Royal Holloway, University of London, UK

Doctoral Thesis

Reliable Confidence Measures and Well-Calibrated Probabilistic Outputs in Classification Algorithms

Author:
Antonis Lamprou

Supervisor:
Dr. Harris Papadopoulos,
Prof. Alexander Gammerman

A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy in the

Computational Learning Research Centre
Department of Computer Science

July 20, 2016

Declaration of Authorship

I, Antonis Lamprou, hereby declare that this thesis and the work presented in it is entirely my own. Where I have consulted the work of others, this is always clearly stated.

Signed: _____

Date: _____

"We both step and do not step in the same rivers. We are and are not."

Heraclitus of Ephesus

ROYAL HOLLOWAY, UNIVERSITY OF LONDON, UK

Abstract

Department of Computer Science

Doctor of Philosophy

Reliable Confidence Measures and Well-Calibrated Probabilistic Outputs in Classification Algorithms

by Antonis Lamprou

The Machine Learning research area is widely used in several predictive systems, where observations from the past can be used to create predictions about future events. Machine Learning can be applied to any area where classification or regression is used. Nonetheless, most Machine Learning algorithms do not provide any measures of valid confidence. Conformal Prediction (CP) is a framework which uses underlying Machine Learning algorithms, and can provide valid measures of confidence for predictions. Additionally, the Venn Prediction (VP) framework, which is an extension to the CP framework, provides well-calibrated probabilistic outputs. This thesis explores and provides new methods for valid measures of confidence and probabilistic outputs, based on the Conformal and Venn Prediction frameworks. We introduce a new Conformal Predictor based on Genetic Algorithms and compare our approach with other methods. Additionally, the CP framework is extended for multi-label applications where predictions can contain more than one possible classifications. Furthermore, a new Venn Predictor based on Inductive CP is introduced, which greatly improves the computational efficiency of VP. We conduct experiments on our methods and examine their performance and validity. Finally, we examine the applications of osteoporosis risk assessment, the diagnosis of childhood abdominal pain, and the evaluation of the risk of stroke based on ultrasound images of atherosclerotic carotid plaques. Our experimental results on all our methods demonstrate the reliability and usefulness of our confindence and probabilistic outputs.

Acknowledgements

I am grateful for all the help I have received from my supervisors Prof. Alex Gammerman and Dr. Harris Papadopoulos during my PhD course. I would also like to express my thankfulness to my advisor Prof. Volodya Vovk for his helpful discussions, and Dr. Ilia Nouretdinov for his contribution to my work.

Additionally, I would like to express my appreciation to the medical doctors and nursing stuff who have participated in my research throughout the PhD course, and have helped me gain important knowledge and gather data that contributed to my work.

Moreover, I would like to thank Dr. Efthyvoulos Kyriacou for providing access to medical data, and for the discussions we had during my research.

Contents

List of Figures

List of Tables

Abbreviations

AA	Algorithmic Adaptation
AF	Amaurosis Fugax
ANN	Artificial Neural Network
ANN-CP	Artificial Neural Network Conformal Predictor
ANN-TVP	Artificial Neural Network Transductive Venn Predictor
ANN-IVP	Artificial Neural Network Inductive Venn Predictor
ATL	Advanced Technology Laboratories
BMD	Bone Mineral Density
BR-MLCP	Binary Relevance Multi-Label Conformal Predictor
BS	Brier Score
CA	Cumulative Accuracy
CAP	Cumulative Accuracy Probability
CA-125	Cancer Antigen 125
CBFS	Correlation Based Feature Selection
CLAP	Cumulative Lower Accuracy Probability
CP	Conformal Prediction
CUAP	Cumulative Upper Accuracy Probability
CSFS	Chi-Squared Feature Selection
DEXA	Dual Energy X-Ray Absorptiometry
FDTA	Fractal Dimension Texture Analysis
FN	False Negatives
FP	False Positives
FPS	Fourier Power Spectrum
GA	Genetic Algorithm
GA-CP	Genetic Algorithm Conformal Predictor
GLDS	GrayLevel Difference Statistics
ICP	Inductive Conformal Prediction
IGFS	Information Gain Feature Selection
IR	Isotonic Regression

IVP	Inductive Venn Prediction
k-NN	k-Nearest Neighbours
k-NN-CP	k-Nearest Neighbours Conformal Predictor
L	Large
LOOCV	Leave One Out Cross Validation
M	Medium
ML	Medium-Large
ML-CP	Multi-Label Conformal Predictor
NB	Naive Bayes
NBC	Naive Bayes Classifier
NBC-CP	Naive Bayes Classifier Conformal Predictor
NGTDM	Neighbourhood Gray Tone Difference Matrix
PAC	Probably Approximately Correct
PCA	Principal Component Analysis
PNN	Probabilistic Neural Network
PT	Pattern Transformation
RBF	Radial Basis Function
RUNL	Run Length Statistics
S	Small
SF	Statistical Features
SFM	Statistical Feature Matrix
SGLDM	Spatial Gray Level Dependence Matrices
SM	Small-medium
SVM	Support Vector Machine
SVMFS	Support Vector Machine Feature Selection
SVM-CP	Support Vector Machine Conformal Predictor
SVM-TVP	Support Vector Machine Transductive Venn Predictor
SVM-IVP	Support Vector Machine Inductive Venn Predictor
SVM-LR	Support Vector Machine Logistic Regression
SVM-IR	Support Vector Machine Isotonic Regression
TEM	Laws Texture Energy Measures
TIA	Transient Ischaemic Attack
TP	True Positives
TN	True Negatives
TVP	Transductive Venn Prediction
VC	Vapnik-Chervonenkis
VP	Venn Prediction
WBCD	Winsconsin Breast Cancer Diagnosis
WHO	World Health Organisation

Chapter 1

Introduction

1.1 Machine Learning

Machine Learning is a research area in Computer Science and Artificial Intelligence, where algorithms are developed to learn from past experience in order to be able to predict future cases. Machine Learning is divided into two main categories: supervised learning, and unsupervised learning. In supervised learning, a data instance typically consists of an input vector and a desirable output value. A supervised algorithm is trained on a training set of data and creates a model that maps the input vectors to the output values of the data. The algorithm can be assessed on how well it can predict the output values of new data. In unsupervised learning, the output values of the data are unknown, and the task is to divide the given data into clusters. In this thesis, we are interested in supervised learning applications.

Machine Learning is based on the statistical learning theory. This theory assumes that for a vector space X of all possible inputs, and a vector space Y of all possible output values there is some unknown probability distribution over the product space $Z = X \times Y$. The data are assumed to be identically and independently distributed (i.i.d.). A learning algorithm is trained to find a function $f : X \mapsto Y$ such that $f(\vec{x}) \sim y$. For simplicity purposes, $\vec{x}$ will be hereafter simplified to x.

Applications of statistical learning are either classification or regression problems. In classification, the output values of the data are discrete and are usually mapped into classes, whereas in regression the output of each data point is continuous. In any of these problems, a loss function (or error function) is defined which indicates how well an algorithm can predict the output of the data. In regression, a common loss function is the square loss function $V(f(x, y) = (y - f(x))^2$. In classification, a natural loss function is the 0-1 loss function, where $V = 1$ if the predicted output is different from the actual output, and $V = 0$ if the prediction is correct. When training an algorithm, the main criteria is to minimise the loss function. In this thesis, we are interested in classification applications. In Figure 1.1, we illustrate a two dimensional linear classification task. The linear classifier is depicted as a dashed line, and the task is to classify instances in the two-dimensional space as "squares" or "circles". A new instance with an unknown label to be predicted is shown as x. According to the linear classifier, the given instance should be labelled as a "circle", since it falls in the class with the "circle" points.

A common problem in Machine Learning is overfitting. When a learning algorithm creates a model that minimises the loss on the training data, but fails

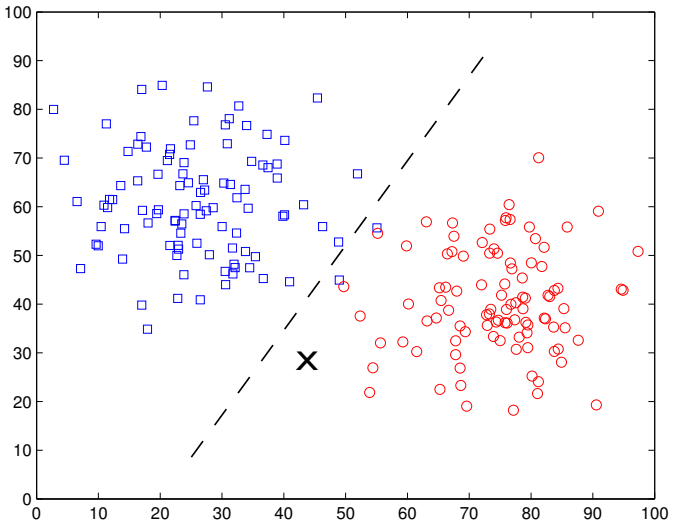


FIGURE 1.1: Classification task.

to generalise on new data, such that the loss increases, it is said that the algorithm has overfitted. In order to avoid overfitting, algorithms are usually constructed to generalise as much as possible. In statistical learning theory, the Vapnik-Chervonenkis (VC) dimension of a classifier is utilised to find the upper bound probability of test loss (i.e. loss on unseen test data), given the training loss. The VC dimension is a measure of the capacity of a learning classification algorithm. As it can be shown by the VC theory, it is not always the case that the test loss would decrease while we increase the capacity of a classifier. Therefore, one should control the capacity of a classifier, such that the loss is minimised.

A practical approach for evaluating a classifier is through cross-validation experiments. In cross-validation, the data is divided into training set and test set several times, while for each time a different block of the data is used as

the test set. Through cross-validation, one can assess on how well the classifier generalises and avoids overfitting.

In the following subsections, we describe five popular Machine Learning algorithms which can be used for classification. Namely, the algorithms are: k-Nearest Neighbours (k-NN), Artificial Neural Networks (ANNs), Support Vector Machines (SVMs), Naive Bayes (NB) and Genetic Algorithms (GAs). A thorough description of the aforementioned algorithms can be found in [3].

1.1.1 k-Nearest Neighbours

The k-Nearest Neighbours algorithm is one of the simplest algorithms which performs well in many classification applications. In k-NN classification, a training set of the form $(x_1, y_1), \ldots, (x_n, y_n)$ is given, and the goal is to predict the class of a new instance x_{n+1}. The algorithm finds the k nearest neighbours of the new instance in the training set, and assigns the class of the new instance to the most common label found amongst the k neighbour training instances. When $k = 1$, the new instance is simply assigned to the class of the single nearest neighbour. The distance between two instances can be any distance measure. Typically, the Euclidean distance can be used:

$$D = \sqrt{\sum_{j=1}^{m} (q_j - p_j)^2}, \tag{1.1}$$

where q and p are two instances of data with Cartesian coordinates $j = 1, ..., m$. Once all distances are found between the new instance x_{n+1} and all other instances, the k-NN algorithm sorts the distances from smaller to larger and

outputs as a prediction the label which occurs the most amongst the first k labels.

1.1.2 Artificial Neural Networks

Artificial Neural Networks (ANNs) imitate the work of biological neural networks which are located in the brain of living organisms. ANNs consist of a set of interconnected units, called neurons. Abstractly, each neuron has several inputs and an output value, and the outputs can be connected to other neurons in the network (i.e. an output becomes the input of other neurons). In ANNs, there can be several layers of interconnected neurons. Typically there is an input layer, one or more hidden layers, and an output layer. Networks that only forward the output to a next layer are called feed forward neural networks. Figure 1.2 depicts such a neural network. Other structures of ANNs exists, such as recurrent networks where connections between the neurons form a directed cycle.

Each neuron in the network accepts several input values and generates an output based on its input weights w_i, a transfer function net and an activation function ϕ. A neuron representation is depicted in Figure 1.3. A common transfer function is the sum of the weighted inputs:

$$net = \sum_{i=1}^{n} w_i x_i, \tag{1.2}$$

and a common activation function which produces the output o is the logistic function

$$o = \phi(net) = (1 + e^{-net})^{-1}. \tag{1.3}$$

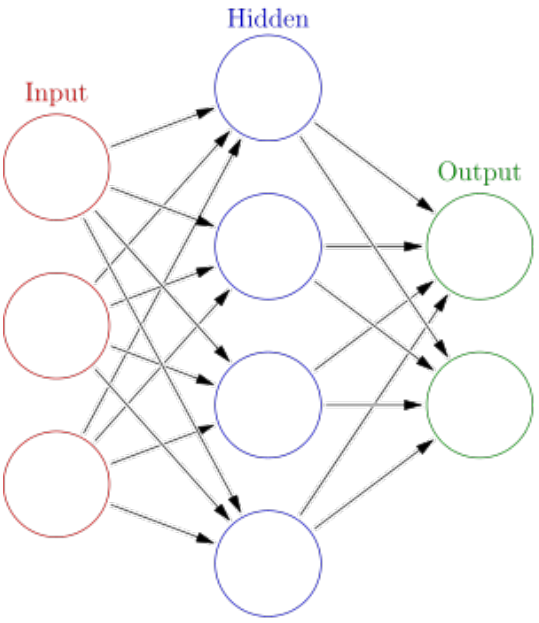


FIGURE 1.2: Feed Forward Neural Network.

Initially, all neuron weights in an ANN are assigned to random values. In supervised learning, the input values are typically the attributes of an input instance x_i. The output value of the ANN can provide a classification Y_i for the given instance. In a binary classification problem, the output layer consists of a single neuron, whereas in a multi-class problem each class can be represented by an output layer neuron, where the neuron which outputs the highest value is selected for classification. The classifications of the network are evaluated, and the weights are re-calculated in order to minimise the training loss. A common approach for minimising the loss function is the use of the Backpropagation method with gradient descent optimisation. The main idea behind Backpropagation is as follows: If the activation functions of a network are differentiable, then the activations of its output units will be differentiable functions of its inputs and weights. Accordingly, if the error function used

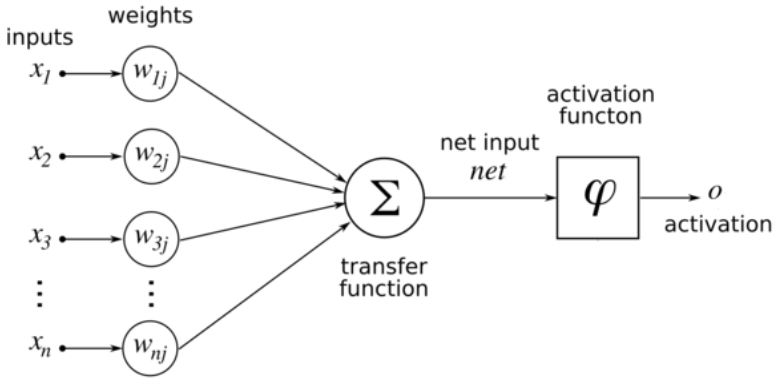


FIGURE 1.3: Artificial Neuron

for its training is a differentiable function of its outputs, such as the sum of squares-error, then the errors produced will be differentiable functions of the weights. The Backpropagation method provides a computationally efficient way for evaluating the derivatives of the error function with respect to the weights. These derivatives can then be fed into an optimisation technique, such as gradient descent, in order to produce weight values which minimise the error. The process of Backpropagation is repeated until the training loss (given a training set of instances) is minimised to a threshold value, or when a terminating criterion has been met. Once the process is finished, the ANN is said to have been trained and can therefore output classification predictions for future instances.

1.1.3 Support Vector Machines

In supervised learning, a Support Vector Machine (SVM) is an algorithm which tries to identify a hyperplane that gives the maximum margin between two classes of data instances. In order to achieve this, SVMs find the boundary

instances (called support vectors) of a class given a candidate hyperplane, and calculate the margin between the boundary instances and the hyperplane. The hyperplane that maximizes the margin between the classes is selected for classification. Figure 1.4 shows a two dimensional separating hyperplane with the boundary vectors being highlighted. The hyperplane is written as $w \cdot x - b = 0$, where w is the normal vector to the hyperplane. The margin is the region between the two hyperplanes $w \cdot x - b = 1$ and $w \cdot x - b = -1$ as shown in Figure 1.4. This margin is described as $2/||w||$, where $||w||$ is the normalised normal vector w, and the goal is to minimise $||w||$, subject to the constraint $y_i(w \cdot x_i - b) \geq 1, \forall(1 \leq i \leq n)$. The constraint prevents data points from falling into the margin. In order to solve the minimization task, the Lagrange multipliers method can be used, which offers a strategy for finding the local minima of a function subject to constraints. When the optimization goal is reached it means that the SVM has found a hyperplane which maximizes the margin from the support vectors.

In the case of multi-class tasks, several binary SVMs can be combined to form a multi-class classifier. In other terms, the multi-class problem is reduced to multiple binary tasks. Each binary SVM can then form a separating hyperplane between one of the classes against the rest (one-against-all), or between every pair of classes (one-against-one). The results of all binary SVMs are then combined to produce a multi-class classifier.

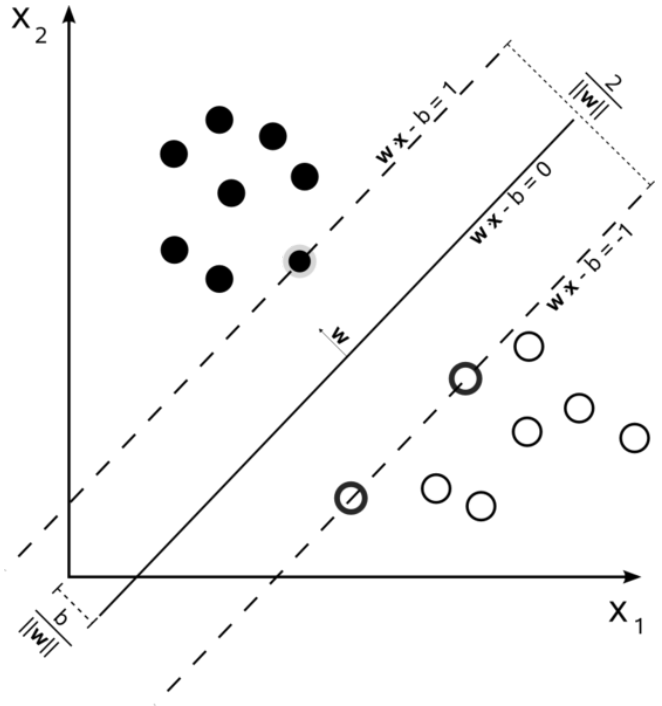


FIGURE 1.4: Support Vector Machines.

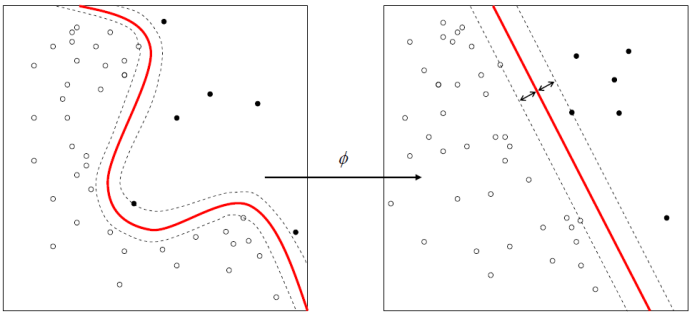


FIGURE 1.5: Kernel mapping.

For non-linear classification, the data are mapped into a higher dimensional space where a linear separation can be made, using a nonlinear kernel map $\phi : X \to V$. A depiction of a kernel mapping is shown in Figure 1.5. The SVM

algorithm is not changed, except that every dot product in the maximization task is replaced by a non-linear kernel map.

1.1.4 Naive Bayes Classifier

The Naive Bayes classifier is a probabilistic classifier which is based on Bayes' theorem. The assumption to be made when using such a classifier is that the attributes of the data instances are conditionally independent. Given a training set of data, the classifier multiplies the probabilities of the attributes given their class, and outputs the probability $P(y_i|x_i)$ of label y_i given instance x_i.

In Naive Bayes, the target function $f : X \mapsto Y$ is translated as $P(Y|X)$. When we apply Bayes rule, the representation is given as:

$$P(Y = y_k|X = x_i) = \frac{P(X = x_i|Y = y_k)P(Y = y_k)}{\sum_{k=1}^{c} P(X = x_i|Y = y_k)P(Y = y_k)}, \tag{1.4}$$

where x_i is an instance vector, and y_k is a classification of all possible c classifications. Assuming conditional independence, we have

$$P(X = x_i|Y = y_k) = \prod_{j=1}^{m} P(X = x_i^j|Y = y_k), \tag{1.5}$$

where x_i^j is the jth attribute of instance x_i. Equation (1.4) is therefore rewritten as

$$P(Y = y_k|X = x_i) = \frac{\prod_{j=1}^{m} P(X = x_i^j|Y = y_k)P(Y = y_k)}{\sum_{k=1}^{c} \prod_{j=1}^{m} P(X = x_i^j|Y = y_k)P(Y = y_k)}. \tag{1.6}$$

Given a new instance x_{n+1}, the Naive Bayes Classifier will output as prediction the label y_{n+1} with the highest probability.

1.1.5 Genetic Algorithms for classification

Genetic Algorithms (GAs) are population based optimisation algorithms, which mimic natural evolution mechanisms such as cross-over reproduction, mutation, and selection. In Machine Learning classification, candidate solutions in the population are parametric classifiers. Initially, a GA generates a random population of possible solutions. Each solution is evaluated against a fitness or objective function, and a selection process selects the fittest individuals in order to create a successive population. Typically, the fittest individuals are selected probabilistically using the following:

$$P(s_i) = \frac{Fitness(s_i)}{\sum_{l=1}^{L} Fitness(s_l)}, \tag{1.7}$$

where L is the number of individuals in the population. For classification, the fitness score is generally defined as the accuracy of an individual over the training set.

Individual solutions are represented as chromosomes. A chromosome can consist of a sequence of binary digits. The candidate individuals are combined to generate offspring through cross-over and mutation operations. In cross-over, the bits of two individuals are mixed together to generate new individuals, whereas in mutation the bits of a single individual are altered randomly. The

entire process of a GA is repeated until a termination criteria is met. Termination conditions are either a predefined number of generations, or when a solution is found that satisfies some minimum threshold.

Algorithm 1: Genetic Algorithm

Input: O, G, L, r, m
O: The objective (fitness) function;
G: The number of generations for termination;
L: The size of the population;
r: The cross-over rate;
m: The mutation rate;
Initialise random population Pop of size L;
for *Individual* s_i *in* Pop **do**
  Find fitness score of individual: $Fitness(s_i) = O(s_i)$;
end
while *generations* $< G$ **do**
  1. Selection: Select $(1 - r)S$ individuals using $P(s_i) = \frac{Fitness(s_i)}{\sum_{l=1}^{L} Fitness(s_l)}$;
  2. Crossover: Probabilistically select $\frac{rL}{2}$ pairs from Pop according to $P(s_i)$, and produce offspring with cross-over operation. Add individuals in Pop_{new};
  3. Mutate: Choose m percent from Pop_{new} randomly and alter randomly selected bits;
  4. Update Pop with Pop_{new};
  5. **for** s_i *in* Pop **do**
    $Fitness(s_i) = O(s_i)$;
  end
end
Output:
$\arg\max_{s_i} O(s_i)$ from final Pop

In Algorithm 1, we formalise a prototype GA. The algorithm accepts as inputs the objective function O, the number of maximum generations G, the population size L, the cross-over rate r, and the mutation rate m. Initially, the GA will generate a random population of chromosomes. The chromosomes will be evaluated against the objective function, which will generate fitness scores

for the chromosomes. The algorithm proceeds by selecting the fittest chromosomes and performs cross-over and mutation to generate new offspring. The new offspring are added to the population and a percentage of non-selected individuals are removed in order to generate a successive population. The process is repeated until the number of generations reaches G. The fittest chromosome from the final population is then selected as the generated solution of the GA. Later in Chapter 2, we provide a more detailed description of a particular GA, where we define the chromosome's representation and the objective function.

1.2 Motivation

A major problem in Machine Learning is the reliability of the predictive outputs of learning algorithms. Although a loss function can indicate the general accuracy of a learning algorithm, the credibility (or reliability) of each individual prediction is unknown. There exist only a few theories to alleviate the problem of reliability in the predictions. Most methods that use such theories are categorised into probabilistic classification or provide some kind of confidence in the predictions. There are currently two major approaches, the Bayesian theory and the Probably Approximately Correct (PAC) theory. Nevertheless, these approaches have some important drawbacks that can hinder application. For Bayesian theory, a priori assumptions need to be made about the data. If prior knowledge is not available, the Bayesian estimated confidence intervals can be misleading. For example, at the 95% of confidence, the loss rate can be much more than the expected 5%. In the case of applying PAC theory,

the data used must be particularly clean, something that is not always true in practical applications.

In this thesis, we investigate the Conformal Prediction (CP) framework [4] which is a novel framework that can be used for obtaining reliable confidence measures in predictive Machine Learning systems. Unlike other approaches to confidence measures, a Conformal Predictor can guarantee that the confidence measures will be valid, and thus the error can be controlled by setting a prior confidence level such that the error rate of the predictions will be bounded by the given level. For example, if the confidence level is pre-set to 95%, then the predictor will output prediction regions to satisfy this level. The overall error of the predictor, i.e. the percentage of times the true label is not included in the output prediction region, will not exceed the 5% that is allowed due to the pre-defined confidence level. In classification, a prediction region may contain more than one label, and the number of labels in the region depends on how certain the predictor is about the data, at the given confidence level. If necessary, Conformal Predictors can be forced to output single predictions instead of regions, and can complement such predictions with valid confidence measures. Under the i.i.d. assumption, a CP is guaranteed to give valid confidence values (i.e. the confidence values of the predictions will be bound to the true classification loss, or regression loss). An elaborate comparison of the Bayesian framework with CP is made in [5]. The weakness of the bounds of PAC methods is presented in [6].

Furthermore, we examine the use of Venn Prediction (VP), which is an extension to the CP framework. Venn Predictors are algorithms that output multi-probability values for each prediction. Similar to CP, the probabilistic

outputs of Venn Predictors are guaranteed to be valid under the i.i.d. assumption. In chapter 3, we compare VP with other probabilistic methods. Unlike VP, the results of currently known probabilistic methods demonstrate that their probabilistic outputs are not always well-calibrated.

1.2.1 Contribution

The contribution of the thesis is divided into three categories. First, we examine the CP framework, and we construct a new CP algorithm based on Genetic Algorithms. Additionally, we extend the CP framework to multi-label classification which is a new area of research in the field of Machine Learning. In multi-label classification, an instance in the data can have multiple output values (or labels). Current multi-label classification algorithms do not provide valid confidence measures in their predictions.

Secondly, we investigate Venn Predictors and we extend the VP framework to Inductive Venn Prediction. An important drawback of VPs is their computational inefficiency, especially in the case of large datasets. In this thesis, we give a description of the Transductive Venn Prediction (TVP) framework, and we introduce Inductive Venn Prediction (IVP) which is a novel approach for improving the computational efficiency of VPs. Inductive CP methods have been successfully used in the past in [7–9].

Thirdly, we conduct thorough experiments on all of our methods, and we apply our methods on three real-world applications: the assessment of the risk of stroke based on ultrasound images of atherosclerotic carotid plaques, the

diagnosis of childhood abdominal pain, and the assessment of the risk of Osteoporosis using data that have been collected from monitored patients.

1.2.2 Publications

Work presented in this thesis has been published by the author in several conference proceedings and international journals. A list of publications is given below.

1. "Inductive Venn Prediction". Annals of Mathematics and Artificial Intelligence, Springer, 2014.

2. "Evaluation of the risk of stroke with Confidence Predictions based on ultrasound carotid image analysis". International Journal on Artificial Intelligence Tools, Vol. 21, World Scientific, 2012.

3. "Reliable Confidence Measures for Medical Diagnosis with Evolutionary Algorithms". IEEE Transactions on Information Technology in Biomedicine, Vol. 15, No. 1, 93–99. IEEE, 2011.

4. "Osteoporosis Risk Assessment with Well-Calibrated Probabilistic Outputs". Artificial Intelligence Applications and Innovations (AIAI 2013), IFIP Advances in Information and Communication Technology Volume 412, 2013.

5. "Calibrated Probabilistic Predictions for Biomedical Applications". In Proceedings of the 12th IEEE International Conference on BioInformatics and BioEngineering (BIBE 2012).

6. "Reliable Probability Estimates Based on Support Vector Machines for Large Multiclass Datasets". In Proceedings of the 1st Workshop on Conformal Prediction and its Applications (COPA 2012), IFIP AICT 382, 182–191. Springer, 2012.

7. "Assessment of Stroke Risk Based on Morphological Ultrasound Image Analysis with Conformal Prediction". In Proceedings in Artificial Intelligence Applications and Innovations, 2010. AIAI2010.

8. "Evolutionary Conformal Prediction for Breast Cancer Diagnosis". In Proceedings of the 9th International Conference on Information Technology and Applications in Biomedicine, 2009. ITAB 2009.

1.3 Thesis Structure

In Chapter 2, we give a detailed description of the CP Framework, and we introduce two new CP methods. In the first method, we propose the use of Genetic Algorithms. In the second method, we introduce Multi-Label Conformal Predictors (ML-CP), a setting which is currently attracting more research interest. We provide experimental results of both introduced methods. The results demonstrate the reliability and efficiency of our methods.

In Chapter 3, we describe the VP framework, and we introduce IVP, which can greatly improve the computational efficiency of VPs in general. We provide experimental results, which demonstrate the reliability of the introduced method, and we compare the efficiency of the results with TVPs and three other probabilistic methods.

In Chapter 4, we apply CP and VP on three medical-diagnostic problems. The first medical problem is the assessment of the risk of stroke based on ultrasound images of atherosclerotic carotid plaques. We describe the problem, and provide details of the dataset that we have used in our research. Further, we provide experimental results of several CP algorithms on the specific problem, and we discuss the importance of the CP framework to the area of the assessment of the risk of stroke. The second medical problem that we investigate is Osteoporosis Risk Assessment based on well known risk factors of Osteoporosis. We describe the problem in detail, and the data that we have collected from actual Osteoporosis patients. We experiment on the collected data with several VP algorithms, and provide results which demonstrate the reliability and efficiency of our methods. Finally, we apply CP and VP on childhood abdominal pain diagnosis. We conduct experiments on real world data for childhood abdominal pain with collaboration with expert physicians, and we discuss the importance of the confidence measures and probabilistic outputs that we provide in our results.

In Chapter 5, we conclude and provide a summary of the thesis. In the conclusion, we also indicate possible work that can be conducted in the future, following the work presented in this thesis.

Chapter 2

Conformal Prediction

In this chapter, we explain in detail the CP framework and introduce a new Conformal Predictor based on Genetic Algorithms. Additionally, we extend the CP framework for multi-label classification. We provide experimental results which demonstrate the reliability of the confidence measures and the efficiency of our methods.

2.1 Introduction

The Conformal Prediction (CP) [4] framework can be used for obtaining reliable confidence measures in Machine Learning applications. The confidence measures are valid under the assumption that the data used are identically and independently distributed (i.i.d.). The CP framework was first proposed in [10] and later improved in [11], [12], and more recently in [13]. CPs are built using classical machine learning algorithms, called underlying algorithms, and

complement the predictions of the underlying algorithms with measures of confidence. Many CPs have been built to date, based on various algorithms such as Support Vector Machines [11], k-Nearest Neighbours for classification [14] and for regression [15], and Random Forests [16]. The computational efficiency of CPs has also been greatly improved using Inductive Conformal Prediction (ICP) [9], as demonstrated in applications to Ridge Regression [17], k-Nearest Neighbours [9], and more recently in applications to Artificial Neural Networks [7, 18]. The CP framework has been successfully applied to medical diagnostic problems, such as ovarian cancer diagnosis [19], breast cancer diagnosis [20], classification of leukaemia subtypes [21], and acute abdominal pain diagnosis [22, 23]. Other applications of CPs include information fusion [24], and feature selection [25].

