ROYAL HOLLOWAY AND BEDFORD NEW COLLEGE,
UNIVERSITY OF LONDON

THESIS SUBMITTED IN FULFILMENT OF THE REQUIREMENTS FOR THE
DEGREE OF DOCTOR OF PHILOSOPHY
IN MACHINE LEARNING

Conformal and Venn Predictors for large, imbalanced and sparse chemoinformatics data

Paolo TOCCACELI

supervised by
Prof. Alexander GAMMERMAN

Declaration of Authorship

I, Paolo TOCCACELI, 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:

Abstract

Conformal Predictors (CP) (Vovk, Gammerman, and Shafer, 2005) are a novel approach to dealing with the uncertainty of predictions. Whereas conventionally one obtains predictions first and then estimates their error, CPs allow the error rate to be chosen. The predictions output by CP are provably guaranteed to exhibit the chosen error rate (barring statistical fluctuation). This distinctive property of CP is referred to as validity and is achieved under minimal assumptions. CPs are of very wide applicability. In fact, rather than a self-contained method, CPs can be viewed as framework, in which virtually any ML method can be plugged in. In this work, we first set the context by defining formally the notion of Conformal Prediction. We then introduce its variants, identify the desirable properties of CPs, and survey the state-of-the-art. Next, we consider Venn Predictors (VP) (Vovk, Gammerman, and Shafer, 2005), which are calibrated probabilistic predictors, i.e. predictors that output probability estimates that are guaranteed to correspond to actual relative frequencies (again barring statistical fluctuation). The advantages of CPs and Venn-ABERS Predictors (VAP) (Vovk, Petej, and Fedorova, 2015) — a form of VP for binary classification — are illustrated by considering their application to chemoinformatics problems. The difficulties posed by large-scale, highly imbalanced, sparse datasets common in this domain were met by the careful implementation of CP and VAP and by scaling the computations over distributed processing architectures. The results confirm that the methods produce predictions with validity and calibration properties, despite the challenges posed by the domain. A further section of the thesis explores how the desirable properties of CPs can be improved by their combination. The problem of CP combination can be viewed as a special form of the problem of p-value combination which has been studied extensively in the context of Classical Statistical Hypothesis Testing. A selection of combination methods from the literature is discussed and a new method based on the Neyman-Pearson Lemma (NPL) is introduced. It is conjectured that the property of Uniform Maximum Power of the NPL translates into maximal efficiency of the resulting Conformal Predictor. The various combination methods are compared and contrasted on a synthetic dataset and, more importantly, on real-world datasets and the results show that some combination methods are indeed synergistic. The implementation of the methods and its challenges are also discussed. In particular, the NPL-based method hinges upon the estimation of a ratio of probability densities, for which only Prof. Vapnik's V-matrix method appeared to provide the required accuracy. The last part of the thesis describes Conformal Predictive Distributions (CPDs), a novel non-parametric framework for probabilistic prediction in a regression setting (Vovk et al., 2019). CPDs, a variant of CP for regression, can output Predictive Distributions with guaranteed coverage under the assumptions of i.i.d. and unrestricted randomness. We present

a concrete instance of the CPD framework, known as Kernel Ridge Regression Predictive Machine (KRRPM) (Vovk et al., 2018) and we discuss its application in the domain of chemoinformatics on real-world data from a major pharmaceutical company.

Acknowledgements

This thesis would not have been possible without the supervision and guidance of Prof. Alexander Gammerman. He first insisted that I embarked on this adventure, then steered me clear of various navigational hazards throughout and finally kept me focused on reaching the destination.

My gratitude extends also to Prof. Vladimir Vovk, Prof. Zhiyuan Luo, and Dr. Ilia Nouretdinov for sharing their insights and for their collaborative spirit.

The research presented in this thesis was partly carried out in the context of the ExCAPE project, which received funding from the European Union's Horizon 2020 Research and Innovation programme under Grant Agreement no. 671555. We are grateful for the help in conducting experiments to the Ministry of Education, Youth and Sports (Czech Republic) that supports the Large Infrastructures for Research, Experimental Development and Innovations project "IT4Innovations National Supercomputing Center – LM2015070".

We are grateful to AstraZeneca for their financial support for the collaborative project entitled: "Machine Learning for Chemical Synthesis" (R10911). In particular, I am indebted to Claus Bendtsen (Quantitative Biology) for his keen interest and his thoughtful suggestions and to Lars Carlsson and Ernst Helgee-Ahlberg (formerly of AstraZeneca) for providing data and for useful discussions. Finally, I would like to thank the examiners Dr. Yuri Kalnishkan and Prof. Henrik Boström for their meticulous review, for all their constructive suggestions, and for keeping me honest.

Contents

List of Figures

List of Tables

List of Abbreviations

ABERS	**A**yer **B**runk **E**wing **R**eid **S**ilverman
CDF	**C**umulative **D**istribution **F**unction
CP	**C**onformal **P**redictor
CPU	**C**entral **P**rocessing **U**nit
ECDF	**E**mpirical **C**umulative **D**istribution **F**unction
ExCAPE	**Ex**ascale **C**ompound **A**ctivity **P**rediction **E**ngines
GM	**G**eometric **M**ean
HPC	**H**igh **P**erformance **C**omputing
HTS	**H**igh **T**hroughput **S**creening
ICP	**I**nductive **C**onformal **P**redictor
I.i.d.	**I**ndependent and **i**dentically **d**istributed
IVAP	**I**nductive **V**enn-**A**BERS **P**redictor
kNN	**k** **N**earest **N**eighbors
KRRPM	**K**ernel **R**idge **R**egression **P**redictive **M**achine
MICP	**M**ondrian **I**nductive **C**onformal **P**redictor
ML	**M**achine **L**earning
NCM	**N**on **C**onformity **M**easure
NP	**N**eyman **P**earson
NPL	**N**eyman **P**earson **L**emma
PDF	**P**robability **D**ensity **F**unction
QSAR	**Q**uantitative **S**tructure-**A**ctivity **R**elationship
RAM	**R**andom **A**ccess **M**emory
RBF	**R**adial **B**asis **F**unction
RV	**R**andom **V**ariable
SHT	**S**tatistical **H**ypothesis **T**esting
SV	**S**upport **V**ector
SVC	**S**upport **V**ector **C**lassifier
SVM	**S**upport **V**ector **M**achine
TFLOPS	**T**era **FL**oating-point **O**perations **P**er **S**econd
VAP	**V**enn-**A**BERS **P**redictor
VP	**V**enn **P**redictor
XGB	e**X**treme **G**radient **B**oosting

Chapter 1

Introduction

1.1 Outline of the thesis

The research in this thesis was motivated by the fact that many established Machine Learning methods do not output predictions with good statistical properties. It is known for instance that neural networks output estimated probabilities that differ from relative frequencies (Guo et al., 2017). Conformal methods offer a principled way to generate predictions with good statistical properties under minimal assumptions. This distinctive feature of conformal methods attracted interest in the domain of chemoinformatics and resulted in our participation to an EU project and in a research collaboration with a major pharmaceutical company. Our work focused on the following three problems:

1. Conformal methods should cater for the peculiar characteristic of the data sets prevalent in chemoinformatics, namely size, imbalance and sparseness.
2. Conformal methods should be applicable at scale.
3. Conformal methods should take advantage of the benefits of ensembling.

