictors, the more likely the hypothesis is, the higher p-value $p(y, \mathbf{x}^*)$ is. The approach assumed that the additional information must be categorical and finite, so that we can have finite combination of the label and additional values to analyse. If the values are continuous, some discretization methods should be used to transform continuous values to nominal.

Learning in off-line mode

In off-line mode, the new test example will not be included in training set. Conformal predictors can be applied in off-line mode, but can not have a guarantee of validity. In CP, suppose we are given a training set $\{\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_n\}$ of examples $\mathbf{z}_i = (\mathbf{x}_i, y_i)$ and the problem is to predict the labels y_{n+1} for new example $\mathbf{x}_{n+1}$. The off-line conformal predictor outputs the region prediction:

$$\Gamma^{\varepsilon}(\mathbf{x}_1, y_1, ..., \mathbf{x}_n, y_n, \mathbf{x}_{n+1}) := \{y \in Y : \frac{|\{j = 1, \ldots, n+1 : \alpha_j \geq \alpha_{n+1}\}|}{n+1} > \varepsilon\} \tag{4.6}$$

where the nonconformity scores are computed from a nonconformity measure A:

$$\alpha_j = A(\wr \mathbf{z}_1, ..., \mathbf{z}_{j-1}, \mathbf{z}_{j+1}, \ldots, \mathbf{z}_n, (\mathbf{x}_{n+1}, y) \int, \mathbf{z}_j) \tag{4.7}$$

$$\alpha_{n+1} = A(\wr \mathbf{z}_1, ..., \mathbf{z}_n \int, (\mathbf{x}_{n+1}, y)) \tag{4.8}$$

It is true that

$$P\{y_{n+1} \notin \Gamma^{\varepsilon}(\mathbf{x}_1, y_1, ..., \mathbf{x}_n, y_n, \mathbf{x}_{n+1})\} \leq \varepsilon \tag{4.9}$$

for every example in test set which are drawn independently from the distribution Q, but the events in Equation 4.9 are not independent and

$$\frac{|\{m = n+1, ..., n+k : y_m \notin \Gamma^{\varepsilon}(\mathbf{x}_1, y_1, ..., \mathbf{x}_n, y_n, \mathbf{x}_m)\}|}{k} \tag{4.10}$$

can be significantly above ε even when k is very large. The validity is not kept. Thus, in off-line mode, only single prediction is considered.

Algorithm 6 Conformal Prediction with Additional Information in Off-line Mode

Require: training example sequence $\{\mathbf{z}_1 = (\mathbf{x}_1, \mathbf{x}_1^*, y_1), \ldots, \mathbf{z}_n = (\mathbf{x}_n, \mathbf{x}_n^*, y_n)\}$

Require: new example $\mathbf{x}_{n+1}$

Require: conformity measure A

Require: significance level ε

for $y \in Y$ **do**

for $\mathbf{x}^* \in \mathbf{X}^*$ **do**

$\mathbf{z}_{n+1} = (\mathbf{x}_{n+1}, \mathbf{x}^*, y)$

for $i = 1 : n$ **do**

$\alpha_i = A(\lbag \mathbf{z}_1, ..., \mathbf{z}_{i-1}, \mathbf{z}_{i+1}, \ldots, \mathbf{z}_n, (\mathbf{x}_{n+1}, y) \rbag, \mathbf{z}_i)$

end for

$\alpha_{n+1} = A(\lbag \mathbf{z}_1, ..., \mathbf{z}_n \rbag, (\mathbf{x}_{n+1}, y))$

$p(y, \mathbf{x}^*) = \frac{\#\{i=1,\ldots,n+1 : \alpha_i \geq \alpha_{n+1}\}}{n+1}$

end for

$p(y) = \max_{\mathbf{x}^*} p(y, \mathbf{x}^*)$

end for

return single prediction $\hat{y}_{n+1} = \arg\max_y p(y)$

return region prediction $\Gamma^{\varepsilon}_{n+1} = \{y : p(y) > \varepsilon\}$

Learning in on-line mode

In on-line mode, the advantage of CP is its validity. We assume that once the classifier made the predictions, the true label and additional values of the new object $\mathbf{x}$ are revealed. The challenge of our method is how to combine a number of extended p-values $p(y, \mathbf{x}^*)$ into $p(y)$ and to maintain the validity

property.

Since only one of the hypotheses is true, selecting the maximum of the extended p-values is the only way to hold the validity:

$$\max_{\mathbf{x}^*} p(y, \mathbf{x}^*) \geq p(y, \mathbf{x}_{true}), y \in Y, \mathbf{x}^* \in \mathbf{X}^*$$

Thus:

$$Prob\{\max_{\mathbf{x}^*} p(y, \mathbf{x}^*) \leq \varepsilon\} \leq Prob\{p(y, \mathbf{x}_{true}) \leq \varepsilon\}$$

So:

$$Prob\{\max_{\mathbf{x}^*} p(y, \mathbf{x}^*) \leq \varepsilon\} \leq \varepsilon$$

Region prediction:

$$\Gamma^{\varepsilon} = \{y : p(y) > \varepsilon\}$$

The procedure of conformal prediction with additional information can be summarized as Algorithm 7.

Algorithm 7 Conformal Prediction with Additional Information in On-line Mode

Require: training example sequence $\{\mathbf{z}_1 = (\mathbf{x}_1, \mathbf{x}_1^*, y_1), \ldots, \mathbf{z}_n = (\mathbf{x}_n, \mathbf{x}_n^*, y_n)\}$

Require: new example $\mathbf{x}_{n+1}$

Require: conformity measure A

Require: significance level ε

for $y \in Y$ **do**

for $\mathbf{x}^* \in \mathbf{X}^*$ **do**

$\mathbf{z}_{n+1} = (\mathbf{x}_{n+1}, \mathbf{x}^*, y)$

for $i = 1 : n+1$ **do**

$\alpha_i = A(\wr \mathbf{z}_1, ..., \mathbf{z}_{i-1}, \mathbf{z}_{i+1}, \ldots, \mathbf{z}_n, (\mathbf{x}_{n+1}, y) \int, \mathbf{z}_i)$

end for

$p(y, \mathbf{x}^*) = \frac{\#\{i=1,\ldots,n+1 : \alpha_i \geq \alpha_{n+1}\}}{n+1}$

end for

$p(y) = \max_{\mathbf{x}^*} p(y, \mathbf{x}^*)$

end for

return single prediction $\hat{y}_{n+1} = \arg\max p(y)$

return region prediction $\Gamma^{\varepsilon}_{n+1} = \{y : p(y) > \varepsilon\}$

4.4 Results and Discussion

The algorithm is firstly applied on the Abdominal Pain dataset for binary classification. Our results in off-line mode are compared with the results of classifications based on SVM+ method. Only CP method is applied to make predictions in on-line mode, since SVM+ is designed in off-line mode.

To verify that our method is suitable for multiple-class classification problems, it is applied on the Dermatology data. Classifications are implemented in both off-line mode and on-line mode.

4.4.1 Results of the Applications for Abdominal Pain Diagnosis

Among 9 diseases in Abdominal Pain dataset, we only make predictions for APP and DYS disease groups for illustration purposes. APP and DYS are two diseases which have the largest number of patients. Because we use one-against-rest method to transfer multiple class problem into a set of binary classes, this could lead to the situation where there are unbalanced examples in the two classes. And this will affect the performance of classifications by SVM and SVM+. For each disease, the features identified by medical experts are used as additional features and the rest are considered as usual features. In the experiments, we varied the number of usual features, from 30 to 120, to provide different predictive power of usual features. For each trial, usual features are randomly chosen.

The implementation of conformal predictor is based on Nearest Neighbour algorithm for classification is performed in both off-line and on-line mode.

Off-line mode

Results of classification on Abdominal Pain dataset in off-line mode by CP method, SVM and SVM+ method are shown in Figures 4.2, 4.4. 4.3, and 4.5. The experiment is repeated 10 times, and new features are selected as usual features every time. The accuracy of predictions for APP increased as the

size of usual features growing. The accuracy of predictions for DYS disease raised at the beginning and started falling when it reached the highest point. The changes of accuracy of predictions with additional information always match the changes of accuracy of predictions without additional information. Sometimes the accuracy of predictions with additional information slightly falls when the other one keeps rising. This is caused by the situation that some of additional features are correlated with some usual features. The intervals between the accuracy with the help of additional information and the other is much larger at the beginning when the size of usual features is smaller than when the size of usual features is big at the end. In these experiments, both CP method and SVM+ successfully utilize additional information for better classification performance, compared to the classification without additional information.