2.2 Conformal Prediction Framework

Provided a training dataset, CPs output predictions for new instances together with valid confidence measures, based on the assumption that the given data are identically and independently distributed (i.i.d.). CPs generate prediction regions (sets of possible labels for a new instance), such that the error rate of the prediction regions is guaranteed to not exceed a given significance level in the long run. Additionally, CPs can be configured to output single predictions (instead of prediction regions), together with valid confidence measures. We explain how this is done in the following paragraphs.

A training set is of the form $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, where x_i is a vector of real-valued attributes and $y_i \in \{Y_1, Y_2, \ldots, Y_c\}$ is a label given to the instance

x_i. Given a new instance x_{n+1} with unknown label, the target is to find the likelihood of correctness for each possible label $Y_k \in \{Y_1, Y_2, \ldots, Y_c\}$ that can be given to x_{n+1}. A set of steps are performed for each assumed label, in order to obtain the likelihood:

1. The new instance x_{n+1} is appended in the training set together with the assumed label Y_k.

2. An underlying machine learning algorithm is trained on the extended training set

$$\{(x_1, y_1), \ldots, (x_{n+1}, Y_k)\}. \tag{2.1}$$

3. The underlying algorithm is transformed in order to generate a non-conformity score for each of the instances in (2.1). A non-conformity score indicates how different (or strange) an instance x_i is for its given label y_i, compared to the other instances in (2.1). In subsection 2.2.1, we explain how classical machine learning algorithms can be transformed in order to generate non-conformity scores.

4. The following p-value function is used to calculate how likely the assumed label is of being correct:

$$p(Y_k) = \frac{\#\{i = 1, \ldots, n+1 : a_i \geq a_{n+1}\}}{n+1}, \tag{2.2}$$

which compares the non-conformity score a_{n+1} of (x_{n+1}, Y_k) with all the other non-conformity scores of the rest of the instances in the extended training set.

Given the true label y_{n+1}, the p-value function in (2.2) satisfies the following property for all probability distributions P and for any significance level ε:

$$P(p(y_{n+1}) \leq \varepsilon) \leq \varepsilon. \tag{2.3}$$

In fact, the p-value function is a test function which measures how likely the dataset is of being i.i.d. If the p-value is lower than a given ε, it is because we either have non i.i.d. data, or because some event has happened with probability less than or equal to ε. Based on the assumption that the data are i.i.d., we realise that if we include in our predictions all assumed labels that provide a p-value greater than a given significance level ε, then the probability of missing the true label of an instance will be less than or equal ε. In the case that all p-values are less than ε, the label with the highest p-value is included to ensure that that the prediction regions will always contain at least one prediction. This step does not increase the probability of error. The definition of a prediction region is given as

$$R = \{Y_k : p(Y_k) > \varepsilon\} \cup \left\{ arg \max_{k=1,\ldots,c} (p(Y_k)) \right\}. \tag{2.4}$$

Predictions are now called prediction regions, since they may contain more than one possible labels. In the long run, these regions will make errors at a rate of at most ε. Therefore, the confidence is calculated as $1 - \varepsilon$. The formal definition of the CP algorithm is given in Algorithm 2.

By preference, we may output only single labels (forced predictions) instead of prediction regions. In forced prediction, only the label with the highest p-value is given as a prediction, together with a confidence measure which is 1 minus

Algorithm 2: Conformal Predictor

Input: training set $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, new instance x_{n+1}, possible labels $Y_k \in \{Y_1, Y_2, \ldots, Y_c\}$, significance level ε

for $k = 1$ **to** c **do**

Train the underlying algorithm on the extended set $\{(x_1, y_1), \ldots, (x_n, y_n), (x_{n+1}, Y_k)\}$;

Supply the input patterns $x_1, \ldots, x_{n+1}$ to the underlying algorithm to obtain the respective non-conformity scores $a_1, \ldots, a_{n+1}$;

Calculate the p-value $p(Y_k) = \frac{\#\{i=1,\ldots,n+1 : a_i \geq a_{n+1}\}}{n+1}$;

end

Output:

Prediction Region $R = \{Y_k : p(Y_k) > \varepsilon\} \cup \{arg\max_{k=1,\ldots,c}(p(Y_k))\}$

the second largest p-value. The confidence measure indicates how likely the prediction is of being correct, with respect to the rest of the possible labels.

In Table 2.1, we give an example of a prediction region which contains a single label and a prediction region which contains two labels for the 95% level of confidence ($\varepsilon = 0.05$). A prediction region which contains only one label is a certain prediction. The algorithm can be certain for its prediction at the required level of confidence. For the instance x_1, the second p-value is 0.0145, which is less than the significance level 0.05. Therefore, the CP discards the second label at 95% level of confidence and gives a certain prediction, which is the label that obtains the highest p-value. In contrast, for instance x_2, the second largest p-value is 0.1920 and is greater than the significance level. In this case, the second largest p-value cannot be discarded and thus the CP gives an uncertain prediction region with both possible labels at the required level of confidence. If we decrease the confidence level to 80.80% (or lower), we then have a certain prediction (or forced prediction), but the lower confidence in this

case indicates the inability of the algorithm to produce a certain prediction for 95% confidence.

Instance	$\mathbf{x_1}$	$\mathbf{x_2}$
$\mathbf{p(Y_1)}$	0.8623	0.1920
$\mathbf{p(Y_2)}$	0.0145	0.3768
Actual label	Y_1	Y_2
Prediction region for $\varepsilon = 0.05$	$\{Y_1\}$	$\{Y_2, Y_1\}$
Forced prediction	$\{Y_1\}$	$\{Y_2\}$
Confidence	98.55%	80.80%

TABLE 2.1: Example of prediction regions for 95% level of confidence ($\varepsilon = 0.05$) and forced predictions with confidence measures.

2.2.1 Non Conformity Measures

A non-conformity measure is a way of scoring how strange an instance is for its label compared to the other instances that are given in a training set. Every non-conformity measure that we derive defines a CP, and can be used in (2.2) in order to calculate p-values.

Non-conformity measures should be designed in order to produce efficient prediction regions. In general, any non-conformity measure can be used in the p-value function defined in (2.2) without violating the validity property mentioned in inequality (2.3). Nonetheless, our goal is to generate small sized prediction regions at high levels of confidence. For efficiency, our non-conformity measures should have the following two properties:

- Ranking: The non-conformity scores should rank instances according to typicalness w.r.t the training set.

- Diversity: The non-conformity scores should be maximized in diversity w.r.t. the training set.

By using Machine Learning algorithms, we intend to define non-conformity measures that satisfy the aforementioned properties. In this section, we describe how we may derive non-conformity measures using Artificial Neural Networks, Support Vector Machines, Naive Bayes Classification, and k-Nearest Neighbours.

2.2.1.1 Artificial Neural Networks

Artificial Neural Networks (ANNs) are networks of interconnected nodes. Each connection is associated with a weight which determines the intensity of the information travelling through that connection. These weights are adjusted during training to reduce the output error of the network. The output layer of an ANN has a neuron o_k for each possible class, and given an instance x_i we predict the class Y_k corresponding to the output neuron which gives the highest value.

We expect that the more conforming an instance is for its given label, the higher the corresponding o_k value would be. As proposed in [18], we can build a CP based on ANNs (ANN-CP) using the non-conformity measure

$$a_i = 1/o_t, \tag{2.5}$$

for any (x_i, y_i) where $y_i = Y_t$, and $o_t = [0, 1]$. Alternatively, we can use the following non-conformity measure which is again defined in [18]:

$$a_i = \frac{max_{k=1,\ldots,c:k\neq t} o_k}{o_t \gamma}. \tag{2.6}$$

In this definition, we use as a numerator the maximum of the output units which do not correspond to the label of the given instance, since a higher value from such units indicate a more strange instance. The γ parameter is a constant that is used to adjust the sensitivity of the output.

2.2.1.2 Support Vector Machines

Support Vector Machines (SVMs) identify boundary instances for each class, and fix a separating hyperplane that maximises the margin between them. In the case of a non-linear separation, SVMs use a kernel mapping function, where the instances are mapped to a higher dimensional space such that a linear separation can be made. For the purpose of building a CP using SVM (SVM-CP), we use the distance of each instance from the separating hyperplane together with the class that it belongs to, in order to produce non-conformity scores. For $Y = \{-1, 1\}$, we use the non-conformity measure

$$a_i = -y_i h(x_i), \tag{2.7}$$

where $h(x_i)$ is the output of the SVM for the given instance x_i. The output $h(x_i)$ is negative if the instance belongs to class -1, and positive if it belongs to class 1. If the prediction is correct, then the further the instance is from the hyperplane, the less the non-conformity score will be, since more typical

instances should be further away from the hyperplane. In contrast, if the prediction is incorrect, the non-conformity score will increase as the distance from the hyperplane increases.

The original SVM works only for binary classification problems. We can use a more general SVM-CP (defined in [4]), which can work for both binary and multi-class problems. For a general approach, we can use the one-against-the-rest procedure, which transforms a multi-class problem into several binary sub-problems. A class is selected for each sub-problem, and the instances are labelled with {-1,1} depending on whether they belong to the selected class. A non-conformity score is generated for each instance in each sub-problem, using (2.7). Finally, the average of the scores of each instance is calculated to get a final non-conformity score.

2.2.1.3 Naive Bayes Classifier

The Naive Bayes Classifier (NBC) is named after Bayes' Theorem, and the "naive" assumption of attribute independence. The assumption that attributes are independent is a simplistic one. Nevertheless, Naive Bayes works very well on many real-world datasets, particularly when combined with attribute selection procedures that remove redundant, and hence nonindependent, attributes. The classifier calculates and multiplies the probabilities of the attributes given each class, and outputs the probability of label y_i given instance x_i. We can use the output probability to define a non-conformity measure and build a CP based on the NBC (NBC-CP):

$$a_i = -P(y_i|x_i). \tag{2.8}$$

As $P(y_i|x_i)$ increases, the instance is less strange (or more typical), since the probability assigned by the NBC to the correct class is higher.

2.2.1.4 Nearest Neighbours

The k-Nearest Neighbours (k-NN) method computes the distance of a test instance from the other instances that are provided in the training set, and finds its k nearest instances. The prediction of the algorithm is the class of the majority of the k instances. In the case of building a CP based on k-NN (k-NN-CP), we use the distances of the k nearest instances to define a non-conformity measure. The simplest approach is to calculate the total of distances of the k instances that belong to the class of instance x_i, since the nearer the instance is to its class, the less strange it is. Nonetheless, for a more efficient non-conformity measure we also take into consideration the distances of the k nearest instances that belong to other classes, since the nearer the instance x_i is to the other classes the more strange it is. We build a k-NN-CP using the non-conformity measure defined in [9, 14]:

$$a_i = \frac{\sum_{j=1,\ldots,k} s_{ij}}{\sum_{j=1,\ldots,k} o_{ij}}, \tag{2.9}$$

where s_{ij} is the jth shortest distance of x_i from the instances of the same class, and o_{ij} is the jth shortest distance of x_i from the instances of other classes.

2.3 Genetic Algorithm approach

In this section, we propose a Genetic Algorithm which can be used to generate non-conformity scores. A Genetic Algorithm (GA) is a search method for optimization and search problems. GAs have been used widely for evolving decision rules in order to make predictions for medical diagnosis and other applications. Particularly, fuzzy systems have been used together with GAs in [26] for breast cancer diagnosis, and in [27, 28] for more general applications. We are interested in evolving fuzzy-systems for our GA approach, since we wish to generate decision rules that can output degrees of membership, and thus be able to produce non-conformity scores based on the typicalness (or strangeness) of the instances.

GAs are inspired by natural evolution: a population of encoded candidate solutions (called "chromosomes") is evolved through generations using genetic-like operations, such as crossover and mutation. At each generation, solutions are selected probabilistically based on their fitness, in order to generate offspring and create the next generation. The initial population is generated randomly, and at each generation every candidate solution is evaluated against an objective function in order to gain a fitness score. In a learning system, the objective function is typically the measure of the performance of a candidate solution over a training set of instances.

In our work, we use the "Pittsburgh" approach [29], where each "chromosome" in the population is a rule-set, and each rule-set is composed by a variable number of rules. We evolve a population of rule-sets for each class of the data for a given number of generations, and then we select the best rule-set from

each of the final populations in order to form the final model. Next, we describe how we have developed our GA, and later in this section we explain how we have built a CP using our GA approach.

2.3.1 Chromosome representation

We use fuzzy rules for our GA implementation, since fuzzy rules can give degrees of membership and are useful for calculating non-conformity scores. The rules are connected with the fuzzy-OR operator to form rule-sets. A "chromosome" represents a fuzzy rule-set of the form:

```
IF rule1 OR ... OR ruleR THEN consequent.
```

Each rule is composed by one or more simple fuzzy expressions connected by fuzzy operators, and *consequent* is an expression that assigns a value to the output of the rule-set. The way a fuzzy rule is determined, and the respective fuzzy operators are explained later in this section. The output value denotes the membership of a given instance to the class of the rule-set. Each rule tests all J attributes of an instance, and each attribute is represented by L binary bits in the "chromosome". The number of bits for each attribute is defined by the number of fuzzy-sets, which are described next. A complete "chromosome" has size $s = R \times J \times L$ bits, where R is the number of rules in the rule-set.

2.3.1.1 Fuzzy sets

Fuzzy rules are determined by a fuzzy space. A fuzzy space is defined by L fuzzy-set membership functions. Figure 2.1 depicts 5 triangular fuzzy-sets with linguistic names: Small ("S"); Small-Medium ("SM"); Medium ("M"); Medium-Large ("ML"); and Large ("L"). The input of the fuzzy space in Figure 2.1 is a real-valued attribute within the range [0, 1]. The output of the fuzzy space is the set of the membership values of the fuzzy-sets for the given input value. For example, given an input value of 0.2, the output of the fuzzy space in Figure 2.1 will be $\{0.2, 0.8, 0, 0, 0\}$. In our GA implementation we use the depicted fuzzy-space in Figure 2.1, which contains 5 triangular membership functions. Other membership functions can be used, such as the Gaussian membership function

$$\mu_l(x) = \exp\left(-\frac{(c_l - x)^2}{2\sigma_l^2}\right), \tag{2.10}$$

where c_l and σ_l are the centre and width of the lth fuzzy set respectively. Nonetheless, the Gaussian membership function requires more processing to be calculated. For simplicity purposes, we have excluded other membership functions from our GA implementation.

2.3.1.2 Fuzzy rules

A fuzzy rule indicates the fuzzy-sets considered for some real-valued attributes of an instance x_i (each attribute denoted as x_{ij}). In our work, the considered fuzzy-sets are connected with the union operator in order to have a single output value instead of a set of membership values. Moreover, the resulting

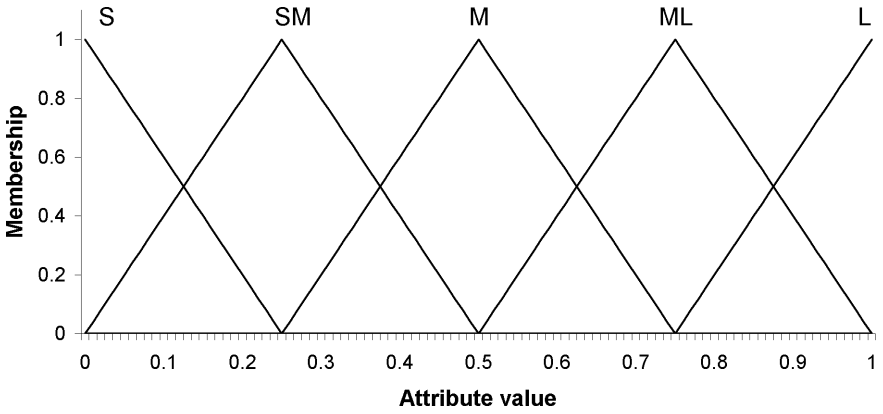


FIGURE 2.1: Triangular fuzzy-set membership functions within a fuzzy space for real valued attributes of the range [0, 1].

outputs of all of the attributes in each rule are connected with the fuzzy-AND operator, and the resulting outputs of several rules (which form a rule-set) are connected with the fuzzy-OR operator. The fuzzy-AND operator can be described as the minimum output, and the fuzzy-OR operator is the equivalent maximum function. For the fuzzy space in Figure 2.1, a rule-set of two rules with two attributes can be the statement in Figure 2.2.

IF [($x_{i1} \in$ "S" $\cup$ "SM") AND ($x_{i2} \in$ "ML")] OR
[($x_{i1} \in$ "S") AND ($x_{i2} \in$ "L")]
THEN
output = class [weight]

FIGURE 2.2: A rule-set of two fuzzy rules with two attributes in each one.

The first rule in the rule-set of Figure 2.2 takes into account the memberships of attribute x_{i1} to the "S" or "SM" fuzzy-sets, and for attribute x_{i2} the membership to the "ML" fuzzy-set. The second rule requires the membership of

attribute x_{i1} to the S fuzzy-set, and for attribute x_{i2} the membership to the "L" fuzzy-set. The rule-set in Figure 2.2 can be represented as a "chromosome" of binary bits:

$$\underbrace{11000}_{g_1}\underbrace{00010}_{g_2}\ \underbrace{10000}_{g_1}\underbrace{00001}_{g_2},$$

where each g_j is a "gene" of the "chromosome" requiring the fuzzy-set memberships of attribute j. Each "gene" contains one bit for each fuzzy-set, and the l-th bit in a "gene" represents the l-th fuzzy-set in the fuzzy space. If the membership of a fuzzy-set is required, the corresponding bit is enabled by setting it to 1, otherwise the bit is set to 0. The fuzzy logical operators of the rule are not encoded in the "chromosomes", since these are pre-defined. The class of the rule-set is also not encoded in the "chromosomes", since we evolve a population of rule-sets for each class separately.

2.3.1.3 Output weight

Here, we formally define the output of a rule-set. The union membership β of a real-valued attribute x_{ij} to "gene" g_j is calculated as the sum of memberships of the enabled fuzzy-sets:

$$\beta_{g_j}(x_{ij}) = \sum_{l=1}^{L} \left[\mu_l(x_{ij}) \times g_{jl}\right], \tag{2.11}$$

where $g_{jl} = 1$ for the enabled fuzzy-sets, and $g_{jl} = 0$ for the disabled fuzzy-sets. The $\mu_l(x_{ij})$ function returns the membership degree of attribute x_{ij} to the lth fuzzy-set, given a fixed fuzzy-space.

Further, the membership degrees β of all attributes J of instance x_i are combined, to find the compatibility degree of the instance x_i to the rule r:

$$\lambda_r(x_i) = \min\left\{\beta_{g_j}(x_{ij}) : j = 1, ..., J\right\}, \tag{2.12}$$

where the minimum function is the equivalent fuzzy-AND operation. Finally, the output of a rule-set containing R rules given instance x_i is defined as the weight:

$$w(x_i) = \max\left\{\lambda_r(x_i) : r = 1, ..., R\right\}, \tag{2.13}$$

where the maximum function is the equivalent fuzzy-OR operation.

2.3.2 Genetic Operations

Genetic Operations are carried out during the evolution process of the Genetic Algorithm. Crossover and mutation are used for the generation of new "chromosomes" and the objective function is used for evaluating each "chromosome". Finally, the termination criteria are explained, which are responsible for terminating the GA after a number of generations. The details of these operations are explained in the following sections.

2.3.2.1 Crossover

We use two point variable size crossover between two "chromosomes" in order to generate two offspring "chromosomes". In two point crossover we randomly

1st parent with one rule	10100 10111
2nd parent with two rules	11001 01101 10011 01010
1st offspring	10001 01101 10011 01111
2nd offspring	11100 10010

TABLE 2.2: Crossover between two parents with different number of rules

take two points, a and b which are used as follows: for generating the first offspring, we accumulate the first bits from positions 1 to a of the first parent, the bits from point a to point b of the second parent, and the bits of the first parent from point b to the end of the "chromosome". For generating the second offspring, we apply the same operation with the only difference that the positions of the parents are swapped.

When the two parents have different number of rules, we restrict the points we can select in the "chromosomes" in a way that the resulting offspring will always be valid "chromosomes". The corresponding points have to be on the same bits within each rule of the "chromosome". In Table 2.2, we demonstrate an example of crossover between two parents with different number of rules. The underlined bits denote the bits that are selected for generating the first offspring, whereas the rest of the bits will generate the second offspring. Notice for the second parent, that the first swapping point is identical as the first parent, and the second point is given to the second rule.

2.3.2.2 Mutation

We have additionally used a mutation operation in our GA. Mutation is a simple flip operation of a random bit in a "chromosome", and the probability of the flip operation to happen is given by the mutation rate. The mutation

operation is greatly important in GAs, since populations otherwise tend to converge to local maxima due to the crossover operation.

2.3.2.3 Objective function

The objective function measures the accuracy of a rule-set based on a training set of instances. To calculate the accuracy of a rule-set, we evaluate the rule-set against all training instances and we find the weight value returned for each instance. For the instances with the same class as the rule-set, such that $y_i = k$ the following formulae are used to calculate the True Positives (TP) and False Negatives (FN):

$$TP = \sum_{i=1,..,n+1:y_i=k} \sqrt{w^k(x_i)} \tag{2.14}$$

$$FN = \sum_{i=1,..,n+1:y_i=k} \sqrt{1 - w^k(x_i)}. \tag{2.15}$$

Regarding the examples which belong to a different class from the rule-set, such that $y_i \neq k$, we apply the same formulae, (2.14) and (2.15), for False Positives (FP) and True Negatives (TN) respectively. The fitness score of the rule-set after processing the training set is defined as

$$fitness = \frac{TP}{TP + FN} + \frac{TN}{FP + TN}. \tag{2.16}$$

2.3.2.4 Termination Criteria

The main termination criterion of a GA is the number of generations required until a candidate decision rule can be chosen as a final decision rule. The number of generations can be decided and pre-defined by the user. Another approach is to let the algorithm decide when a good enough solution has been found.

In order to achieve a termination decision automatically, we specify a convergence criterion. In detail, the termination criterion method obtains the score of the fittest decision rule at each generation, and if the score converges (i.e. there is no major increase or decrease in the fitness score for some pre-defined number of generations) then the algorithm terminates and the fittest rule is selected as the final decision rule.

2.3.3 GA Classifier

The GA can predict the label of a given instance by selecting the class which gives the highest weight (using the corresponding evolved rule-set of each class of the data). In this case, we use a modified version of the weight function in (2.13), which includes the quality of each rule in the evolved rule-set:

$$w_e(x_i) = \max \{\lambda_r(x_i)Q : r = 1, ..., R\} \tag{2.17}$$

We calculate the quality Q of each rule over the training set as the proportion of the compatibility degrees of the instances that have the same label as the

rule, over the sum of the compatibility degrees of all the instances. The defined quality has allowed us to increase the accuracy of our GA classifier.

2.3.4 GA-CP

In order to build a CP using our GA method (GA-CP), we need to define non-conformity measures. The fittest rule-sets evolved from the GA can generate non-conformity scores for given instances. The most natural way to define a non-conformity measure for a pair (x_i, y_i) is to reverse the weight function of the rule-set, since the higher the weight the less strange (or more typical) the instance would be for its given label. Therefore, we define:

$$a_i = -w_e^{y_i}(x_i), \tag{2.18}$$

where $w_e^{y_i}$ is the weight of the corresponding (evolved) rule-set of the class given to instance x_i. Other non-conformity measures can be defined with slightly more sophisticated calculations. Generally, we would like to use as much information we can derive from the rule-sets. Therefore, we may include the weights returned by all of the rule-sets which are evolved for each class. Our second definition of a non-conformity measure is:

$$a_i = \frac{\sum_{k=1,\ldots,c:c\neq y_i} w_e^k(x_i)}{w_e^{y_i}(x_i) + \gamma}, \tag{2.19}$$

where $w_e^{y_i}(x_i)$ is the weight of the rule-set for the class y_i of the given instance x_i, c is the number of all possible labels, and γ is a chosen constant which

adjusts the level of sensitivity to changes of $w_e^{y_i}(x_i)$. We have also defined a non-conformity measure based on the same information as above, using subtraction instead of division, as it is a smoother function to changes of $w_e^{y_i}(x_i)$:

$$a_i = \left[\sum_{k=1,\ldots,c:c\neq y_i} w_e^k(x_i) \right] - w_e^{y_i}(x_i)\gamma. \tag{2.20}$$

We can use non-conformity measures (2.18), (2.19), or (2.20) to provide predictions with confidence levels, or provide sets of predictions (prediction regions) for instances, given desirable error rates. The method is defined as follows: the new instance to be predicted is appended in the training set, together with an assumed label. The GA is applied on the extended training set and the fittest rule-set for each class after a given number of generations is selected. Then, based on the evolved rule-sets, a non-conformity score is calculated for each instance using (2.18), (2.19) or (2.20). The p-value of the assumed label is generated. The same process is repeated for every possible label of the new instance, and a set of p-values is generated. Finally, we may output a prediction region for any level of confidence as described in section 2.2. The implementation of our GA-CP is publicly available at [30].

2.4 Multi-label Conformal Prediction

In this section, we extend the CP framework for multi-label classification. In multi-label classification an instance can belong to multiple classes in parallel. Applications include image tagging, document classification, gene function

categorization, and music classification. For example, in document classification, a specific document which contains both religious and political issues can have both labels: one label for class "politics", and one for class "religion". Multi-label algorithms are generally categorized into two groups based on the transformation method that is used. One group is using Pattern Transformation (PT), where the multi-labelled data are split into several single labelled data, and then traditional machine learning algorithms can be applied for classification. The second group is using Algorithmic Adaptation (AA), where the underlying algorithm is transformed in order to construct a mutli-label classifier. An overview of multi-label classification is provided in [31]. In a related study, a CP was developed for multi-label classification using powersets [32]. The powerset method (which falls into the PT group) transforms the multi-label classification task into single label classification by mapping each combination of the available labels into single labelled classes. Another study, which follows another PT approach, can be found in [33] and [34], where the multi-labelled data are decomposed into multiple binary labelled datasets (Binary Relevance approach), and a CP is applied on each subset.

Unlike the work found in the literature, we propose a confidence measure using the Hamming loss metric, which is the most common evaluation measure in the setting of multi-label classification. Our proposed confidence measure allows us to produce multi-label prediction regions with at most ε chance of having a Hamming loss more than some threshold h. In other words, we can guarantee under i.i.d. assumption, that Hamming loss in our multi-label predictions will not exceed h given some confidence $1-\varepsilon$. In the next section, we describe the developed Binary Relevance Multi-Label Conformal Predictor (BR-MLCP),

and we provide an upper bound of Hamming loss using the CP framework and Chebychev's inequality.

2.4.1 Developed Algorithm

In multi-label classification, a training set of the form $D = \{(x_1, \psi_1), ..., (x_n, \psi_n)\}$ is given, where x_i is an input vector of real-valued attributes, and the instances can be labelled as $\psi_i \subseteq \{Y^1 \times Y^2 \times ... \times Y^c\}$, where each $Y^k \in \{y_k^1, y_k^0\}$. Instances that belong to class Y^k are labelled y_k^1, and y_k^0 otherwise. One possible approach to solve a multi-label problem is to decompose it into c single-label binary classification problems (Binary Relevance approach in [34]). The original dataset D is copied into datasets $D_1, ..., D_c$, and for each D_k we label as y_k^1 the instances that originally have label y_k^1 in the multi-label ψ_i, and y_k^0 otherwise.

We use a CP on each dataset D_k separately, and given a new instance x_{n+1} and a desirable significance level ε_k, each CP provides a prediction region r_k for class Y^k (as in usual single-label classification). The prediction region r_k states whether the new instance belongs to class Y^k or not, or whether there is uncertainty at the given significance level. We then combine all r_k to provide the prediction region for the multi-label classification task:

$$R = r_1 \times ... \times r_c. \tag{2.21}$$

As shown in property (2.3), the probability of each r_k missing the true binary label for class Y^k, given a significance level ε_k, is at most ε_k. According to the

Bonferroni general inequality, which can be applied for multiple tests, we may state that the probability of the true multi-label ψ_{n+1} not being in R is at most the sum of the upper bound probabilities of the individual r_k sets missing the true binary label:

$$P(\psi_{n+1} \notin R) \leq \sum_{k=1}^{c} \varepsilon_k. \tag{2.22}$$

Therefore, we have multi-label prediction regions, for which the error rate is

$$\varepsilon \leq \sum_{k=1}^{c} \varepsilon_k. \tag{2.23}$$

Consequently, for a confidence level $1-\varepsilon$ in R we set the significance level for each $r_k, k = 1, \ldots, c$ to

$$\varepsilon_k = \frac{\varepsilon}{c}. \tag{2.24}$$

Alternatively, we may set each ε_k to the second largest p-value provided by each CP, such that each r_k contains a single prediction for the new instance. Thus, the final prediction region R will also contain a single multi-label, which can be considered as a forced prediction for the new instance. The prediction of the multi-label can be complemented with confidence measure $1-\varepsilon$. The algorithm of the Binary Relevance Multi-Label Conformal Predictor (BR-MLCP) is given in [34] and in Algorithm 3. Our implemented version is publicly available at [35].

2.4.1.1 Prediction Regions with Hamming loss

In inequality (2.23), we consider the error rate with respect to each multi-label prediction as a whole prediction. If the prediction contains even a single binary

Algorithm 3: Binary Relevance Multi-Label Conformal Predictor (BR-MLCP[34])

Input: training set $D = \{(x_1, \psi_1), ..., (x_n, \psi_n)\}$, new instance x_{n+1}, possible labels $\{Y^1, Y^2, ..., Y^c\}$, significance level ε

for $k = 1$ **to** c **do**

  for $b = 0$ **to** 1 **do**

    $D_k = \left\{(x_1, Y_1^k), ..., (x_n, Y_n^k), (x_{n+1}, y_k^b)\right\}$;

    Train the underlying algorithm on the extended set D_k;

    Supply the input patterns $x_1, \ldots, x_{n+1}$ to the underlying algorithm to obtain the respective non- conformity scores $a_1, \ldots, a_{n+1}$;

    Calculate the p-value $p(y_k^b) = \frac{\#\{i=1,\ldots,n+1 : a_i \geq a_{n+1}\}}{n+1}$;

  end

  $r_k = \left\{y_k^b : p(y_k^b) > \varepsilon/c\right\} \cup \left\{arg\max_{k=1,\ldots,c}(p(y_k^b))\right\}$;

end

Output:

Prediction Region $R = r_1 \times ... \times r_c$.

miss-classification, then the whole multi-label prediction is considered wrong. A more common evaluation metric is used for multi-label prediction, which is the Hamming loss metric. Given two sets a and b their Hamming loss is calculated as

$$H(a, b) = \#\left\{k : a_k \neq b_k\right\}. \tag{2.25}$$

Given the true multi-label ψ_i of an instance x_i, and a prediction region R_i, the Hamming loss of R_i is defined as

$$HL(\psi_i, R_i) = \min_{\pi \in R_i} H(\psi_i, \pi). \tag{2.26}$$

We can state that an error occurs only when the Hamming loss of a prediction region is above a pre-defined value. Let us denote for $k = 1, ..., c$, $e_k = 1$ if

there is a loss in the prediction of class Y^k, and $e_k = 0$ otherwise. By setting the significance level of the kth CP to ε_k for $k = 1, \ldots, c$, we have

$$P\left(e_1 = 1\right) \leq \varepsilon_1; ...; P\left(e_c = 1\right) \leq \varepsilon_c. \tag{2.27}$$

If we allow a Hamming loss level h, then the overall prediction is wrong when $e_1 + ... + e_c \geq h + 1$ by definition. As a result of (2.27), the expected value of $e_1 + ... + e_c$ is at most $\varepsilon_1 + ... + \varepsilon_c$. Consequently, by Chebyshev's inequality we get:

$$P\left(e_1 + ... + e_c \geq h + 1\right) \leq \frac{\varepsilon_1 + ... + \varepsilon_c}{h + 1}. \tag{2.28}$$

In order to show that the upper bound in (2.28) is optimal, let us assume the case where $\varepsilon_1 = \cdots = \varepsilon_c$, and

$$P\left(\sum_{k=1}^{c} e_k = m\right) = 0, \tag{2.29}$$

for $m > 0$ and $m \neq h + 1$. This means that the probability of each possible combination of exactly $h + 1$ losses becomes

$$\frac{\varepsilon_k}{C_{c-1}^h}, \tag{2.30}$$

since each e_k has at most probability ε_k and this is divided between the C_{c-1}^h possible combinations of other losses, which together with e_k result in exactly $h + 1$ losses. There are C_c^{h+1} possible combinations that give exactly $h + 1$ losses, therefore the total probability of having a Hamming loss of more than h is

$$\frac{\varepsilon_k}{(C_{c-1}^h)} \cdot C_c^{h+1} = \frac{c\varepsilon_k}{(h + 1)}. \tag{2.31}$$

This is equal to (2.28) when $\varepsilon_1 = \cdots = \varepsilon_c$.