While the application of Conformal Predictors to chemoinformatics was already not a novel idea at the outset — as it had already been proposed as early as in (Norinder et al., 2014) — the widening of the domain of applicability required new techniques (such as Mondrian CP). We believe that the application of calibrated probabilistic predictors (Venn-ABERS) and of Conformal Predictive Distributions to chemoinformatics is novel.
An approach to the scalability of CP was explored in (Capuccini et al., 2015) within the limits of the Spark framework (Zaharia et al., 2016). Our focus was instead on exploiting High Performance Computing systems (colloquially referred to as supercomputers) and on the search for efficiency.
As to the combination of CP, (Balasubramanian, Chakraborty, and Panchanathan, 2015) provided a first comprehensive attempt. The preservation of validity and the search for methods with a theoretical grounding seemed the most logical direction in which to conduct further research.

The thesis is articulated as follows. After framing the problem of prediction under uncertainty in Section 1.2, we introduce Conformal Prediction (CP) in Chapter 2, providing formal definitions as well as the intuition behind it. The application of CP to the problem of predicting the biological activity of chemical compounds is discussed in Chapter 3, where we show results on challenging real-world public-domain datasets, with high imbalance, high sparseness, and high dimensionality. Probabilistic prediction — in the form of Venn Predictors — is presented in Chapter 4, where we also present an example of its application to the compound activity prediction problem. Chapter 5 explores the topic of CP Combination, i.e. the idea of combining CPs in order to obtain one with better properties, examining existing ideas and proposing new methods. A method with potentially optimal efficiency is presented in Chapter 6 and its performance is evaluated on a synthetic dataset,

under different correlation levels. In Chapter 7, we show how a variant of CP can produce probabilistic predictions, referred to as Conformal Predictive Distributions, in a regression setting. The method's advantages and limitations are discussed using an application to real world data sets from a major pharmaceutical company.

1.2 Context

Uncertainty permeates the world. This bare fact sits uncomfortably with Mankind's instinctive urge to explain the world through a mechanistic interplay of cause and effect. Our minds are programmed to look for a reason behind all occurrences. All cultures seem to have created at some point elaborate superstitions to satisfy their need for an explanation. Some thinkers, however, realized very early the illusory nature of a fully deterministic account. As early as the first century BC, Lucretius found it necessary to introduce the notion of "clinamen", an unexplained "deviation" in the otherwise fully deterministic motion of atoms that Democritus and then Epicurus postulated. A discussion of the implications of determinism vs. randomness on free-will, ethics, etc., while fascinating, would be outside the scope of this work. Let's just conclude these initial considerations by observing how, in the present age, uncertainty is central to Physics. In Quantum Mechanics, the laws governing the wave function are deterministic, but the observations accompanied by the collapse of the wave function can only be predicted in probabilistic terms.

Returning closer to the main subject, it may be useful to distinguish two mechanisms by which the ideal goal of exact prediction is thwarted: aleatoric variability and epistemic uncertainty. The latter refers to the incomplete knowledge of the mechanism that generates the data, where as the former warns us that, even if we had perfect knowledge of the observation-generating process, its nature may be intrinsically stochastic.

If we accept that a degree of uncertainty is an inescapable fact of life, then it can be argued that the primary goal of prediction should to estimate probabilities as accurately as possible.

The term 'probability' is encountered on a daily basis, but its precise meaning remains elusive. The mathematician and philosopher Bertrand Russell noted in 1929 that "Probability is the most important concept in modern science, especially as nobody has the slightest notion what it means". To this day, there seem to be no consensus on a single definitive notion. The "Interpretations of Probability" entry in the authoritative "Stanford Encyclopedia of Philosophy" lists six interpretations, although one could argue that most of the debate is between Frequentists and Bayesians. While it would seem that the tales of fierce, no-holds-barred, ad-hominem confrontation between Bayesians and Frequentists are but a caricature of what are now civilized exchanges of opinions between open-minded scientists, fundamental differences persist. It is fair to say that in this work when the term "probability" is used, what the author has in mind is a limit of relative frequencies in

repeated experiments. It is well-understood that this view has limitations. In many cases, one deals with probabilities for which no actual experiment can be performed and one has to resort to a rather contrived thought-experiment to express a limit of a relative frequency. But the definition of such an experiment helps to clarify exactly what we are expressing. Let's consider for instance Bertrand's chord Paradox (Figure 1.1). A seemingly clear question in the geometrical setting can legitimately admit multiple different answers. If this happens in the Platonic realm of rigorously defined pure ideas such as geometry, we can imagine the potential for confusion and misunderstanding in fields such social sciences, biology, etc.

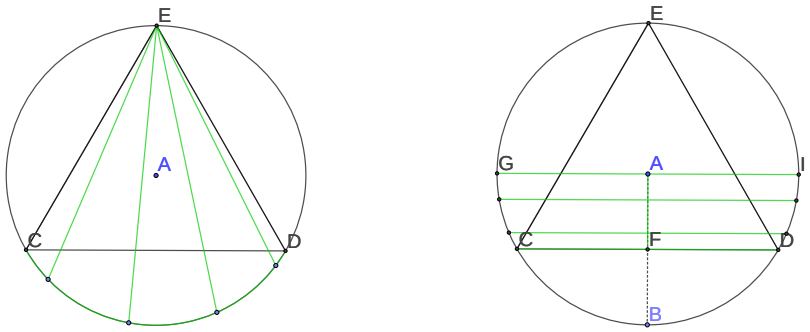


FIGURE 1.1: Bertrand's (chord) Paradox. What is the probability of a chord being longer than the side of an inscribed equilateral triangle? We illustrate here 2 of the 3 seemingly legitimate arguments Bertrand showed in 1889 in his "Calcul des probabilitès". In the diagram on the left, we consider the chords constructed by connecting point E with points on the circumference. The chords we seek are those corresponding to points on the arc CD in green. So the probability is $\frac{1}{3}$. In the diagram on the right, we consider instead the chords constructed by taking the perpendicular to a radius. The chords longer than the side are those corresponding to the points in one half of the radius (segment AF). So the probability is $\frac{1}{2}$. (Bertrand also showed a third construction leading to yet another value of probability).

Moreover, it has long been suspected that human brains are not very well-equipped to process probabilities, but it was only recently that the cognitive illusions and the semantic biases that affect our probabilistic reasoning were clearly exposed (Kahneman, 2011). This inability is insidious and not confined only to the general population. In the outcry that followed Marilyn Vos Savant's discussion of the Monty Hall's dilemma even academics and professional statisticians embarrassed themselves by sanctimoniously rebutting the correct answer (Savant, 1997; Rosenhouse, 2009).

That is not to say that the Bayesian notion is without merit. A subjective probability can be manipulated in a way that is consistent with Kolmogorov's axioms just as well the frequentist notion. Also, and perhaps more importantly, the challenge brought by empiricists to the idea of an "unobservable" belief to which a numerical

value is attached by way of an unknown process that potentially differs from one individual to the other can be addressed. Indeed the ardent subjectivist Bruno De Finetti proposed in the 1930s to obtain an *objective personal belief* by using a betting framework. Specifically, De Finetti would define the probability of an event A is the price p at which an individual is willing to trade "tickets" that pay out 1 if event A occurs. Although it does not solve all the problems, this notion is a step in the right direction. In recent years, (Shafer and Vovk, 2019) used a similar device in an attempt to unify frequentist and subjectivist approaches at a level deeper than the level of axioms. By also bringing together ideas from Cournot, Ville, Von Mises, and others, they have reframed the notion of probability into a game-theoretic context. In a way that recalls De Finetti's set-up, the expected values in a probability model are viewed as prices of future payoffs in a game. The key point (referred to as the fundamental interpretative hypothesis) is that if a winning gambling strategy exists, the probability model has to be rejected.