Figure 4.2: Results of classification for APP disease by CP methods

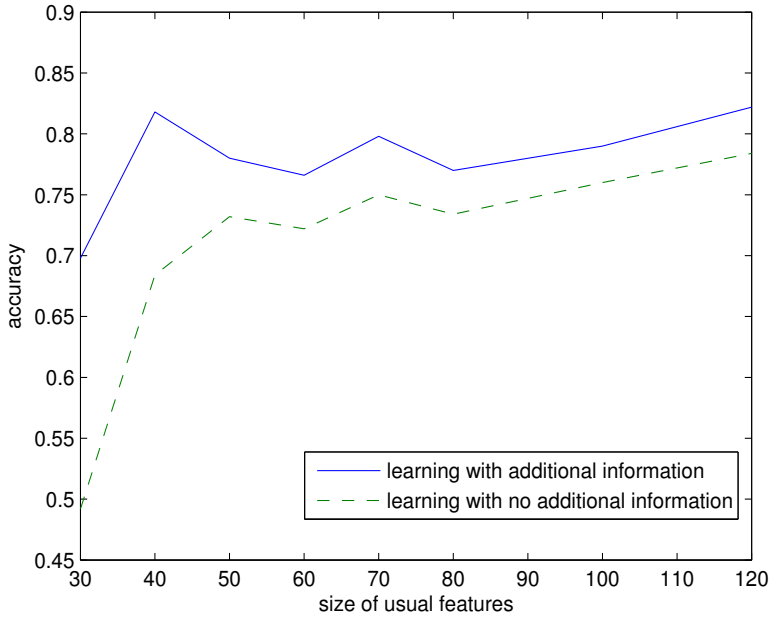


Figure 4.3: Results of classification for APP disease by SVM and SVM+

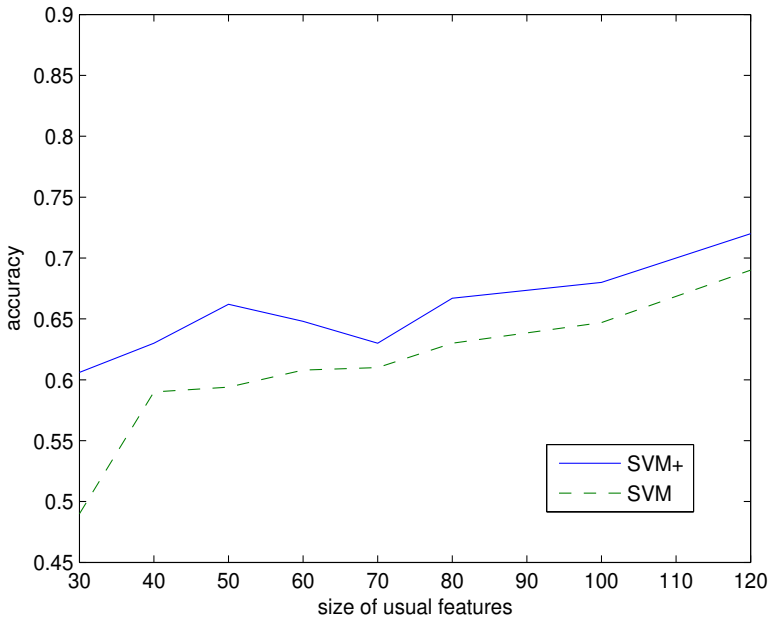


Figure 4.4: Results of classification for DYS disease by CP methods

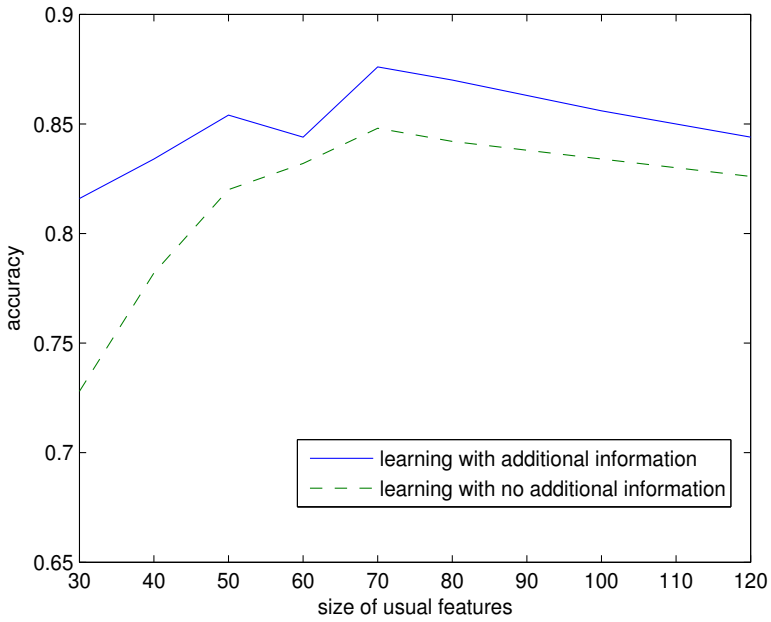
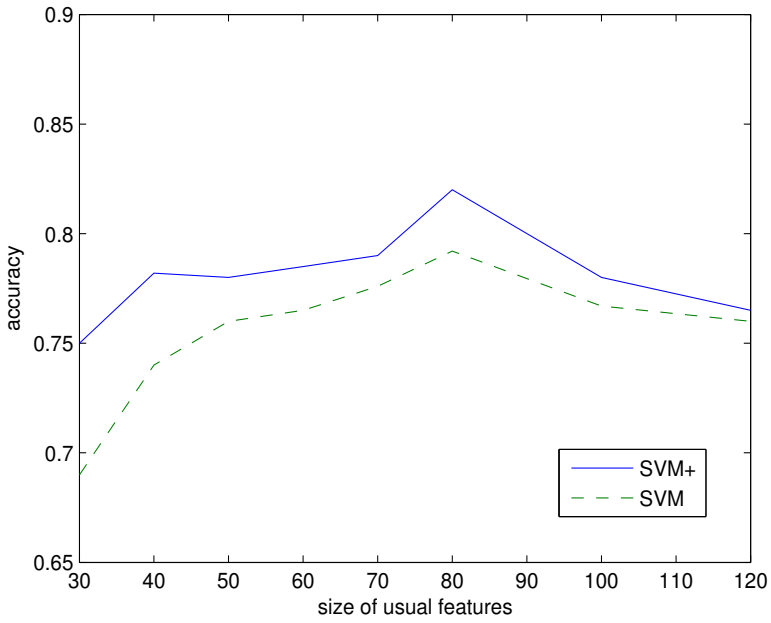


Figure 4.5: Results of classification for DYS disease by SVM and SVM+



On-line mode

Results of classification by conformal predictor in on-line mode are shown as Table 4.3. When the size of usual features is small, the percentages of errors of region predictions with usual feature exceed the corresponding significance level, while region predictions with additional information do not. In the case when $\varepsilon = 0.2$ the region predictions with additional information have smaller size than the one with usual information only; in other cases the size of predictions with additional information is no bigger than the other.

Table 4.3: Region predictions of Abdominal Pain dataset in on-line mode

Disease	Size of	Additional	$\varepsilon = 0.05$		$\varepsilon = 0.10$		$\varepsilon = 0.20$	
	usual features	information		uncertain	error	uncertain	error	uncertain
APP	30	Yes	0.04±0.01	0.49±0.15	0.05±0.02	0.46±0.07	0.19±0.02	0.12±0.04
		No	0.43±0.12	0.10±0.42	0.38±0.11	0.10±0.21	0.26±0.13	0±0.18
	40	Yes	0.05±0.01	0.71±0.05	0.10±0.01	0.28±0.03	0.16±0.03	**0.05±0.03**
		No	0.14±0.15	0.61±0.03	0.19±0.06	0.18±0.17	0.19±0.01	0.18±0.09
	50	Yes	0.04±0.02	0.71±0.05	0.08±0.01	0.43±0.06	0.18±0.02	**0.04±0.05**
		No	0.06±0.02	0.71±0.08	0.10±0.03	**0.39±0.05**	0.19±0.03	0.19±0.06
	60	Yes	0.03±0.01	0.57±0.07	0.08±0.02	0.35±0.11	0.15±0.04	**0.01±0.05**
		No	0.04±0.02	0.57±0.05	0.07±0.02	0.35±0.14	0.17±0.03	0.19±0.04
	70	Yes	0.02±0.02	0.58±0.06	0.06±0.01	0.33±0.09	0.15±0.01	**0.01±0.05**
		No	0.03±0.02	0.59±0.09	0.10±0.01	0.33±0.15	0.16±0.04	0.13±0.06
	80	Yes	0.03±0.02	0.57±0.07	0.07±0.03	0.33±0.05	0.12±0.03	**0.01±0.04**
		No	0.05±0.01	0.57±0.07	0.08±0.01	0.33±0.04	0.14±0.05	0.14±0.02
	100	Yes	0.03±0.01	0.48±0.04	0.07±0.01	0.25±0.02	0.17±0.02	**0.01±0.05**
		No	0.05±0.01	0.48±0.04	0.06±0.04	0.25±0.02	0.16±0.02	0.09±0.02
	120	Yes	0.03±0.01	0.40±0.03	0.06±0.02	0.20±0.04	0.12±0.01	**0.01±0.02**
		No	0.04±0.01	0.40±0.03	0.06±0.02	0.20±0.03	0.13±0.06	0.07±0.02
DYS	30	Yes	0.05±0.01	0.25±0.14	0.06±0.04	0.22±0.09	0.20±0.01	0.01±0.10
		No	0.21±0.17	0.08±0.05	0.21±0.13	0.09±0.08	0.32±0.0	0±0.03
	40	Yes	0.04±0.01	0.40±0.04	0.08±0.02	0.16±0.07	0.18±0.02	**0.02±0.07**
		No	0.09±0.03	0.36±0.06	0.12±0.01	0.12±0.05	0.17±0.06	0.07±0.05
	50	Yes	0.03±0.02	0.45±0.05	0.08±0.01	0.18±0.04	0.12±0.03	**0.01±0.05**
		No	0.05±0.02	0.45±0.07	0.08±0.02	0.18±0.07	0.10±0.01	0.11±0.03
	60	Yes	0.04±0.01	0.33±0.06	0.07±0.02	0.18±0.06	0.15±0.01	**0.05±0.03**
		No	0.04±0.01	0.33±0.06	0.07±0.01	0.18±0.06	0.18±0.06	0.07±0.05
	70	Yes	0.03±0.02	0.35±0.05	0.06±0.01	0.19±0.03	0.13±0.04	**0.07±0.01**
		No	0.03±0.02	0.35±0.05	0.06±0.01	0.19±0.03	0.17±0.05	0.08±0.04
	80	Yes	0.03±0.01	0.36±0.02	0.06±0.02	0.19±0.04	0.12±0.04	**0.05±0.02**
		No	0.03±0.01	0.36±0.02	0.06±0.02	0.19±0.04	0.13±0.03	0.07±0.03
	100	Yes	0.02±0.02	0.33±0.04	0.07±0.01	0.16±0.03	0.15±0.02	**0.05±0.01**
		No	0.02±0.02	0.33±0.04	0.07±0.01	0.16±0.03	0.15±0.01	0.07±0.03
	120	Yes	0.02±0.01	0.30±0.04	0.09±0.01	0.11±0.05	0.13±0.02	**0.01±0.03**
		No	0.02±0.01	0.30±0.04	0.09±0.01	0.11±0.05	0.14±0.03	0.05±0.02

4.4.2 Results of the Applications for Dermatology Diagnosis

We apply the CP method on this dataset to demonstrate the advantage of our method for multiple-class classification. SVM+ just works for binary classification, so there is no comparison here.