As a result of (2.28) in order to produce multi-label prediction regions with at most ε chance of having a Hamming loss more than h, the significance level of the kth CP for $k = 1, \ldots, c$ should be set to

$$\varepsilon_k = \frac{\varepsilon(h+1)}{c}. \tag{2.32}$$

An alternative measure of error could be set with Hamming loss. Let us define Hamming loss HP as the percentage of errors amongst all predicted labels. By property (2.3), the probability of each loss in $H(\psi_i, R_i)$ is at most ε_k, and using equation (2.22), the percentage of Hamming loss is

$$HP \leq \sum_{k=1}^{c} \frac{\varepsilon_k}{c}. \tag{2.33}$$

Therefore, we may set our multi-label CP to provide prediction regions such that the percentage of Hamming loss will be at most HP, at a confidence level $1 - HP$.

2.5 Experiments

In this section, we evaluate the methods presented in sections 2.3 and 2.4, and we provide experimental results. There are four sets of experiments, each done on a different dataset. The first and second sets of experiments evaluate the Genetic Algorithm Conformal Predictor (GA-CP), and the third and fourth sets of experiments evaluate the Multi-Label Conformal Predictor (MLCP).

#	Attribute name	Values
1	Clump Thickness	1-10
2	Uniformity of Cell Size	1-10
3	Uniformity of Cell Shape	1-10
4	Marginal Adhesion	1-10
5	Single Epithelial Cell Size	1-10
6	Bare Nuclei	1-10
7	Bland Chromatin	1-10
8	Normal Nucleoli	1-10
9	Mitoses	1-10
10	Class	{1 (benign), 2 (malignant)}

TABLE 2.3: Attributes contained in the WBCD data

We have used four separate datasets: the Wisconsin Breast Cancer Diagnosis (WBCD) dataset [36]; the Ovarian Cancer NCI PBSII dataset [37]; and two multi-label datasets, one that can be used for classifying music into emotions [38], and another for yeast (Saccharomyces cerevisiae) gene function classification [39].

2.5.1 Experiments for breast cancer diagnosis

We have conducted experiments on the Wisconsin Breast Cancer Diagnosis (WBCD) dataset [26, 36, 40], which is a popular dataset for the domain of breast cancer diagnosis. The WBCD dataset was recorded at the University of Wisconsin Hospital, and contains attributes which are computed from digitized images of fine needle aspirate of breast mass. The instances in the dataset can be classified as *benign* or *malignant*. The attributes of the instances in the dataset are listed in Table 2.3.

The WBCD dataset that we use here is the same version as in [26], which contains 699 instances. From the 699 instances, we have discarded 16 cases which contain unknown values, thus the results shown in this section are on the remaining 683 instances. Moreover, we have applied a random permutation on the dataset, and we have normalized the attributes to real values within the range [0, 1].

2.5.1.1 Experimental settings and results

We apply ten-fold cross validation on the dataset. In cross validation, we split the dataset into ten equally sized blocks, and repeat our experiments ten times, where each time we leave out one block as the test-set. We evaluate our GA-CP based on the accuracy and confidence of the predictions which were generated using the test-set. In this section, we show the average results of the ten folds.

Our GA classifier (described in section 2.3.3) evolved variable sized rule-sets with a maximum of 4 rules in each rule-set. From the study in [26], it seems that 4 rules are sufficient for the WBCD dataset. Moreover, if we had allowed a maximum of more rules the search space for the GA would increase dramatically and would make the problem much more difficult. Additionally, there is the danger of overfitting when using many rules on a limited training set. The rest of the parameters that we have used for our GA are: population size=100; generations=100; crossover rate=0.8; mutation rate=0.01; and elitism rate=0.2. The elitism rate is the percentage of the population that is copied to the next generation. The crossover rate is the probability that the crossover operation will be applied on two selected "chromosomes". We have also increased the probability of the initial population to contain 1's in the

Method	Accuracy %
Setiono[40]	97.21
Taha and Ghosh[41]	96.19
Pena Reyes and Sipper[26]	97.80
SVM	96.78
Our GA classifier	97.20

TABLE 2.4: Accuracy comparison with other methods on the WBCD dataset.

"chromosomes" to p=0.9. We have identified from empirical results that our GA converges sooner when the rate of 1's is higher. In this case, the rule-sets in the initial population have most of the fuzzy-sets enabled, giving higher membership values in general (see (2.11)). As the GA evolves the population, some fuzzy-sets are ruled out moving from general rule-sets towards specific ones.

In Table 2.4, the accuracy of our GA classifier is compared with the best results from other work which has been conducted on the WBCD dataset. We also include the accuracy results of the Support Vector Machine (SVM) classifier, which is a popular classifier. Our main goal is not to provide better accuracy; instead, we aim to retain accuracy, while we provide some confidence information in each prediction. From the results in Table 2.4, we can confirm that our GA implementation is as accurate as the rest of the methods, giving 97.20% accuracy.

The experiments for the GA-CP (described in section 2.3.4) were done with two slightly different versions of the algorithm. The two versions differ in their termination criteria. We define GA-CP-T, which is the GA-CP algorithm where the number of generations is specified by the user, and GA-CP-C where

the termination of the algorithm is decided automatically upon convergence as described in section 2.3.2.4. We have changed the population size of GA-CP-C to 20, since a smaller population when converging to a solution tends to give better results. The number of generations used by the convergence criterion for comparing the fitness scores was set to 50.

In Table 2.5, we show the certainty and error rates given four confidence levels: 99%; 98%; 95%; and 90%. Here, we compare the confidence measures of GA-CP-T, GA-CP-C and SVM-CP defined in 2.2. We use non-conformity measure (2.20) for the two versions of GA-CP, and $\gamma = 1.25$ which was chosen empirically. Regarding the SVM-CP, we used a Radial Basis Function (RBF) kernel with spread parameter $spread = 1$, and complexity $c = 1$. The certainty in the results is measured in terms of how many "prediction regions" contain only a single prediction (i.e. only a single p-value is above a given significance level for the new instance we wish to predict). Considering that the validity of a CP is given, we are more interested about the certainty in the results, since it is a way to calculate the quality of our confidence measures. Therefore, we wish to have as many certain predictions as possible, given high confidence levels.

The GA-CP-T gives 84.9% certainty rate in the predictions given the 99% confidence level, which is a high rate, but slightly lower than the certainty rate of the SVM-CP. This is still a satisfactory result, as the SVM is a well-known strong classifier. Nevertheless, a GA can rapidly overfit the data depending on the number of generations. Overfitting can greatly affect the certainty rates, but not the validity of the confidence measures. The GA-CP-C which terminates the algorithm automatically, gives a slightly better result of 86.2%

Method	Confidence Level	Certainty	Error
GA-CP-T	99%	84.9%	1.0%
	98%	95.7%	2.2%
	95%	99.1%	4.2%
	90%	99.5%	9.5%
GA-CP-C	99%	86.2%	1.0%
	98%	96.3%	2.2%
	95%	99.5%	4.2%
	90%	99.1%	9.5%
SVM-CP	99%	91.9%	0.8%
	98%	96.9%	1.9%
	95%	100%	4.5%
	90%	100%	9.8%

TABLE 2.5: Certainty and error rates on the WBCD data-set using GA-CP-T, GA-CP-C, and SVM-CP.

certainty rate and satisfactory results for the rest of the confidence levels. The effect of overfitting is more evident in the results of the second set of experiments presented in subsection 2.5.2, where GA-CP-T has performed poorly and GA-CP-C has performed remarkably well.

There are two further observations. First, as the confidence level is increased the certainty rate drops, since less error is allowed at the higher levels of confidence. An uncertain prediction is an indication to the user that a particular instance cannot be predicted with the desirable level of confidence. Secondly, the error rates (which are for the prediction regions that did not contain a correct prediction) confirm the validity of the CP, since for a given confidence level $1 - \varepsilon$ the error rate is near ε.

In Figure 2.3, we show the decoding of the final "chromosome" of one of the ten folds, for class *benign*. The class *benign* seems to be easier for the GA to solve, as the rule-set contains only a single rule, whereas for class *malignant* the final

IF ($x_{i1} \in$ "S" $\cup$ "SM" $\cup$ "M") AND ($x_{i2} \in$ "S" $\cup$ "SM")
AND ($x_{i4} \in$ "S" $\cup$ "SM" $\cup$ "M" $\cup$ "ML")
AND ($x_{i6} \in$ "S" $\cup$ "SM" $\cup$ "M")
AND ($x_{i7} \in$ "S" $\cup$ "SM" $\cup$ "M" $\cup$ "ML")
THEN output = benign

FIGURE 2.3: Final rule-set of one of the ten folds for class *benign*.

Instance #	1	2	3	4
p-value for $Y_1 = benign$	1.0000	0.0049	1.0000	0.0130
p-value for $Y_2 = malignant$	0.0032	1.0000	0.0260	1.0000
Correct prediction	*benign*	*malignant*	*benign*	*malignant*
Confidence in prediction	99.68%	99.51%	97.40%	98.70%

TABLE 2.6: The p-values of 4 instances in the WBCD dataset, as generated by the GA-CP.

rule-set contains 4 rules (not shown here). We would like to emphasize that the rule does not include all attributes, as some attributes in the "chromosome" have all fuzzy-sets enabled, and thus making such attributes irrelevant. In this case, the attributes X_3, X_5, X_8, and X_9 are irrelevant for class benign (the names of the attributes are given in Table 2.3). We also note that the decoding of the rule can be read easier if we use the NOT operator. For an expression like $x_{i4} \in$ "S"$\cup$"SM"$\cup$"M"$\cup$"ML", we may use the transformation $x_{i4} \notin$ "L". The readability of the rules is an advantage of the GA-CP over the SVM-CP, as it offers more information to the user about how the predictions are obtained.

In Table 2.6, we list the results of 4 instances in the dataset generated by our GA-CP, showing the p-value for each assumed label Y_h and the resulting confidence in each prediction. We have chosen 2 instances for each class with different p-values, in order to be able to demonstrate the difference in the

confidence of each prediction. The resulting confidence in each prediction is 1 minus the second largest p-value, as explained in section 2.2, and it is the maximum confidence that can be achieved for giving a single prediction. For example, if we pre-set the confidence level to 99%, both labels will be included in the prediction regions for the predictions of instances 3 and 4 in Table 2.6, and thus giving uncertain predictions for level 99%. Nevertheless, the two p-values in the uncertain predictions can still give an indication (but with lower confidence) of which of the two labels is more likely to be the correct one. In contrast, for instances 1 and 2 we can set the confidence level to 99% and we can reject the second label from the prediction regions, since the second p-value is smaller than the significance level (for 99% confidence the significance level is 0.01).

2.5.2 Experiments for ovarian cancer diagnosis

We have conducted experiments on the UKOPS dataset [37] for ovarian cancer diagnosis, which contains proteomic patterns identified in serum that can distinguish ovarian cancer. The dataset contains 170 healthy and 67 malignant instances, each of which consists of 109 attributes generated by mass spectroscopy. The data is normalised such that each attribute has a real value within the range [0,1].

For the experiments, we have used an attribute selection method in order to reduce the 109 attributes in the data. We have applied the Correlation-based Feature Subset (CBFS) selection method [42], using best-first search. The selection method reduced the data to 13 attributes that had low intercorrelation and high correlation with the 2 classes of the data.

We would like to note that one of the attributes identified in the data is the well known biomarker for ovarian cancer CA-125 (Cancer Antigen). Elevated levels of CA-125 in blood serum can indicate ovarian cancer. The CA-125 attribute was automatically selected by the CBFS algorithm together we the rest of the 12 selected attributes. More studies on ovarian cancer diagnosis have been conducted in [43, 44].

2.5.2.1 Experimental settings and results

We have applied the same strategy for the evaluation of the GA-CP as in section 2.5.1. Nevertheless, we have changed the parameters of the GA to fit with the characteristics of the dataset. For the GA-CP-T, the number of generations has been changed to 150. For both GA-CP-T and GA-CP-C, the population size has been changed to 20, and the number of rules to 3. We have found from empirical results that the GA would overfit when we increased the number of rules or the population size. Moreover, the crossover probability and the selection rate has been changed to 0.7, while the probability of 1s in the initial population has also been changed to 0.7.

In Table 2.7, we compare the GA classifier accuracy with the accuracy of some well known methods that we have applied on the same dataset. All methods achieved similar accuracy with our GA. We used a Radial Basis Function (RBF) kernel with $spread = 0.1$ for the SVM, and for the k-NN we set $k = 5$. These parameters were empirically chosen. The number of False Negatives (FNs) predicted by our method is 10, giving a sensitivity rate of 85.1%, and the number of False Positives (FPs) is 0, giving a specificity rate of 100%. The UKOPS dataset seems to be more complex in relation to the WBCD dataset,

Method	Accuracy %	False Negatives	False Positives
SVM	95.43	11	0
Naive Bayes	95.43	4	7
C4.5	96.20	6	3
k-NN	94.09	13	1
Our GA	95.78	10	0

TABLE 2.7: Accuracy comparison between our method and other methods on the UKOPS dataset with 13 selected attributes.

which can be seen from the accuracy difference between the two datasets. As a result, the lower accuracy is reflected on the certainty rates achieved on this dataset, which are listed in Table 2.8. At 99% confidence, the GA-CP-T has only 52% certain predictions which is much lower than the 84.9% achieved on the WBCD dataset. Nevertheless, if we lower our confidence to 98%, we see an increase of the certainty rates to about 85%. GA-CP-T has not performed as well as the SVM-CP for the 99% of confidence, but for the 95% level and below the results are satisfactory. On the other hand, GA-CP-C has performed extremely well, with a 77% certainty rate for the 99% confidence level. The results show that automatic convergence can play an important role for avoiding overfitting or underfitting in GAs, which highly affects the certainty rates in the results. The validity of the confidence measures is not affected, as it is demonstrated by the resulting error rates.

In Table 2.9, we list the results of 4 chosen instances giving their p-value for each assumed label, together with the resulting confidence of each prediction. For instance 4 in Table 2.9, we have a lower confidence in the prediction since the p-value of the wrong label $Y_1 = healthy$ is higher than usual. Therefore, the predictor is less confident for the first prediction, since the second prediction gives a relatively higher p-value. Even if we set the confidence level to 97%,

Method	Confidence Level	Certainty	Error
GA-CP-T	99%	52.3%	0.8%
	98%	84.8%	2.1%
	95%	96.9%	5.0%
	90%	100%	10.0%
GA-CP-C	99%	77.0%	0.4%
	98%	85.6%	1.7%
	95%	96.2%	4.6%
	90%	100.0%	9.7%
SVM-CP	99%	74.6%	0.7%
	98%	89.8%	1.9%
	95%	98.3%	4.6%
	90%	99.1%	9.2%

TABLE 2.8: Certainty and error rates on the UKOPS dataset using GA-CP-T, GA-CP-C and SVM-CP.

Instance #	1	2	3	4
p-value for $Y_1 = healthy$	1.0000	0.0655	1.0000	0.0337
p-value for $Y_2 = malignant$	0.0047	0.0419	0.0192	1.0000
Actual class	*healthy*	*malignant*	*healthy*	*malignant*
Prediction	*healthy*	*healthy*	*healthy*	*malignant*
Confidence in prediction	99.53%	95.81%	98.08%	96.63%

TABLE 2.9: Listing the p-values of 4 instances in the UKOPS dataset, as generated by our GA-CP.

we would still get an uncertain prediction for this instance. In fact, we cannot get a certain prediction for this instance unless we set the confidence level to at most 96.63%. In contrast, for instance 1 in Table 2.9, we have a high confidence in the prediction, since the second p-value is very low. We have also included instance 2 which is a false negative. We can see that both p-values for this instance are very low, which is a rare event. Such a result may give an indication to the user that such an instance requires further examination before making a final diagnosis.

2.5.3 Experiments on Multi-Label datasets

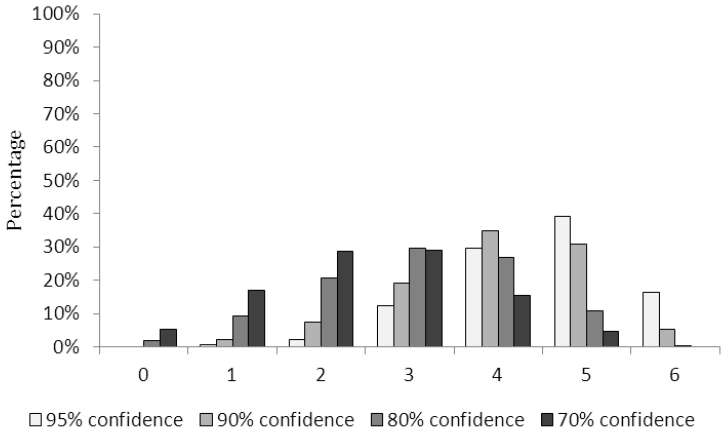
Here, we evaluate the BR-MLCP and the proposed confidence measure. In our evaluation process, we copy the original datasets into binary class datasets as explained in section 2.4.1, and for each subset we apply the Correlation-Based Feature Selection (CBFS) method in [42], in order to reduce the number of features. We then apply 10-fold cross validation on each of the reduced datasets. The folds are identical for all datasets. Each test instance on each dataset is given a possible label (y_k^1 or y_k^0), and the test instance is added to the training set. The underlying algorithm is trained on the extended training-set and provides non-conformity scores. A p-value is then generated for each possible binary label given to the test instance. Once we have p-values from all CPs, we apply equation (2.21) to provide a prediction region for the test instance, given a pre-defined confidence level, or a forced prediction.

2.5.3.1 Music into emotions dataset

We experiment on a multi-label dataset for classifying music into emotions [38]. The Music Emotions dataset contains 593 songs with a total of 72 rhythmic and timbre features in each song. There are 6 possible classes that each song can belong to. The classes and the number of instances in each one are listed in Table 2.10. As baseline, we provide the average Hamming loss of our forced predictions which is 18.77%. This result is comparable with the results provided in [38], which give an overall Hamming loss of 19.43% for the related Binary Relevance algorithm.

Label	Class	# of instances
1	amazed-surprised	173
2	happy-pleased	166
3	relaxing-calm	264
4	quiet-still	148
5	sad-lonely	168
6	angry-fearful	189

TABLE 2.10: Class distribution for the Emotions dataset.



Confidence level	95%	90%	80%	70%
Error rate	4.28%	8.50%	17.34%	26.16%

FIGURE 2.4: Percentages of prediction regions with number of uncertain labels for different levels of confidence, and their respective error rates on the Emotions dataset.

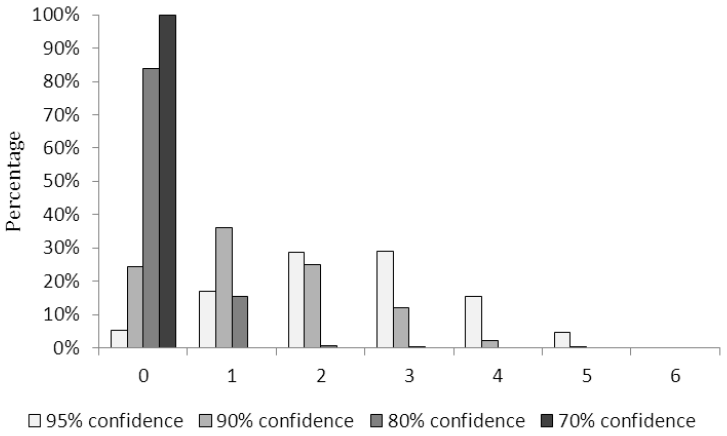
In Figure 2.4, we provide the results of the BR-MLCP using (2.23). The figure shows the distribution of the prediction regions according to the number of uncertain labels, at four different levels of confidence (95%, 90%, 80%, and 70%). When a prediction region has 0 uncertain labels, the size of the prediction region is 1 (contains a single multi-label prediction). When we have 1 uncertain label, the prediction region size is 2, since the region contains a multi-label prediction for each of the 2 possible values of the uncertain binary

label. Generally, for n uncertain binary labels, the prediction region size is 2^n. The error rates presented in Figure 2.4 demonstrate the validity of the BR-MLCP, since they are always below the rate given by the confidence level. Thus, we demonstrate the ability to control the error rate of BR-MLCP. Nevertheless, when we have a high confidence level, we lose some certainty in the predictions. In the figure, we can see that for 95% level of confidence the number of certain predictions is 0, and a significant percentage of predictions contained all 6 labels as uncertain labels. The algorithm provides uncertain results when there is not enough information to give a single result for a given confidence level.

It is admitted that for a multi-label problem, the error measure that was defined in (2.23) is strict. Nonetheless, if we lower the confidence level, we get more certainty in the predictions. For example, at 80% and 70% levels of confidence, we have a significant amount of prediction regions with less uncertain labels.

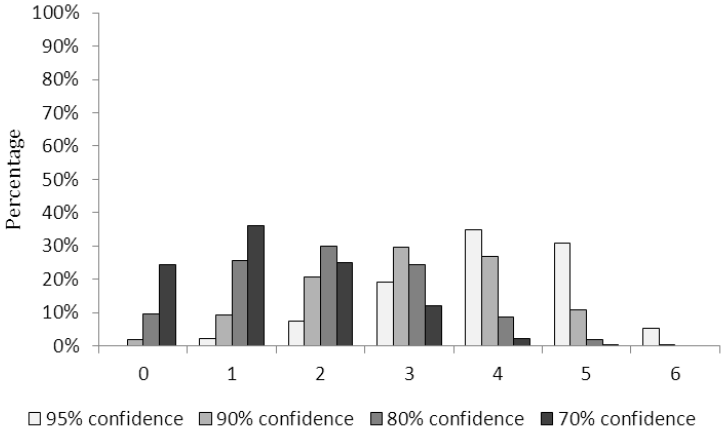
In Figure 2.5, we provide the results of the BR-MLCP using (2.33). Here the error measure is less strict, and thus we get satisfactory certainty in our prediction regions. The error is measured in terms of Hamming loss. As shown in the figure, the Hamming loss in the prediction regions does not exceed the allowed rate given by the confidence level. Therefore, we demonstrate that the BR-MLCP can control the Hamming loss in the prediction regions and provide useful prediction regions. Additionally, the Hamming loss at 70% confidence does not exceed 18.77%. We also notice that at this confidence level, we have 100% certain predictions.

In Figure 2.6, we provide the results of the BR-MLCP using (2.28). Here, we have an error when the Hamming loss h of a prediction region exceeds 1. The



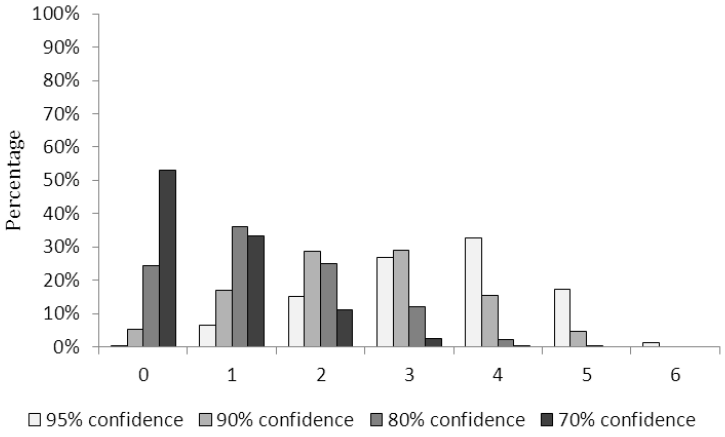
Confidence level	95%	90%	80%	70%
HP	4.87%	9.91%	17.54%	18.77%

FIGURE 2.5: Percentages of prediction regions with number of uncertain labels for different levels of confidence, and their respective Hamming loss on the Emotions dataset.



Confidence level	95%	90%	80%	70%
$h > 1$	0.49%	1.43%	5.53%	12.01%

FIGURE 2.6: Percentages of prediction regions with number of uncertain labels for different levels of confidence, and their respective error rates (using (2.28) with $h = 1$) on the Emotions dataset.



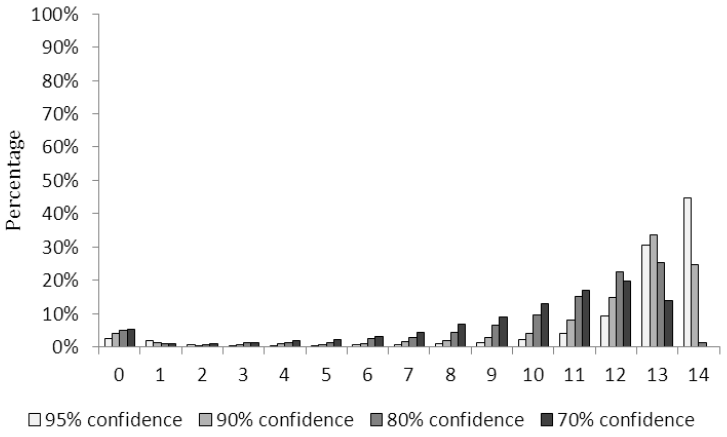
Confidence level	95%	90%	80%	70%
$h > 2$	0.02%	0.19%	1.45%	4.76%

FIGURE 2.7: Percentages of prediction regions with number of uncertain labels for different levels of confidence, and their respective error rates (using (2.28) with $h = 2$) on the Emotions dataset.

results demonstrate the validity of the BR-MLCP using equation (2.28). We consider a multi-label prediction as a correct classification when there is at most 1 wrong label. Thus, we have better certainty in the results compared with the results given in Figure 2.4. In Figure 2.7, we provide the results when we set $h > 2$. As expected, this less strict metric allows for more certainty in the results. At 70% confidence, we have near 50% certain predictions (with 0 uncertain labels), whereas in the previous case when $h > 1$, the certainty at 70% confidence was around 25%.

2.5.3.2 Gene Function Classification dataset

We have experimented on a relatively larger dataset in order to evaluate the BR-MLCP method. We have used a dataset for yeast (Saccharomyces cerevisiae) gene function classification [39]. The dataset contains 2417 genes with

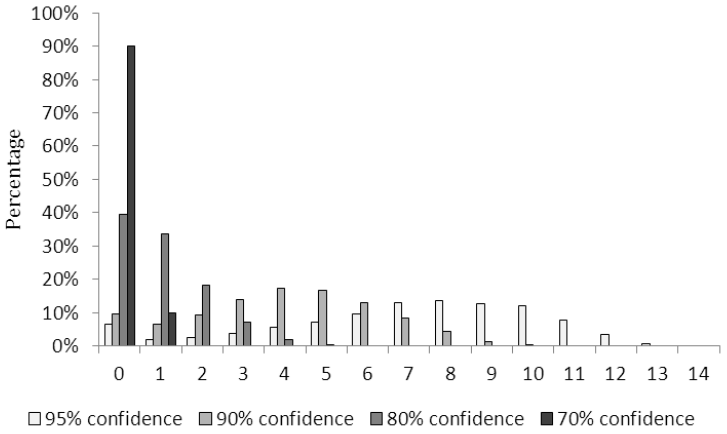


Confidence level	95%	90%	80%	70%
Error rate	3.41%	7.32%	14.72%	20.63%

FIGURE 2.8: Percentages of prediction regions with number of uncertain labels for different levels of confidence, and their respective error rates on the Yeast dataset.

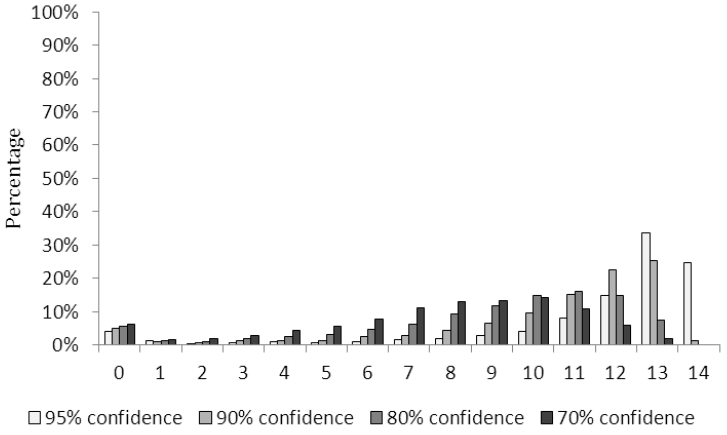
103 features in each gene. Each instance can be classified into 14 possible functional groups. Since one gene can have many functional classes this is a multi-label problem. We apply the same evaluation process on this dataset as with the Emotions Dataset. The baseline Hamming loss with forced predictions on this dataset was 19.32%, which is comparable with the best Hamming loss of 19.5% reported in [39]

In Figure 2.8, we provide the results of the BR-MLCP using (2.23). As shown in the figure, the percentage of prediction regions which contained a certain multi-label prediction is near 5%. This is true for any given confidence level. As explained previously, using (2.23) as an error measure can be very strict for multi-label problems. This becomes more clear when the number of classes is larger. Nevertheless, the BR-MLCP can still provide valid prediction regions, as it is demonstrated by the error rates provided with the results.



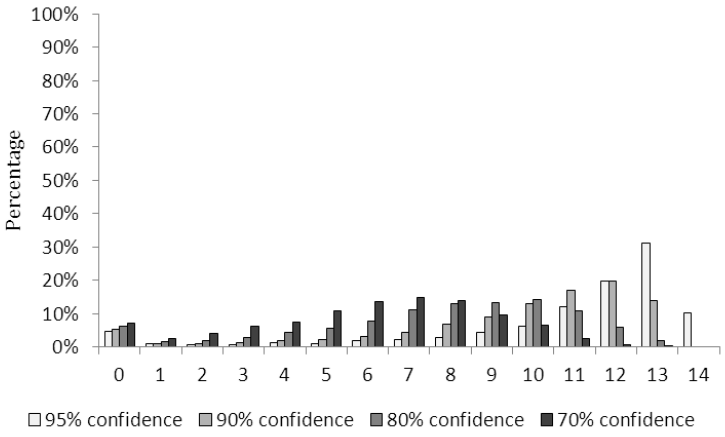
Confidence level	95%	90%	80%	70%
HP	4.71%	9.16%	16.22%	18.99%

FIGURE 2.9: Percentages of prediction regions with number of uncertain labels for different levels of confidence, and their respective Hamming loss on the Yeast dataset.