1.3 Hedging predictions

The usefulness of the notion of probability could be seen in the role that it plays in processes that use the available information to derive decisions that are optimal in some sense toward the achievement of a certain goal. Assigning a probability to the events allows a mathematical treatment of the problem of deciding a course of action. Machine Learning methods can complement bare predictions (i.e. the value of the label, be it discrete or continuous, in supervised learning setting) with an estimate of probability. Bayesian methods do this directly, but the actual reliability of their estimate depends on how well the assumptions match reality. In general, cross validation can produce estimates of the error that are robust as long as the validation is truly representative of the test set. In the conventional approach, in summary, the error rate is estimated and fixed after the model is trained.

Conformal Predictors differ from the conventional approach in that they allow to choose the error rate of the predictions, as opposed to just producing an estimate of it. The predictions are guaranteed to exhibit the chosen error rate (barring statistical fluctuations), in a sense that will be made precise later. A key difference between the conventional approach and CP is that in the latter the prediction is not a single value, but a set. In the context of CP, each prediction can take a multiplicity of values, from none (the empty set is a possible prediction) to the entire domain of the labels (in which case, the prediction is hardly of any use).

The conventional approach and CP can be seen as two alternative ways of hedging a prediction. The conventional approach produce single-valued predictions and complements them with the estimated error rate, whereas in Conformal Prediction the error rate is chosen (as opposed to estimated) and the predictions are made narrower or wider so that the targeted error rate is achieved. The smaller the target error rate, the wider the CP predictions are going to be and vice versa.

1.4 Application: Chemoinformatics

In this thesis we will discuss the application of conformal and probabilistic prediction to problems of chemoinformatics, which can be defined as "the discipline organizing and coordinating the increasing application of computers in chemistry" (Polanski, 2009). More specifically we will focus on the prediction of biological, physicochemical, or pharmacokinetic properties of compounds. It is perhaps helpful to the general reader if we devote a few words to put the problem into its wider context, with the aim to provide just enough background for the reader to make sense of the objectives of the present research.

1.4.1 Drug discovery and development

The cost of developing a new pharmaceutical drug has increased steadily over the years. The statistics are a subject of controversy, but if one takes the overall Research and Development yearly expenditure for pharmaceutical sector and divides it by the number of new drugs approved by the Federal Drug Administration every year, the result is of the order of billions of US dollars (Wouters, McKee, and Luyten, 2020). Despite the constant flux of technological advances that have introduced techniques that were unthinkable a few years earlier, the pharmaceutical industry is faced with increased costs in bringing to fruition its research efforts.

Simplifying massively, the development of a new drug can be described as a pipeline, divided into the sequential stages of drug discovery, drug development, and clinical trials (further divided into Phase I, II, and III). Researchers first select a biological target of interest (often an enzyme — a protein that acts as a catalyst for a chemical reaction — that is involved in a disease that we seek to treat) and then, starting from libraries of millions of compounds, identify those with promising activity towards that target. To be a drug, a compound has to have a number of desirable properties in terms of absorption, distribution, metabolism, excretion, and last but not least toxicity[1]. During drug development, medicinal chemists modify the structure of the hits — the active compounds identified during drug discovery — through a process of trial-and-error guided by their expert judgement in the pursuit of those properties. When this objective is reached, the few candidate drugs that emerge from this process go through trials to establish dosage, safety, and end efficacy.

In this thesis the universe of discourse (as far as applications are concerned) will be drug discovery and development. In those two stages, the costs are driven primarily by the large number of lab tests that are performed.

[1] Absorption, distribution, metabolism, excretion properties are pharmacokinetic properties and often referred to as ADME properties. Informally, pharmacokinetics refers to "what the body does to the drug". Physicochemical properties include solubility and hydrophilicity. By biological properties, we'll tend to refer to activity (inhibitory or enhancing) towards a target and to toxicity.

1.4.2 The role of Machine Learning

A variety of approaches have been pursued to reduce the number of lab tests during drug discovery and development. One broad class is predicated on the application of physics and geometry to the calculation, for instance, of the affinity with which a ligand binds to the active site of an enzyme. While this approach is undeniably sound for a scientific perspective, the computational difficulties that it entails are enormous. Another broad class takes instead a statistical approach to the prediction of properties. The Quantitative Structure-Activity Relationship (QSAR) techniques are firmly in this second class. They rest on the idea that if a suitable description of the relevant features of the molecular structure is found, then it is possible to create a statistical model of wide applicability to predict properties. QSAR techniques have been proposed since the 1960s (Dearden, 2017), but came into their own in recent years with the widespread availability of vast amounts of computing power and the advent of Machine Learning.

Applying Machine Learning techniques presents some challenges. The first is to assemble a training set that (a) is sufficiently representative of the problem domain and (b) describes the objects (compounds) by means of features that are relevant for the specific prediction task. In the case of chemoinformatics, (a) means performing assays for the property of interest to a large set of compounds. The chemical space, i.e. the set of all possible molecules, is immensely vast, even if we limit ourselves to the so-called small molecules[2]. An estimate often quoted for its size is 10^{60} (Bohacek, McMartin, and Guida, 1996, page 43), but in recent years this has been revised down to 10^{33} or even 10^{23} (Polishchuk, Madzhidov, and Varnek, 2013, Table 1). Be as it may, it is clear that any training set produced with the current technologies can only cover a negligible fraction of such a space. The challenge posed by (b) is still the subject of an active area of research. In this study, the features describing a compound are extracted from the topology of the molecule, viewed as a labelled graph. This will be explained with some examples in Section 3.1.

One legitimate question is whether statistical modelling of biological properties is at all possible or useful. One could argue that whenever we employ statistical modelling, we rely heavily on some assumption of regularity on some neighborhood in the problem space. There is unfortunately a phenomenon called the "activity cliff" which consists in heavy non-linearity in the properties (Maggiora, 2006). Whereas small changes in the structure of a molecule generally result in correspondingly small changes in the in property under study, in a few cases a further small structural change can turn out to have surprisingly disproportionate effects. This seems to militate against the assumption of regularity which inevitably underpins any statistical approach.

[2]While the notion of small molecule is used widely in pharmacology, there is no universally accepted definition. Just to provide some element of reference, one definition limits the molecular weight to less than 900 Daltons.

Despite these potential issues and the controversy that they bring about, Machine Learning techniques appear to have found a role as useful tools for discovering and developing new drugs (Schneider et al., 2020).

Chapter 2

Conformal Predictors

2.1 Notation

In the sequel, we will operate, unless otherwise stated, within the setting of *supervised learning*, using the follow notation and conventions.
A training set $\mathbf{Z} = \{z_1, z_2, \ldots, z_\ell\}$ contains *examples* $z_i = (x_i, y_i)$ consisting of an *object* $x_i \in \mathbf{X}$ and a *label* $y_i \in \mathbf{Y}$. We use the $\ell + 1$ index to denote the test object $x_{\ell+1}$.

We distinguish between two modes of operation, namely *batch* vs. *online*. In the batch mode, the training set is provided once, in its entirety. The model is trained on it and then used to make an indefinite number of predictions. The actual labels of the test objects, possibly revealed after prediction, are not used to create new training examples. The order of the training examples and the test examples is irrelevant.