The dataset contains 34 features and 6 classes. The diseases in this data are psoriasis, seboreic dermatitis, lichen planus, pityriasis rosea, cronic dermatitis and pityriasis rubra pilaris. Patients were first evaluated clinically with 12 features. Afterwards, skin samples were taken for the evaluation of 22 histopathological features. The values of the histopathological features are determined by an analysis of the samples under a microscope.

In the dataset constructed for this domain, the family history feature has the value 1 if any of these diseases has been observed in the family, and 0 otherwise. The age feature simply represents the age of the patient. Every other feature (clinical and histopathological) was given a degree in the range of 0 to 3, where, 0 indicates that the feature was not present, 3 indicates the largest amount possible, and 1, 2 indicate the relative intermediate values. In Chapter 2, we selected a relevant feature subset for this dataset.

In this experiment, we assume these selected features are additional and the unselected ones are usual. We increase the number of usual features, from 5 to 20, to provide different predictive power of usual features. For each trial, usual features are randomly chosen. Conformal predictor underlying 1-Nearest Neighbour algorithm is used for classification in both off-line and on-line mode. 10-fold cross-validation is used to divide the dataset into training set and test set to reduce the classification bias.

Off-line mode

Results are shown by Figure 4.6. The accuracy of classification with additional information is much higher than the accuracy of classification without one. And the increasing of accuracy of predictions with additional information keeps in line with the increasing of the other one.

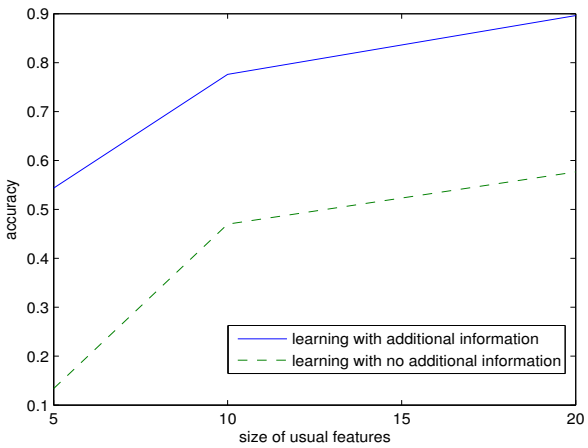


Figure 4.6: Results of predictions on Dermatology dataset in off-line mode

On-line mode

Results of classification in on-line mode are shown as Table 4.4. With the help of additional information, the percentages of errors in region predictions never exceed the corresponding significance level, and the size of predictions are always smaller.

Table 4.4: Region predictions of Dermatology dataset in on-line mode

Size of usual features	Additional information	$\varepsilon = 0.05$		$\varepsilon = 0.10$		$\varepsilon = 0.20$	
		error	uncertain	error	uncertain	error	uncertain
5	Yes	0±0	1±0	0.05±0.01	0.79±0.06	0.11±0.01	**0.57±0.03**
	No	0±0.04	1±0	0.55±0.06	0.53±0.14	0.66±0.05	0±0
10	Yes	0.02±0.01	0.84±0.11	0.09±0.04	0.41±0.01	0.19±0.04	**0.06±0.02**
	No	0.03±0.02	0.84±0.12	0.11±0.04	0.42±0.02	0.16±0.05	0.22±0.03
20	Yes	0.05±0.01	0.23±0.02	0.09±0.03	**0.04±0.02**	0.17±0.06	**0.01±0.01**
	No	0.05±0.01	0.23±0.02	0.07±0.02	0.13±0.02	0.19±0.06	0.12±0.01

4.4.3 Discussion

Learning with additional information, which is only available in training set, is a challenge for traditional machine learning system. Based on SVM algorithm, Vapnik developed SVM+ method to realize it. In SVM+, the additional information is used to generate a correction function to help find a better hyperplane for separating. We repeated Vapnik's experiments for hand-written digits recognition. Results showed SVM+ utilized additional information to improve the performance of predictions. We are interested in incorporating additional information within conformal predictors. Based on the rationale of conformal predictions, the new method treat the additional information as unknown label and calculate the level of conformity of the example with hypothetical additional information and class label are compared to old examples to make prediction. It assumes that the additional information is discrete and finite.

In the experiments, the method was applied in both off-line and on-line mode, for binary classification and multiple-class classification. In off-line mode, the method utilized the additional information for more accurate pre-

dictions. Results showed that additional information provide more help when usual features do not carry enough information for separating and if additional information was correlated with usual features, utilizing this information may not have too much advantages. Thus, the performance of learning with additional information can be affected by its quality. In on-line mode, it had been proved that the validity is kept. Furthermore, when region predictions with usual features only were not valid, the predictions with additional information can still hold the validity. With the help of additional information, the efficiency of region predictions had been improved.

Our method successfully incorporates additional information within conformal predictors to improve the predictive performance for binary and multiple-class classification in both on-line and off-line mode.

Chapter 5

Conformal Predictors with Missing Information

In many practical applications, some feature values in a dataset may be missing and learning algorithms should be able to deal with such missing information. However, most existing algorithms are designed under the assumption that there are no missing values in datasets. Traditional ways deal with features with missing values in the pre-processing step, either ignoring them or imputing them. In this chapter, we propose a novel method which embeds missing information imputation method in the conformal predictor for directly making classification.

5.1 Background

5.1.1 Missing Information

Incomplete data is an unavoidable problem in dealing with most of the real world data sources. In this chapter, we address the problem when missing values only exist in test set. Additional information for training examples discussed in Chapter 3 could be considered as missing values in test examples. The algorithm we used for dealing with missing values is similar to Algorithm 6. The difference between additional information and missing information is that the additional information is usually related to a fixed subset of features, but missing information could happen to any features.

Data quality is a major concern in machine learning, as even a small amount of missing information in the features can cause serious problems and may lead to wrong conclusions [88]. Little and Rubin discussed three types of missing values and their underlying mechanisms [66]:

- Missing completely at random (MCAR). It occurs when the probability of an example having a missing value for an attribute does not depend on either the known values or the missing data.

- Missing at random (MAR). When the probability of an example having a missing value for an attribute may depend on the known values, but not on the value of the missing data itself.

- No missing at random (NMAR). When the probability of an example having a missing value for an attribute could depend on the value of that attribute.

In case of the MCAR mode, the assumption is that the distributions of missing and complete data are the same, while for MAR mode they are different, and the missing data can be predicted by using the complete data [66]. MAR mechanism is assumed by most of the existing missing data imputation methods [65]. Furthermore, considering these three mechanisms, it is only in the MCAR case where the analysis of the remaining complete data could give a valid inference (classification) due to the assumption of same distributions. In the case where the underlying mechanism is unknown, the user can perform a statistical test introduced by Chen and Little to determine whether the missing values were introduced as MCAR [20].

5.1.2 Existing Methods of Treating Features with Missing Values

In general, missing data treatment methods can be divided into the following two categories, ignoring/discarding method and imputation method [66].

Ignoring / Discarding methods

Ignoring is the most simple and low cost solution of handling missing value. However, this method is practical only when the dataset contains relatively small number of examples with missing values [36]. There are two main ways to discard data with missing values. Many researchers studied these two methods and used different notions [97, 40, 88, 16]. In this chapter we use Ignoring for deleting examples and Discarding for deleting attributes. The first one is to completely ignore examples with missing values. The key disadvantage of this method is the waste of the training data. The second

method is known as discarding attributes. This method first determines the extent of missing values on each attribute, and then deletes the attributes with high level of missing values. Before deleting any attribute, it is necessary to evaluate its relevance to the classification. Unfortunately, relevant attributes should be kept even though they have high degree of missing values [16]. Both methods should be applied only if missing data are missing completely at random (MCAR) where the features are independent from each other.

Imputation methods

Imputation is a class of procedures that aims to fill in the missing values with estimated ones. Single imputation is the filling in of a single value for each missing observation. However, it has two disadvantages, 1) imputing a single value does not capture the sample variability, 2) there is uncertainty associated with the model used for imputation. Multiple imputation procedures suggest multiple, usually likelihood ordered, choices for each missing value. They are computationally more expensive compared to single imputation procedures, but are not associated with the two drawbacks mentioned above. Simultaneous imputation is the filling in of missing values for multiple features in an example at the same time [36].

Imputation methods are traditionally developed based on statistical algorithms [36]. Statistical methods range from simple data driven methods such as mean imputation to complex model based methods that perform parameter estimation, such as likelihood based imputation [36].

Mean imputation is the most common imputation method which replaces

missing values of discrete attributes by the most common value (i.e. the mode value), and missing values of continuous attributes by their average value (i.e. the mean value [36]). Hot-decking imputation [36, 16] is the simplest statistical imputation method. In the method, the missing value is filled in with a value from an estimated distribution for the missing value from the current data. Hot-decking is typically implemented in two stages. In the first stage, the data are partitioned into clusters. In the second stage, each example with missing data is associated with one cluster. The complete cases in a cluster are used to fill in the missing values. This can be done by calculating the mean or mode of the attribute within a cluster. Imputation with parameter estimation is based on non-missing attributes as well. Maximum likelihood procedures are used to estimate the parameters of a model defined for the complete data. In the presence of missing data maximum likelihood procedures use variants of the expectation-maximization algorithm for parameter estimation [30].