Confidence level	95%	90%	80%	70%
$h > 1$	1.75%	3.90%	8.97%	14.97%

FIGURE 2.10: Percentages of prediction regions with number of uncertain labels for different levels of confidence, and their respective error rates (using (2.28) with $h = 1$) on the Yeast dataset.



Confidence level	95%	90%	80%	70%
$h > 2$	0.62%	1.41%	4.41%	8.52%

FIGURE 2.11: Percentages of prediction regions with number of uncertain labels for different levels of confidence, and their respective error rates (using (2.28) with $h = 2$) on the Yeast dataset.

In Figure 2.9, we use the Hamming loss measure defined in (2.33). Here, the results are promising. For 70% confidence, we have around 90% of certain multi-label predictions, and for 80% confidence, we have approximately 40% certainty rates. This demonstrates that we can predict for a given number of cases a multi-label classification with Hamming loss less than the given confidence level. As it is expected, the Hamming loss for any given confidence level is below the allowed rate, since we use the CP framework.

In Figure 2.10, we test the BR-MLCP with the Hamming loss measure defined (2.28). As the number of classes is larger, the error measure for $h > 1$ can be considered strict, and thus the results are similar to that of Figure 2.8. The strictness of $h > 1$ loss is also reflected on the error rates which are shown in Figure 2.10. The rates are much lower than the expected allowed rate given by each confidence level. In Figure 2.11 where we set $h > 2$, the results have

slightly improved, yet the number of certain predictions is very low.

2.6 Summary

The CP framework provides reliable measures of confidence to predictions of Machine Learning algorithms. We gave an overview of the CP framework and have given examples of non-conformity measures that can be constructed with the use of conventional algorithms. We have developed and presented a CP based on GAs, and we have applied our method on two medical diagnosis problems: breast cancer diagnosis, and ovarian cancer diagnosis. Our experimental results demonstrate the efficiency of our GA-CP method on both problems. Additionally, we explain the advantage of our GA-CP method over other CPs, which is the easier readability of the generated rule-sets. We have extended the CP framework for multi-label classification, and have applied the defined BR-MLCP algorithm on two multi-label datasets: one for classifying music into emotions, and another for Yeast gene function classification. We have experimented with three measures of error. Hamming loss, which is a widely used measure of error for multi-label problems, was shown to be a more informative measure of error. As it was demonstrated, our proposed confidence measure allows us to reliably control the Hamming loss in our predictions.

Chapter 3

Venn Prediction

In this chapter, we explain in detail the Venn Prediction (VP) framework and propose Inductive Venn Prediction (IVP) based on the idea of Inductive Conformal Prediction (ICP). We provide experimental results which demonstrate the reliability of the probabilistic outputs and the efficiency of our the proposed IVP. We compare the results of our proposed IVP with the results of Logistic Regression (Platt's method), Binning, and Isotonic Regression (IR) methods. Furthermore, we compare and discuss the results of both VP and IVP.

3.1 Introduction

Venn Prediction is a novel machine learning framework that can be combined with conventional classifiers for producing well calibrated multiprobability predictions under the assumption that the data used are identically and independently distributed (i.i.d.). In particular, multiprobability predictions are a set

of probability distributions for the true classification of a new instance (of unknown classification). In effect this set defines lower and upper bounds for the conditional probability of the new instance belonging to each one of the possible classes. These bounds are guaranteed (up to statistical fluctuations) to contain the corresponding true conditional probabilities.

A major drawback of VPs is their computational inefficiency, especially in the case of large datasets. In this chapter, we give a description of the original VP (or Transductive VP) framework, and we introduce Inductive Venn Prediction (IVP) which is a novel approach for improving the computational efficiency of VPs. Inductive methods have been successfully used in the past with CPs in [7–9].

The Transductive VP (TVP) framework has been introduced in [45] where the interested reader can find a detailed description of the framework. Since then, VPs have been developed based on k-Nearest Neighbours [46], Nearest Centroid [47] and Neural Networks [48, 49]. Furthermore, VPs based on SVMs have been developed in [50, 51], and have been compared with three other methods that produce probabilistic outputs. Namely, the three methods are Platt's method [52], Binning [53] and Isotonic Regression [54].

3.1.1 Related Work

Here, we describe Binning, Isotonic Regression, and Platt's method, which are three different methods found in the literature that can provide probabilistic predictions. As it will be shown later in this chapter, these three methods

do not guarantee that the probabilistic outputs will always be well-calibrated. The three aforementioned methods use SVMs as their underlying algorithms.

3.1.1.1 Binning

The binning method [53] sorts the training instances according to their SVM scores, and then divides them into b equal sized sets, or bins, each having an upper and lower bound. Given a test instance x_i, it is placed in a bin according to its classifier score. The corresponding probability $P(Y_j = 1|x_i)$ is the fraction of positive training instances that fall within that bin. There is no imposed lower or upper bound on SVM scores. Therefore, when using this method it is possible for some scores from the test instances to fall below or above the low and high scores, respectively, of the training instances. If this happens the corresponding probability $P(Y_j = 1|x_i)$ is that of the nearest bin to the score of x_i.

3.1.1.2 Isotonic Regression

Isotonic regression has been used in order to map the SVM scores into probability estimates in [54]. An isotonic function $g(i)$ has a monotonically increasing trend, which means that for all i, j:

$$i > j \implies g(i) > g(j) \, and \, i < j \implies g(i) < g(j). \tag{3.1}$$

If the scores of the SVM are ranked correctly, we can assume that the probability $P(Y_j = 1|x_i)$ will be increasing as the SVM scores increase. Therefore,

we can use isotonic regression to map SVM scores into probability estimates. The most common algorithm used for isotonic regression is the Pair-Adjacent-Violators (PAV) algorithm.

The PAV algorithm learns the probability estimate $g(x_i)$ for each ranked instance x_i. First, we set $g(x_i) = 1$ if x_i is a positive instance, and $g(x_i) = 0$ otherwise. If g is already isotonic the function has been learned. Otherwise, there must be an instance where $g(x_{i-1}) > g(x_i)$. The two instances x_{i-1} and x_i are called pair-adjacent violators, because they violate the isotonic assumption. The values of $g(x_{i-1})$ and $g(x_i)$ are then replaced by their average, so that their values no longer violate the isotonic assumption. This process is repeated until an isotonic set of values is obtained. In the end, we have a list of probability estimates together with the adjacent SVM scores of the training instances. When a new instance arrives, we assign the mapped probability estimate based on the score that x_i has obtained from the SVM decision rule. Normally, there will be intervals of scores with the same probability estimates. Since there are no imposed boundaries on the SVM scores, the lowest interval begins from $-\infty$ and the highest interval ends at $+\infty$.

3.1.1.3 Logistic Regression

Platt introduced a method in [52] to estimate posterior probabilities based on the decision function f by fitting a sigmoid to the output value of f:

$$P(Y_j = 1|f(x_i)) = \frac{1}{1 + \exp(Af(x_i) + B)}, \tag{3.2}$$

where $Y_j \in \{-1, 1\}$. The best parameters A and B are determined so that they minimise the negative log-likelihood of the training data. Platt uses a Levenberg-Marquardt (LM) optimisation algorithm to solve this. As indicated in [52], any method for optimisation can be used.

3.2 Venn Prediction Framework

In this section, we describe the VP framework. Typically, we have a training set [1] of the form $\{z_1, \ldots, z_n\}$, where each $z_i \in Z$ is a pair (x_i, y_i) consisting of the object x_i and its classification y_i. For a new object x_{n+1} we intend to estimate the probability of $y_{n+1} = Y_j$ for all possible classifications $Y_j \in \{Y_1, \ldots, Y_c\}$. The main idea behind Venn prediction is to divide all instances into a number of categories and calculate the probability of x_{n+1} belonging to each class $Y_j \in \{Y_1, \ldots, Y_c\}$ as the frequency of Y_j in the category that contains it. However, as we don't know the true class of x_{n+1}, we assign each one of the possible classes to it in turn, and for each assigned classification Y_k we calculate an empirical probability distribution for the true class of x_{n+1} based on the instances

$$\{(x_1, y_1), \ldots, (x_n, y_n), (x_{n+1}, Y_j)\}. \tag{3.3}$$

The VP framework assigns each one of the possible classifications Y_j to x_{n+1} and divides all instances $\{(x_1, y_1), \ldots, (x_{n+1}, Y_j)\}$ into a number of categories based on what is called a *Venn taxonomy*. For $n \in \mathbb{N}$, an n-taxonomy is a measurable function $K : Z^n \times Z \to \mathbf{K}$, where $\mathbf{K}$ is a measurable space, that is

[1] The training set is in fact a multiset, as it can contain some instances more than once.

equivariant with respect to permutations in the sense of

$$i = \pi(i) \implies K((z_1, \ldots, z_n), z_i) = K((z_{\pi(1)}, \ldots, z_{\pi(n)}), z_{\pi(i)}), \tag{3.4}$$

for all $i = 1, \ldots, n$ and any permutation π of $(1, \ldots, n)$. The set $\mathbf{K}$ is usually finite; we will refer to its elements as categories. Every taxonomy defines a different VP. Typically each taxonomy is based on a traditional machine learning algorithm, called the *underlying algorithm* of the Venn predictor. The output of this algorithm for each attribute vector $x_i, i = 1, \ldots, n+1$ after being trained on the set (3.3), is used to assign (x_i, y_i) to one of a predefined set of categories $\kappa_i \in \mathbf{K}$. For example, a Venn taxonomy that can be used with every traditional algorithm puts in the same category all instances that are assigned the same classification by the underlying algorithm. In subsection 3.2.1.2, we define a taxonomy based on the output of the Support Vector Machine (SVM) classifier.

After assigning the category $\kappa_i^{Y_j} = K((z_1, \ldots, z_n, (x_{n+1}, Y_j)), z_i)$ to each instance in the extended set (3.3), the empirical probability of each classification Y_k in $\kappa_{n+1}^{Y_j}$ will be

$$p^{Y_j}(Y_k) = \frac{\left|\{i = 1, \ldots, n+1 | \kappa_i^{Y_j} = \kappa_{n+1}^{Y_j} \ \& \ y_i = Y_k\}\right|}{\left|\{i = 1, \ldots, n+1 | \kappa_i^{Y_j} = \kappa_{n+1}^{Y_j}\}\right|} \tag{3.5}$$

This is an empirical probability distribution for the true class of x_{n+1}. After assigning all possible classifications to x_{n+1} we get a set of probability distributions $P_{n+1} = \{p^{Y_j} : Y_j \in \{Y_1, \ldots, Y_c\}\}$ that compose the multi-probability prediction of the VP. As proved in [4] the predictions produced by any Venn

predictor are automatically valid multiprobability predictions. This is true regardless of the taxonomy of the VP. Of course the taxonomy used is still very important as it determines how efficient, or informative, the resulting predictions are. We want the diameter of multiprobability predictions and therefore their uncertainty to be small and we wish that the predictions are as close as possible to zero or one.

The maximum and minimum probabilities obtained for each label Y_k amongst all distributions $\{p^{Y_j} : Y_j \in \{Y_1, \ldots, Y_c\}\}$, define the interval for the probability of the new instance belonging to Y_k. We denote these probabilities as $U(Y_k)$ and $L(Y_k)$, respectively. The VP outputs the prediction $\hat{y}_{n+1} = Y_{k_{best}}$, where

$$k_{best} = \arg\max_{k=1,\ldots,c} \overline{p(k)}, \tag{3.6}$$

and $\overline{p(k)}$ is the mean of the probabilities obtained for label Y_k amongst all probability distributions. The probability interval for this prediction is $[L(Y_k), U(Y_k)]$. In Algorithm 4 we define the Transductive Venn Predictor algorithm.

3.2.1 Inductive Venn Prediction

Here, we describe the proposed IVP method. The transductive nature of the original VP framework is computationally inefficient, since it requires training the underlying algorithm for every possible class of each new test instance. To address this problem we follow the idea of the Inductive Conformal Prediction, and propose an efficient IVP. Our approach splits the available training instances into two parts, the proper training set with q instances and the calibration set with the remaining $r = n - q$ instances. We then use the proper

Algorithm 4: Transductive Venn Predictor.

Input: training set $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, new instance x_{n+1}, possible classes $\{Y_1, \ldots, Y_c\}$.

for $j = 1$ **to** c **do**

  Assume classification Y_j for x_{n+1}.

  Train the multiclass underlying algorithm on the extended training set $\{(x_1, y_1), \ldots, (x_{n+1}, y_{n+1})\}$;

  Supply the input patterns $x_1, \ldots, x_{n+1}$ to the trained underlying algorithm to obtain the outputs $o_1, \ldots, o_{n+1}$;

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

    Assign κ_i to (x_i, y_i) according to the underlying algorithm classification output o_i;

  end

  for $k = 1$ **to** c **do**

    $p^{Y_j}(Y_k) = \frac{\left|\{i=1,\ldots,n+1 | \kappa_i^{Y_j} = \kappa_{n+1}^{Y_j} \ \&\ y_i \in Y_k\}\right|}{\left|\{i=1,\ldots,n+1 | \kappa_i^{Y_j} = \kappa_{n+1}^{Y_j}\}\right|}$;

  end

end

for $k = 1$ **to** c **do**

  $\overline{p(Y_k)} := \frac{1}{c} \sum_{j=1}^{c} p^{Y_j}(Y_k)$;

end

$k_{best} = \arg\max_{k=1,\ldots,c} \overline{p(Y_k)}$;

Output:

Prediction: $\hat{Y} = Y_{k_{best}}$;

The probability interval for $\hat{Y}$: $[\min_{k=1,\ldots,c} p^{Y_k}(\hat{Y}), \max_{k=1,\ldots,c} p^{Y_k}(\hat{Y})]$.

training set to train the underlying algorithm and the calibration set to calculate the set of probability distributions for each new instance. The main advantage of the IVP method is that the underlying algorithm is trained only once on the training set, and the probability distributions are calculated from the calibration set for every class of the test instance. There is no more the requirement to re-train the algorithm for every possible class of the test instance. The original taxonomy function K is transformed to another taxonomy

$K' : Z^{r+1} \times Z \to \mathbf{K}$ such that

$$\begin{gathered} K'_{r+1}((z_{q+1}, \ldots, z_{n+1}), z_i) = \\ K_q((z_1, \ldots, z_q), z_i), i = q+1, \ldots, n+1. \end{gathered} \tag{3.7}$$

In this definition we assume that the proper training set $\{z_1, \ldots, z_q\}$ is a fixed part of K' and therefore K' is a valid Venn taxonomy.

After assigning the category $\kappa_i^{Y_j} = K'((z_{q+1}, \ldots, z_n, (x_{n+1}, Y_j)), z_i)$ to each instance in the calibration set $i = q+1, \ldots, n+1$, the empirical probability of each classification Y_k in $\kappa_{n+1}^{Y_j}$ will be

$$p^{Y_j}(Y_k) = \frac{\left|\{i = q+1, \ldots, n+1 | \kappa_i^{Y_j} = \kappa_{n+1}^{Y_j} \ \& \ y_i = Y_k\}\right|}{\left|\{i = q+1, \ldots, n+1 | \kappa_i^{Y_j} = \kappa_{n+1}^{Y_j}\}\right|} \tag{3.8}$$

3.2.1.1 Online mode

In the online mode there is no fixed training set. On each step of the algorithm, a new instance is predicted and then it is added to the calibration set. Therefore, as the IVP makes predictions, new instances are considered for calibrating the probabilistic outputs. After a number of m predictions, we remove $m - s$ of the instances from the calibration set and we add them to the training set (where s is chosen such that both the training set and calibration set grow with equal rate on each update step). The algorithm is then re-trained on the training set and proceeds on predicting new instances. In contrast with the TVP, the IVP is re-trained only once every m steps, while the TVP is retrained on every step of the algorithm for every possible class of a new instance. In the online mode, we are able to test the probabilistic outputs of the algorithm

and examine whether the actual accuracy falls near the probability estimates. We must mention here, that the independence of error across updates of the training set is not entirely retained, an assumption that has to be made for any VP (which falls under the i.i.d. assumption). Validity is retained within in each update, since the training set is fixed, nonetheless the independence of error is violated on each update of the training set. One should be careful when deciding the size of m as to not affect the independence of error. Nonetheless, the independence of error may be affected, but the expected results would be well-calibrated in practice. This is demonstrated in section 3.3 and in Chapter 4, where we provide experimental results of the IVP.

3.2.1.2 Taxonomy

As explained in section 3.2, the validity of a TVP is guaranteed under the i.i.d. assumption, regardless of the taxonomy used. For example, a taxonomy that puts all instances in one large category would still give a valid predictor. Nevertheless, the performance of each VP is highly affected by the information provided from the categories defined in a taxonomy. A VP with a non efficient taxonomy would give very wide probabilistic bounds, whereas the better the taxonomy the narrower the probabilistic bounds will be.

In choosing the partitions that determine a taxonomy, we face a problem that is often called the problem of the reference class. We want the categories into which we divide the given instances to be large, in order to have a reasonable sample size for estimating the probabilities. In parallel, we want the categories to be small and homogeneous. In other words, we have two kinds of inefficiencies: too many categories in our taxonomy is a kind of overfitting, and

it is punished by a large diameter for the multiprobability prediction. Too few categories is a kind of underfitting, and it is punished by predictions that are not close enough to zero or one. When defining a taxonomy, our goal is to minimize the two inefficiencies in parallel, and find a taxonomy that gives large enough categories, but not too large.

In this work, our taxonomy is based on the classification output o_i of a conventional classification algorithm. Therefore, $\kappa_i^{Y_j} = f(x_i)$, where $f(x_i)$ is the classification output of the underlying algorithm after being trained on $z_1, \ldots, z_q$ where each $z_i = (x_i, y_i)$. This taxonomy will give c categories. The reasoning behind this definition of taxonomy matches our goal for finding an efficient taxonomy. If the classifier is fitted well on the training dataset then each category should contain sufficient information for the VP to perform well, while keeping the size of each category as small as possible, with respect to the number of classes in the dataset. The IVP algorithm is presented in Algorithm 5. In our implementation, we have used the SVM classifier with Sequential Minimal Optimisation (SMO) as our underlying algorithm [55]. The IVP was implemented in JAVA using the WEKA data mining software [56]. Our implementation is publicly available at [57].

3.2.1.3 Time efficiency

The nature of the TVP algorithm makes it inefficient in the case of large datasets. The algorithm has a training phase (learning phase) for every new instance and every possible class of the instance. This time inefficiency problem is removed from IVP algorithm, because of the use of the calibration set. The training phase of the algorithm needs to be performed only once, and then

Algorithm 5: Inductive Venn Predictor.

Input: proper training set $\{(x_1, y_1), \ldots, (x_q, y_q)\}$, calibration set $\{(x_{q+1}, y_{q+1}), \ldots, (x_n, y_n)\}$, new instance x_{n+1}, possible classes $\{Y_1, \ldots, Y_c\}$.

Train the multiclass underlying algorithm on the proper training set $\{(x_1, y_1), \ldots, (x_q, y_q)\}$;

for $j = 1$ **to** c **do**

  Assume classification Y_j for x_{n+1}.

  Supply the input patterns $x_{q+1}, \ldots, x_{n+1}$ to the trained underlying algorithm to obtain the outputs $o_{q+1}, \ldots, o_{n+1}$;

  for $i = q + 1$ **to** $n + 1$ **do**

    Assign κ_i to (x_i, y_i) according to the underlying algorithm classification output o_i;

  end

  for $k = 1$ **to** c **do**

    $p^{Y_j}(Y_k) = \frac{\left|\{i=1,\ldots,n+1 | \kappa_i^{Y_j} = \kappa_{n+1}^{Y_j} \ \&\ y_i \in Y_k\}\right|}{\left|\{i=1,\ldots,n+1 | \kappa_i^{Y_j} = \kappa_{n+1}^{Y_j}\}\right|}$;

  end

end

for $k = 1$ **to** c **do**

  $\overline{p(Y_k)} := \frac{1}{c} \sum_{j=1}^{c} p^{Y_j}(Y_k)$;

end

$k_{best} = \arg\max_{k=1,\ldots,c} \overline{p(Y_k)}$;

Output:

Prediction: $\hat{Y} = Y_{k_{best}}$;

The probability interval for $\hat{Y}$: $[\min_{k=1,\ldots,c} p^{Y_k}(\hat{Y}), \max_{k=1,\ldots,c} p^{Y_k}(\hat{Y})]$.

for every new instance the calibration set is being used to calculate the probabilistic outputs. This modification of the algorithm not only removes computationally expensive calculations, but also maintains the property that the probabilistic outputs will be well-calibrated, under the i.i.d. assumption. The time efficiency of the IVP becomes prominent when the underlying algorithm is expensive in terms of time efficiency. For example, if the underlying algorithm requires $O(n)$ time, the TVP method will reuse the underlying algorithm $n * c$ times, which makes the time requirement of the TVP to $O(n * n * c) = O(n^2)$ for small numbers of c. The IVP method, will only use the algorithm once,

thus leaving the time requirements to $O(n)$. As we will see in our experimental results in section 3.3, the difference between TVP and IVP has great impact in practice.

3.3 Experiments

We have conducted experiments with the proposed IVP algorithm in order to compare the results with its transductive counterpart, and with the three methods described in section 3.1.1. In the following subsections, we describe the datasets used, the online mode experiments, and the offline mode (10-fold cross validation) experiments.

3.3.1 Datasets

- Car Evaluation dataset

  The Car Evaluation dataset was derived from hierarchical decision model [58] and is available at [36]. The dataset contains 1728 instances with 6 features for each instance. There are 4 classes for this dataset which describe the car acceptability based on features that represent the price, technology, and comfort of a car.

- Red Wine quality dataset

  The Red Wine quality dataset contains 1599 instances of physiochemical features of red variants of the "Vinho Verde" wine [59]. Each instance has a quality score from 1 to 10. In this work, we have used the scores as 10 different classes from 1 to 10. This dataset is particularly difficult

and requires some pre-processing to remove redundant features, or even reduce the number of classes.

In our experiments, we have intentionally left the dataset to its original state in order to demonstrate the reliability of our probability estimates on difficult problems. The Red Wine quality dataset was used in the online experiments for its complexity. Nevertheless, it was not used in the offline experiments, since the large number of classes was prohibitive (in terms of time efficiency) for the evaluation of the TVP method.

- Spambase dataset

  The Spambase dataset which is available at [36], contains 4601 instances of email messages. There are 57 attributes which describe the content of each email. The emails can be classified into two classes: spam or non-spam.

- MiniBooNE dataset

  The MiniBooNE particle identification dataset (Booster Neutrino Experiment) [36, 60] contains 130065 instances of electron neutrinos and muon neutrinos. Each instance contains 50 real valued attributes which describe signal events. This dataset was used only with the IVP online method, in order to demonstrate its ability to handle large datasets.

3.3.2 Online experiments

In order to demonstrate the validity of the probabilistic outputs of our method, we conduct experiments in the on-line mode. Initially all instances are test instances and they are added to the training set after a prediction for each

one is made. The online experiments are carried out as follows: First, we compare the online results on the Car Evaluation dataset with the IVP and the three algorithms that were described in section 3.1.1, namely SVM with Logistic Regression (SVM-LR) which is Platt's method, SVM with Binning, and SVM with Isotonic Regression (SVM-IR). We also compare the IVP with its counterpart TVP method. Secondly, we compare the results of all the aforementioned methods on the Wine Quality dataset. Thirdly, we compare the IVP method with TVP on the Spambase dataset, and finally we conduct a large scale experiment on the MiniBooNE dataset to evaluate the scalability of the proposed IVP.

For the VPs, we graph the Cumulative Lower Accuracy Probability (CLAP), the Cumulative Upper Accuracy Probability (CUAP), and the Cumulative Accuracy (CA) curves:

$$CLAP(t) = \frac{1}{t}\sum_{i=1}^{t} U_i(Y_{k_{best}}), \tag{3.9}$$

$$CUAP(t) = \frac{1}{t}\sum_{i=1}^{t} L_i(Y_{k_{best}}), \tag{3.10}$$

$$CA(t) = \frac{1}{t}\sum_{i=1}^{t} Acc_i, \tag{3.11}$$

where t is the number of test instances that have been added to the training set, and $Acc_i = 1$ when the prediction for instance x_i is correct and 0 otherwise. We also plot the Cumulative Mean Accuracy Probability (CMAP) curve, which is the mean of the CLAP and CUAP curves. Since VPs provide well calibrated probabilistic outputs, it is expected that the CA curve will fall within

or near the CLAP and CUAP curves. For classical probabilistic predictors (the SVM-LR algorithm) we plot the CA and the Cumulative Accuracy Probability (CAP) curves. The CAP curve is similarly calculated as the CA curve:

$$CAP(t) = \frac{1}{t}\sum_{i=1}^{t}\max_{j=1}^{c} f(x_{ij}), \tag{3.12}$$

where $f(x_{ij})$ is the probability estimate given for a prediction. The CA curve should fall near the CAP curve, if the algorithm provides well-calibrated probabilities.

In Figures 3.1 and 3.2, we show the online results of SVM with Logistic Regression (SVM-LR), SVM with Binning, SVM with Isotonic Regression (IR), and SVM-IVP on the Car Evaluation dataset. The underlying SVM algorithm that we have used works with the RBF kernel. We experiment with each algorithm two times, one with a RBF parameter set to an optimal value, and another with a RBF parameter set to the optimal value divided by 10 (we do this in order to test the difference in the results when the predictors do not perform so well). The optimal value for each experiment was chosen based on offline tests (10-fold cross validation) that have been conducted with a standard SVM predictor. The standard SVM predictor was tested with the RBF parameter ranges of [0.1, 1] with steps of 0.1, and [1, 5] with steps of 1. The number of bins for the SVM Binning method was set to $b = 10$. In our experiments with the IVP we have set $q = \lceil 0.7(n-1) \rceil$. The RBF spread parameter chosen for this dataset was 0.2. In the figures, we expect the curves in each plot to be relatively near, if the probabilities produced by the corresponding methods are well calibrated. As it is shown, this is true only for the IVP in

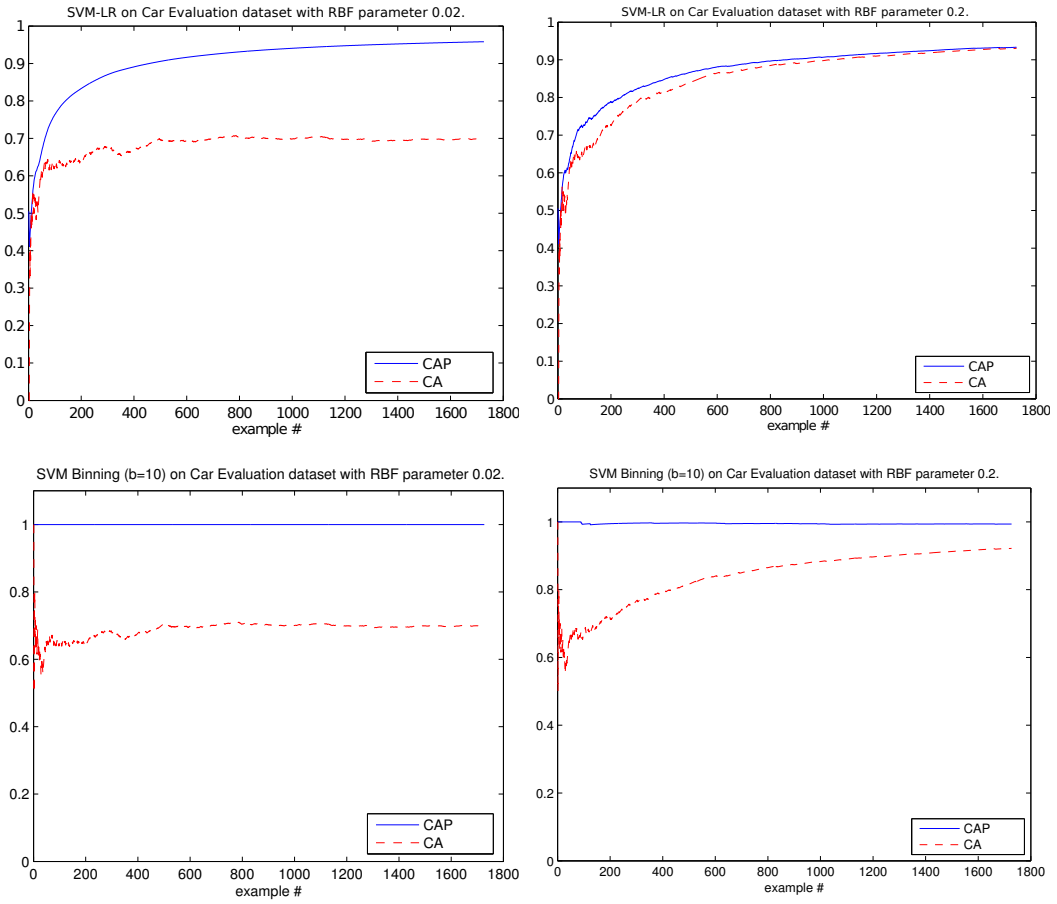


FIGURE 3.1: Online experiments of SVM-LR and SVM Binning on the Car evaluation dataset. RBF parameter is 0.02 on the left column and 0.2 on the right column.

both experiments and for SVM-LR method only with the optimal RBF parameter. When the RBF parameter is 0.2 the accuracy is around 90% for all methods, which is the expected accuracy on this dataset. In contrast, when we set the RBF parameter to 0.02 the accuracy is reduced to around 70% (which is near the percentage of the first class), while the probabilistic outputs are near 100% for all methods except the IVP. As shown in Figure 3.2, the IVP probabilistic outputs are automatically lowered to around 68%, which is near

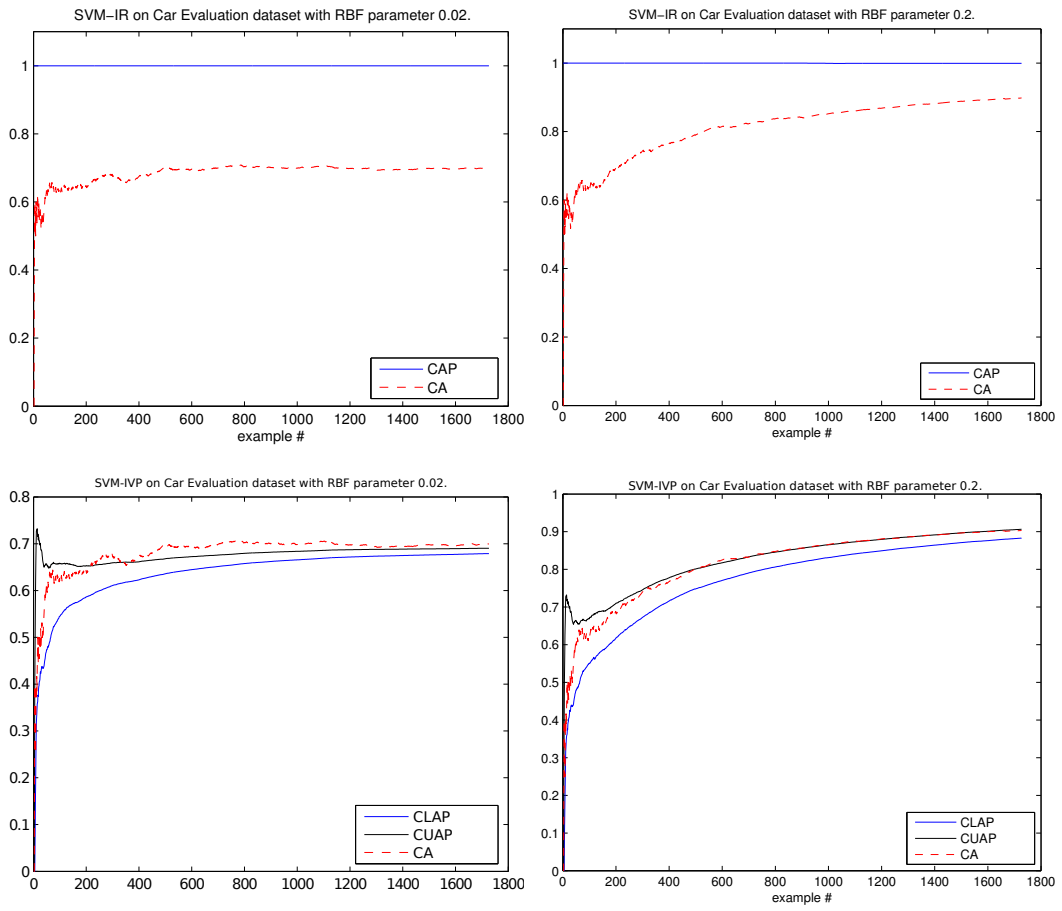


FIGURE 3.2: Online experiments of SVM with Isotonic Regression (SVM-IR) and SVM-IVP on the Car evaluation dataset. RBF parameter is 0.02 on the left column and 0.2 on the right column.

the actual accuracy. This indicates that the Logistic Regression, Binning, and Isotonic Regression methods cannot always guarantee that their probabilistic outputs will be well calibrated, while the IVP method can always guarantee this property under the i.i.d. assumption.