In the online mode, instead, the training examples are presented in a sequence. At step i, the ML method will have a training set $\{(x_1, y_1), \ldots, (x_{i-1}, y_{i-1})\}$ and will be presented with a test object x_i. It will output a prediction $\hat{y}_i$ and the actual label y_i will be revealed. The example $z_i = (x_i, y_i)$ will be added to the training set and the execution will move on to the next step $i + 1$.

2.2 Introduction

The most straightforward form of prediction consists in a providing a single value, i.e. what we will refer to as a *bare prediction*. While this may be adequate in some simple applications, it is easy to convince ourselves that in fact it omits valuable information. The bare prediction does not convey the strength of the statistical evidence that supports its specific value; the strength of the evidence could be simply marginally lower for other values, but one would not be aware to this fact. Additional information should complement the bare prediction.

Conformal Predictors convey this information in a novel way. They offer a principled, efficient, and flexible way to obtain predictions that guarantee a given error rate, under minimal assumptions. The predictions are sets, discrete in case of classification and continuous in the case of regression.

As noted in (Hedging predictions in Machine Learning, 2007), "the problem of hedged prediction is intimately connected with the problem of testing randomness". The theoretical foundations of CP can be traced to the universal test of randomness by Per Martin-Löf.

The predictions produced by the conformal algorithm are (a) invariant with respect to the old examples, (b) correct with the advertised probability, and (c) nested. They are optimal among all region predictors with these properties.

2.3 I.i.d. Assumption and Exchangeability

The properties of CPs are guaranteed by a theoretical apparatus that rests on minimal assumptions. In this section, we will define such assumptions

I.i.d. assumption

Let's consider n random variables $X_1, \ldots, X_n$ taking values in a space I and let F_{X_i} be the distribution of X_i. The $X_1, \ldots, X_n$ variables are independent and identically distributed (i.i.d.) if and only if

$$F_{X_1}(x) = F_{X_k}(x) \qquad \forall k \in \{1, \ldots, n\} \text{ and } \forall x \in I \tag{2.1}$$

$$F_{X_1,\ldots,X_n}(x_1, \ldots, x_n) = F_{X_1}(x_1) \cdot \cdots \cdot F_{X_n}(x_n) \qquad \forall x_1, \ldots, x_n \in I \tag{2.2}$$

This can be paraphrased by saying that the marginal probability of each variable is the same as that of any other of the n variables and that the joint distribution is simply the product of the marginal distributions for each variable. The i.i.d. assumption also means that the probability distribution of any Random Variable (RV) does not depend on the values of the other RVs. Informally, one could say that the knowledge of the values taken by the other RVs is of no help in predicting the distribution of any RV.

The i.i.d. assumption is pervasive in Machine Learning. The setting of Statistical Learning Theory stipulates "random vectors $x \in R^n$ drawn independently from a fixed but unknown probability distribution $F(x)$" (Vapnik, 1995, Page 17). Indeed, the vast majority of ML algorithms relies, explicitly or implicitly, on the assumption that the test examples be drawn from the same distribution as the training examples. There are also, however, approaches that do away even with this seemingly minimal requirement[1].

One such example is the framework of Prediction with Expert Advice (Cesa-Bianchi and Lugosi, 2006), in which the Forecaster is presented by Nature with a sequence of examples (with concealed label). At each step, the Forecaster predicts the label, which is then revealed by Nature and a Loss is computed as a function of predicted and actual labels. The loss is cumulated over the sequence. The key point here is that no assumption is made on their distribution of the examples that Nature presents to the Forecaster. In fact, the choice of the examples that are presented can even be adversarial. In such a hostile environment, one might as well wonder if prediction is at all possible in any meaningful way. Prediction with Expert Advice posits the existence of a number of Experts (or reference forecasters) that make their predictions available as Advice to the Forecaster before Nature reveals the true label. The problem that can be studied is that of determining rules that minimizes Regret, i.e. the difference between the loss deriving from the choice of the Forecaster and that of the best Expert (see (Herbster and Warmuth, 1998; Vovk, 2001; Kalnishkan et al., 2015; Korotin, V'yugin, and Burnaev, 2019)

Finally, let's conclude these remarks with an epistemological observation. Recourse to the i.i.d assumption (or perhaps just the weaker assumption discussed in the next paragraph) appears inevitable whenever the methods are fundamentally

[1] In addition to the discussion that follows, another example of such approaches is the study of stochastic processes, in which successive examples are not necessarily i.i.d.

applying the inductive method, as is the case of statistical learning methods, which are prevalent in the current (year 2020) incarnation of Artificial Intelligence. The dominant AI paradigm until the 1990, which is now often referred to Good Old Fashioned AI (GOFAI) (Haugeland, 1989, p.112) was chiefly concerned with the manipulation of symbols and as such can be viewed as an application of the deductive method. In Symbolic AI, inference did not seek justification in the regularity assumption that underpins the inductive method, but was a mechanical application of principles of logic and by its nature independent of the statistics of the data (we prove that the sum of the internal angle of triangle is 180 degrees as opposed to looking at many triangles and concluding that the sum appears to be 180 degrees). Just as the limits of GOFAI became apparent and the interest fizzled out, especially as the statistical approach gained traction, there seems to be a growing awareness of the limitations of the current "inductivist" approach. Despite all the successes, the state-of-the-art still relies largely on the training set to contain examples sufficiently correlated to any possible test object to get a useful prediction. Hence, larger and larger training sets are needed to cover all possible test objects that the model will be asked to predict on. Even if the technological advances and great engineering feats allow further scaling in this direction, it's hard not to wonder if this "blind" reliance on correlation is the right path forward. While it has been taught for decades in any statistics course that correlation does not imply dependence, it is only relatively recently that the extent of the implications have been clearly articulated. A new field of research has sprung to focus on the idea of causation (Pearl, 2009). Its main advocate, Judea Pearl, popularized the notion (Pearl and Mackenzie, 2018) that observational data alone cannot provide in itself a way to distinguish the causal direction of an observed correlation. Data must be complemented by information on the mechanism of its production to be used for inference. While it is outside of the scope of this thesis to discuss any further the topic of causality and the controversy that surrounds it, it is important to note that a causal model holds the promise of lessening the dependence on the i.i.d. assumption that the current statistical "inductivist" approach suffer from.

Exchangeability

The variables $z_1, \ldots, z_N$ are exchangeable if for every permutation τ of the integers $1, \ldots, N$,

$$\Pr(z_1, \ldots, z_n) = \Pr\left(z_{\tau(1)}, \ldots, z_{\tau(n)}\right)$$

that is, the variables $w_1, \ldots, w_N$, where $w_i = z_{\tau(i)}$, have the same joint probability distribution as $z_1, \ldots, z_N$.

Exchangeability is effectively a property of the probability measure over the N random variables. To put it informally, the value of the probability measure does not depend on the order of its arguments.

It is straightforward to prove that i.i.d. variables are also exchangeable (the joint distribution is the product of N identical univariate distributions, hence the order is irrelevant).

The converse, however, is not true. An example of a sequence of RVs that is exchangeable but not independent is illustrated in the next subsection.

2.3.1 Polya's urn

As an example of a sequence of RV that is not i.i.d. but is exchangeable, consider the following case (adapted from (Lauritzen, 2007)), called Polya's urn. It's a variation on the setting of sampling with replacement.

Consider an urn with b black balls and w white balls. Draw a ball at random and note its colour (this is the realization of RV X_i). Replace the ball together with a balls of the same colour. Repeat the procedure n times.