In recent years, machine learning algorithms have been employed to develop imputation methods [65, 35]. Machine learning algorithms are applied to find a predictive model to produce values that will substitute the missing data. The methods assume the missing values are missing at random (MAR) [65]. Several different types of machine learning algorithms were used, such as decision trees [65, 37], probabilistic methods [21, 34], and rule-based methods [41], however the underlying methodology was the same. One of the most recent developments was a missing data imputation framework that was developed to improve the quality of imputation methods [35]. This framework serves as a wrapper that can be applied with most existing im-

putation methods (referred to as base methods) to improve their accuracy of imputation while preserving the asymptotic computational complexity of the base method.

Imputation using k-Nearest Neighbour algorithm is the most basic and widely used machine learning method [16, 88]. Given an example $\mathbf{x}$ with missing values, this method selects the k closest cases that do not have missing values for the attributes to be imputed. The basic version of the k-nearest neighbour algorithm assumes that all instances correspond to points in the n-dimensional space of $\mathbf{X}$. The nearest neighbours of an example are often defined in terms of the standard Euclidean distance. The example $\mathbf{x}$ can be described by the feature vector $(x_1, x_2, ..., x_n)$ where x_r denotes the value of the rth feature of the example $\mathbf{x}$. In k-Nearest Neighbour learning the concept function can be either discrete-valued or real-valued [16]. Let us first consider learning discrete-valued concept functions of the form $f : \mathbf{X} \rightarrow Y$, where Y is the finite set $\{y_1, ..., y_s\}$ of class values. To estimate missing value with kNN, consider $\mathbf{x}^q$ represents an example having missing value on the qth input feature. Before making prediction, $\mathbf{x}_q$ is treated as an unknown label and then looking for the k examples that are nearest to $\mathbf{x}_q$. Let $\mathbf{x}_1, ..., \mathbf{x}_k$ denote the k examples that are nearest to $\mathbf{x}^q$,

$$\hat{y}_q = \max_{y \in Y} \sum_{i=1}^{k} \delta(y, y_i)$$

where $\delta(y, y_i) = 1$ if $y = y_i$ and $\delta(y, y_i) = 0$ otherwise.

The value $\hat{y}_q$ returned by this function as its estimate of y_q is just the mode of the true concept function f among k training examples nearest to $\mathbf{x}^q$. The kNN algorithm can be adapted to approximating real-valued concept functions [16]. To accomplish this, we have the algorithm calculating the

mean value of the k nearest examples, rather than calculating their most common value. More precisely, to approximate a real-value target function $f : \mathbf{X} \rightarrow R$ we replace the function by

$$\hat{y}_q = \frac{\sum_{i=1}^{k} y_i}{k}.$$

5.1.3 Issues in Handling Missing Value Methods

There are several techniques to address the problem of missing information, but no one is absolutely better than the others. Different solutions are needed for different situations. The choice between the different approaches discussed above largely depends upon the nature and quantity of the available data, the intended use of the data, and the underlying mechanism of the missing data problem [65].

As mentioned earlier, ignoring/discarding method is feasible only in situations where these examples with missing values constitute a negligible percentage of the total data, and no significant bias is introduced by their elimination. Imputation method needed to employ the relationships that can be identified in the valid values of the dataset to assist in estimating the missing values. Mean or mode imputation method is usually regarded as inadequate, because the standard deviation of the sample is underestimated even when data are MCAR [41]. Imputation with the prediction model approach requires that there is correlation among the attributes. If there are no relationships among attributes in the dataset, then the model will not be precise for estimating missing values. Furthermore, the distributions of missing and complete data could be different and this may lead to the generation of an incorrect classifier.

5.2 Conformal Predictors with Missing Information

In this work, we propose a novel method to deal with missing values. The method is designed to embed the imputation of missing value in Conformal Predictors for the cases when missing values are in test set. The imputation procedure will be treated as a part of classification. Therefore it does not need to train anther classifier for producing substitutes or make any assumptions on the mechanism of the missing values.

5.2.1 Data Representation

Examples in training set are composed by object $\mathbf{x}$ and label y. Objects of examples in test set have non-missing part $\mathbf{x}^{*1}$ and missing part $\mathbf{x}^{*2}$.

The problem with missing values in test set can be summarized as following:

Given a set of i.i.d. training examples,

$$\mathbf{Z} = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)\}, \mathbf{x}_i \in \mathbf{X}^d, y_i \in Y$$

where $\mathbf{x}_i$ is a d-dimensional vector. The problem is to map the test example $\mathbf{x}_s^{*1}$ with missing attributes $\mathbf{x}_s^{*2}$ to a predicted label $\hat{y}_s$, where $\mathbf{x}_s^{*1} \in \mathbf{X}^p$, $\mathbf{x}_s^{*2} \in \mathbf{X}^m$, $\mathbf{X}^p \bigcup \mathbf{X}^m = \mathbf{X}^d$, $p + m = d$ and $1 \leq m, p < d$. We will extend conformal predictors to solve it in the following section.

5.2.2 Conformal Predictions with Missing Information in Off-line Mode

The prediction of conformal predictors is based on the hypothesis testing. Based on the rationale of CP, if we know a set of values containing the missing value, we could calculate the probability of each hypothesis value. When making predictions for a test example $\mathbf{x}_s^{*1}$, firstly, the attributes which are missing are treated as target attributes $\mathbf{X}^{*2}$. According to the target attributes values in the training set, we could obtain a set of candidate values for each missing attribute in the testing examples. Then, each candidate value is combined with a label and the nonconformity score about how similar the example with a combination $(\mathbf{x}_s^{*1}, \mathbf{x}^{*2}, y)$ is to the entire training examples will be calculated as α_s, shown in Equation 5.1. For each example in training set, its nonconformity score is computed as α_i, $i = 1, ..., n$, shown as Equation 5.2. Finally, extended p-values for all candidates $(y, \mathbf{x}^{*2})$ are calculated to show the relevance of the example $\mathbf{x}_s$ with hypothesis values to the training examples.

$$\alpha_s = A(\wr \mathbf{z}_1, ..., \mathbf{z}_n \int, (\mathbf{x}_s^{*1}, \mathbf{x}^{*2}, y)) \tag{5.1}$$

$$\alpha_i = A(\wr \mathbf{z}_1, ..., \mathbf{z}_{i-1}, \mathbf{z}_{i+1}, \ldots, \mathbf{z}_n, (\mathbf{x}_s^{*1}, \mathbf{x}^{*2}, y) \int, \mathbf{z}_i) \tag{5.2}$$

$$p(y, \mathbf{x}^{*2}) = \frac{\#\{i = 1, ..., n, s : \alpha_i \geq \alpha_s\}}{n+1}, y \in Y, \mathbf{x}^{*2} \in \mathbf{X} \tag{5.3}$$

The procedure of conformal predictions with missing information in off-line mode can be summarized as Algorithm 8. The algorithm assumes the target attribute values must be finite and discrete. The more target attribute

values we have, the more complex the combination will be. The complexity will affect the speed of the processing.

Algorithm 8 Learning with Missing Information

Require: training example sequence $\{\mathbf{z}_1 = (\mathbf{x}_1, y_1), ..., \mathbf{z}_n = (\mathbf{x}_n, y_n)\}$

Require: new example $\mathbf{x}_s = (\mathbf{x}_s^{*1})$

Require: nonconformity measure A

$\mathbf{X}^{*2} = \mathbf{X}/\mathbf{X}_s^{*1}$

for $y \in Y$ **do**

for $\mathbf{x}^{*2} \in \mathbf{X}^{*2}$ **do**

$\mathbf{z}_s = (\mathbf{x}_s^{*1}, \mathbf{x}^{*2}, y)$

for $i = 1 : n$ **do**

$\alpha_i = A(\wr \mathbf{z}_1, \ldots, \mathbf{z}_{i-1}, \mathbf{z}_{i+1}, \ldots, \mathbf{z}_n, (\mathbf{x}_s^{*1}, \mathbf{x}^{*2}, y) \int, \mathbf{z}_i)$

end for

$\alpha_s = A(\wr \mathbf{z}_1, \ldots, \mathbf{z}_n \int, (\mathbf{x}_s^{*1}, \mathbf{x}^{*2}, y))$

$p(y, \mathbf{x}^{*2}) = \frac{\#\{i=1,...,n,s:\alpha_i \geq \alpha_s\}}{n+1}$

end for

end for

return single prediction $\hat{y}_s = \arg\max_{\{\mathbf{x}^{*2}, y\}} p(y, \mathbf{x}^{*2})$

5.3 Results and Discussion

In order to evaluate the effectiveness of our approach, we have conducted extensive experiments on Abdominal Pain dataset and SPECT heart dataset for binary classification, on Dermatology dataset and Nursery School Ranking dataset for multiple-class classification. It has been pointed out that numerous datasets have significant number of missing values, in some cases

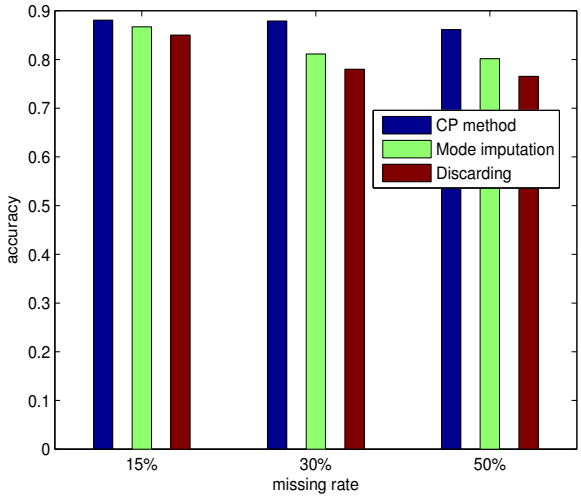
up to 50% [65, 53]. For ease of comparison, we use completely random missing mechanism to generate missing values with missing rates of 15%, 30%, and 50%, respectively. As discussed in Section 4.2, only mean or mode imputation method could be applied if data are missing completely at random (MCAR). The results have been compared with the results by mode imputation method. Our result was also compared with the results of the methods which discard all of the attributes with missing values.