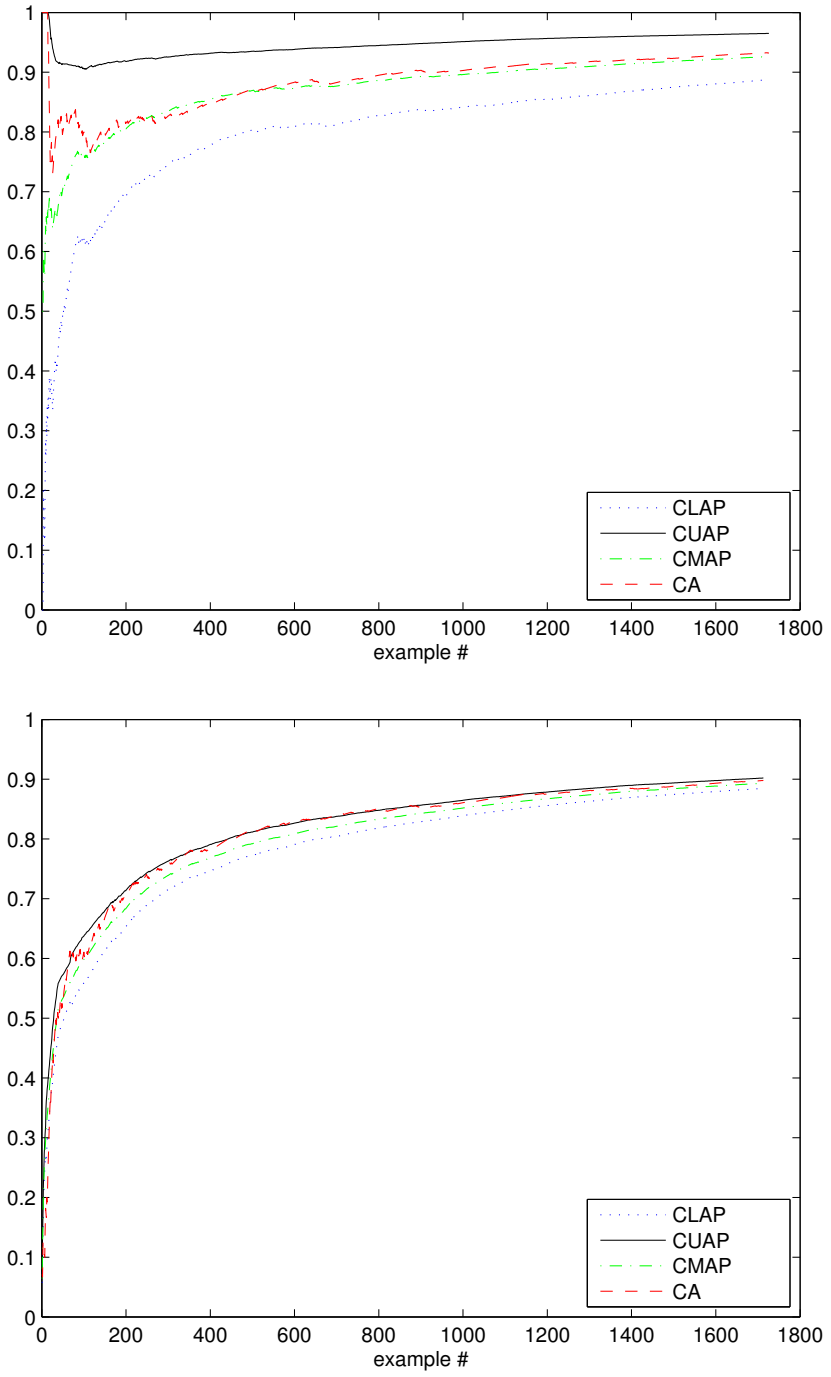


FIGURE 3.3: Online experiments with TVP (top), and IVP (bottom) on the Car evaluation dataset.

Algorithm	**Accuracy**	**Duration (hh:mm:ss)**	**Probabilities**
SVM-LR	90.63%	00:24:03	93.21%
TVP	93.11%	00:52:38	88.61% – 96.39%
IVP	89.70%	00:03:37	88.36% – 90.10%

TABLE 3.1: Comparison of online results on the Car evaluation dataset.

In Figure 3.3, we conduct more experiments and compare the probabilistic outputs of the Transductive VP and the proposed IVP method on the Car Evaluation dataset. Both VPs provide well-calibrated probability bounds, while the IVP method gives even more narrow results. In Table 3.1, we show the probabilistic outputs and timing results that were recorded at the end of the online experiments. We include the results of SVM-LR as the baseline algorithm. For the SVM-LR algorithm there is about 3% difference for the estimated probability and accuracy, while the TVP provides well calibrated results with an interval of about 8%. The IVP provides well calibrated results with a much better interval of about 2–3%. The accuracy of the IVP remains at the same level as with the SVM-LR method, although the TVP performs better in terms of accuracy with 93.11%. The IVP accuracy is expected to be lower than the TVP accuracy, since there is a number of instances removed from the training set to be used as the calibration set. The great advantage of the IVP method is the time efficiency, which is compared with the rest of the methods. As we can see in Table 3.1, the SVM-LR algorithm required 24 minutes and the TVP 52 minutes to finish the experiment, while the IVP required only 3 minutes to finish. The IVP method is much faster since the training of the underlying algorithm is required once every $m = 20$ steps in this experiment. TVP uses the training set on each update for retraining, while the IVP uses the calibration set without retraining. Moreover, the training set of the IVP method is

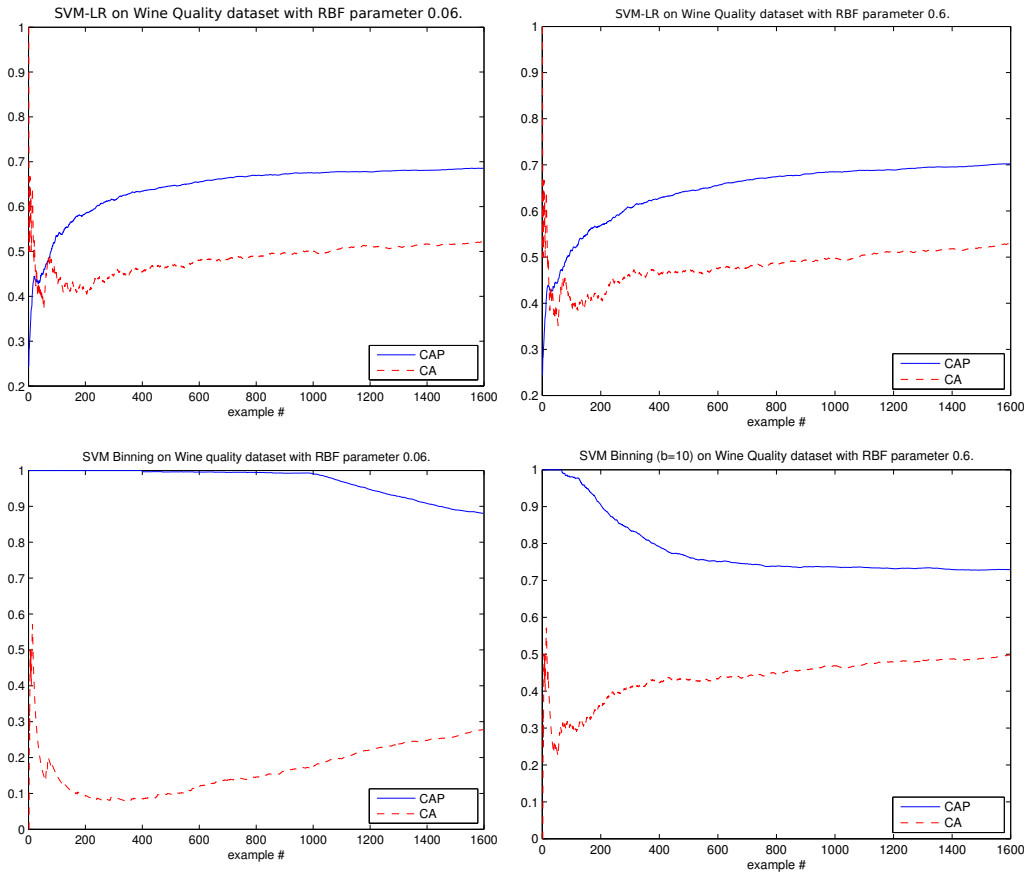


FIGURE 3.4: Online experiments of SVM-LR and SVM Binning on the Wine quality dataset. RBF parameter is 0.06 on the left column and 0.6 on the right column.

slightly smaller, since there is a percentage used for the calibration set.

Figures 3.4 and 3.5 show the online results of the four algorithms on the Wine Quality dataset. The optimal RBF parameter was set to 0.6. The lower and upper probability interval is very tight in the case of the IVP, while the TVP provides very wide probability bounds, as it is shown in Figure 3.6. The SVM-LR, SVM Binning, and SVM-IR methods provide misleading probability estimates, since there is a discrepancy of at least 10% between the average

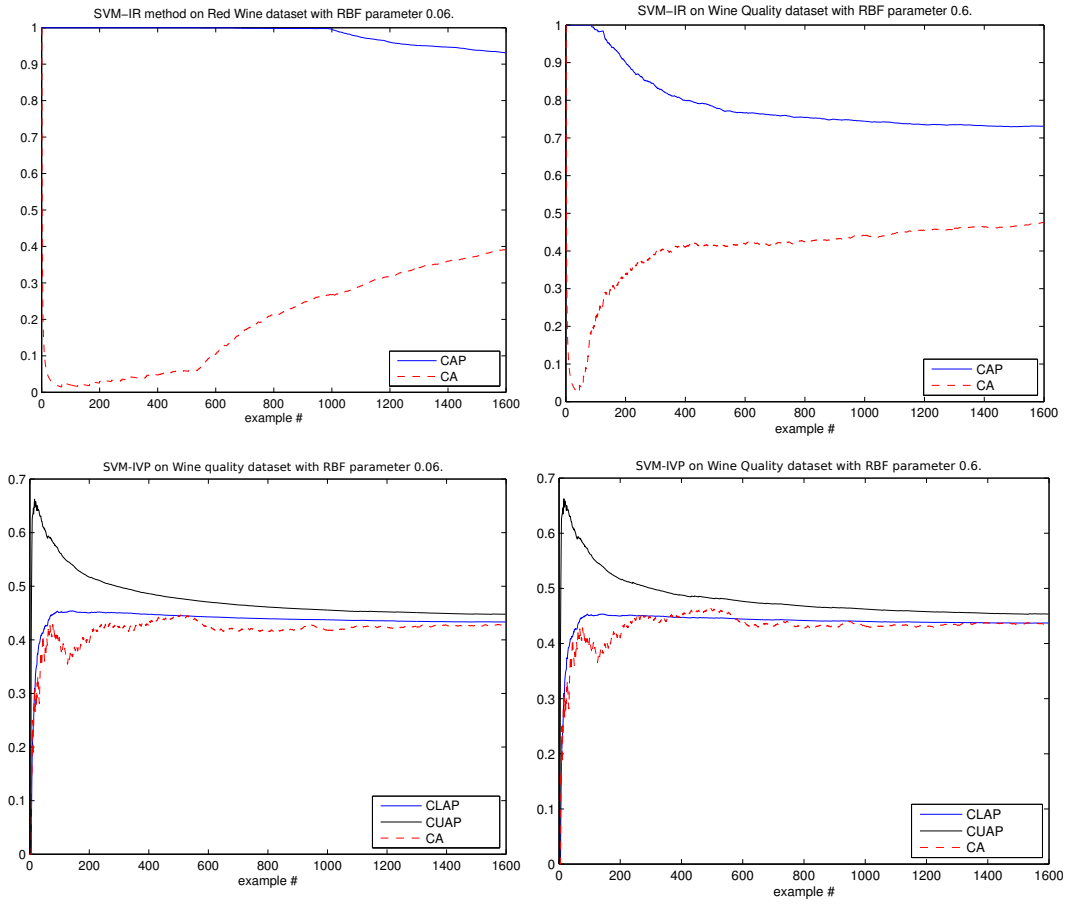


FIGURE 3.5: Online experiments of SVM with Isotonic Regression (SVM-IR) and SVM-IVP on the Wine quality dataset. RBF parameter is 0.06 on the left column and 0.6 on the right column.

probability and average accuracy. This is not the case for TVP, since the probabilistic outputs are valid in the sense that they do not give any misleading information. In Figure 3.6, we compare the results of the TVP with those of the IVP method. It is surprising how the IVP provides such tight probabilistic outputs, even when the TVP does not perform so well. A possible explanation of this result is that the IVP method calculates the probabilities using only the calibration set and the underlying algorithm is trained only once every m steps.

Algorithm	**Accuracy**	**Duration (hh:mm:ss)**	**Probabilities**
SVM-LR	51.84%	01:26:42	62.99%
TVP	47.78%	10:04:10	19.34% – 88.85%
IVP	48.59%	00:56:26	48.81% – 50.11%

TABLE 3.2: Comparison of online results on the Wine Quality dataset.

Algorithm	**Accuracy**	**Duration (hh:mm:ss)**	**Probabilities**
SVM-LR	87.97%	04:22:43	89.99%
TVP	81.54%	07:53:20	80.00% – 81.44%
IVP	86.50%	00:26:57	86.75% – 87.22%

TABLE 3.3: Comparison of online results on the Spambase dataset.

The only thing that changes during each test is the assumed label of the test instance. The change of the assumed label does not affect the outputs of the algorithm on the training set, and the instances in the calibration set remain in the same category. Therefore, we should not expect a lot of difference in the probabilities calculated. In contrast, the TVP method re-trains the training set for every assumed label of the test instance, and the categorization for calculating the probabilities might change drastically. In Table 3.2 we show the end results and durations of the SVM-LR, SVM-TVP, and SVM-IVP algorithms. Again, the IVP is faster compared with SVM-LR and TVP.

We conduct further experiments using the Spambase dataset for comparing the TVP and IVP methods. The online results are shown in Figure 3.7 and Table 3.3. The IVP algorithm provides better accuracy than its transductive counterpart. Since the dataset is larger here, we notice that when using the calibration set for estimating probabilities, we get more accurate results. The number of steps before a training update for the IVP on the Spambase dataset

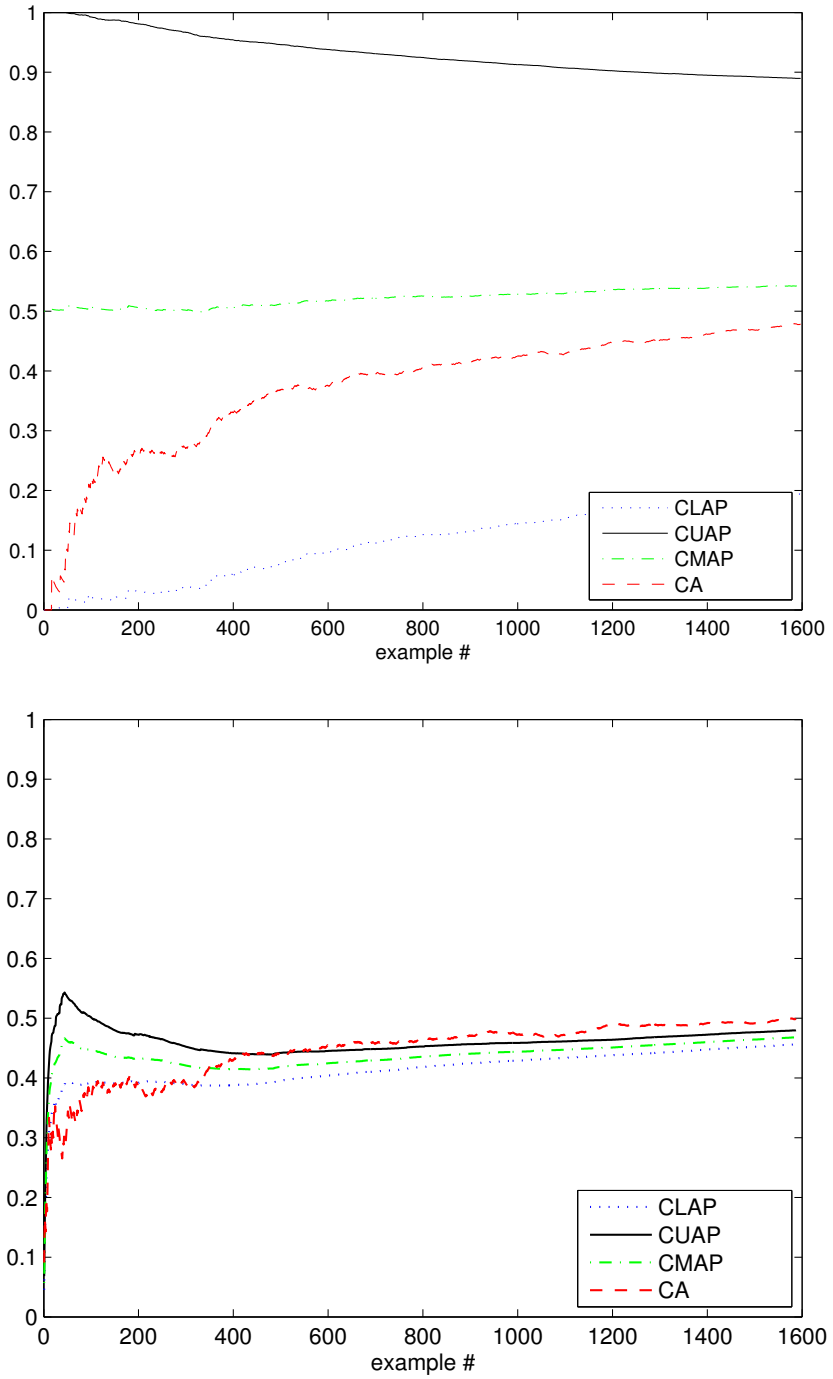


FIGURE 3.6: Online experiments with TVP (top), and IVP (bottom) on the Wine Quality dataset.

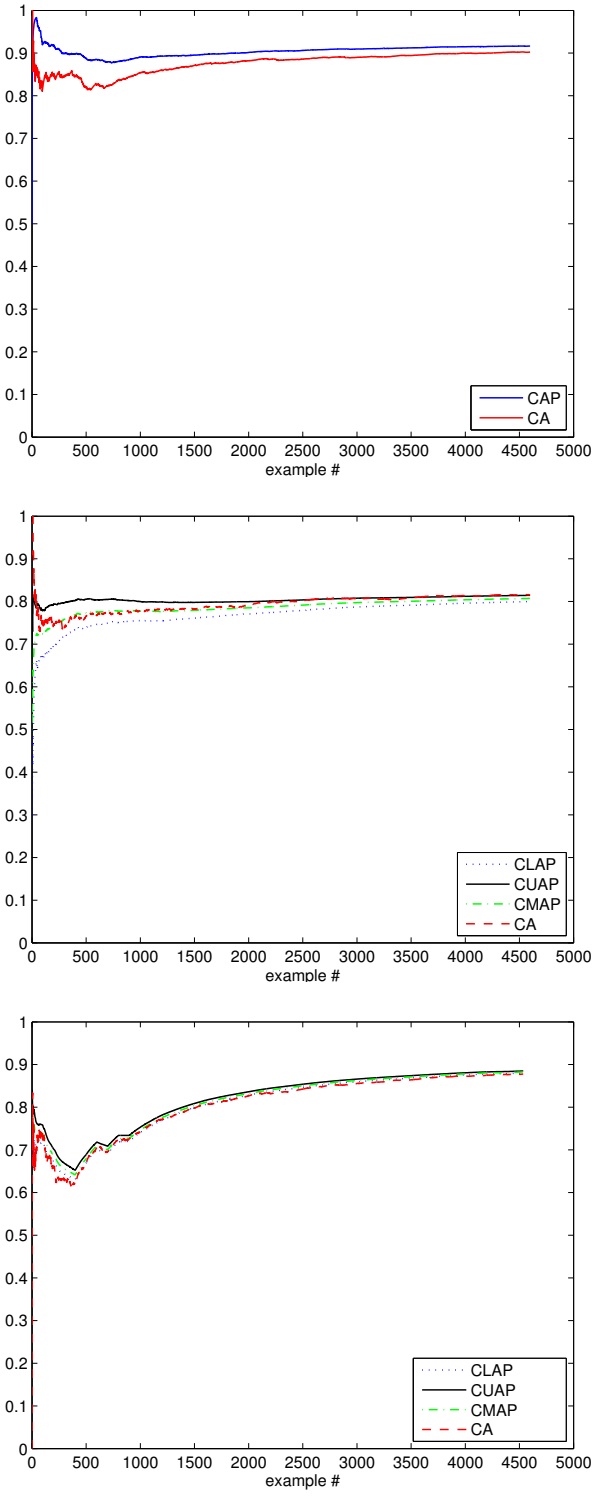


FIGURE 3.7: Online experiments with SVM-LR (1st), TVP (2nd), and IVP (3rd) on the Spambase dataset.

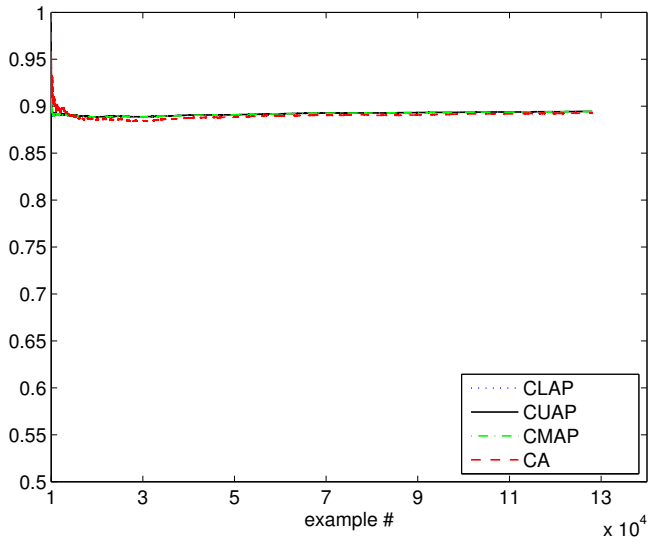


FIGURE 3.8: Online experiment with IVP on the MiniBooNE dataset.

was chosen to $m = 100$ because of the large size of the dataset. The IVP method outperforms the TVP and SVM-LR methods in terms of time efficiency. The total duration of the IVP experiment required 56 minutes, while TVP required 10 hours, and SVM-LR required 1 hour and 26 minutes.

We have additionally performed an experiment on the IVP method on a larger scale problem. We have used the MiniBooNE particle identification dataset, which contains 130065 particle instances. The dataset was not possible to be tested with the TVP method due to the time inefficiency problem of the method. Since in this experiment we do not compare the IVP with other methods, we have used the C4.5 decision tree classifier [61] as the underlying algorithm of the IVP, which runs faster. As it is shown in Figure 3.8, the IVP method has provided well-calibrated and accurate results. The number of steps before each training update was chosen to $m = 10000$, which allowed us to overcome the time inefficiency problem.

3.3.3 Offline experiments

We have performed 10-fold cross validation experiments with the Car evaluation and Spambase datasets in order to evaluate the results of the IVP method and compare it with the TVP. Our intention here is to compare the probabilistic intervals that the two algorithms give, based on the training set size. Therefore, the SVM-LR, SVM-IR, and SVM-Binning algorithms have not been used here. Moreover, we have not used the Wine Quality dataset in the offline experiments, due to the large number of classes of the dataset, which made the TVP method prohibitively slow. The Spambase dataset has only two classes, which allowed us to evaluate the TVP and IVP methods together in the offline mode.

We compare TVP and IVP using different sizes of the datasets in order to evaluate which method performs better on various data sizes. Since IVP uses a percentage from the training set as the calibration set, we expect the IVP to give lower accuracy when the dataset is small, and as the data size increases, we expect the accuracy of the IVP to match the accuracy of the TVP. For the IVP method, we have used 30% of the training set as calibration set.

In Table 3.4 and Figure 3.9 (top), we show the results of the IVP on the Car Evaluation dataset. In each row of the table we show the results with a different number of instances in the training set. The results in each row are the averages of ten 10-fold cross validation runs. We have started the experiments with only 100 instances and increased the number of instances in each experiment by 100. As expected, the IVP always provides well calibrated probabilistic outputs, regardless of the number of instances in the training set.

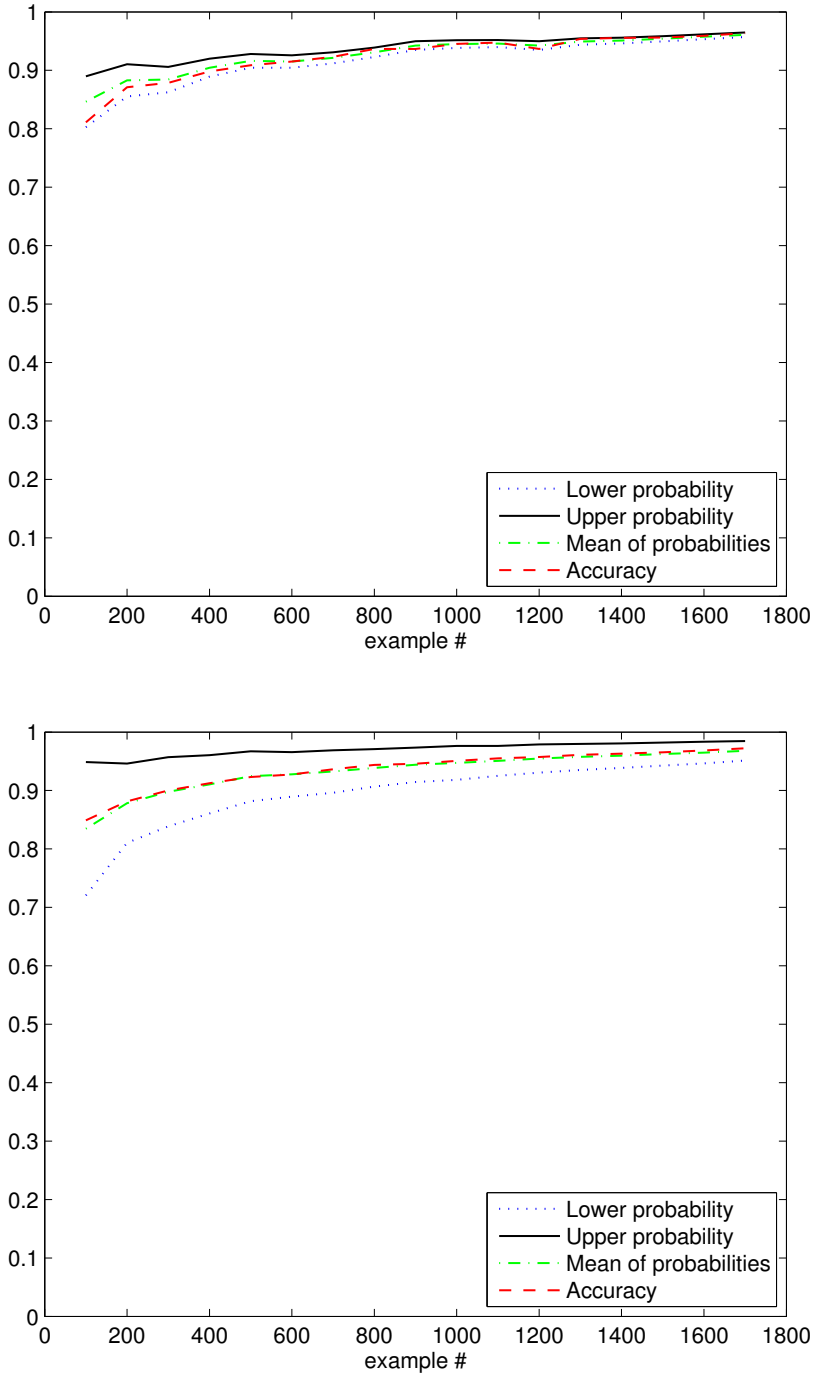


FIGURE 3.9: IVP (top) and TVP (bottom) 10-fold cross validation results on the Car evaluation dataset.

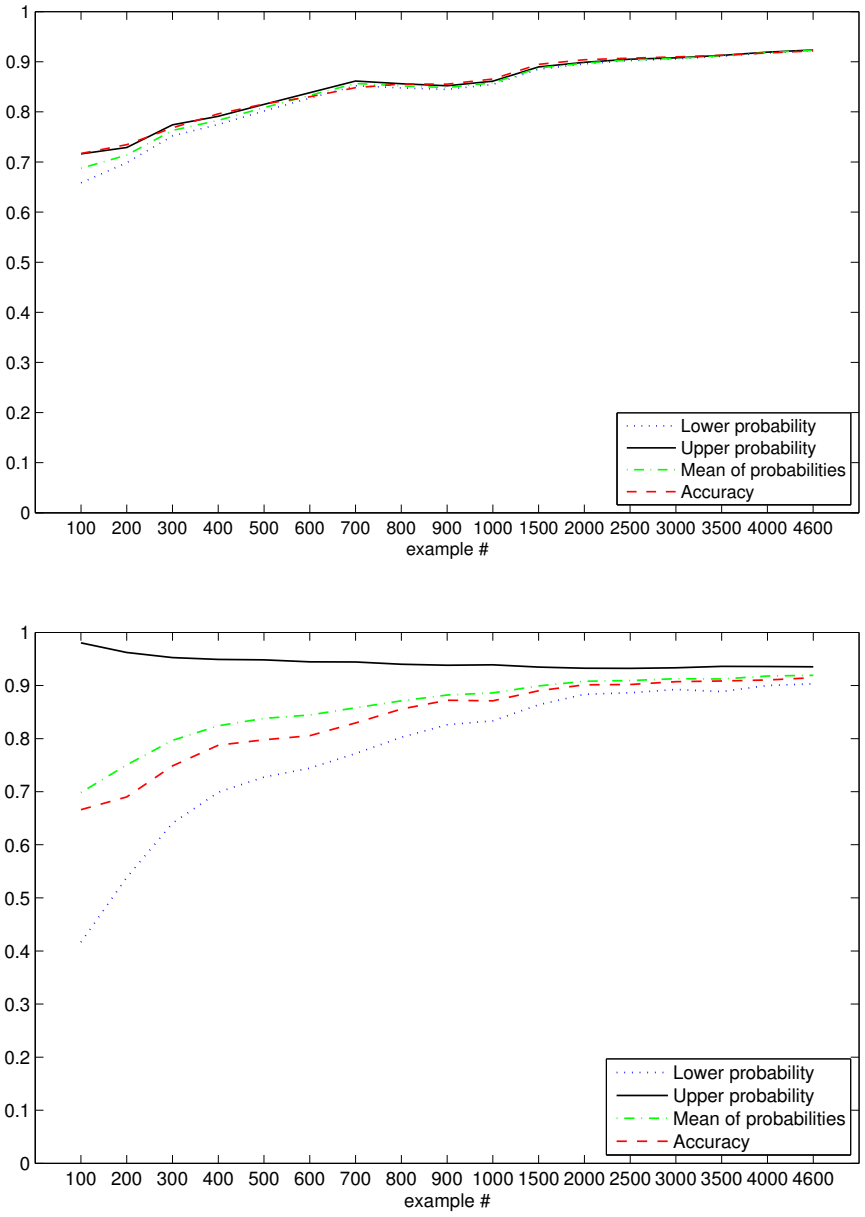


FIGURE 3.10: IVP (top) and TVP (bottom) 10-fold cross validation results on the Spambase dataset.