Let's consider the joint probability $\Pr(X_1, \ldots, X_n)$ with an example first.
$\Pr(X_1 = white, X_2 = black, X_3 = black, X_4 = white)$ is the product of:

- $\frac{w}{w+b}$ for draw 1
- $\frac{b}{w+b+a}$ for draw 2
- $\frac{b+a}{w+b+2a}$ for draw 3
- $\frac{w+a}{w+b+3a}$ for draw 4

We can see that the denominator in $\Pr(X_1, \ldots, X_n)$ does not depend on the outcomes at the various draws. It is $w + b + na$, where n is the number of draws, whereas the numerator is of the form $\left(1 + \prod_{i=0}^{r-1}(w + ia)\right)\left(1 + \prod_{j=0}^{s-1}(b + ja)\right)$, where r and s are the counts of occurrences of white ball draws and black ball draws, respectively. The joint probability is thus a function only of r and s (and their sum n) and the order of the outcomes of the draws is irrelevant. The latter is indeed the defining property of exchangeable sequences.

We can also show that the RVs in the sequence are not independent.

$$\begin{aligned}
\Pr(X_1 = white, X_2 = black) &= \Pr(X_1 = white) \cdot \Pr\left(X_2 = black \mid X_1 = white\right) \\
&= \frac{w}{w+b}\frac{b}{w+b+a} \qquad (2.3)\\
\Pr(X_1 = white) &= \frac{w}{w+b} \qquad (2.4)\\
\Pr(X_2 = black) &= \Pr(X_1 = white) \cdot \Pr\left(X_2 = black \mid X_1 = white\right) \\
&\quad + \Pr(X_1 = black) \cdot \Pr\left(X_2 = black \mid X_1 = black\right) \\
&= \frac{w}{w+b}\frac{b}{w+b+a} + \frac{b}{w+b}\frac{b+a}{w+b+a} \qquad (2.5)\\
&= \frac{b}{w+b}
\end{aligned}$$

and we can see that $\Pr(X_1 = white, X_2 = black) \neq \Pr(X_1 = white) \cdot \Pr(X_2 = black)$

2.4 Conformal Predictors

At the root of Conformal Prediction, we have the notion of Non Conformity Measure (NCM). The NCM expresses how dissimilar (or non-conform) an example appears to be with respect to a collection of examples. The NCM is the elementary tool we will use to assess randomness. Note that the NCM is not defined by the CP framework. The NCM is left to the user to define. It is meant to be defined so that its values are larger, the more out-of-place the example appears. Figure 2.1 illustrates the intuitive notion of Non Conformity[2].

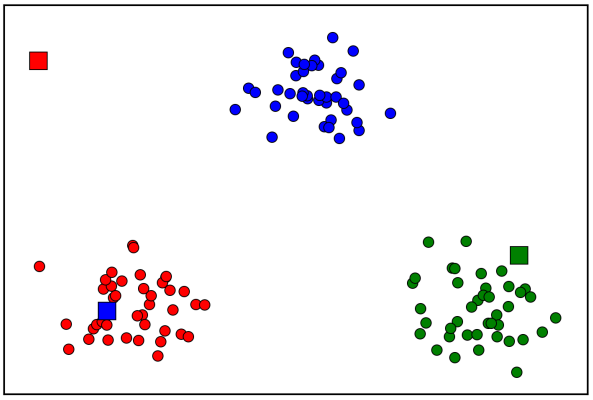


FIGURE 2.1: An illustration of Non Conformity. The round markers represent the collection of examples, where the colours red, green, blue correspond to the labels. The square markers represents test examples. The test example at the right (green square) does not look out of place, so a good Non Conformity measure would assign a (relatively) low value to it, whereas the blue square in the middle of the red cluster at bottom left would have a high NCM. The case of the red marker at the top left is not as definite as those of the previous examples. The NCM would take an intermediate value.

The randomness of an example is then assessed by using the NCM in relative terms, rather than using its absolute value. To judge the randomness of an example, we determine the proportion of the examples in the collection that have a larger NCM than the example in hand. Low values of this proportion mean that it is rare to find examples that look more out-of-place, whereas high values signify that the majority of the examples would look more out-of-place.

The region prediction for a test object (of which we do not know the actual label) is built by the following procedure. For each possible value of the label, we construct a hypothetical example, made up of the test object and that hypothetical label (thereby forming a *hypothetical completion*), and we assess the randomness of these hypothetical completions. Only the labels for which the corresponding hypothetical completion exhibits a degree of randomness relative to the training set (measured as

[2]CPs can be formulated equally well in terms of a Conformity Measure. The reason for using a possibly less intuitive Non Conformity measure are explained in Section 2.4.9

proportion of the training set with larger NCM) higher than the chosen significance level are included in the region prediction.

A key result proved in (Vovk, Gammerman, and Shafer, 2005, Theorem 8.1) establishes that, provided that data is exchangeable, a region predictor computed using this rule has an error rate that reflects the significance level (barring statistical fluctuation).

2.4.1 Formal definition

A rigorously formal definition of CP has inevitably to take into account a number of technicalities. Here, we will attempt to strike a balance between clarity and formality.

We will specify CP with reference to on-line mode as defined in Algorithm 1 because it is in this mode that the theoretical results are stated and proved in the literature, but it is possible to extend it in some sense also to the batch case.

Algorithm 1: On-line protocol

Data: Sequence of examples: $z_1 = (x_1, y_1)$, $z_2 = (x_2, y_2), \ldots$
Result: Prediction sets: $\Gamma_{\epsilon,1}$, $\Gamma_{\epsilon,2}, \ldots$
Cumulative error counts: $\mathrm{Err}_{\epsilon,1}, \mathrm{Err}_{\epsilon,2}, \ldots$

$\mathrm{Err}_0 = 0$;
for $i = 1, 2, \ldots$ **do**
  Nature presents x_i ;
  Predictor outputs $\Gamma_{\epsilon,i}$;
  Nature reveals y_i ;
  $\mathrm{err}_i = \begin{cases} 1 & \text{if } y_i \notin \Gamma_{\epsilon,i} \\ 0 & \text{otherwise} \end{cases}$;
  $\mathrm{Err}_{\epsilon,i} = \mathrm{Err}_{\epsilon,i-1} + \mathrm{err}_i$;
end

Let's assume that the training set is made up of a sequence of ℓ examples $z_i := (x_i, y_i) \in \mathbf{Z} = \mathbf{X} \times \mathbf{Y}$ and $x_{\ell+1}$ is a test object taken from the same exchangeable distribution as the training examples.

We will use the notion of *bag* or *multi-set*. A bag of size $\ell \in \mathbb{N}$ is a collection of ℓ elements some of which may be identical; a bag differs from a set in that repetition is allowed. We will indicate a bag with the following notation $\wr z_1, \ldots, z_\ell \int$. The set of all possible bags of k elements from a set $\mathbf{Z}$ will be denoted as $\mathbf{Z}^{(k)}$ (note the brackets around the exponent).

We call Non-Conformity Measure (NCM) a real-valued measurable function $A(z; \wr z_1, \ldots, z_k \int)$, $A : \mathbf{Z} \times \mathbf{Z}^{(k)} \to \mathbb{R}$. The notation may seem strange, but it distinguishes clearly between the collection of examples $\wr z_1, \ldots, z_k \int$ and the example z for which we want to measure the non-conformity with respect to the collection. Also, the use of a bag emphasizes that the order of the elements is irrelevant. Let's call α_i

the NCM of z_i according to:

$$\alpha_i = A((x_i, y_i), \lbag z_1, \ldots, z_\ell, z_{\ell+1} \rbag / \lbag (x_i, y_i) \rbag) \qquad i = 1, \ldots, \ell + 1 \tag{2.6}$$

.