5.3.1 Results for Abdominal Pain Diagnosis and SPECT Heart Diagnosis

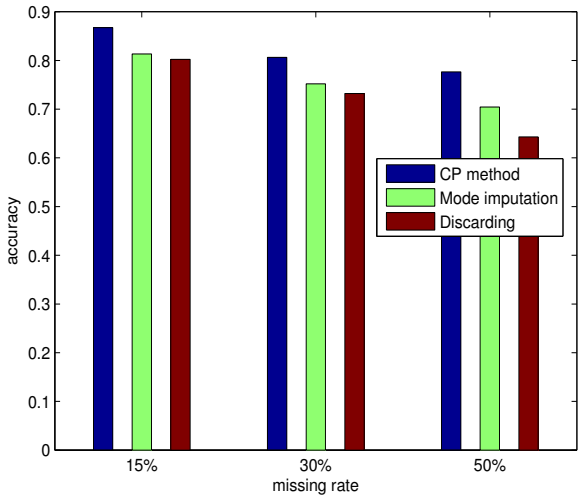
We evaluate our algorithm for binary classification on Abdominal Pain dataset and SPECT heart dataset. As we described earlier in Section 2.4.1, Abdominal Pain dataset contains 6387 examples with 135 attributes. All attributes have two possible values, 0 or 1. SPECT heart dataset is about diagnosis of cardiac Single Proton Emission Computed Tomography (SPECT) images. The database of 267 SPECT image sets (patients) was processed to extract features that summarize the original SPECT images. 44 continuous feature patterns were created for each patient. The patterns were further processed to obtain 13 discrete feature patterns. The features in this dataset have more than two possible values. kNN algorithm is used as the underlying algorithm for conformal predictions. 10-fold cross-validation is applied to reduce classification bias and repeated 10 times.

Figure 5.1 shows the results of applications for APP disease diagnosis and DYS disease diagnosis. Figure 5.2 shows the results for SPECT heart diagnosis. As it can be seen, the classifications by our method always have

the highest accuracy.



(a) Results of classification for APP disease



(b) Results of classification for DYS disease

Figure 5.1: Results of predictions on Abdominal Pain dataset in off-line mode
(a) the classification for APP disease and (b) the classification for DYS disease

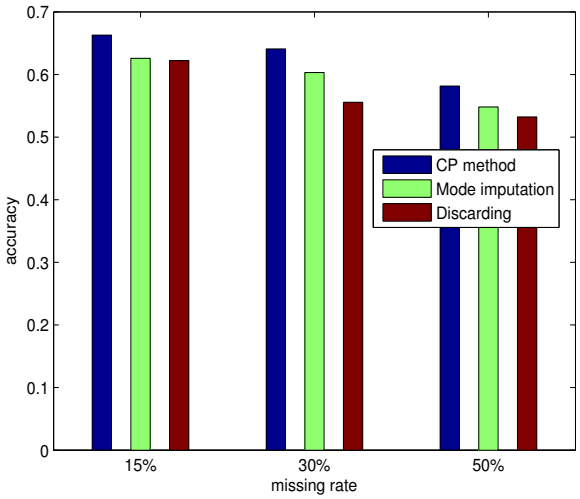


Figure 5.2: Results of predictions on SPECT Heart dataset in off-line mode

5.3.2 Results for Dermatology Diagnosis and Nursery Schools Ranking

These two datasets are used to test the performance of multiple-class classification by our approach. The Dermatology dataset has been used in Section 3.4.2. It contains 34 features and 6 classes. The Nursery dataset was derived from a hierarchical decision model originally developed to rank applications for nursery schools. It is composed by 12960 examples, 8 features and 5 classes. The feature values range from 1 to 5. For our experiments kNN algorithm is used as the underlying algorithm for conformal predictions. 10-fold cross-validation is applied to reduce classification bias and repeated 10 times.

Results for Nursery School Ranking and Dermatology are showed by Figures 5.3 and 5.4, respectively. It can be seen that the classifications by our

method always have the higheset accuracy. The accuracy of classifications by mode imputation drops quickly when the missing rate is growing from 30% to 50%.

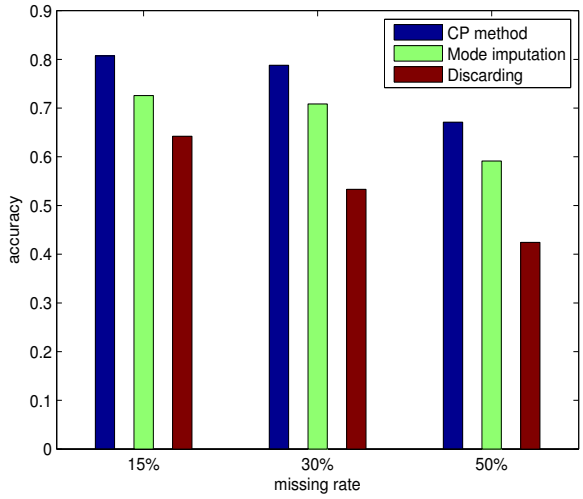


Figure 5.3: Results for Nursery dataset in off-line mode

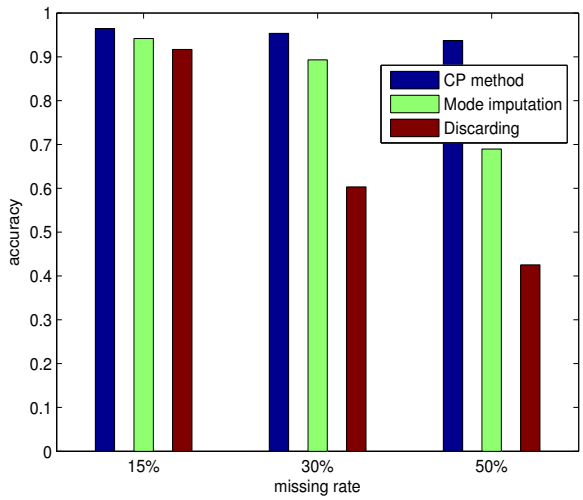


Figure 5.4: Results for Dermatology dataset in off-line mode

5.3.3 Discussion

Features with missing values are a common problem in real-world datasets. Most current algorithms treat features with missing values in the pre-processing step, either discarding them or imputing them. In machine learning, imputation method with prediction model has been well developed and successfully applied on industrial and medicine datasets [65, 48]. However, missing values in these methods are assumed to be in MAR mode. Furthermore, an extra classifier is needed for imputation of missing values. In this chapter, we propose an embedded method for dealing with missing values based on Conformal Predictors. We assume missing values are only existing in test set. One advantage of our approach is making predictions directly with missing values. Furthermore, there is no assumption on the mechanism on missing values.

Our method was compared with both mode imputation method and discarding on four datasets. These datsets provide a wide variety of sizes, feature types, and the number of classes. Experimental results show our method performed better than these methods. Our method successfully embedded the processing of dealing with missing value in making conformal predictions in the off-line mode.

Chapter 6

Conclusion and Future Works

This chapter summarizes the contributions of the thesis and discusses some issues that could be investigated in the future as an extension of this research.

6.1 Summary of Outcomes

Feature selection by conformal predictors

Feature subset selection aims to find the optimal feature subset by eliminating as many irrelevant and redundant features as possible for reducing dimensionality and maximizing accuracies of classifications. CP outputs validity region predictions in on-line mode and performance is evaluated by the efficiency. We presented IACM method which is specifically designed for CP to find the feature subset to help with improving classification efficiency. In our method, average confidence measurement of true label is used as evaluation criterion since it shows how well class groups are separated from each other. Meanwhile, the analysis of irrelevancy and redundancy is introduced as stopping criterion. We tested the effectiveness of IACM on Abdominal

Pain data set and four other data sets from UCI. Results showed that classification using features selected by IACM always provided the most efficient region predictions.

Conformal predictors with additional information

Dealing with additional information is a challenge for current machine learning framework. Inspired by LUPI, we proposed a novel method to incorporate additional information within CP. We introduced additional information to individual example in test set by analyzing how familiar the example with additional values is to the entire examples in training set. The method is not only available for classification in off-line mode but also proved to be valid in on-line mode. But it restricts that the additional information must be discrete and finite. The results of the applications showed that our method successfully utilize additional information to improve the predictive performance for binary and multiple-class classification in both on-line and off-line mode.

Conformal predictors with missing information

Incomplete data is a common problem in the applications on real-life data source. Existing methods deal with missing information before classification. Methods based on machine learning algorithms need to train an extra classifier for imputing missing values and require the missing features are correlated with each other. The method was applied on four different data sets. Results demonstrated that our method do not need to impute missing values and can directly make predictions with them.

6.2 Main Contributions

To the best of author's knowledge, the following pieces of work represent original contribution to the field:

- Using the measurement of confidence as a criterion to select feature subset, which is specificly designed for efficient region predictions of Conformal Predictors.
- Creation of a new method for incorporating additional information within conformal predictors. Such method can utilize additional information to provide better classification performance in both off-line and on-line modes.
- Dealing with missing information in a new way, which embeds the imputation of missing information in conformal predictors.

6.3 Future Prospects

There are various issues in the applications of machine learning algorithm for real-life problems. The contributions in this work attempt to adapt conformal predictions to address the problems related to feature handling. The main purpose of the near future work is: 1) to find theoretical proofs for IACM feature selection method; 2) to study how the training data size can affect the performance of learning with additional information; 3) to apply the proposed method for handling additional information and missing information in"lazy teacher" mode.