# of instances	Accuracy	Lower Prob.	Upper Prob.	BS
100	81.10%	80.28%	88.98%	0.2805
200	87.10%	85.52%	91.04%	0.2055
300	87.87%	86.26%	90.58%	0.1964
400	89.80%	88.91%	91.99%	0.1595
500	90.88%	90.43%	92.80%	0.1524
600	91.52%	90.43%	92.57%	0.142
700	92.30%	91.20%	93.09%	0.1361
800	93.65%	92.27%	93.90%	0.1145
900	93.66%	93.50%	94.98%	0.1136
1000	94.52%	93.84%	95.15%	0.1002
1100	94.76%	93.97%	95.18%	0.0962
1200	93.66%	93.50%	94.98%	0.1136
1300	95.38%	94.39%	95.46%	0.0867
1400	95.55%	94.61%	95.59%	0.0824
1500	95.61%	94.95%	95.86%	0.0825
1600	95.91%	95.33%	96.17%	0.0775
1728	96.37%	95.69%	96.48%	0.0699

TABLE 3.4: IVP 10-fold cross validation results on the Car evaluation dataset.

Nevertheless, as the training set grows, the accuracy increases and the interval of the probabilities becomes smaller. We can also see clearly that the accuracy always falls within the probability estimates. We have calculated the Brier Score (BS) in each experiment, which indicates the quality of the probability estimates. The BS is calculated as

$$BS = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{c}(f(x_{ij}) - o_{ij})^2, \tag{3.13}$$

where $f(x_{ij})$ is the mean of the probabilities obtained for class j. The value o_{ij} is set to 1 if instance x_i belongs to class j, and 0 otherwise. The constant c is the number of classes and N is the number of instances. As shown in the results,

# of instances	Accuracy	Lower Prob.	Upper Prob.	BS
100	84.90%	72.05%	94.88%	0.2243
200	88.15%	81.04%	94.62%	0.1743
300	90.00%	83.87%	95.70%	0.1500
400	91.25%	86.07%	96.04%	0.1327
500	92.30%	88.18%	96.71%	0.1139
600	92.73%	88.95%	96.57%	0.1103
700	93.62%	89.61%	96.89%	0.0989
800	94.37%	90.67%	97.08%	0.0887
900	94.57%	91.46%	97.35%	0.0846
1000	95.06%	91.81%	97.62%	0.0759
1100	95.50%	92.52%	97.64%	0.0724
1200	95.74%	93.06%	97.89%	0.0671
1300	96.10%	93.53%	97.98%	0.0629
1400	96.32%	93.85%	98.07%	0.0596
1500	96.53%	94.27%	98.21%	0.0571
1600	96.86%	94.64%	98.35%	0.0526
1728	97.23%	95.12%	98.45%	0.0468

TABLE 3.5: TVP 10-fold cross validation results on the Car evaluation dataset.

the BS decreases as the training set grows. A smaller BS indicates better quality of the probability estimates. In Table 3.5 and Figure 3.9 (bottom), we show the results of the TVP method on the same dataset. Comparing the results of the TVP with the IVP method, we can see that the TVP method provides slightly higher accuracy whether the data size is small or large (about 1% difference) and slightly better BS. Nonetheless, although the TVP provides well-calibrated probabilities, it gives intervals that are much wider than those of the IVP method, especially when the training set is small. On the Car Evaluation dataset with 100 instances, the IVP probability interval has 8.7% width, while the TVP interval has 22.83% width. On the same dataset with all instances (1728), the IVP probability interval has 0.79% width and the TVP

# of instances	Accuracy	Lower Prob.	Upper Prob.	BS
100	71.70%	65.86%	71.62%	0.3867
200	73.45%	69.84%	72.89%	0.3667
300	76.80%	75.20%	77.42%	0.3249
400	79.60%	77.48%	79.10%	0.3018
500	81.58%	80.18%	81.49%	0.2822
600	83.02%	82.75%	83.84%	0.2554
700	84.83%	85.20%	86.14%	0.2327
800	85.57%	84.78%	85.60%	0.2310
900	85.52%	84.48%	85.21%	0.2315
1000	86.55%	85.46%	86.12%	0.2162
1500	89.47%	88.52%	88.96%	0.1864
2000	90.41%	89.56%	89.89%	0.1728
2500	90.72%	90.25%	90.52%	0.1675
3000	90.97%	90.56%	90.78%	0.1631
3500	91.29%	91.08%	91.27%	0.1579
4000	91.77%	91.75%	91.91%	0.1511
4601	92.16%	92.22%	92.36%	0.1455

TABLE 3.6: IVP 10-fold cross validation results on the Spambase dataset.

interval has 3.33% width. In Table 3.6 and Figure 3.10 (top) we show the results of the IVP method on the Spambase dataset. Again, the probability estimates interval of the IVP method is very tight regardless of the size of the training set, while the probability estimates of the TVP method, shown in Table 3.7 and Figure 3.10 (bottom), are generally wider. On the Spambase dataset with 100 instances, the IVP diameter is 5.76% and the TVP diameter is 56.39%. On the same dataset with all instances (4601), the IVP diameter is 0.14% and the TVP diameter is 3.21%. From these results, we see how the IVP outperforms the TVP method in terms of smaller probability intervals. The accuracy of the IVP matches the accuracy of the TVP method (in fact, in some cases the IVP provides slightly higher accuracy). Therefore, accuracy is

# of instances	Accuracy	Lower Prob.	Upper Prob.	BS
100	66.60%	41.64%	98.03%	0.3820
200	69.00%	53.85%	96.22%	0.3389
300	74.87%	64.06%	95.26%	0.2888
400	78.75%	69.90%	94.92%	0.2525
500	79.78%	72.76%	94.83%	0.2392
600	80.55%	74.44%	94.45%	0.2353
700	82.96%	77.19%	94.43%	0.2087
800	85.58%	80.25%	94.00%	0.1885
900	87.23%	82.60%	93.81%	0.1792
1000	87.12%	83.34%	93.91%	0.1819
1500	89.00%	86.36%	93.48%	0.1644
2000	90.15%	88.35%	93.25%	0.1565
2500	90.19%	88.64%	93.24%	0.1540
3000	90.73%	89.24%	93.34%	0.1487
3500	90.86%	88.86%	93.61%	0.1444
4000	91.05%	90.01%	93.58%	0.1411
4601	91.46%	90.31%	93.52%	0.1395

TABLE 3.7: TVP 10-fold cross validation results on the Spambase dataset.

retained, while more effective probabilistic outputs are provided by the IVP.

In Tables 3.8 and 3.9, we present the final results of the 10-fold cross validation experiments of the TVP, and IVP algorithms on the Car evaluation and Spambase datasets. Here, we also compare the time duration of each experiment. We can see that the accuracy of the two methods on both datasets is around the same level, while the probabilistic outputs of the IVP method are narrower and well-calibrated. The time efficiency of IVP over TVP is once again demonstrated in the results.

Algorithm	**Accuracy**	**Duration (hh:mm:ss)**	**Probabilities**
TVP	93.11%	02:17:47	88.61% – 96.39%
IVP	89.70%	00:05:49	88.36% – 90.10%

TABLE 3.8: Comparison of offline results on the Car evaluation dataset.

Algorithm	**Accuracy**	**Duration (hh:mm:ss)**	**Probabilities**
TVP	91.46%	72h+	90.31% – 93.52%
IVP	92.16%	04:50:22	92.22% – 92.36%

TABLE 3.9: Comparison of offline results on the Spambase dataset.

3.4 Summary

In this chapter, we gave a detailed description of the Transductive Venn Prediction framework. We have developed the Inductive version of the Venn Prediction framework which can provide well calibrated probabilistic outputs based on the only assumption that the data used are identically and independently distributed. We have performed extensive experiments with an SVM Inductive Venn Predictor (IVP) on four datasets and we have compared its probabilistic outputs and computational efficiency to those of the SVM with Logistic Regression (SVM-LR), SVM with Binning, SVM with Isotonic Regression (SVM-IR), and SVM Transductive Venn Predictor (TVP) on three of the four datasets. The fourth larger dataset was used to demonstrate the scalability of the IVP. In the comparison, it is shown that our IVP outperforms SVM-LR, SVM Binning, and SVM-IR in terms of reliability. We have additionally compared the results of IVP with TVP, and we have demonstrated that the proposed IVP outperforms the corresponding TVP method in terms of time efficiency.

Chapter 4

Applications

In this chapter, we apply the Conformal and Venn Prediction frameworks on real-life applications and we examine the practical aspects of the confidence and probabilistic outputs of our algorithms. In particular, we examine three applications:

1. Assessment of the risk of stroke, based on ultrasound images of atherosclerotic carotid plaques.
2. Osteoporosis risk assessment, based on known factors.
3. Diagnosis of childhood abdominal pain.

4.1 Assessment of the risk of stroke

Visual classification of high-resolution ultrasound has made the non-invasive visualisation of the carotid bifurcation possible, and has thus been used in the

study of arterial wall changes. Clinical applications of carotid bifurcation ultrasound include: i) identification and grading of stenosis of extracranial carotid artery disease often responsible for ischaemic strokes, Transient Ischaemic Attacks (TIAs) or Amaurosis Fugax (AF); ii) follow-up after carotid endarterectomy; iii) evaluation of pulsatile neck mass; iv) investigation of asymptomatic neck bruits where severe internal carotid artery stenosis is used as a predictive factor for future stroke; v) cardiovascular risk assessment where the presence of carotid bifurcation of atherosclerotic plaques is associated with increased cardiovascular mortality. During the last 20 years, the introduction of computer aided methods and image standardisation has improved the objective assessment of carotid plaque echogenicity and heterogeneity[62], and has largely replaced subjective assessment that had been criticized for its poor reproducibility [63].

Until now several studies presenting classification models for carotid ultrasound images have been presented, see for example [2, 64–66], but none of these methods provide any valid confidence measures on this problem. In order to address this, we propose the use of Conformal Prediction to assess the risk of stroke based on morphological ultrasound images.

For classification, our system is based on a set of morphological features and a set of classical image texture features, extracted from 274 ultrasound images of carotid plaques. Images used are base-line images, which means that they were collected before any event happened. From these images, 137 were classified as asymptomatic, while 137 are symptomatic (an event of Stroke, TIA or AF happened at some stage during a monitoring period of 8 years). We apply the Conformal Prediction framework on both categories of features, using four

different classifiers: Artificial Neural Network (ANN); Support Vector Machine (SVM); Naive Bayes Classifier (NBC); and k-Nearest Neighbours (k-NN). We compare the results and show the reliability and practicality of the confidence measures obtained for the classification of atherosclerotic carotid plaques.

4.1.1 Atherosclerotic Carotid Plaque Data

A total of 274 carotid plaque ultrasound images associated with retinal or hemispheric symptoms (33 stroke, 60 TIA, and 44 AF) were used in this work. Patients with cardioembolic symptoms or distant symptoms (more than 6 months) were excluded from the study. Asymptomatic plaques were truly asymptomatic if they had never been associated with symptoms in the past, or symptomatic if they had been associated with retinal or hemispheric symptoms (Stroke, TIA or AF). The ultrasound images were collected in the Irvine Laboratory for Cardiovascular Investigation and Research, Saint Mary's Hospital, UK, using an Advanced Technology Laboratories (ATL model HDI 3000 - Seattle, USA) duplex scanner with a linear broadband width 4-7 MHz (multi-frequency) transducer, at a resolution of 20 pixels/mm. The gray scale images (gray levels 0-255) were normalized manually by adjusting the image linearly. The plaque identification and segmentation tasks are quite difficult and were carried out manually by a physician or vascular ultrasonographer who are experienced in scanning, both actions are described in [67].

4.1.1.1 Texture Features

Texture features, shape parameters, and morphological features were extracted from the manually segmented ultrasound plaque images. Seven different texture features sets were extracted from the plaque segments using the algorithms described in [2]. The algorithms used in these studies namely are 1) Statistical Features (SF), 2) Spatial Gray Level Dependence Matrices (SGLDM), 3) Gray Level Difference Statistics (GLDS), 4) Neighbourhood Gray Tone Difference Matrix (NGTDM), 5) Statistical Feature Matrix (SFM) method, 6) Laws Texture Energy Measures (TEM), 7) Fractal Dimension Texture Analysis (FDTA), 8) Fourier Power Spectrum (FPS) features, and 9) Run Length Statistics (RUNL).

4.1.1.2 Morphological features

Morphological features are motivated from the need to study the basic structure of the plaque. We have used two morphological analysis methods in order to quantify morphological features of the plaques. The first one was based on a multilevel approach where the image intensity was thresholded at three different levels, while the second one was based on gray scale morphological analysis.

Morphological features of plaques are strongly associated with events. For example black (echolucent) plaques with white big blobs are considered to be very dangerous. From a structural perspective, morphological methods allow us to provide size distributions for different components of the plaque. A detailed analysis of morphological features extracted from the plaques can be

found in [1]. In this work, we have used the group of L-images as described in [1]. This group gave the best accuracy results.

4.1.2 Experiments and Results

We have experimented on both the morphological data and the texture data described, and we have compared the results of the classical algorithms used in this study with the corresponding CPs. Before conducting our experiments we have applied Principal Component Analysis (PCA) on the datasets and selected the features which accounted for 98% of each dataset's variance. For evaluating our algorithms, we have applied the Leave-One-Out Cross Validation (LOOCV) technique. Both of these choices were made in order to be able to have similar results with [1] which have conducted research on the morphological data. In LOOCV, a test instance is left out from the training set and after training, a prediction is made for the left-out instance. This experiment is repeated for every instance in the dataset, and the predictions are then evaluated with the true labels of the instances. The ANN-CP was structured with one hidden layer, and all units had sigmoid activation functions. We have used 30% learning rate and a momentum rate of 20%. In each experiment, the ANNs were trained for 500 epochs with 10% validation set, which was used to stop training when the performance on the validation set was deteriorating. For the SVM-CP, we have used a Radial Basis Function (RBF) kernel mapping, and complexity $c = 1$. The complexity parameter c of SVMs allows us to control the trade-off between errors on the training data and the complexity or capacity of the model. When c is small, more errors are allowed. The aforementioned parameters were chosen similarly with the

work that was done in [1], in order to be able to perform a comparison. In Table 4.1, we show the results of the four CPs described in section 2.2.1 on the morphological data. We have conducted experiments with different parameter values of the underlying algorithm of each CP. The parameter for the ANN-CP is the number of neurons of the hidden layer, for the SVM-CP the spread of the RBF kernel, and for the k-NN-CP we set the number of nearest neighbours considered. We should note here that the NBC-CP (also included in Table 4.1) has no parameters. We also report the certainty and error rates of each CP for the confidence levels 95%, 85%, and 75%. The certainty rates correspond to the prediction regions that contained only a single label, and the error rates correspond to the prediction regions that did not contain the true label. The certainty rates show the efficiency of each CP. High rates of certainty show a better quality in our confidence measures. We highlight the results which give the best average percentage of accuracy and certainty rates.

In Table 4.2, we compare the accuracy results of the four CPs with the results of the corresponding classical algorithms. We have selected for each algorithm the parameters which are highlighted in Table 4.1. We have also calculated the True Positive Rates (TPR), and True Negative Rates (TNR). A TN in our case is a plaque which has been correctly classified as asymptomatic, and a TP a plaque which is correctly classified as symptomatic.

In Table 4.3, we give the results of the CPs on the texture data. The structure of the results is identical to that of Table 4.1. In Table 4.4, we compare the accuracy, TPR and TNR of the classical algorithms with the corresponding CPs.

		Certainty			Error		
CP	**Acc.**	**95%**	**85%**	**75%**	**95%**	**85%**	**75%**
ANN neurons							
0	73.36%	30.29%	67.15%	90.15%	4.74%	13.50%	22.26%
1	**72.26%**	**35.04%**	**68.61%**	**89.42%**	**4.74%**	**14.23%**	**23.72%**
2	70.80%	32.48%	64.96%	88.32%	4.74%	14.23%	22.26%
3	71.90%	31.02%	68.98%	90.15%	4.74%	14.60%	23.36%
4	71.90%	32.85%	66.42%	88.69%	4.74%	13.87%	22.63%
5	71.53%	33.21%	66.42%	89.05%	5.11%	14.23%	22.63%
6	71.53%	34.31%	65.69%	88.32%	5.11%	13.50%	22.26%
7	71.90%	33.21%	65.69%	88.69%	4.74%	13.87%	22.63%
8	70.80%	31.75%	63.87%	87.96%	4.74%	14.60%	22.26%
9	70.80%	32.48%	64.96%	88.32%	4.74%	13.87%	22.63%
10	70.80%	33.94%	64.60%	88.69%	5.47%	13.50%	22.26%
SVM spread							
0.10	**73.72%**	**19.34%**	**54.74%**	**85.40%**	**4.74%**	**14.96%**	**24.82%**
0.11	73.72%	17.52%	54.38%	85.04%	4.74%	14.96%	24.82%
0.12	72.99%	18.61%	54.01%	85.40%	4.74%	14.96%	24.82%
0.13	72.99%	19.34%	54.38%	85.04%	4.74%	14.96%	24.82%
0.14	72.26%	20.44%	54.38%	85.77%	4.38%	14.96%	24.82%
0.15	72.26%	21.53%	54.74%	84.67%	4.74%	14.96%	24.82%
0.16	71.17%	21.17%	54.01%	85.04%	4.74%	14.96%	24.82%
0.17	70.80%	20.44%	54.74%	83.94%	4.74%	14.96%	24.82%
0.18	70.07%	21.17%	55.47%	83.94%	4.74%	14.96%	24.82%
0.19	69.71%	22.26%	55.84%	82.85%	4.74%	14.96%	24.82%
0.20	69.71%	22.99%	55.84%	82.12%	4.74%	14.96%	25.18%
k**-NN** k							
5	67.88%	28.83%	57.30%	82.48%	4.74%	14.96%	24.82%
6	67.52%	28.83%	56.57%	86.13%	4.74%	14.96%	24.82%
7	68.25%	29.20%	58.03%	87.59%	4.74%	14.96%	24.82%
8	69.71%	29.93%	56.93%	87.96%	4.74%	14.96%	24.82%
9	71.53%	29.20%	56.93%	87.23%	4.74%	14.96%	24.82%
10	71.53%	29.20%	56.57%	87.96%	4.74%	14.96%	24.82%
11	71.17%	29.20%	58.76%	87.96%	4.74%	14.96%	24.82%
12	70.44%	29.56%	59.12%	88.32%	4.74%	14.96%	24.82%
13	70.07%	29.56%	62.41%	88.32%	4.74%	14.96%	24.82%
14	**70.80%**	**29.56%**	**63.50%**	**89.05%**	**4.74%**	**14.96%**	**24.82%**
15	70.07%	29.20%	63.14%	89.78%	4.74%	14.96%	24.82%
NBC							
-	67.52%	21.90%	59.85%	81.75%	4.74%	14.96%	24.82%

TABLE 4.1: Results of four CPs on the morphological data. We show the accuracy, and the certainty and error rates for three levels of confidence.

Method	Classifier			CP		
	Accuracy	**TNR**	**TPR**	**Accuracy**	**TNR**	**TPR**
ANN	71.16%	59.90%	82.50%	72.26%	60.06%	83.2%
SVM	73.72%	63.50%	83.94%	73.72%	63.50%	83.94%
NBC	68.24%	54.70%	81.80%	67.52%	63.64%	74.49%
***k*-NN**	70.07%	59.10%	81.00%	70.80%	58.39%	83.21%

TABLE 4.2: Comparing Accuracy, True Negative Rate (TNR), and True Positive Rate (TPR) of four classifier algorithms with the corresponding CPs, on the morphological data.

4.1.3 Discussion

As expected, the error rates confirm the validity of the CPs as they are always near the pre-set significance levels, regardless of the non-conformity measures defined and parameters that have been chosen for each algorithm. On the morphological data, the SVM-CP provides the best average of accuracy, while the ANN-CP gives much higher certainty rates. The results of the ANN-CP are improved even more when the size of the hidden layer is limited to a single neuron. At 95% level of confidence the ANN-CP gives 35.04% of certain prediction regions. This means that a significant amount of patients will get a prediction in which the error will not exceed the 5% that is allowed. Given the difficulty of the task, this is arguably a useful result. Moreover, as we decrease the confidence level, the certainty rates increase dramatically.

The accuracy between the classifiers and their corresponding CPs have no significant difference, as expected. We highlight here that our aim is not to improve accuracy. We show that CPs can provide more information in each prediction while accuracy is retained. As shown in Table 4.2, all algorithms provide higher TPRs and lower TNRs on the morphological data. This means

		Certainty			Error		
CP	**Acc.**	**95%**	**85%**	**75%**	**95%**	**85%**	**75%**
ANN neurons							
0	70.80%	34.67%	67.15%	88.69%	4.38%	12.77%	22.99%
1	69.34%	34.67%	67.15%	87.59%	4.74%	14.96%	24.09%
2	71.17%	32.12%	66.42%	86.50%	4.01%	14.23%	22.99%
3	70.07%	34.31%	66.79%	89.78%	4.38%	14.23%	24.82%
4	71.17%	33.21%	67.15%	90.15%	4.38%	13.50%	24.45%
5	70.07%	34.31%	66.42%	89.42%	4.74%	13.87%	24.45%
6	70.80%	32.85%	66.06%	89.05%	4.74%	13.50%	23.72%
7	71.17%	31.02%	67.15%	88.69%	4.01%	13.87%	22.99%
8	71.17%	33.94%	66.79%	89.78%	5.11%	13.50%	24.45%
9	71.53%	32.85%	67.52%	89.78%	4.74%	13.14%	24.09%
10	**71.53%**	**34.31%**	**67.15%**	**90.15%**	**4.74%**	**13.50%**	**24.09%**
SVM spread							
1	64.96%	17.88%	53.65%	77.74%	4.74%	14.96%	24.45%
2	68.25%	18.25%	55.84%	82.12%	4.74%	14.96%	24.45%
3	69.71%	25.55%	57.30%	81.02%	4.74%	14.96%	24.82%
4	69.71%	28.47%	55.47%	84.67%	4.74%	14.96%	24.82%
5	69.34%	28.83%	57.30%	84.67%	4.74%	14.96%	24.82%
6	69.71%	30.29%	59.85%	86.50%	4.74%	14.60%	24.82%
7	69.34%	31.39%	61.31%	86.86%	4.74%	15.33%	24.82%
8	69.34%	31.75%	61.68%	87.59%	4.74%	14.96%	24.82%
9	70.07%	31.75%	62.77%	87.96%	4.74%	14.96%	24.82%
10	**69.34%**	**32.12%**	**63.87%**	**88.32%**	**4.74%**	**14.96%**	**24.82%**
k-NN k							
20	70.07%	32.85%	68.98%	89.78%	4.74%	14.96%	24.82%
21	70.44%	33.21%	69.34%	89.05%	4.74%	14.96%	24.82%
22	70.07%	33.94%	69.34%	89.42%	4.74%	14.96%	24.82%
23	70.07%	33.94%	69.71%	89.42%	4.74%	14.96%	24.82%
24	70.80%	33.94%	69.71%	90.15%	4.74%	14.96%	24.82%
25	70.80%	33.94%	70.80%	90.15%	4.74%	14.96%	24.82%
26	**70.80%**	**34.31%**	**70.44%**	**90.15%**	**4.74%**	**14.96%**	**24.82%**
27	71.17%	33.94%	68.61%	90.15%	4.74%	14.96%	24.82%
28	71.53%	33.94%	68.61%	90.15%	4.74%	14.96%	24.82%
29	71.17%	33.58%	68.61%	90.15%	4.74%	14.96%	24.82%
30	71.53%	33.94%	68.61%	90.15%	4.74%	14.96%	24.82%
NBC							
-	69.34%	27.01%	57.66%	83.94%	4.74%	14.96%	24.82%

TABLE 4.3: Results of four CPs on the texture data. We show the accuracy, and the certainty and error rates for three levels of confidence.

Method	**Classifier**			**CP**		
	Accuracy	**TNR**	**TPR**	**Accuracy**	**TNR**	**TPR**
ANN	68.97%	66.40%	71.50%	71.53%	66.12%	82.42%
SVM	69.70%	64.23%	75.18%	69.34%	63.50%	76.64%
NBC	70.07%	80.30%	59.90%	69.34%	74.77%	65.87%
***k*-NN**	70.43%	70.10%	70.80%	70.80%	65.69%	75.91%

TABLE 4.4: Comparing Accuracy, True Negative Rate (TNR), and True Positive Rate (TPR) of four classifier algorithms with the corresponding CPs, on the texture data.

that patients with symptomatic plaques will have more chance to be identified, whereas asymptomatic plaques could be miss-classified as symptomatic. This kind of wrong predictions could yield unnecessary complications, such as surgery, in which other risks may be introduced. For this reason, a valid confidence measure in each prediction could play an important role for this application.

The accuracy on the texture data is slightly lower than that of the morphological data, but with no significant change in the certainty and error rates. As shown in Table 4.3, the best average of accuracy and certainty rates is provided by the ANN-CP, which gives 71.53% accuracy and 34.31% certainty at the 95% level of confidence. The size of the hidden layer to achieve this result is set to 10 neurons, which is contrary to the size of a single neuron that has been set for the morphological data. This result suggests that the texture data is more complex than the morphological data. In order to achieve good results, the range of the parameters for the SVM-CP and k-NN-CP has been changed to a RBF parameter of $1 - 10$, and $k = 20, \ldots, 30$ respectively. The k-NN-CP has also performed well in these experiments giving similar results with those of the ANN-CP.

On the texture data, the TNR has increased in most of the algorithms, while the TPR has decreased (see Table 4.4). Therefore, more asymptomatic plaques are identified with the texture data, rather than symptomatic plaques. The miss-classification of a symptomatic plaque is critical for the patient, and thus again, a confidence measure in this kind of predictions seems to be important. It is remarkable that the NBC gives high TNR and low TPR, which is contradictory to what the rest of the algorithms give. The k-NN method gives a balanced result, where both TNR and TPR lie at about the same level. The results of the CPs are satisfactory, as the accuracy is preserved while extra information is provided.

In Table 4.5, we compare our accuracy on the morphological data with the results of [1], which describes work on the same data using identical experimental settings as ours. We also compare our accuracy results on the texture data with the results of [2]. We would like to note that the dataset used in [2] is an older version of our dataset, which contains only 230 instances. Moreover, the experimental settings in [2] are slightly different than ours. Nevertheless, we are still able to show that the accuracy obtained here is very close to the best accuracy obtained in [2]. We show the best accuracies achieved by the SVM and the Probabilistic Neural Network (PNN) classifiers used in [1], on the L-image group of the morphological data. We also include the results of our SVM and ANN CPs which are highlighted in Table 4.1. For the texture data, we show the results of the 10 combined SVM classifiers and the 10 combined k-NN classifiers used in [2]. We compare these with our results of the k-NN-CP and ANN-CP as highlighted in Table 4.3. From the comparison, we are able to see that the accuracies on the morphological data and the texture data remain at about the same level with the accuracies of the two previous studies. Thus,

Accuracy	**TNR**	**TPR**
74.09%	67.55%	88.37%
Certainty		
95%	**85%**	**75%**
40.88%	68.98%	86.50%
Error		
95%	**85%**	**75%**
5.47%	14.23%	24.45%

TABLE 4.6: Results of ANN-CP using both morphological and texture data.

we show that our CPs preserve accuracy while they provide important extra information (the confidence measures) for the expert physicians.

Method	**Accuracy on morphological data**	**Method**	**Accuracy on texture data**
SVM[1]	73.72%	**Comb. k-NN**[2]	68.8%
PNN[1]	70.44%	**Comb. SVM**[2]	73.1%
SVM-CP	73.72%	**k-NN-CP**	70.80%
ANN-CP	72.26%	**ANN-CP**	71.53%

TABLE 4.5: Comparing accuracy of the classifiers in [1] and [2] with the accuracy of our CPs on the morphological and texture data.

4.1.4 Combined data

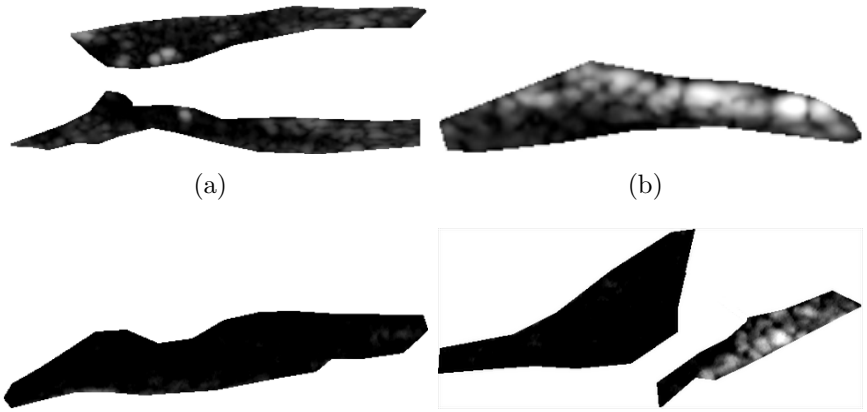
Based on the results in Tables 4.1 and 4.3, we have built a combined CP which works on both morphological and texture data in parallel. An ANN-CP is trained on the morphological data with 1 hidden neuron, and another ANN-CP is trained on the texture data with 10 neurons. We have chosen

the ANN-CPs, since they provide the best results of accuracy and certainty rates. Given that the TPR of the ANN-CP is higher on the morphological data (see Table 4.2), when the assumed label of a test instance is symptomatic (positive), we use the ANN-CP which was trained on the morphological data to get a p-value. Otherwise, we get a p-value from the ANN-CP of the texture data which gives a better TNR. As shown in Table 4.6, the accuracy rates have improved. Moreover, the certainty rates have increased, while the error rates remain near the predefined significance levels.

4.1.5 Output for selected images

One of the problems with current methods is the low percentage of correct classification results (around 70-75%). This is because of the nature of the problem. We are working on base line images, events recorded are for a period of eight years after images were captured. Thus, even though some of the plaques can be characterized by the experts as dangerous, we may have events that did not occur during the monitoring period. Instead, events may have occurred later or not at-all.