Given these NCM values, it is possible to compute for a test example $(x_{\ell+1}, \bar{y}_{\ell+1})$ with hypothetical label $\bar{y}_{\ell+1}$ a *p-value* defined as:

$$p_{\bar{y}} := \frac{|\{i = 1, \ldots, \ell + 1 : \alpha_i \geq \alpha_{\ell+1}\}|}{\ell + 1} \tag{2.7}$$

In words, the p-value of a hypothetical completion $(x_{\ell+1}, \bar{y}_{\ell+1})$ is the fraction of the elements in the training bag augmented with the hypothetical completion itself whose NCM is greater than or equal to the NCM of the hypothetical completion[3].

The prediction region Γ_ϵ for a test object $x_{\ell+1}$ for a chosen significance level $\epsilon \in [0, 1]$ is the set of labels for which the p-value exceeds the significance level:

$$\Gamma_\epsilon(x) := \{y \mid p_y > \epsilon\} \tag{2.8}$$

It is customary to use the term *confidence* for the quantity $1 - \epsilon$.

We say that an error occurs when the region prediction Γ_ϵ does not contain the actual label, i.e. $y_i \notin \Gamma_\epsilon$. We will refer to the count of the errors up to and including step n as $\mathrm{Err}_{\epsilon,n}$.

As stated in previous sections, it is possible to state *validity* guarantees for CPs. With the definition of p-value in Eq. 2.7, it can be proved that the CP has a *conservative asymptotic validity* property is that the rate of errors converges almost surely[4] to a value that is less than or equal to the significance level, i.e.

$$\lim_{n\to\infty} \frac{\mathrm{Err}_{\epsilon,n}}{n} \leq \epsilon \qquad \text{a.s.} \tag{2.9}$$

To achieve exact validity, eq. 2.7 must be modified so that ties (i.e. the occurrences of multiple α_i equal to $\alpha_{\ell+1}$) are broken with a element of randomness.

$$p_{\bar{y}} := \frac{|\{i = 1, \ldots, \ell + 1 : \alpha_i > \alpha_{\ell+1}\}| + \tau|\{i = 1, \ldots, \ell + 1 : \alpha_i = \alpha_{\ell+1}\}|}{\ell + 1} \tag{2.10}$$

where $\tau \sim U[0, 1]$, i.e. τ is an RV uniformly distributed on $[0, 1]$ (this RV is to be "drawn" independently for each test object). With this more complex definition of p-value (referred to as *smoothed* p-value), it can be proved that the asymptotic validity becomes exact, i.e.

$$\lim_{n\to\infty} \frac{\mathrm{Err}_{\epsilon,n}}{n} = \epsilon \qquad \text{a.s.} \tag{2.11}$$

[3] An equivalent formulation might have used the bag unchanged and simply added one at the numerator. This formulation however will become preferable when we introduce the smoothed conformal predictor.

[4] The 'almost surely' qualification is a technicality that can be informally explained as the fact that the probability of encountering a sequence of examples such that the assert is not verified is vanishing

The validity property can be formally proved as a consequence of the following key theorem (Gammerman and Vovk, 2007, Theorem 1)

Theorem 1. *Suppose the examples* $(x_1, y_1), (x_2, y_2), \ldots$ *are generated independently from the same probability distribution.*
For any smoothed Conformal Predictor working in the on-line prediction protocol and any significance level $\epsilon \in [0, 1]$*, the Random Variables* $err_1, err_2, \ldots$ *are independent and take value 1 with probability* ϵ

Both conservative and exact validity guarantees as stated above in Eq. 2.9 and 2.11 are asymptotic and it may be argued that they may not be relevant in any finite-sample regime that we may encounter in practice. There exists also a finite-sample guarantee (Vovk, Gammerman, and Shafer, 2005, p.27), which can be derived by applying Hoeffding's inequality:

$$\forall n > 0, \forall \delta > 0 \qquad \mathbb{P}\left[\frac{\mathrm{Err}_{\epsilon,n}}{n} \geq (\epsilon + \delta)\right] \leq e^{-2n\delta^2} \tag{2.12}$$

In words, this finite-sample guarantee states that for any choice of $\delta > 0$, the probability that the actual observed error rate exceeds the targeted ϵ by δ is bounded by $e^{-2n\delta^2}$.

2.4.2 Some comments

The definition of CP in the previous section appears to suggest that a prediction region for a test object is computed by examining as many cases as possible values of the label. Obviously, this would be infeasible for regression problems because in that setting the label can take infinitely many values. However, for some definitions of NCMs, it turns out that it is possible to compute (theoretically exact) prediction regions in a finite number of steps. Two observations are relevant here: (a) there is a finite number of possible values that the p-value can take and (b) what we need to compute is really what values of the label correspond to a given p-value. The former is a direct consequence of the definition itself of CP p-value. The latter is a direct consequence of the definition of prediction region as the set of labels for which the p-value is greater than the significance level. For some choices of NCM, the relationship between p-values and hypothetical labels can be computed in a finite number of steps. One such example is the Ridge Regression Confidence Machine (Nouretdinov, Melluish, and Vovk, 2001).

The prediction regions $\Gamma_\epsilon(x)$ can contain any subset of the label space $\mathbf{Y}$, including the entire label space $\mathbf{Y}$ and the empty set. The latter happens when there is no label $\bar{y}_{\ell+1}$ that would create with the test object $x_{\ell+1}$ an example whose conformity (with the respect to the examples in the bag) would be sufficient to be included in the prediction. One way to interpret this is that the object is an anomaly[5], in that no label

[5]One should keep in mind that, as discussed later in Section 2.4.9, the p-values for the correct labels are distributed uniformly. So, empty prediction sets could occur also, with probability that depends on the significance level, when the object is not an anomaly.

assignment would "seem right". An empty prediction is automatically counted as a prediction error.

It has to be noted that conservative validity, in the sense of a guarantee that the predictions will exhibit an error rate that does not exceed the chosen significance level ϵ, can be banally obtained by predicting always the entire set $\mathbf{Y}$ as $\Gamma_\epsilon(x)$. Of course, such predictions would be totally uninformative and completely useless. In fact we want the prediction regions $\Gamma_\epsilon(x)$ to be as small as possible (without being empty)[6].

Developing further this line of thought, we can identify two main desiderata in set prediction:

- Validity: the error rate corresponds to the chosen significance level.
- Efficiency: the prediction sets are as small as possible

There is obviously a trade-off in general between these two goals, as making the prediction sets smaller makes missing the correct label more likely. By using CPs, one can take advantage of the fact that validity is guaranteed, so that all efforts can be focused solely on improving efficiency[7].