- Theoretical proofs.

The effectiveness of IACM feature selection method was studied empirically, more stronger theoretical proofs are needed to guarantee that IACM could work well on other datasets.

- Additional information.

  In Chapter 3, additional information was incorporated within CP and expected to help improve the discrimination power of the classifier. In Section 3.4, experimental results showed that for region predictions additional information only helped when ε is 0.2. It would be interesting to investigate the dependence between the training set size and classification performance using additional information.

- "Lazy teacher" mode.

  In previous work we studied conformal predictors and applied it in on-line mode and off-line mode. In the on-line mode, the feedback is assumed to be given immediately for every object, when prediction was made. This has not always happened in real world applications as there is no such an expert to give the right answer for each question. Although the proposed method could be extended to off-line mode, the validity of prediction would not be guaranteed. Thus, it would be more practical if the method could still work well when the feedback is given with a delay. Our method could be developed further for a new learning protocol, so called lazy teacher [92] where labels are assumed to come with a delay.

Appendix A

Datasets

In this appendix, we list all datasets used for this thesis.

A.1 Abdominal Pain Dataset

The dataset consists of a training set of 4387 patient records and a test set of 2000 patient records, with 9 categories of diseases and 33 types of symptoms [42]. List of diagnostic groups of Abdominal Pain data set and list of all 33 symptoms and their values are shown as below:

Group	Diagnosis	Number of Examples
1	Appendicitis (APP)	844
2	Diverticulitis (DIV)	143
3	Perforates Peptic Ulcer (PPU)	130
4	Non-Specific Abdominal Pain (NAP)	2835
5	Cholecystitis (CHO)	572
6	Intestinal Obstruction (INO)	417
7	Pancreatitis (PAN)	96
8	Renal Colic (RCO)	473
9	Dyspepsia (DYS)	877

Symptom	Values
Sex	male, female
Age	0-9,10-19,20-29,30-39,40-49,50-59,60-69,70+
Pain-site onset	right upper quadrant, left upper quadrant, right lower quadrant left lower quadrant, upper half, lower half, right half, left half central, general, right loin, left loin, epigastric
Pain-set present	right upper quadrant, left upper quadrant, right lower quadrant left lower quadrant, upper half, lower half, right half, left half central, general, right loin, left loin, epigastric
Aggravating factors	movement, coughing, inspiration, food, other, nil
Relieving factors	lying still, vomiting, antacids, milk/food, other, nil
Progress of pain	getting better, no chance, getting worse
Duration of pain	under 12 hours, 12-24 hours, 24-48 hours, over 48 hours
Type of pain	steady, intermittent, colicky, sharp
Severity of pain	moderate, severe
Nausea	nausea present, no nausea
Vomiting	present, no vomitting
Anorexia	present, normal appetite
Indigestion	history of dyspepsia, no history of dyspepsia
Jaundice	history of jaundice, no history of jaundice
Bowel habit	no change, constipated, diarrhoea, blood, mucus
Micturition	normal, frequent, dysuria, haematuria, dark urine
Previous pain	similar pain before, no similar pain before
Previous Surgery	yes, no
Drugs	being taken, not being taken
Mood	normal, distressed, anxious
Colour	normal, pale, flushed, jaundiced, cyanosed
Abdominal scar	present, absent
Abdominal distension	present, absent
Site of Tenderness	right upper quadrant, left upper quadrant, right lower quadrant left lower quadrant, upper half, lower half, right half, left half central, general, right loin, left loin, epigastric,noe
Rebound	present, absent
Guarding	present, absent
Rigidity	present, absent
Abdominal masses	present, absent
Murphy's test	positive, negative
Bowel sounds	normal, decreased/absent, increased
Rectal examination	tender left side, tender right side, generally tender, mass felt, normal

A.2 Datasets from UCI Dataset Repository

The following data sets we used are all from UCI Machine Learning Repository [14], shown as Table A.2.

Table A.1: Summary of data sets

Name	Feature Type	Number of Features	Number of Examples	Number of Classes
Breast Cancer	real	30	500	2
SPECT Heart	categorical	13	267	2
LSVT Voice rehabilitation	real	310	126	2
Dermatology	categorical	34	366	6
Nursery school	categorical	8	12960	5

Appendix B

Feature Subset Significance

In this appendix, we introduce the Feature Subset Significance method [13].

Let m be the total number of features available and denote $F = \{1, \ldots, m\}$. Let A and B are two feature subsets derived from independent processes, $A \subseteq F$ and $B \subseteq F$. Denote $a = |A|$ and $b = |B|$. Let $c = |A \cap B|$ be the number of overlapped features between A and B. The significance of the process that generate B is calculated as:

$$P_{FS}(m, a, b, c) = \sum_{i=c}^{min(a,b)} \binom{b}{i} \frac{(a)_i (m-a)_{(b-i)}}{(m)_b}$$

where $(n)_k = \frac{(n)!}{(n-k)!}$ and $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ are the number of permutations and combinations of k from n items.

The smaller the significance, the less likely that B is derived just by chance, with respect to reference subset A.

Appendix C

Classification Algorithms

In this appendix, we briefly discuss all classification algorithms used in this thesis.

C.1 k-Nearest Neighbor Algorithm

k-Nearest Neighbor (kNN) is based on the principle that the examples within a dataset will generally exist in close proximit to other examples that have similar properties. If the examples are tagged with a classification label, then the value of the label of an unclassified example can be determined by observing the class of its nearest neighbor. The kNN locates the k nearest examples to the query example and determines its class by identifying the single most frequent class label. There are three methods used commonly for calculating distance, which are City block metric, Euclidean distance and Chebyshev distance.

- City Block Metric: $d_{st} = \sum_{j=1}^{n} |\mathbf{x}_{sj} - \mathbf{x}_{tj}|$

- Euclidean Distance: $d_{st} = \sqrt[2]{\sum_{j=1}^{n} |\mathbf{x}_{sj} - \mathbf{x}_{tj}|^2}$
- Chebyshev Distance: $d_{st} = \max_j |\mathbf{x}_{sj} - \mathbf{x}_{tj}|$

C.2 Nearest Centroid Classifier and NCM Underlying NC Classifier

C.2.1 Nearest Centroid Classifier

Nearest Centroid algorithm is similar to the kNN algorithm. A centroid is computed for each class label as the mean position of the training examples with the label. Then an object is labeled by the nearest class.

Given a set of n training examples $(\mathbf{x}_i, y_i)$, $\mathbf{x}_i \in \mathbf{X}, y_i \in Y$. The centroid for each class label y could be calculated as:

$$\mu_y = \frac{1}{|C_y|} \sum_{i \in C_y} \mathbf{x}_i \tag{C.1}$$

where $C_y = \{i : y_i = y\}$. Then, given a new example $\mathbf{x}_{n+1}$, we predict its classification by measuring the distance between object and centroid as:

$$\hat{y} = \arg\min_y D(\mu_y, \mathbf{x}_{n+1})$$

where D is a distance metric.

C.2.2 NCM Underlying NC Classifier

In Nearest Centroid algorithm, the prediction for a new object $\mathbf{x}_i$ is the label of the closest centroid from training examples. If D is a distance metric,

the nonconformity measure underlying NC classifier for an example $(\mathbf{x}_i, y)$ is defined as [13]:

$$\alpha_i^y = \frac{D^y(\mathbf{x}_i, \mu_y)}{\min_{-y} D^{-y}(\mathbf{x}_i, \mu_{-y})}$$

where $D^y(\mathbf{x}_i, \mu_y)$ denotes the distance between an example $\mathbf{x}_i$ and the centroid of examples with the same label, μ_y. $D^{-y}(\mathbf{x}_i, \mu_{-y})$ denotes the distance between an example $\mathbf{x}_i$ and the centroid of examples with the other labels, μ_{-y}.

C.3 Support Vector Machines and NCM Underlying SVM Classifier

C.3.1 Support Vector Machines

Support Vector Machines (SVMs) are the widely used supervised machine learning algorithms. Suppose given examples belonging to two classes, in SVMs, we want to know whether we can separate these examples with a $(d-1)$-dimensional hyperplane. This is called a linear classifier. There are many hyperplanes that might classify the examples. One reasonable choice as the best hyperplane is the one that represents the largest separation, or margin, between two classes [25]. So we choose the hyperplane so that the distance from it on the nearest example on each side is maximized [25].

Given some training examples $\{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)\}$, $\mathbf{x}_i \in \mathbf{X}^d$ and $y_i \in \{1, -1\}$. If the training examples are linear separable, we can select two hyperplanes in a way that they separate the examples and there are no points

between them and then try to maximize their distance. The hyperplanes can be described by the equations:

$$(\mathbf{w} \cdot \mathbf{x}) - b = 1; (\mathbf{w} \cdot \mathbf{x}) - b = -1; \tag{C.2}$$

By using geometry, we find the distance between these two hyperplanes is $\frac{2}{||\mathbf{w}||}$, so we want to minimize $||\mathbf{w}||$. As to prevent examples from falling into margin, the following constraint are added: for all $1 < i < n$,

$$y_i(\mathbf{w} \cdot \mathbf{x}_i - b) \geq 1$$

The optimization problem presented in the preceding section is difficult to solve because it depends on $||\mathbf{w}||$, so we alter $||\mathbf{w}||$ with $\frac{1}{2} \parallel \mathbf{w} \parallel^2$. This is a quadratic programming optimization problem. By introducing Lagrange multipliers α, the previous constrained problem can be expressed as:

$$\arg\min_{(\mathbf{w},b)} \frac{1}{2}||\mathbf{w}||^2$$

for any $i = 1, ..., n$, subject to

$$y_i(\mathbf{w} \cdot \mathbf{x}_i - b) \geq 1$$

In 1995, Cortes and Vapnik suggested a modified maximum margin idea that allows for mislabeled examples [25]. If there exists no hyperplane that can completely split the examples, the soft margin method will choose a hyperplane that splits the examples as cleanly as possible, while still maximizing the distance to the nearest cleanly split examples. The method introduces non-negative slack variables, ξ_i, which measure the degree of misclassification of examples, as described as following equation:

$$y_i(\mathbf{w} \cdot \mathbf{x}_i) - b \geq 1 - \xi_i, 1 \leq i \leq n \tag{C.3}$$

The objective function is then increased by a function which penalizes non-zero ξ_i, and the optimization becomes a trade off between a large margin and a small error penalty. If the penalty function is linear, the optimization problem becomes:

$$\arg \min_{(\mathbf{w},\xi,b)} \{\frac{1}{2} \parallel \mathbf{w} \parallel^2 + C \sum_{i=1}^{n} \xi_i\}$$

for any $i = 1, ..., n$, subject to

$$y_i(\mathbf{w} \cdot \mathbf{x}_i - b) \geq 1 - \xi_i$$

$$\xi_i \geq 0$$

The constant C here is a parameter responsible for the trade-off between $||\mathbf{w}||^2$ and $\sum_{i=1}^{l} \xi_i$. If $C = +\infty$, the separation do not tolerate any slack, if $C = 0$, we do not care about slack at all, thus any $\mathbf{w}$ satisfies the constraints.