Figure 4.1 shows four plaques that were used in our experiments. Figure 4.1(a) shows a plaque that was classified as symptomatic by the ANN-CP, but with low confidence (70.80%). In this example, if we raise the required classification confidence above 70.80%, then the plaque classification changes to uncertain. The expert physician assessed this plaque as average risk. Thus, the CP classification, although wrong, shows low confidence in the prediction. A more accurate prediction is given in Figure 4.1(b). In this case, the plaque was classified as asymptomatic with very high confidence (99.64%), in agreement



(a) (b)

(c) (d)

FIGURE 4.1: (a) Plaque that was classified as low confidence (70.8%) symptomatic. The subject was asymptomatic but was classified as an average risk image by the expert physician. (b) Plaque that was classified as high confidence (99.64%) asymptomatic. This subject was asymptomatic and classified as low risk for symptoms by the expert physician. (c) Plaque that was classified as low confidence (69.34%) symptomatic. This subject had an AF event and was classified as low risk for stroke but high risk for AF by the expert physician. (d) Plaque that was classified as high confidence (99.64%) symptomatic. This subject had a stroke event and was classified as high risk for symptoms by the expert physician.

with the expert physician. A symptomatic example is given in Figure 4.1(c). In this example, the plaque was classified as symptomatic but with low confidence (69.34%). If we raise the confidence requirement to above 69.34%, then the plaque classification changes to uncertain. However, this is one of the plaques that resulted in AF and was classified as low risk for stroke but high risk for AF by the expert physician. A more accurate symptomatic classification is given in Figure 4.1(d). In this example, the plaque was classified as symptomatic with high confidence (99.64%). Furthermore, this is a plaque associated with a stroke event and was identified as a dangerous plaque by the expert physician.

4.1.6 Summary

The classification of symptomatic and asymptomatic atherosclerotic plaques is a crucial task as it can be used to predict the risk of stroke. In this work, we have applied the Conformal Prediction framework on four machine learning algorithms in order to assign reliable confidence measures to the recognition of symptomatic or asymptomatic plaques; thus assess the risk of stroke. Our results demonstrate the validity of the produced confidence measures and their importance in the application of stroke prediction.

The proposed methods provide the expert physicians with a reliable confidence measure for each prediction, which can be trusted based only on the i.i.d. assumption. As the confidence measures that we provide are valid (in the sense that they are proven to be correct), the expert physician needs to have no further knowledge about the methods in order to be able to trust the confidence measure in each prediction.

4.2 Osteoporosis risk assessment

Osteoporosis is a systemic skeletal disease characterized by low bone density and microarchitectural deterioration of bone tissue with a consequent increase in bone fragility. Early osteoporosis is not usually diagnosed and remains asymptomatic; it does not become clinically evident until fractures occur. Loss of bone density occurs with advancing age and rates of fracture increase markedly with age, giving rise to significant morbidity and some mortality.

Osteoporosis is three times more common in women than in men, partly because women have a lower peak bone mass and partly because of the hormonal changes that occur at the menopause. Estrogens have an important function in preserving bone mass during adulthood, and bone loss occurs as levels decline, usually from around the age of 50 years [68].

4.2.1 Osteoporosis data

The World Health Organisation (WHO) has defined the disease of Osteoporosis as a Bone Mineral Density (BMD) which is lower than 2.5 standard deviations from the average of young healthy adults. Furthermore, BMD that is 1 standard deviation lower is defined as Osteopenia, which is a precursor to Osteoporosis [68]. DEXA stands for Dual Energy X-ray Absorptiometry, and is a standard test for BMD. DEXA scanners throw an X-ray beam at the lumbar vertebrae and measure the shadow cast by the bones. In Figure 4.2 we include a sample image of the lumbar spine of a DEXA scan. Software in the machine estimates the amount of calcium in the bone based on the darkness of the shadow. The result is expressed as a number of grams per square centimeter, which is defined as the Bone Mineral Density (BMD). In Table 4.7, we show how the BMD is mapped to a t-score value compared against the average of young healthy adults.

We have collected data from various clinics in Cyprus. The data were collected during the research project named "Development of New Venn Prediction Methods for Osteoporosis Risk Assessment" (research contract TPE/ORIZO/0609(BIE)/24), which was funded by the Research Promotion Foundation of Cyprus (and co-funded by the Structural Funds of the European Union).

BMD	1.44	1.32	1.20	1.08	0.96	0.84	0.72	0.60
YA T-Score	2	1	0	-1	-2	-3	-4	-5

TABLE 4.7: Young Adult (YA) T-score based on the Bone Mineral Density (BMD) according to the Wolrd Health Organisation (WHO).

The project started on the 1st of September 2011 and its duration was 24 months. We have gathered data using a questionnaire that was given to patients to complete after at least one DEXA scan. The patients may have previous history of osteoporosis and may already follow therapy. The questionnaire was constructed by physicians and contains questions that are relevant to Osteoporosis risk factors. Each case is classified as "Normal" or "Risk of Osteoporosis" based on the patient's spine t-score that was given by the DEXA scan. According to the WHO, patients with a t-score above -1 are diagnosed as healthy, therefore we have classified patients into two classes: "Normal" for patients with t-score above -1, and "Risk of Osteoporosis" otherwise. In Table 4.8, we give the list of attributes of this dataset.

4.2.2 Data Preprocessing

We have performed an initial analysis of the data collected in order to find which attributes contain the necessary information for correct classification. Specifically, we have tried various Feature Selection methods that exist in the literature, such as Correlation Based Feature Selection (CBFS) [42], Principal Component Analysis (PCA) [69], Chi-Squared Feature Selection (CSFS), Information Gain Feature Selection (IGFS) [70], and Feature Selection with SVM (SVMFS) [71]. For each feature selection method, we have experimented with

#	**Attribute name**	**Type**	#	**Attribute name**	**Type**
1	Gender	Binary	35	Receive Thyroxine	Binary
2	Age	Numeric	36	Receive Estrogens	Binary
3	Weight	Numeric	37	Neurogenic Anorexia	Binary
4	Height	Numeric	38	Malabsorption syndrome	Binary
5	Start of Menstruation	Numeric	39	Chronic liver diseases	Binary
6	End of Menstruation	Numeric	40	Inflammatory bowel diseases	Binary
7	Pregnacies	Numeric	41	Transplantation	Binary
8	Smoking now	Binary	42	Chronic renal failure	Binary
9	Smoking in the past	Binary	43	Prolonged immobilization	Binary
10	No smoking	Binary	44	Cushing's syndrome	Binary
11	Years of past smoking	Numeric	45	Epilepsy	Binary
12	Years of current smoking	Numeric	46	Insulin Dependent	Binary
13	Cigarettes per day	Numeric	47	Ovariectomy before menopause	Binary
14	Alchohol intake per day	Numeric	48	Chronic gastrointestinal disorders	Binary
15	Caffeine intake per day	Numeric	49	Paget's Disease	Binary
16	History of fracture	Binary	50	Hyperthyroidism	Binary
17	Hip fracture	Binary	51	Parathyroid gland disease	Binary
18	Spine fracture	Binary	52	Receive Steroids	Binary
19	Wrist fracture	Binary	53	Receive Thyroxine	Binary
20	Low energy	Binary	54	Anticonvulsants (for seizures)	Binary
21	High energy	Binary	55	Diuretics	Binary
22	Sports	Binary	56	Heparin	Binary
23	History of osteoporosis	Binary	57	Chemotherapy	Binary
24	Osteoporosis in family	Binary	58	Treatment of osteoporosis	Binary
25	Loss of height	Binary	59	Alendronati	Binary
26	Kyphosis	Binary	60	Risedronati	Binary
27	End of menstrual bleeding	Binary	61	Zoledronati	Binary
28	Arthritis	Binary	62	Raloxifeni	Binary
29	Secondary Osteporosis	Binary	63	Strontio	Binary
30	Breast feeding	Binary	64	Parathormoni	Binary
31	Avoidance of milk	Binary	65	Denosoymapi	Binary
32	Avoidance of sex	Binary	66	Kalsitonini	Binary
33	Diarrhea	Binary	67	Calcium + Bitamin D	Binary
34	Receive Cortisone	Binary	68	Calcium	Binary

TABLE 4.8: Table of attributes in the Osteoporosis dataset.

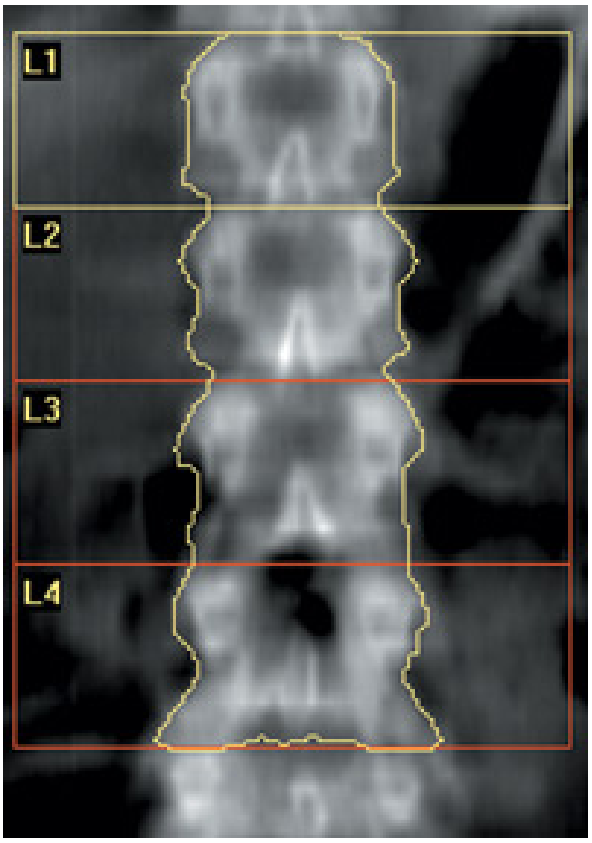


FIGURE 4.2: Image of the Lumbar Spine AP (Anterior Posterior) from a DEXA Scan.

the data using SVM with Sequential Minimal Optimisation (SMO) and RBF kernel [55] in order to compare the results. For each method we performed a 10-fold cross validation experiment. In 10-fold cross validation, we repeat 10 experiments. In each experiment, we remove 10% of the data and we train the underlying algorithm on the remaining 90% of the data. We then evaluate the method on the 10% data which have been removed from the training phase. We repeat the same process for each fold, by removing non-overlapping blocks of the data in each experiment.

In Table 4.9, we compare the results of the five methods. The results show the

FS method	Accuracy	TP	FP
SVMFS	72.23%	61.00%	18.00%
PCA	68.12%	60.00%	25.00%
IGFS	69.92%	62.00%	24.00%
CSFS	70.17%	55.00%	18.00%
CBFS	69.66%	51.00%	15.00%

TABLE 4.9: Results of the 5 feature selection methods.

RANK	CBFS	CSFS
1	Cortizone	Previous therapy
2	Heparine	Weight
3	Previous therapy	End of menstruation cycle
4	Alendronati	Calcium
5	Zoledronati	Age
6	Calcium	Alendronati
7	Weight	Cortizone
8	End of menstruation cycle	Heparine
9		Smoking
10		Smoked in the past

TABLE 4.10: Features selected by the CBFS and CSFS methods.

accuracy of the classifier, the True Positive (TP) rates and False Positive (FP), rates. In Tables 4.10 and 4.11, we show the selected attributes for each of the methods. For CBFS, we show the best subset of attributes, while for the rest of the methods, we rank the top 10 attributes based on their score that they gained from the feature selection methods.

4.2.3 Experiments

In this section we describe and analyse the results of four algorithms, the ANN-TVP, ANN-IVP, SVM-TVP and SVM-IVP. We experiment with both

RANK	IGFS	PCA	SVMFS
1	Previous therapy	Previous therapy	End of menstruation
2	Weight	Weight	Heparine
3	End menstruation	End menstruation	Cortizone
4	Calcium	Calcium	Weight
5	Heparine	Age	Calcium
6	Alendronati	Alendronati	Smoking
7	Age	Cortizone	Previous therapy
8	Cortizone	Heparine	Age
9	Smoking	Smoking	Smoked in the past
10	Smoked in the past	Smoked in the past	Alendronati

TABLE 4.11: Features selected by the IGFS, PCA, and SVMFS methods.

the online and offline settings. In the online setting, we are able to evaluate the validity of the probabilistic outputs of the algorithms. First, we conduct experiments on the whole set of attributes, and later we compare the results with several feature selection methods.

4.2.3.1 Artificial Neural Network Venn Predictor

We evaluate ANN-TVP and ANN-IVP on 10-fold cross validation experiments. In the results, we show the average accuracy, and the average lower and upper probability bounds. Since VPs provide well-calibrated probabilistic outputs, the accuracy of the VPs is expected to fall within the lower and upper probability bounds. Moreover, we show the BS of the experiments, which is calculated on the mean of the lower and upper probability bounds. As we can see in Table 4.12, the accuracy of the ANN-TVP is 67.23%, which is within the bounds of 65.97% - 70.19%.

Method	Accuracy	Low bound	Upper bound	Brier Score
ANN-TVP	67.23%	65.97%	70.19%	0.4314

TABLE 4.12: Offline results of the ANN-TVP on the Osteoporosis dataset.

Method	Accuracy	Low bound	Upper bound	Brier Score
ANN-IVP	62.47%	57.17%	58.61%	0.4715

TABLE 4.13: Offline results of the ANN-IVP on the Osteoporosis dataset.

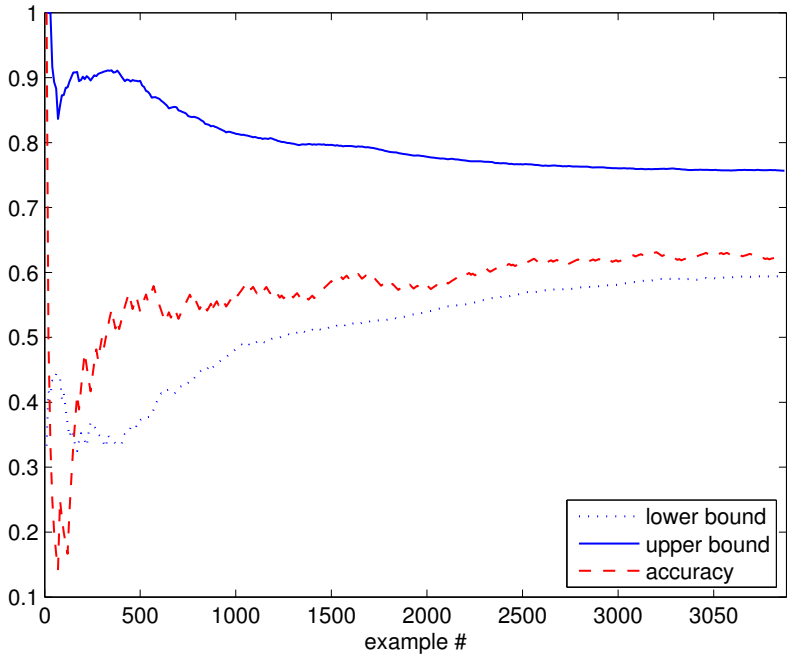


FIGURE 4.3: Online experiments with ANN-TVP on the Osteoporosis dataset.

In Table 4.13, we show the average accuracy, and the average lower and upper probabilities given by the ANN-IVP in the offline setting (10-fold cross validation). The actual accuracy is within the lower and upper bounds provided, but the results are less accurate than those of the ANN-TVP method.

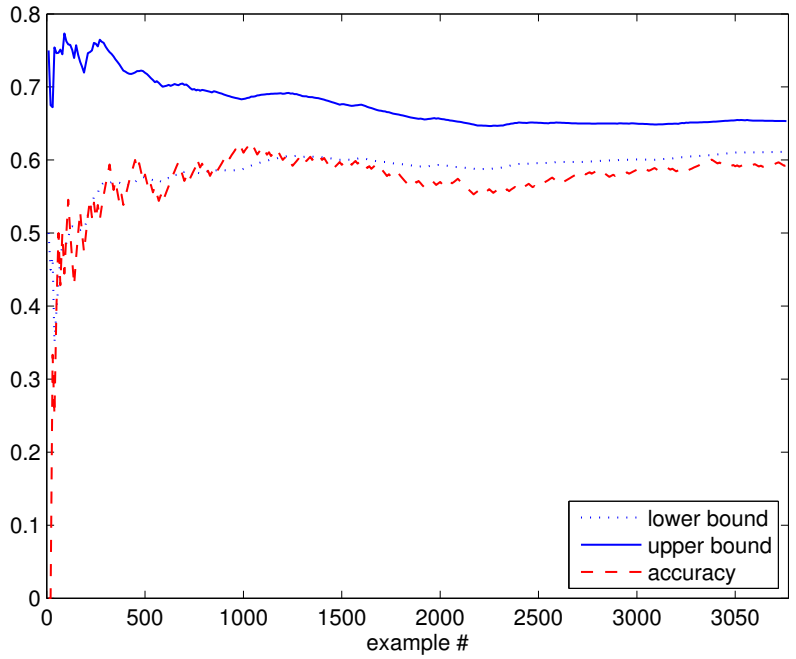


FIGURE 4.4: Online experiments with ANN-IVP on the Osteoporosis dataset.

In Figure 4.3, we show the online results of the ANN-TVP on the Osteoporosis dataset. The online experimental results match the offline results. As expected the actual accuracy of the Venn Predictor falls within the lower and upper bounds. In Figure 4.4, we show the online results of the ANN-IVP. Here, the accuracy is slightly lower than the low probability bound provided, but the results are still well-calibrated and even more narrower than those of the ANN-TVP method. The reason that accuracy might fall slightly outside the lower and upper bounds could be explained by the nature of the IVP method. The experiments start with an empty training set and an empty calibration set. This might deviate the cumulative results from the starting point.

Method	Accuracy	Low bound	Upper bound	Brier Score
SVM-TVP	65.71%	64.21%	71.83%	0.4434

TABLE 4.14: Offline results of the SVM-TVP on the Osteoporosis dataset.

Method	Accuracy	Low bound	Upper bound	Brier Score
SVM-IVP	64.78%	62.98%	65.09%	0.4616

TABLE 4.15: Offline results of the SVM-IVP on the Osteoporosis dataset.

4.2.3.2 Support Vector Machine Venn Predictor

We perform 10-fold cross validation experiments on our Osteoporosis dataset with the SVM-TVP and SVM-IVP methods. In the results (Table 4.14), we show the average accuracy, and the average lower probabilities and upper probabilities of the SVM-TVP method. The results of the SVM-TVP are similar to those of the ANN-TVP.

In Table 4.15, we show the offline results of the SVM-IVP. The results of the SVM-IVP compared with the results of the SVM-TVP method are similar, although the probabilistic bounds given by the SVM-IVP seem to be narrower.

In Figure 4.5, we show the online results of the SVM-TVP on the Osteoporosis dataset. As with the online results of the ANN-TVP, the results demonstrate the validity of the Venn Predictor, regardless of the underlying algorithm used. In Figure 4.6, we show the online results of the SVM-IVP algorithm. The accuracy is within the bounds of the Venn Predictor and the width of the bounds is much narrower than the width of the bounds provided by the counterpart SVM-TVP algorithm.

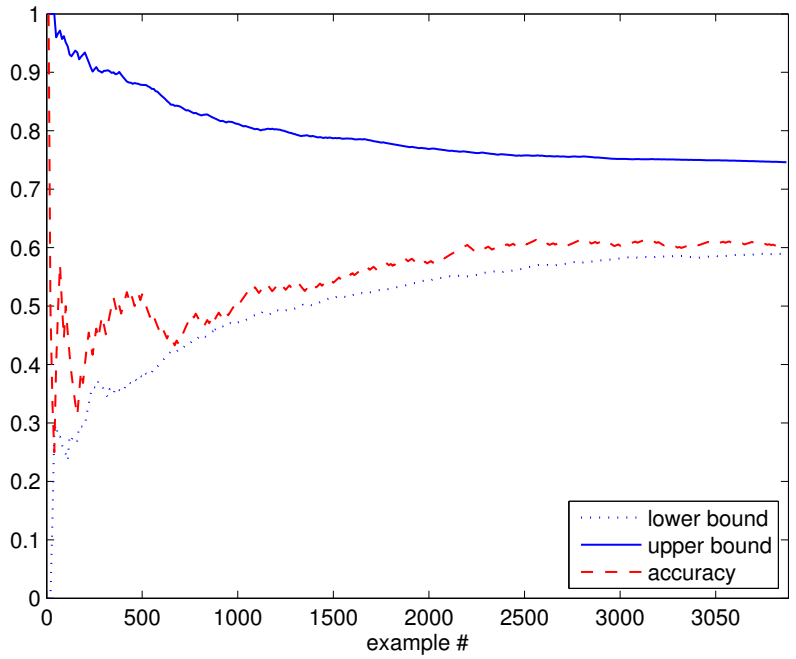


FIGURE 4.5: Online experiments with SVM-TVP on the Osteoporosis dataset.

4.2.3.3 Comparison of Feature Selection Methods

We compare the results of the four methods ANN-TVP, SVM-TVP, ANN-IVP, and SVM-IVP on the five Feature Selection Methods that we have used in our data pre-processing. Specifically, we have tried the following Feature Selection methods: Correlation Based Feature Selection (CBFS) [42], Principal Component Analysis (PCA) [69], Chi-Squared Feature Selection (CSFS), Information Gain Feature Selection (IGFS) [70], and Feature Selection with SVM (SVMFS) [71]. For each feature selection method and for each VP we perform a 10-fold cross validation experiment.

In Tables 4.16, 4.17, 4.18, 4.19, and 4.20, we show the results for each feature

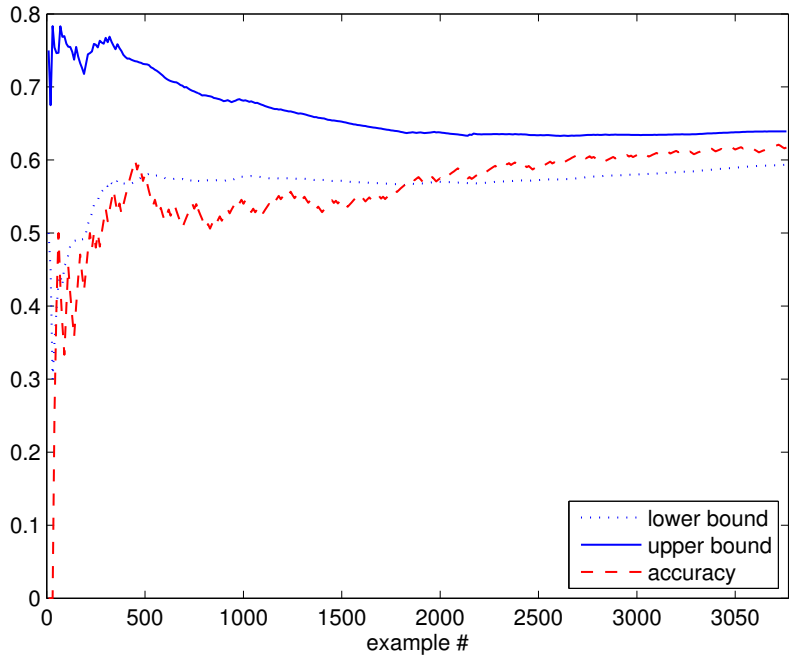


FIGURE 4.6: Online experiments with SVM-IVP on the Osteoporosis dataset.

selection method. In each table, we show the average accuracy, the average lower probability, the average upper probability and the Brier Score for each 10-fold cross validation experiment.

From the results, we can see that the IGFS and SVMFS methods have given the best results, while the SVM-IVP method with SVMFS has the best results amongst all Venn Predictors with a 72.49% accuracy.

After discussions with the medical doctors who were involved in the project, we have conducted further experiments by manually removing specific attributes of the dataset. We have removed attributes which describe previous therapy of Osteoporosis, in order to evaluate the accuracy of our methods, without such

Method	ANN-TVP	SVM-TVP	ANN-IVP	SVM-IVP
Accuracy	66.58%	69.41%	68.89%	72.49%
Lower Probability	64.44%	69.84%	65.62%	67.41%
Upper Probability	69.20%	75.04%	67.53%	69.30%
BS	42.56%	41.82%	43.44%	41.14%

TABLE 4.16: Comparison of the VPs on the Osteoporosis dataset with SVMFS data preprocessing.

Method	ANN-TVP	SVM-TVP	ANN-IVP	SVM-IVP
Accuracy	62.72%	58.35%	61.70%	59.64%
Lower Probability	60.72%	23.13%	63.51%	59.66%
Upper Probability	73.22%	96.71%	65.41%	61.59%
BS	46.50%	48.33%	46.34%	48.25%

TABLE 4.17: Comparison of the VPs on the Osteoporosis dataset with PCA.

Method	ANN-TVP	SVM-TVP	ANN-IVP	SVM-IVP
Accuracy	69.92%	69.92%	68.12%	68.89%
Lower Probability	66.51%	67.01%	65.80%	66.69%
Upper Probability	72.07%	75.13%	67.68%	68.61%
BS	42.21%	40.86%	43.44%	42.66%

TABLE 4.18: Comparison of the VPs on the Osteoporosis dataset with IGFS.

information. Specifically, we have removed the attributes with the related medical therapy information such as "Alendronati", "Zoledronati", "Risedronati", "Raloxifeni", "Strontio", "Parathormoni", "Denosoumapi", and "Kalsitonini" (see Table 4.8). In Table 4.21, we show the results of our VPs, ANN-TVP, SVM-TVP, ANN-IVP, and SVM-IVP.

Method	ANN-TVP	SVM-TVP	ANN-IVP	SVM-IVP
Accuracy	61.95%	59.38%	65.30%	58.87%
Lower Probability	53.14%	25.98%	66.42%	63.96%
Upper Probability	83.36%	97.68%	68.31%	65.80%
BS	46.26%	45.12%	45.91%	48.95%

TABLE 4.19: Comparison of the VPs on the Osteoporosis dataset with CSFS.

Method	ANN-TVP	SVM-TVP	ANN-IVP	SVM-IVP
Accuracy	68.89%	66.32%	66.84%	68.64%
Lower Probability	69.06%	67.99%	65.27%	65.44%
Upper Probability	71.81%	71.07%	67.15%	67.35%
BS	43.40%	43.95%	44.62%	43.33%

TABLE 4.20: Comparison of the Venn Predictors on the Osteoporosis dataset with CBFS data preprocessing.

Method	ANN-TVP	SVM-TVP	ANN-IVP	SVM-IVP
Accuracy	61.58%	60.05%	56.58%	64.20%
Lower Probability	48.62%	29.62%	59.04%	60.00%
Upper Probability	82.21%	96.21%	60.51%	61.71%
BS	46.56%	44.74%	47.84%	45.84%

TABLE 4.21: Comparison of the Venn Predictors on the Osteoporosis dataset with manually removed attributes.

4.2.4 Summary

We have applied Venn Prediction to the problem of Osteoporosis Risk Assessment. We have evaluated our method on real-world data that we have collected from various clinics in Cyprus. Our results demonstrate that our method provides well-calibrated probabilistic outputs in the predictions that can be useful

in practice. Precisely, patients may get a prognosis based on Osteoporosis risk factors before the performance of a DEXA scan.

4.3 Childhood Abdominal Pain Diagnosis

Acute abdominal pain diagnosis in children can be characterized as a classification problem. Although many cases of acute abdominal pain are benign, some can lead to further complications and morbidity. There are many disorders that can cause abdominal pain. The most common medical cause is gastroenteritis, and the most common surgical cause is appendicitis. In most instances, abdominal pain can be diagnosed through the medical history and physical examination. In the acute surgical abdomen, pain generally precedes vomiting, while the reverse is true in medical conditions. Diarrhoea is often associated with gastroenteritis or food poisoning. Appendicitis should be suspected in any child with pain in the right lower quadrant. Signs that suggest an acute surgical abdomen include involuntary guarding or rigidity, marked abdominal distension, marked abdominal tenderness, and rebound abdominal tenderness. If the diagnosis is not clear after the initial evaluation, repeated physical examination is required, and surgical consultation is necessary if a surgical cause is suspected [72].

The application of machine learning methods to the problem of acute abdominal pain (AAP) diagnosis has been the subject of quite a few studies. In [73] two Bayesian methods (Naive and Proper Bayes) were applied to a relatively large dataset consisting of 6387 adult patients. The results of the two Bayesian methods were compared with those of the decision tree algorithm CART and

the preliminary diagnoses of hospital physicians. The Naive Bayes classifier gave 74% correct diagnoses outperforming all other techniques and coming relatively close to the 76% correct diagnoses of the hospital physicians.

Mantzaris et al. [74] studied the application of backpropagation and probabilistic neural networks to a dataset of children with AAP. The experimental results showed that the backpropagation neural networks had a very satisfactory performance which was better than that of the probabilistic neural networks. The same dataset was also used by Anastassopoulos and Iliadis [75] who evaluated the performance of various neural network architectures using different learning rules, transfer functions and optimisation algorithms.

We have applied the Conformal Prediction framework to the problem of Childhood Abdominal Pain Diagnosis, and have created a prototype decision support tool for paediatricians. We provide experimental results on collected data that we have gathered during examination by 10 paediatricians and 4 paediatric surgeons. The data gathering was conducted during research project PLHRO/0506/22: "Development of New Conformal Prediction Methods with Applications in Medical Diagnosis", which was funded by the Research Promotion Foundation of Cyprus.

4.3.1 Dataset

The data were created by questionnaire forms that the physicians were filling for each of their patients during examination. We have collected 804 instances of recorded information, symptoms, and final diagnoses.

We have applied a Feature Selection method to identify which patient information contribute to the classification task. The method used is Correlation-Based Feature Selection (CBFS), which identifies subsets of features that have low intercorrelation, and high correlation with the classes. A list of the data attributes used is presented in Table 4.22.

Name	**Type**	**Name**	**Type**
Gender	Binary	History of jaundice	Binary
Age	Numeric	Similar Pain before	Binary
Pain-site Onset	Binary	Drugs being taken	Binary
Pain-site Present	Binary	Previous Surgery	Binary
Site of Tenderness	Numeric	Rebound	Binary
Aggravating Factors	Numeric	Guarding	Binary
Relieving Factors	Numeric	Rigidity	Binary
Duration of Pain	Numeric	Vomiting	Binary
Progress of Pain	Numeric	Type of Vomitus	Numeric
Type of Pain	Numeric	Bowel Habit	Numeric
Radiation of Pain	Numeric	Bowel Sounds	Binary
Severity of Pain	Numeric	Abdominal Movements	Binary
Nausea	Binary	Murphy's Test	Binary
Anorexia	Binary	Rectal Examination	Numeric
Abdominal Distension	Binary		
Abdominal Masses	Binary		
Abdominal Scar	Binary		
Lapparoscopy Scar	Binary		

TABLE 4.22: List of attributes of the Childhood Abdominal Pain dataset.

Data instances can be classified into one of five predefined categories. The categories are listed in Table 4.23.

Classes
Appendicitis
Gastroenteritis
Urinary Tract Infection
Infantile Colic
Non Specific Abdominal pain
("A mixed bag" including Mesenteric adenitis,
Referred pain, Constipation, Intestinal obstruction)

TABLE 4.23: List of classes of the Childhood Abdominal Pain dataset.

4.3.2 Experiments

We conduct offline (cross-validation) and online experiments on the data collected. We apply four CP algorithms and evaluate the results. Moreover, we conduct experiments with TVP and IVP and provide experimental results.

4.3.2.1 CP experiments

We apply ten-fold cross validation on the dataset in order to evaluate our CPs. In these experiments, we have used four CPs, one based on the NB classifier, one on ANNs, another based on k-NN, and the fourth one using our Genetic Algorithm approach. A comparison of the accuracy provided by the CPs is given in Table 4.24. The parameters of the algorithms were chosen according to empirical results. We compare the accuracy results of the four CPs together with the accuracy provided by the classical algorithm counterparts (the original NB classifier, ANN, k-NN, and GA). The results show that the NB classifier provides the best accuracy at 80.35%.

Method	Accuracy (%)
Naive Bayes	80.35
Naive Bayes CP	79.73
Neural Network	76.74
Neural Network CP	77.24
k-Nearest Neighbours	74.74
k-Nearest Neighbours CP	74.62
Genetic Algorithm	68.36
Genetic Algorithm CP	65.14

TABLE 4.24: Accuracy results on the Childhood Abdominal Pain dataset.