Validity is guaranteed, regardless of the choice of NCM. Even if the NCM is a constant or, say, a random number, the smoothed CP exhibits exact validity. But the predictions are uninformative. It is the efficiency that is determined by the NCM. A Non-Conformity Measure can be in principle extracted from any Machine Learning (ML) algorithm, which is then referred to as the *underlying* ML method. Although there is no universal method to derive an NCM, a default choice is:

$$A((x,y),\{z_1,\ldots,z_k\}) := \Delta(y, f(x)) \tag{2.13}$$

where $f : \mathbf{X} \to \mathbf{Y}'$ is the prediction rule learned on $(z_1,\ldots,z_k)$ and $\Delta : \mathbf{Y} \times \mathbf{Y}' \to \mathbb{R}$ is a measure of dissimilarity between a label and a prediction. Specific examples are provided further on in Chapter 3. Note also that any monotone transformation of an NCM produces the CP that outputs the same predictions as the original CP (by looking at Eq. 2.7 the p-values are exactly the same).

2.4.3 Ideal CP efficiency

As we have seen, CPs have by construction a validity property, so what distinguishes them (as long as the i.i.d. assumption is met) is the distribution of the size of the prediction sets. Here we will focus on a measure of locality of such distribution, namely the average, although other choices are possible. We will speak of the *efficiency* of a CP and we will relate it to the average set size in the sense that the smaller the

[6] We argue that empty prediction sets are not desirable in the specific case in which we are dealing with objects that have a label and we want to have a prediction that identifies that label. In other scenarios, empty prediction sets might convey a useful result.

[7] This echoes a direction advocated in (Tukey, 1986)

average prediction set size, the higher the efficiency. In (Vovk et al., 2016) various measures of CP efficiency are proposed, but here we will restrict our attention to the average prediction set size, which in the paper is referred to as N criterion. In this section, we discuss the lower bounds for the average prediction set size for a binary set-valued predictor that guarantees validity.

Oracle

Let's assume that we can avail ourselves of the services of an Oracle that gives the correct classification for any object. Our constraint is to emit set predictions that do not contain the correct label with relative frequency ϵ, where ϵ is the significance level. The strategy that minimizes the average set size is to output the correct label suggested by the Oracle with relative frequency $1 - \epsilon$ and an empty set with relative frequency ϵ. We therefore achieve an average set size of $1 - \epsilon$.

Bayes classifier

In any non-trivial practical case, we cannot expect to have ideal accuracy. Even with a Bayes Classifier (Hastie, Tibshirani, and Friedman, 2009, page 21), we can only hope to achieve an error rate $0 \leq r \leq 0.5$, which depends on the conditional probability $P(y \mid x)$ of the label given the object. We now adapt the strategy of the Oracle case to the case of a Bayes Classifier.

Let's consider first the case $r = \epsilon$. In this case, if we output prediction sets containing only the label predicted by the Bayes classifier, we achieve the target rate. This is the choice that leads to the smallest average size of the prediction sets. Let's assume that we could achieve the same result by producing a mix of uncertain sets (containing both labels), and empty sets. The relative frequency of empty sets would have to be r (obviously, because errors occur only for empty sets). So the relative frequency of uncertain predictions would be $1 - r$. The expectation of the prediction set size is then $\mathbb{E}[s] = 2(1 - r)$. For $r < 0.5$ the expectation is strictly greater than 1. It is greater than the size of the prediction sets containing only the label predicted by the Bayes classifier.

Let's now examine the two other cases, namely $\epsilon > r$ and $\epsilon < r$.

Without loss of generality, let's assume that we have N test objects. We aim to produce $N(1 - \epsilon)$ correct predictions and $N\epsilon$ errors. We denote with N_p the number of single-label prediction sets, N_u the number of the uncertain predictions, N_e the number of empty sets (with $N_p + N_u + N_e = N$). The average of the prediction set sizes is:

$$\bar{s} = \frac{0 \cdot N_e + 1 \cdot N_p + 2 \cdot N_u}{N}$$

When $\epsilon > r$, we are targeting a rate of errors larger than the one that the classifier produces. The average will be minimised if we produce the correct predictions with as few test objects as possible, so on the basis of the previous case, we will output a fraction of the predictions as single label prediction sets. These will contribute error

at a rate r. So, we have:

$$N\epsilon = N_p r + N_e$$
$$N_u = 0$$

This results in

$$N_p = N\frac{1-\epsilon}{1-r}$$

It is then straightforward to see that

$$\bar{s} = \frac{1-\epsilon}{1-r}$$

When $\epsilon < r$, we are targeting a rate of errors lower than the one that the classifier produces. We will have to inject some correct predictions by means of uncertain predictions. We need to ensure that there are $N\epsilon$ errors. These will arise from the single label predictions at a rate of r. We have:

$$N_p r = N\epsilon$$
$$N_e = 0$$

With straightforward algebraic manipulation, we obtain:

$$\bar{s} = 2 - \frac{\epsilon}{r}$$

2.4.4 Comparison of set-valued predictors

When evaluating the performance of different set-valued predictors (irrespective of whether they are Conformal Predictors or other types), it is essential to keep in mind that there is a trade-off between validity and efficiency. Efficiency should not be evaluated in isolation and the same applies to validity. It may happen that a predictor achieves higher efficiency simply by predicting excessively tight label sets which miss the error target and result in a deviation from validity. Figure 2.3 illustrates the validity and efficiency for some of the CP combination methods discussed in Chapters 5 and 6. CP do have a validity property, but the guarantee hinges on the data being i.i.d. and in practice there are reasons for this minimal of requirements to be violated. In the domain of chemoinformatics, for example, the problem of predicting biological activity of chemical compounds (discussed in Chapter 3) presents in practice this problem: the compounds that are submitted for predictions at time T are not chosen independently from the compounds for which measurements of their biological activity were available at that time T (i.e. the training set). In fact, medicinal chemists tend to explore variations of previously considered chemicals, often adding or removing functional groups, in the hope of obtaining a drug with all the required safety, metabolic, and potency properties.

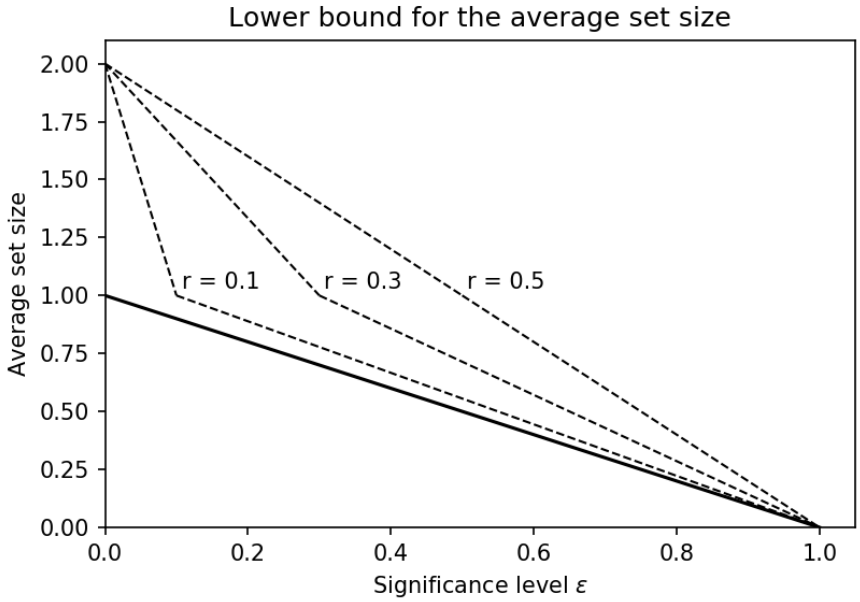


FIGURE 2.2: Lower bounds for the average set size. The solid line shows the minimum set size when classifications come from an Oracle. The dashed lines show three cases where classification are given with different error rates r, namely 0.1, 0.3, 0.5. The case of $r = 0.5$ is, of course, the worst case. For a (randomized or smoothed) CP, it is for instance what happens when the NCMs are all the same.