In 1992, Boser and Vapnik suggested a way to create nonlinear classifiers by applying the kernel trick [2] to maximum-margin hyperplane [7]. In the method, every example is replaced by a nonlinear kernel function. This allows the algorithm to fit the maximum-margin hyperplane in a transformed feature space. The transformation may be nonlinear and the transformed space high dimensional; thus though the classifier is a hyperplane in the high-dimensional feature space, it may be nonlinear in the original input space. Here we introduce some popular kernels.

- Linear Kernel,

$$K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \cdot \mathbf{x}_j$$

- Polynomial Kernel,

$$K(\mathbf{x}_i, \mathbf{x}_j) = (1 + \mathbf{x}_i^T \cdot \mathbf{x}_j)^p$$

- Gaussian Kernel,

$$K(\mathbf{x}_i, \mathbf{x}_j) = exp\left(-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma^2}\right)$$

C.3.2 NCM Underlying SVM Classifier

When solving SVM as a dual form problem, we derive solutions in terms of Lagrange multiplies α_i for each training example $\mathbf{x}_i$ of which $\alpha \geq 0$ is true only for support vectors. Support vectors can be considered the most strange examples as they are the boundary cases that define the shape of the hyperplane. So the Lagrange multipliers can be used as strangeness measure [92]. If an example $\mathbf{x}_i$ is one of the support vectors, $0 < \alpha_i < c$; if an example $\mathbf{x}_i$ is between the margin of the two classes, $\alpha_i = c$; if an example $\mathbf{x}_i$ is out of the margin of the two classes, $\alpha_i = 0$.

Bibliography

[1] H. L. Anderson. Metropolis, monte carlo and the maniac. Los Alamos Science, 14: 96-108, 1986.

[2] M. A. Aizerman, E. M. Braverman, and L. I. Rozonoer. Theoretical foundations of the potential function method in pattern recognition learning. Automation and Remote Control, 25: 821–837, 1964.

[3] D. W. Aha, D. Kibler, and M. K. Albert. Instance based learning algorithms. Machine Learning, 6: 37-66, 1991.

[4] D. Allen. The relationship between variable selection and data augmentation and a method for prediction. Technometrics, 16: 125-127, 1974.

[5] E. Alpaydin. Introduction to machine learning. 2nd ed. Massachusetts Institute of Technology, 2009.

[6] D. Adamskiy, I. Nouretdinov, A. Mitchell, N. Coldham, and A. Gammerman. Applying conformal prediction to the bovine TB dignosing. IFIP Advances in Information and Communication Technology, pages 449–454, 2011.

[7] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifier. In Proceedings of the fifth annual workshop on Computational learning theory, 92: 144–152, 1992.

[8] V. Balasubramanian. Conformal predictions in multimedia pattern recognition. PhD Thesis, Arizona State University, 2010.

[9] V. Balasubramanian, S. S. Ho, and V. Vovk. Conformal prediction for reliable machine learning: Theory, Adaptations, and Applications. Morgan Kaufmann, 2014.

[10] R. E. Bellma. Adaptive control processes: A Guided Tour. Princeton University Press, 1961.

[11] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. Vaughan. A theory of learning from different domains. Machine Learning, 79(1): 151-175, 2010.

[12] A. Blum and P. Langley. Selection of relevant features and examples in machine learning. Artificial Intelligence, 97:245-271, 1997.

[13] T. Bellotti. Confidence machine for microarray classification and feature selection. PhD thesis, Royal Holloway, University of London, 2006.

[14] K. Bache and M. Lichman. UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science, 2013.

[15] T. Bellotti, Z. Luo, and A. Gammerman. Strangeness Minimisation Feature Selection with Confidence Machines. Lecture Notes in Computer Science, 4224: 978–985, 2006.

[16] G. E. A. Batista and M. C. Monard. An analysis of four missing data treatment methods for supervised learning. Applied Artificial Intelligence, 17: 519–533, 2003.

[17] D. A. Bell and H. Wang. A formalism for relevance and its application in feature subset selection. Machine Learning, 41: 175–195, 2000.

[18] R. Carnap. Logical foundations of probability. The University of Chicago Press, 1962.

[19] R. Caruana and D. Freitag. Greedy attribute selection. In Proceedings of the Eleventh International Conference on Machine Learning, pages 28–36, 1994.

[20] H. Y. Chen and R. Little. A testing of missing completely at random for generalized estimating equations with missing data. Biometrika, 86(1): 1–13, 1999.

[21] K. Chan, T. W. Lee, and T. J. Sejnowski. Variational Bayesian learning of ICA with missing data. Neural Comput, 15(8): 1991-2011, 2003.

[22] J. G. Cleary, S. Legg, and I. H. Witten. An MDL estimate of the significance of rules. In Proceedings of ISIS: Information, Statistics, and Induction in Science, pages 43–53, 1996.

[23] J. G. Carbonell, R. S. Michalski, and T. M. Mitchell. An overview of machine learning. TIOGA Publishing Co., 1983.

[24] K. J. Cherkauer and J. W. Shavlik. Growing simpler decision trees to facilitate knowledge discovery. In Proceedings of the Second International

Conference on Knowledge Discovery and Data Mining, pages 315–318, 1996.

[25] C. Cortes and V. N. Vapnik. Support vector networks. Machine Learning, 20: 273–297, 1995.

[26] Y. Chen and Y. Yao. A multiview approach for intelligent data analysis based on data operators. Information Sciences, 178(1): 1-20, 2008.

[27] P. Domingos. A few useful things to know about machine learning. Communications of the ACM, 55(10): 78–87, 2012.

[28] D. Dutton and G. Conroy. A review of machine learning. Knowledge Engineering Review, 12: 341–367, 1996.

[29] M. Dash and H. Liu. Feature selection for classification. Intelligent Data Analysis, pages 131–156, 1997.

[30] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithms. Journal of Royal Statistical Society B39: 1–38, 1977.

[31] P. A. Devijver and J. Kittler. Pattern recognition: a statistical approach. Prentice Hall, 1982.

[32] S. D. Essinger and G. L. Rosen. An introduction to machine learning for students in secondary education. Digital Signal Processing Workshop and IEEE Signal Processing Education Workshop, pages 243–248, 2011.

[33] J. Feyereisl and U. Aickelin. Privileged information for data clustering. Information Science, 194: 4–23, 2012.

[34] A. Farhangfar, L. Kurgan, and W. Pedrycz. Experimental analysis of methods for imputation of missing values in databases. Intelligent Computing: Theory and Applications II Conference, pages 172–182, 2004.

[35] A. Farhangfar, L. Kurgan, and W. Pedrycz. A novel framework for imputation of missing values in databases. IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans, 37(5): 692-709, 2007.

[36] A. Farhangfar, L. Kurgan, and L. Dy. Impact of imputation of missing values on classification error for discrete data. Pattern Recognition 41: 3692–3705, 2008.

[37] A. J. Feelders. Handling missing data in trees: surrogate splits or statistical imputation. In Proceedings of the 3rd European Conference on Principles and Practice of Knowledge Discovery in Data Bases, pages 329-334, 1999.

[38] P. Gardenfors. On the logic of relevance. Synthese, 37: 351-367, 1978.

[39] I. Guyon and A. Elisseeff. An Introduciont to Variable and Feature Selection. Journal of Machine Learning Reserach, 3: 1157–1182, 2003.

[40] J. Grzymala-Busse and W. Grzymala-Busse. Handling missing attribute values. In book: Data Mining and Knowledge Discovery Handbook, pages 37–57, 2005.

[41] J. W. Grzymala-Busse and M. Hu. A Comparison of several approaches to missing values in data mining. Springer, 2001.

[42] A. Gammerman and A. R. Thatcher. Bayesian inference in an expert system without assuming independence. Advances in Artificial Intelligence, pages 182–218, 1988.

[43] A. Gammerman, V. Vovk, B. Burford, I. Nouretdinov, Z. Luo, A. Chervonenkis, M. Waterfield, R. Cramer, P. Tempst, J. Villanueva, M. Kabir, S. Camuzeaux, J. Timms, U. Menon, and I. Jacobs. Serum proteomic abnormality predating screen detection of ovarian cancer. Computer Journal, 52(3): 326–333, 2009.

[44] D. J. Hand. Intelligent data analysis: issues and opportunities. Intelligent Data Analysis, pages 67–79, 1998.

[45] M. A. Hall. Correlation-based feature selection for machine learning. PhD Thesis, University of Waikato, 1999.