Method	Confidence	Certainty	Error
Naive Bayes CP	99%	2.49%	1.12%
	95%	29.73%	4.85%
	90%	58.21%	10.32%
	80%	98.26%	20.40%
Neural Network CP	99%	2.49%	0.75%
	95%	37.69%	4.23%
	90%	57.96%	8.33%
	80%	86.82%	18.28%
k-Nearest Neighbours CP	99%	16.60%	1.00%
	95%	41.02%	4.60%
	90%	62.12%	9.55%
	80%	85.79%	18.95%
Genetic Algorithm CP (2.18)	99%	0.38%	0.74%
	95%	29.87%	4.18%
	90%	45.31%	8.53%
	80%	49.36%	9.41%
Genetic Algorithm CP (2.19)	99%	0.16%	0.73%
	95%	27.58%	4.12%
	90%	44.13%	8.41%
	80%	49.36%	9.41%

TABLE 4.25: Certainty and error results on the Childhood Abdominal Pain dataset.

Method	NB-TVP	NB-IVP
Accuracy	79.85%	79.85%
Lower Probability	78.96%	77.51%
Upper Probability	81.19%	80.33%
BS	0.3329	0.3352

TABLE 4.26: Comparison of the TVP and IVP on the Abdominal Pain Diagnosis dataset.

In Table 4.25, we present the results of the CP methods. Although the NB CP provided the best accuracy, the k-NN CP provides better certainty results at high levels of confidence. The certainty rates are the prediction regions which contained a single prediction. The k-NN CP gives 16.60% certain predictions for 99% level of confidence, whereas the NB and ANN-CP give 2.49% of certainty at the same level of confidence. The GA-CP has given the worst certainty rates, as the fuzzy space provided to the GA may not be representative for the Childhood Abdominal Pain dataset. All CPs give valid error rates which are always below or near the allowed significance level.

4.3.2.2 VP experiments

We have applied our VP and IVP algorithms to the Abdominal Pain Diagnosis dataset. In Table 4.26, we show the average offline results of the two algorithms after 2 separate 10-fold cross validation experiments. We have used the NB classifier as the underlying algorithm for both VPs, since NB is one of the fastest methods for classification. The results of the two algorithms are similar. Both NB-TVP and NB-IVP give an accuracy rate of 79.85%, while the lower and upper probabilities are 78.96% – 81.19% for the NB-TVP and 77.51% – 80.33% for the NB-IVP. The Brier Score (BS) is 33.29% and 33.52%

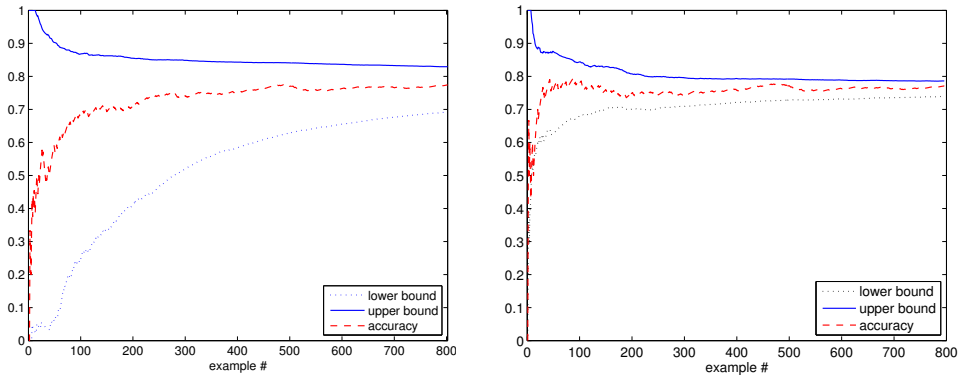


FIGURE 4.7: Online experiments with NB-TVP (left), and NB-IVP (right) on the Abdominal Pain Diagnosis dataset.

respectively. In Figure 4.7, we show the online results of the NB-TVP and NB-IVP. Both algorithms have performed similarly in the online setting. The probabilistic outputs of two VPs become narrower as the training set grows during the online experiments, while the accuracy is within the given bounds.

4.3.3 Summary

The use of CPs and VPs in the case of Childhood Abdominal Pain has shown to be of great importance. The certainty rates and the validity of our methods can provide a decision support tool for the problem of the diagnosis of childhood abdominal pain. A classical ANN or k-NN classifier would not be able to provide certainty in the predictions, whereas a CP can provide certainty and confidence, even with a 77.24% accuracy rate, and a VP can give well-calibrated probabilistic outputs. A physician can rely on the certain predictions of a CP with predefined confidence, whereas for the uncertain predictions the expert physician may take appropriate actions, such as further examinations for the possible diagnoses.

Chapter 5

Conclusion

In this chapter, we make our concluding remarks about the Conformal and Venn Prediction frameworks, the methods that we have developed, and our experimental results. We also provide a list of possible future directions of our research.

5.1 Concluding Remarks

Machine Learning algorithms are incorporated into a wide range of applications, including medical diagnostic systems, robotics, data mining, character and object recognition, and anomaly detection. Many of these applications rely on Machine Learning for providing critical decision support. Nonetheless, Machine Learning algorithms alone, do not guarantee that the correctness of their outputs will always be as expected. In such cases, reliable confidence measures are required to support the decision making process. Even in the case

where algorithms provide some kind of confidence in their outputs, in most of the work found in the literature there is no theory that can support or guarantee that such confidence measures will be well-calibrated. In contrast, the Conformal Prediction (CP) framework, is a novel technique that can provide reliable confidence measures for Machine Learning algorithms, that can guarantee under the i.i.d. assumption, that its confidence measures will be valid (well-calibrated). Additionally, the Venn Prediction (VP) framework, which is an extension of the CP framework, can provide reliable multi-probabilistic outputs for Machine Learning algorithms, which are subsequently guaranteed to be well-calibrated. Probabilistic outputs are not found to be well-calibrated in other Machine Learning methods in the literature.

In this thesis, we have explored and developed new Conformal and Venn Predictors based on Machine Learning algorithms, and we have extensively experimented with our methods. We have provided results for comparison with other learning methods that provide confidence or probabilistic outputs, and we have demonstrated in our experiments, that other methods found in the literature do not always provide reliable or well calibrated results. In contrast, as our experiments show, our CP and VP methods can always provide reliable measures of confidence and well calibrated probabilities.

We have developed a new CP method, based on Genetic Algorithms and Fuzzy-Set theory. Our method was extensively tested and the results of our experiments were presented in this thesis. The results demonstrate the validity of the developed Genetic Algorithm Conformal Predictor (GA-CP). Furthermore, we have extended the CP framework for multi-label classification tasks.

Multi-label classification include applications such as image recognition or document classification, where each instance x_i of such applications may belong into multiple classes, such that the labels of a given instance are more than one. Likewise to traditional learning algorithms, multi-label classifiers found in the literature were not able to guarantee that the confidence in their predictions would be reliable. We have developed a Multi-Label Conformal Predictor (ML-CP), and we have introduced a new confidence measure. We have experimented on two multi-label datasets, and the results have shown that in the same manner to the original CP framework, the extended ML-CP could provide reliable measures of confidence in its outputs.

We have investigated VP and introduced Inductive Venn Prediction (IVP), which greatly improves the computational efficiency of the VP framework. In our experiments, we have thoroughly compared the original Transductive Venn Prediction (TVP) with our IVP method. The results demonstrate the ability of the IVPs to produce well calibrated probabilistic outputs for large datasets, where the original TVP may suffer due to its computational inefficiency problem.

Finally, we have applied our methods on three real-world datasets and we provide experimental results. We have investigated the use of CP for the assessment of the risk of stroke using ultrasound images of atherosclerotic carotid plaques, and we have demonstrated the ability of our methods to provide practical results in difficult cases. We have conducted research on another medical diagnostic problem, which is that of the Osteoporosis Risk Assessment. The dataset that was gathered throughout our research was examined by physicians and then analysed using feature selection algorithms for dimension reduction

of the data. The selected features in our data indicate risk factors of Osteoporosis. In the results, we demonstrate the ability of VPs to provide reliable probabilistic outputs that can indicate risk of Osteoporosis. We have additionally investigated the use of CPs and VPs for the diagnosis of childhood abdominal pain, and have provided experimental results.

5.2 Future work

Our future work may take many possible paths. In this section, we give an outline of three possible directions that our further research may take.

- Genetic Algorithm Approach: We would like to examine the possibilities of improving the confidence values of our GA-CP, using other fuzzy systems. In our implementation we have used standard triangular membership fuzzy-sets for demonstration purposes. We would like to examine other kind of membership functions, such as the trapezoid-fuzzy and the Gaussian membership functions. Moreover, we would like to improve the computational efficiency of our GA-CP, by using Inductive Conformal Prediction. Further, we wish to investigate how the GA algorithm can be incorporated into the VP framework. One possible approach is to define a taxonomy based on the membership output of the fuzzy-space defined in the algorithm. For example, the membership output value that is given by the GA for a given instance can be considered as an input to an identical fuzzy-space, where the fuzzy-space will represent the taxonomy of the data.

- Multi-label Approach: The approach we have taken to build a multi-label CP is one of many possible ones. The decomposition of a multi-label problem into multiple single-label binary classification problems has limitations. This approach does not consider the intercorrelation of the labels or the class imbalance of the datasets. Other approaches may be considered in the future, in order to build reliable CPs for multi-label classification. Furthermore, we would like to examine the extension of the VP framework to multi-label classification. The same approach that we have used to build a ML-CP can be considered for the implementation of a multi-label VP.

- Inductive VP Approach: The probabilistic outputs of the IVP are well-calibrated as expected. In the future, we wish to experiment with more datasets, and use different values of steps m for each re-training of the IVP algorithm, in order to understand how the IVP performs when the assumption of independence of error is violated. Moreover, we aim to try several taxonomies for the developed IVP, and further we wish to investigate other algorithms that can be used to build effective IVPs. The particular IVP proposed in this thesis is based on the SVM classifier, but we expect that our conclusions will be true regardless of which underlying algorithm will be in use.

Bibliography

[1] Efthyvoulos Kyriacou, Marios S. Pattichis, Constantinos S. Pattichis, Andreas Mavrommatis, Christina Christodoulou, Stavros Kakkos, and Andrew Nicolaides. Classification of atherosclerotic carotid plaques using morphological analysis on ultrasound images. *Applied Intelligence*, 30(1): 3–23, 2009. ISSN 0924-669X.

[2] Christina Christodoulou, Constantinos Pattichis, Marios Pantziaris, and Andrew Nicolaides. Texture-based classification of atherosclerotic carotid plaques. *IEEE Transactions on Medical Imaging*, 22(7):902–912, 2003. ISSN 0278-0062.

[3] Thomas M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition, 1997. ISBN 0070428077, 9780070428072.

[4] Volodya Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. New York, Springer, 2005.

[5] Thomas Melluish, Craig Saunders, Ilia Nouretdinov, and Volodya Vovk. Comparing the Bayes and Typicalness frameworks. In *Proceedings of the 12th European Conference on Machine Learning (ECML'01)*, volume 2167 of *Lecture Notes in Computer Science*, pages 360–371. Springer, 2001.

[6] Ilia Nouretdinov, Volodya Vovk, Michael Vyugin, and Alex Gammerman. Pattern recognition and density estimation under the general i.i.d. assumption. In David Helmbold and Bob Williamson, editors, *Computational Learning Theory*, volume 2111 of *Lecture Notes in Computer Science*, pages 337–353. Springer Berlin / Heidelberg, 2001. URL http://dx.doi.org/10.1007/3-540-44581-1_22.

[7] Harris Papadopoulos. Inductive Conformal Prediction: Theory and application to Neural Networks. In Paula Fritzsche, editor, *Tools in Artificial Intelligence*, chapter 18, pages 315–330. InTech, Vienna, Austria, 2008.

[8] Harris Papadopoulos, Kostas Proedrou, Volodya Vovk, and Alex Gammerman. Inductive Confidence Machines for Regression. In *Proceedings of the 13th European Conference on Machine Learning (ECML'02)*, volume 2430 of *LNCS*, pages 345–356. Springer, 2002.

[9] Harris Papadopoulos, Volodya Vovk, and Alex Gammerman. Qualified predictions for large data sets in the case of pattern recognition. In *Proceedings of the 2002 International Conference on Machine Learning and Applications (ICMLA'02)*, pages 159–163. CSREA Press, 2002.

[10] Alexander Gammerman, Volodya Vovk, and Vladimir Vapnik. Learning by transduction. In *In Uncertainty in Artificial Intelligence*, pages 148–155. Morgan Kaufmann, 1998.

[11] Craig Saunders, Alex Gammerman, and Volodya Vovk. Transduction with confidence and credibility. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, volume 2, pages 722–726, Los Altos, CA, 1999. Morgan Kaufmann.

[12] Volodya Vovk, Alex Gammerman, and Craig Saunders. Machine-learning applications of algorithmic randomness. In *Proceedings of the Sixteenth International Conference on Machine Learning*, pages 444–453. Morgan Kaufmann, 1999.

[13] Vineeth N. Balasubramanian, Shen-Shyang Ho, and Vladimir Vovk, editors. *Conformal Prediction for Reliable Machine Learning*. Morgan Kaufmann, Boston, 2014. ISBN 978-0-12-398537-8. doi: http://dx.doi.org/10.1016/B978-0-12-398537-8.00014-6. URL http://www.sciencedirect.com/science/article/pii/B9780123985378000146.

[14] Kostas Proedrou, Ilia Nouretdinov, Volodya Vovk, and Alex Gammerman. Transductive confidence machines for pattern recognition. In *Proceedings of the 13th European Conference on Machine Learning (ECML'02)*, volume 2430 of *Lecture Notes in Computer Science*, pages 381–390. Springer, 2002.

[15] Harris Papadopoulos, Vladimir Vovk, and Alex Gammerman. Regression conformal prediction with nearest neighbours. *Journal of Artificial Intelligence Research*, 40:815–840, 2011.

[16] Devetyarov Dmitry and Nouretdinov Ilia. Prediction with confidence based on a random forest classifier. In Harris Papadopoulos, Andreas Andreou, and Max Bramer, editors, *Artificial Intelligence Applications and Innovations*, volume 339, pages 37–44. Springer Boston, 2010. URL http://dx.doi.org/10.1007/978-3-642-16239-8_8.

[17] Harris Papadopoulos, Kostas Proedrou, Volodya Vovk, and Alex Gammerman. Inductive confidence machines for regression. In *Proceedings*

of the 13th European Conference on Machine Learning (ECML'02), volume 2430 of *Lecture Notes in Computer Science*, pages 345–356. Springer, 2002.

[18] Harris Papadopoulos, Volodya Vovk, and Alex Gammerman. Conformal prediction with neural networks. In *Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'07)*, volume 2, pages 388–395. IEEE Computer Society, 2007.

[19] Tony Bellotti, Zhiyuan Luo, Alexander Gammerman, Frederick W. Van Delft, and Vaskar Saha. Qualified predictions for microarray and proteomics pattern diagnostics with confidence machines. *International Journal of Neural Systems*, 15(4):247–258, 2005.

[20] Alexander Gammerman, Ilia Nouretdinov, Brian Burford, Alexey Chervonenkis, Volodya Vovk, and Zhiyuan Luo. Clinical mass spectrometry proteomic diagnosis by conformal predictors. *Statistical applications in genetics and molecular biology*, 7(2), 2008. URL http://dx.doi.org/10.2202/1544-6115.1385.

[21] Tony Bellotti, Zhiyuan Luo, and Alexander Gammerman. Reliable classification of childhood acute leukaemia from gene expression data using confidence machines. In *Proceedings of IEEE International Conference on Granular Computing (GRC '06)*, pages 148–153, 2006.

[22] Harris Papadopoulos, Alex Gammerman, and Volodya Vovk. Confidence predictions for the diagnosis of acute abdominal pain. In *Artificial Intelligence Applications & Innovations III*, volume 296 of *IFIP International Federation for Information Processing*, pages 175–184. Springer, 2009.

[23] Harris Papadopoulos, Alex Gammerman, and Volodya Vovk. Reliable diagnosis of acute abdominal pain with conformal prediction. *International Journal of Engineering Intelligent Systems for Electrical Engineering and Communications*, 17(2-3):127–137, 2009. ISSN 1472-8915.

[24] Vineeth N. Balasubramanian, Shayok Chakraborty, and Sethuraman Panchanathan. Conformal predictions for information fusion. *Annals of Mathematics and Artificial Intelligence*, 74(1-2):45–65, 2015. ISSN 1012-2443. doi: 10.1007/s10472-013-9392-4. URL http://dx.doi.org/10.1007/s10472-013-9392-4.

[25] Meng Yang, Ilia Nouretdinov, Zhiyuan Luo, and Alex Gammerman. Feature selection by conformal predictor. In Lazaros Iliadis, Ilias Maglogiannis, and Harris Papadopoulos, editors, *Artificial Intelligence Applications and Innovations*, volume 364 of *IFIP Advances in Information and Communication Technology*, pages 439–448. Springer Berlin Heidelberg, 2011. ISBN 978-3-642-23959-5. doi: 10.1007/978-3-642-23960-1_51. URL http://dx.doi.org/10.1007/978-3-642-23960-1_51.

[26] Carlos Andres Pena-Reyes and Moshe Sipper. A fuzzy-genetic approach to breast cancer diagnosis. *Artificial Intelligence in Medicine*, 17(2):131–155, 1999.

[27] Hisao Ishibuchi, Kengo Nozaki, and Naokatsu Yamamoto. Selecting fuzzy rules by genetic algorithm for classification problems. In *Fuzzy Systems, 1993., Second IEEE International Conference on*, pages 1119–1124 vol.2, 1993. doi: 10.1109/FUZZY.1993.327358.

[28] Hisao Ishibuchi and Tomoharu Nakashima. Improving the performance of fuzzy classifier systems for pattern classification problems with continuous attributes. *Industrial Electronics, IEEE Transactions on*, 46(6): 1057–1068, Dec 1999. ISSN 0278-0046. doi: 10.1109/41.807986.

[29] Hisao Ishibuchi, Tomoharu Nakashima, and Tadahiko Murata. Comparison of the Michigan and Pittsburgh approaches to the design of fuzzy classification systems. *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)*, 80(12):10–19, 1997. ISSN 1520-6440. doi: 10.1002/(SICI)1520-6440(199712)80:12⟨10::AID-ECJC2⟩3.0.CO;2-W. URL http://dx.doi.org/10.1002/(SICI)1520-6440(199712)80:12<10::AID-ECJC2>3.0.CO;2-W.

[30] Antonis Lambrou. Genetic Algorithm Conformal Predictor. https://github.com/antonislambrou/GACP, 2016.

[31] Grigorios Tsoumakas and Ioannis Katakis. Multi-label classification: An overview. *Int J Data Warehousing and Mining*, 2007:1–13, 2007.

[32] Harris Papadopoulos. A cross-conformal predictor for multi-label classification. In Lazaros Iliadis, Ilias Maglogiannis, Harris Papadopoulos, Spyros Sioutas, and Christos Makris, editors, *Artificial Intelligence Applications and Innovations*, volume 437 of *IFIP Advances in Information and Communication Technology*, pages 241–250. Springer Berlin Heidelberg, 2014. ISBN 978-3-662-44721-5. doi: 10.1007/978-3-662-44722-2_26. URL http://dx.doi.org/10.1007/978-3-662-44722-2_26.

[33] Huazhen Wang, Xin Liu, Bing Lv, Fan Yang, and Yanzhu Hong. Reliable multi-label learning via conformal predictor and random forest for syndrome differentiation of chronic fatigue in traditional chinese medicine. *PLoS ONE*, 9:e99565, 06 2014.

[34] Huazhen Wang, Xin Liu, Ilia Nouretdinov, and Zhiyuan Luo. *Statistical Learning and Data Sciences: Third International Symposium, SLDS 2015, Egham, UK, April 20-23, 2015, Proceedings*, chapter A Comparison of Three Implementations of Multi-Label Conformal Prediction, pages 241–250. Springer International Publishing, Cham, 2015. ISBN 978-3-319-17091-6. doi: 10.1007/978-3-319-17091-6_19. URL http://dx.doi.org/10.1007/978-3-319-17091-6_19.

[35] Antonis Lambrou. Multi-label Conformal Predictor. https://github.com/antonislambrou/MLCP, 2016.

[36] Arthur Asuncion and David Newman. UCI machine learning repository, 2010. URL http://archive.ics.uci.edu/ml.

[37] John F. Timms, Rainer Cramer, Stephane Camuzeaux, Ali Tiss, Celia Smith, Brian Burford, Ilia Nouretdinov, Dmitry Devetyarov, Aleksandra Gentry-Maharaj, Jeremy Ford, Zhiyuan Luo, Alex Gammerman, Usha Menon, and Ian Jacobs. Peptides generated ex vivo from serum proteins by tumor-specific exopeptidases are not useful biomarkers in ovarian cancer. *Clin Chem*, 56(2):262–271, 2010. doi: 10.1373/clinchem.2009.133363. URL http://www.clinchem.org/cgi/content/abstract/56/2/262.

[38] Alicja Wieczorkowska, Piotr Synak, and ZbigniewW. Raś. Multi-label classification of emotions in music. In MieczysławA. Kłopotek, SławomirT.

Wierzchoń, and Krzysztof Trojanowski, editors, *Intelligent Information Processing and Web Mining*, volume 35 of *Advances in Soft Computing*, pages 307–315. Springer Berlin Heidelberg, 2006. ISBN 978-3-540-33520-7. doi: 10.1007/3-540-33521-8_30. URL http://dx.doi.org/10.1007/3-540-33521-8_30.

[39] André Elisseeff and Jason Weston. A kernel method for multi-labelled classification. In *In Advances in Neural Information Processing Systems 14*, pages 681–687. MIT Press, 2001.

[40] Rudy Setiono. Extracting rules from pruned neural networks for breast cancer diagnosis. In *Artificial Intelligence in Medicine*, pages 37–51, 1996.

[41] Ismail Taha and Joydeep Ghosh. Evaluation ordering of rules extracted from feed forward networks. In *International Conference on Neural Networks (ICNN'97)*, pages 221–226, 1997.

[42] Mark A. Hall. Correlation-based feature selection for machine learning. Technical report, 1998.

[43] Alex Gammerman, Volodya Vovk, Brian Burford, Ilia Nouretdinov, Zhiyuan Luo, Alexey Chervonenkis, Mike Waterfield, Rainer Cramer, Paul Tempst, Josep Villanueva, Musarat Kabir, Stephane Camuzeaux, John Timms, Usha Menon, and Ian Jacobs. Serum proteomic abnormality predating screen detection of ovarian cancer. *The Computer Journal, doi:10.1093/comjnl/bxn021*, 2008.

[44] Emanuel Petricoin, Ali Ardekani, Ben Hitt, Peter Levine, Vincent Fusaro, Seth Steinberg, Gordon Mills, Charles Simone, David Fishman, Elise

Kohn, and Lance Liotta. Use of proteomic patterns in serum to identify ovarian cancer. *The Lancet*, 359(9306):572–577, 2002.

[45] Vladimir Vovk, Glenn Shafer, and Ilia Nouretdinov. Self-calibrating probability forecasting. In Sebastian Thrun, Lawrence K. Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, pages 1133–1140, Cambridge, MA, 2004. MIT Press.

[46] Mikhail Dashevskiy and Zhiyuan Luo. Predictions with confidence in applications. In Petra Perner, editor, *Machine Learning and Data Mining in Pattern Recognition*, volume 5632 of *LNCS*, pages 775–786. Springer, 2009.

[47] Mikhail Dashevskiy and Zhiyuan Luo. Reliable probabilistic classification and its application to internet traffic. In *Advanced Intelligent Computing Theories and Applications. With Aspects of Theoretical and Methodological Issues*, volume 5226 of *LNCS*, pages 380–388. Springer, 2008.

[48] Harris Papadopoulos. Reliable probabilistic prediction for medical decision support. In *Proceedings of the 7th IFIP International Conference on Artificial Intelligence Appications and Innovations (AIAI 2011)*, volume 364 of *IFIP AICT*, pages 265–274. Springer, 2011.

[49] Harris Papadopoulos. Reliable probabilistic classification with neural networks. *Neurocomputing*, 107:59 – 68, 2013. ISSN 0925-2312. doi: 10.1016/j.neucom.2012.07.034. URL http://www.sciencedirect.com/science/article/pii/S0925231212007801.

[50] Antonis Lambrou, Harris Papadopoulos, Ilia Nouretdinov, and Alexander Gammerman. Reliable probability estimates based on Support Vector Machines for large multiclass datasets. In Lazaros Iliadis, Ilias Maglogiannis, Harris Papadopoulos, Kostas Karatzas, and Spyros Sioutas, editors, *Artificial Intelligence Applications and Innovations*, volume 382 of *IFIP Advances in Information and Communication Technology*, pages 182–191. Springer Berlin Heidelberg, 2012. ISBN 978-3-642-33411-5. doi: 10.1007/978-3-642-33412-2_19. URL http://dx.doi.org/10.1007/978-3-642-33412-2_19.

[51] Chenzhe Zhou, Ilia Nouretdinov, Zhiyuan Luo, Dmitry Adamskiy, Luke Randell, Nick Coldham, and Alex Gammerman. A comparison of Venn Machine with Platt's method in probabilistic outputs. In *Proceedings of the 7th IFIP International Conference on Artificial Intelligence Appications and Innovations (AIAI 2011)*, volume 364 of *IFIP AICT*, pages 483–490. Springer, 2011.

[52] John C. Platt. Probabilistic outputs for Support Vector Machines and comparisons to regularized likelihood methods. In *Advances in large margin classifiers*, pages 61–74. MIT Press, 1999.

[53] Joseph Drish. Obtaining calibrated probability estimates from Support Vector Machines, 1998.

[54] Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the 8th ACM International Conference on Knowledge Discovery and Data Mining*, pages 694–699, 2002.

[55] John C. Platt. Fast training of Support Vector Machines using sequential minimal optimization. In Bernhard Schölkopf, Christopher J. C. Burges, and Alexander J. Smola, editors, *Advances in kernel methods*, pages 185–208. MIT Press, Cambridge, MA, USA, 1999. ISBN 0-262-19416-3. URL http://dl.acm.org/citation.cfm?id=299094.299105.

[56] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18, November 2009. ISSN 1931-0145. doi: 10.1145/1656274.1656278. URL http://doi.acm.org/10.1145/1656274.1656278.

[57] Antonis Lambrou. Inductive Venn Prediction. https://github.com/antonislambrou/IVP, 2016.

[58] Marko Bohanec and Vladislav Rajkovič. V.: Knowledge acquisition and explanation for multi-attribute decision making. In *8th International Workshop "Expert Systems and Their Applications"*, 1988.

[59] Paulo Cortez, António Cerdeira, Fernando Almeida, Telmo Matos, and José Reis. Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47(4):547–553, November 2009. ISSN 0167-9236. doi: 10.1016/j.dss.2009.05.016. URL http://dx.doi.org/10.1016/j.dss.2009.05.016.

[60] Byron P. Roe, Hai-Jun Yang, Ji Zhu, Yong Liu, Ion Stancu, and Gordon McGregor. Boosted decision trees as an alternative to Artificial Neural Networks for particle identification. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and*

Associated Equipment, 543(2–3):577 – 584, 2005. ISSN 0168-9002. doi: http://dx.doi.org/10.1016/j.nima.2004.12.018.

[61] John Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993. ISBN 1-55860-238-0.

[62] Gianni Belcaro, Andrew N. Nicolaides, Giuseppe Laurora, Maria Rosaria Cesarone, Mariateresa De Sanctis, Lucrezia Incandela, and Antonio Barsotti. Ultrasound morphology classification of the arterial wall and cardiovascular events in a 6-year follow-up study. *Arterioscler Thromb Vasc Biol*, 16(7):851–856, 1996. URL http://atvb.ahajournals.org/cgi/content/abstract/16/7/851.

[63] Andrew N. Nicolaides, Edward G. Shifrin, Andrew Bradbury, Surinder Dhanjil, Maura Griffin, Gianni Belcaro, and Michael Williams. Angiographic and duplex grading of internal carotid stenosis: Can we overcome the confusion? *Journal of Endovascular Surgery*, 3(2):158–165, 1996.

[64] George Geroulakos, Janine Domjan, Andrew Nicolaides, J. Stevens, Nicos Labropoulos, Ganesh Ramaswami, and Gianni Belcaro. Ultrasonic carotid artery plaque structure and the risk of cerebral infraction on computer tomography. *Journal of Vascular Surgery*, 20(2):263–266, 1994.

[65] Efthyvoulos Kyriacou, Constantinos Pattichis, Marios Pattichis, Christos Loizou, Christodoulos Christodoulou, Stavros Kakkos, and Andrew Nicolaides. A review of noninvasive ultrasound image processing methods in the analysis of carotid plaque morphology for the assessment of stroke

risk. *Information Technology in Biomedicine, IEEE Transactions on*, 14 (4):1027 –1038, 2010. ISSN 1089-7771.

[66] Stavroula Gr. Mougiakakou, Spyretta Golemati, Ioannis Gousias, Andrew N. Nicolaides, and Konstantina S. Nikita. Computer-aided diagnosis of carotid atherosclerosis based on ultrasound image statistics, laws' texture and neural networks. *Ultrasound in Medicine & Biology*, 33(1):26 – 36, 2007. ISSN 0301-5629.

[67] Mark Langsfeld, Anthony C. Gray-Weale, and Robert J. Lusby. The role of plaque morphology and diameter reduction in the development of new symptoms in asymptomatic carotid arteries. *Journal of Vascular Surgery*, 9(4):548–557, 1989.

[68] World Health Organisation. *Prevention and management of Osteoporosis*. Geneva, 2003.

[69] Ian T. Jolliffe. *Principal Component Analysis*. Springer, second edition, October 2002. ISBN 0387954422. URL http://www.worldcat.org/isbn/0387954422.

[70] Chandra Dhir, Nadeem Iqbal, and Lee Soo-Young. Efficient feature selection based on information gain criterion for face recognition. In *Information Acquisition, 2007. ICIA '07. International Conference on*, pages 523–527, 2007. doi: 10.1109/ICIA.2007.4295788.

[71] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. *Mach. Learn.*, 46(1-3):389–422, March 2002. ISSN 0885-6125.

doi: 10.1023/A:1012487302797. URL http://dx.doi.org/10.1023/A:1012487302797.

[72] Alexander Leung and David Sigalet. Acute abdominal pain in children. *American family physician*, 67(11):2321–2328, 2003.

[73] Alexander Gammerman and AR Thatcher. Bayesian diagnostic probabilities without assuming independence of symptoms. *American family physician*, 67(11):2321–2328, 2003.

[74] Dimitrios Mantzaris, George Anastassopoulos, Adam Adamopoulos, and Stefanos Gardikis. A non-symbolic implementation of abdominal pain estimation in childhood. *Information Sciences*, 178(20):3860 – 3866, 2008. ISSN 0020-0255. doi: http://dx.doi.org/10.1016/j.ins.2008.06.015. URL http://www.sciencedirect.com/science/article/pii/S0020025508002405. Special Issue on Industrial Applications of Neural Networks 10th Engineering Applications of Neural Networks 2007.

[75] George C Anastassopoulos and Lazaros S Iliadis. ANN for prognosis of abdominal pain in childhood: use of fuzzy modelling for convergence estimation. In *18th European Conference on Artificial Intelligence (ECAI 2008),-Proc. of the 1st International Workshop on Combinations of Intelligent Methods and Applications (CIMA-08)*, pages 1–5, 2008.