[46] M. A. Hall. Correlation-based feature selection for discrete and numeric class machine learning. Morgan Kaufmann, 2008.

[47] G. H. John, R. Kohavi, and P. Pfleger. Irrelevant features and the subset selection problem. Machine Learning: Proceedings of the Eleventh International Conference, pages 121–129, 1994.

[48] J. M. Jerez, I. Molina, P. J. Gracia-Laencina, E. Alba, N. Ribelles, M. Martin, and L. Franco. Missing dara imputation using statistical and machine learning methods in a real breast cancer problem. Artifical Intelligence in Medicine, 50: 105–115, 2010.

[49] J. M. Keynes. A treatise on probability. London: Macmillan, 1921.

[50] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, 2(12): 1137–1143, 1995.

[51] R. Kohavi. Wrappers for Performance Enhancement and Oblivious Decision Graphs. PhD thesis, Stanford University, 1995.

[52] I. Kononenko and I. Bratko. Information-based evaluation criterion for classifiers performance. Machine Learning, 6: 67-80, 1991.

[53] L. A. Kurgan, K. J. Cios, M. Sontag, and F. J. Accurso. Mining the cystic fibrosis data. In book: Next generation of Data-Mining Applications. IEEE Press, pages 310–311, 2005.

[54] R. Kohavi and G. John. Wrappers for Feature Subset Selection. Artificial Intelligence, special issue on relevance, 97(12): 273-324, 1996.

[55] I. Kojadinovic and T. Wottka. Comparison between a Filter and a Wrapper Approach to variable subset selection in regression problems, ESIT, pages 14–15, 2000.

[56] H. P. Kostas, K. Proedrou, V. Vovk, A. Gammerman, and S. T. Ex. Inductive confidence machines for refression. In Proceedings of the Thirteenth European Conference on Machine Learning, 2430: 345–356, 2002.

[57] K. Kira and L. A. Rendell. A practical approach to feature selection. In Machine Learning: Proceedings of the Ninth International Conference, pages 249–256, 1992.

[58] R. Kohavi and D. Sommerfield. Feature subset selection using the wrapper method: overfitting and dynamic search space topology. In Proceedings of the International Conference on Knowledge Discovery and Data Mining, pages 192-197, 1995.

[59] S. B. Kotsiantis. Supervised machine learning: a review of classification thechniques. In Proceedings of the 2007 Conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies, pages 3–24, 2007.

[60] S. B. Kotsiantis, I. D. Zaharakis, and P. E. Pintelas. Machine learning: a review of classification and combining techniques. Artifical Intelligence Review, 26(3): 159–190, 2007.

[61] P. Langley. Selection of Relevant Features in Machine Learning. In Proceedings of the AAAI Fall Symposium on Relevance, pages 140–144, 1994.

[62] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In Proceedings of the IEEE, 86(11): 2278-2324, 2002.

[63] Y. Lecun, C. Cortes, and J. C. Burges. The mnist data of handwritten digits. [http://yann.lecun.com/exdb/mnist/].

[64] L. Ladha and T. Deepa. Feature selection methods and algorithms. International Journal on Computer Science and Engineering, 3(5): 1787–1797, 2011.

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

[66] R. J. Little and D. B. Rubin. Statistical analysis with missing data. New york: Wiley, 1987.

[67] P. Langley and S. Sage. Induction of selective Bayesian Classifiers. In Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence, pages 399–406, 1994.

[68] P. Langley and S. Sage. Oblivious decision trees and abstract cases. In Working Notes of the AAAI-94 Workshop on Case-Based Reasoning, pages 113–117, 1994.

[69] P. Langley and S. Sage. Scaling to Domains with Irrelevant features. Computational Learning Theory and Natural Learning Systems, 4: 51–63, 1994.

[70] M. Li and P. Vitányi. An introduction to Kolmogorov complexity and its applications. 2nd ed. Springer, 1997.

[71] P. Martin-Löf. The definition of randomness sequence. Information and Control, 1996.

[72] A. W. Moore and M. S. Lee. Efficient algorithms for minimizing cross val- idation error. Machine Learning: Proceedings of the Eleventh International Conference, pages 190–198, 1994.

[73] T. M. Mitchell. Machine learning. McGraw-Hill, 1997.

[74] D. Michie, D. J. Spiegelhalter, and C. C. Taylor. Machine learning, neural and statistical classification. Prentice Hall, 1994.

[75] T. Melluish, C. Saunder, I. Nouretdinov, and V. Vovk. Comparing the bayes and typicalnes frameworks. In Proceedings of the 12th European Conference on Machine Learning, 2167: 360–371, 2001.

[76] R. M. Neal. Bayesian learning for neural networks. 1st ed. Springer, 1996.

[77] P. M. Narendra and K. Fukunaga. A branch and bound algorithm for feature selection. IEEE Transactions on Computers, C26(9): 917–922, 1977.

[78] I. Nouretdinov, V. Vovk, M. V. Vyugin, and A. Gammerman. Pattern recognition and density estimation under the general i.i.d. assumption. In Proceedings of the 14th Annual Conference on Computational Learning Theory and and 5th European Conference on Computational Learning Theory, pages 337-353, 2001.

[79] H. Papadopoulos. Inductive conformal prediction: Theory and application to neural networks. Tools in Artificial Intelligence, pages 315-329, 2008.

[80] H. Papadopoulos, V. Vovk, and A. Gammermam. Conformal prediction with neural networks. Tools with Artificial Intelligence, IEEE International Conference, 2: 388-395, 2007.

[81] J. Pearl. Probabilistic reasoning in intelligent systems: Networks of plausible inference. Morgan Kaufmann, 1988.

[82] K. Proedrou. Rigorous measures of confidence for pattern recognition and regression. PhD Thesis, Royal Holloway College, University of London, 2003.

[83] K. Proedrou, I. Nouretdinov, V. Vovk, and A. Gammerman. Transductive confidence machines for pattern recognition. In Proceedings of the Thirteenth European Conference on Machine Learning, pages 381–390, 2002.

[84] J. R. Quinlan. Induction of decision trees. Machine Learning, 1: 81-106, 1986.

[85] J. R. Quinlan. C4.5: Programs for machine learning. Morgan Kaufmann, 1993.

[86] B. Ribeiro, C. Silva, A. Vieira, A. Gaspar-Cunha, and J. C. das Neves. Financial distress model prediction using SVM+, pages 1-7, 2010.

[87] R. Setiono and H. Liu. Chi2: Feature selection and discretization of numeric attributes. In Proceedings of the Seventh IEEE International Conference on Tools with Artificial Intelligence, pages 388–391, 1995.

[88] N. Suguna and K. G. Thanushkodi. Predicting missing attribute values using k-means clustering. Journal of Computer Science, 7(2): 216–224, 2011.

[89] G. Shafer and V. Vovk. A tutorial on conformal prediction. Journal of machine learning research, 9: 371–421, 2008.

[90] J. F. Timms, U. Menon, D. Devetyarov, A. Tiss, S. Camuzeaux, K. McCurrie, I. Nouretdinov, B. Burford, C. Smith, A. Gentry-Maharaj,

R. Hallett, J. Ford, Z. Luo, V. Vovk, A. Gammerman, R. Cramer, and I. Jacobs. Early detection of ovarian cancer in samples pre-diagnosis using CA125 and MALDI-MS peaks. Cancer Genomics & Proteomics, 8(6): 289–305, 2011.

[91] V. Vapnik and A. Vashiet. A new learning paradigm: Learning using privileged information. Neural Networks, 22(5-6): 544–557, 2009.

[92] V. Vovk, A. Gammerman, and G. Shafer. Algorithmic learning in a random world. Springer-Verlag New York, 2005.

[93] V. Vapnik, A. Vashist, and N. Pavlovitch. Learning using hidden information: Master-class learning. In F. F. Soulie, D. Perrotta, J. Piskorski, and R. Steinberger, editors, NATO Science for Peace and Security Series, D: Information and Communication Security, 19: 3-14, 2008.

[94] H. Vafaie and K. De Jong. Genetic algorithms as a tool for restructuring feature space representations. In Proceedings of the International Conference on Tools with Artificial Intelligence IEEE Computer Society Press, pages 8–11, 1995.

[95] V. Vapnik. Estimation of Dependences Based on Empirical Data (Information Science and Statistics). Springer, 2006.

[96] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik. Feature Selection for SVMs. In S.A. Solla, T.K. Leen, and K.-R. Muller, editors, Advances in Neural Information Processing Systems, 12: 526-532, 2000.

[97] L. Wohlrab and J. Furnkranz. A review and comparison of strategies for handling missing values in separate-and-conquer rule learning. Journal of Intelligent Information Systems, 36: 73–98, 2011.

[98] Y. Wang, M. Yang, G. Wei, R. Hu, Z. Luo, and G. Li. Improved PLS regression based on SVM classification for rapid analysis of coal properties by near-infrared reflectance spectroscopy. Sensors and Actuators B: Chemical, 93: 723–729, 2014.

[99] L. Yu and H. Liu. Feature selection for high-dimensional data: a fast correlation-based filter solution. In Proceedings of the Twentieth International Conference on Machine Learning, pages 856–863, 2003.

[100] L. Yu and H. Liu. Efficient feature selection via analysis of relevance and redundancy. Journal of Machine Learning Research, 5: 1205–1224, 2004.

[101] M. Yang, I. Nouretdinov, Z. Luo, and A.Gammerman. Feature selection by conformal predictor. IFIP Advances in Information and Communication Technology, 364: 439–448, 2011.

[102] M. Yang, I. Nouretdinov, and Z. Luo. Learning by conformal predictors with additional information. IFIP Advances in Information and Communication Technology, 412: 394–400, 2013.

[103] S. Zhang, C. Zhang, and Q. Yang. Data preparation for data mining. Applied Artificial Intelligence, 17: 375-381, 2002.