Conformal Predictors bring a principled approach to the probabilistic prediction paradigm advocated in (Gneiting, Balabdaoui, and Raftery, 2007) in the slight different context of predictive distributions and with different nomenclature: "maximizing the sharpness [...] subject to calibration". In that paper, calibration corresponds *mutatis mutandis* to validity and sharpness to efficiency. In principle, CPs allow to pursue just that: they guarantee validity and make it possible to concentrate the efforts on obtaining smaller prediction sets.

In terms of evaluating different set-valued prediction methods, however, it can be argued that the optimal trade-off between validity and efficiency depends on the specific application. It is quite possible that method A be preferable to method B despite having worse validity deviation, if its efficiency is markedly better. The choice should in general be driven by a loss function that captures the cost of errors (when the actual value outside the predicted set or interval) and of hedging (the undesirability of multiple values as prediction).

2.4.5 Transductive and Inductive CP

CP as described in previous section is referred to as Transductive. The term originates from an idea put forward in (Vapnik, 1995, p.293).

Vapnik refers to the conventional approach to Statistical Learning as Induction: on the basis of a training set, a model is obtained (once for all) and then (Deduction) used to make predictions on any given test object. Deduction is viewed as a form of

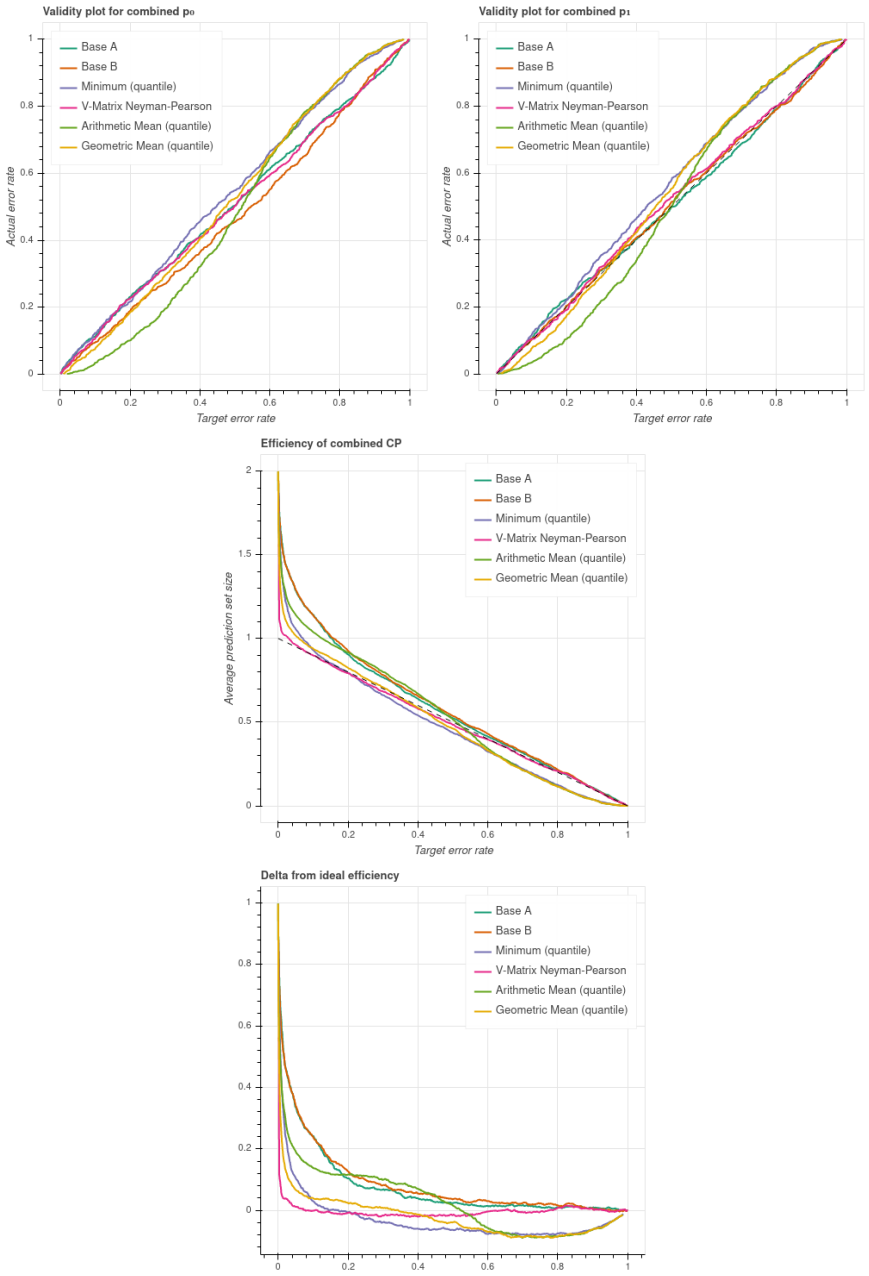


FIGURE 2.3: Efficiency and Validity. This figure shows validity plots and efficiency plots (average prediction set sizes) for some of the CP combination methods discussed in Chapters 5 and 6. In particular this set of charts refers to a case in which the NCMs of the two base CP exhibited negative correlation. This created deviations from validity in the simpler methods, which relied on statistical independence of the underlying ML methods. This example also serves to illustrate the connections between validity and efficiency. Some methods may appear more efficient than others, but they achieve that at the expense of validity. Vice versa, some methods may achieve lower error rates but do so outputting larger prediction sets. This is the case of the green trace, which, for low values of the target error rate, has fewer errors than targeted, but has on average larger predictions sets (see middle row and bottom row). In this example, the best predictor would seem to be the red trace, as it is valid (within statistical fluctuation) and has the smallest average prediction sets.

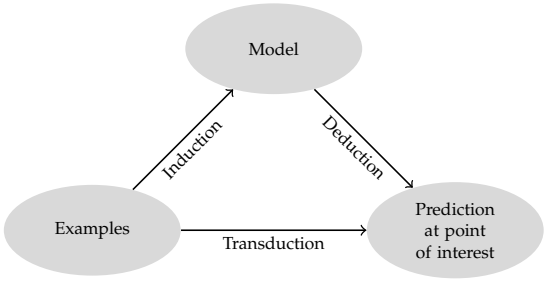


FIGURE 2.4: Induction, Deduction, and Transduction. The conventional approach seeks a model that predicts well for all possible values; in other words, it tries to solve a the problem of estimating a function. Transduction aims at creating a model that predicts well at the points of interest; it tries to estimate values of a function, not the function.

inference that moves from general to particular. The model sought by Induction is of general applicability. Induction can be viewed as a form of inference that moves particular to general. Vapnik observed that we may be solving a more complex problem than warranted. We do not necessarily want a model that performs well for every possible test object. What we want is an accurate prediction for a specific test object (or a finite number of test objects). Vapnik summed up the rationale for transduction as: *"If you are limited to a restricted amount of information, do not solve a particular problem by solving a more general problem"*. Consequently, given a test object, we should seek directly the specific prediction, without the intermediate step of solving the likely more difficult problem of finding a model that is accurate in general, just to apply it on the specific case of interest. Transduction proposes a form of inference that moves from particular to particular. In practical terms, the idea is therefore to exploit the knowledge of the test object during training.

Indeed, the definition of the NCM in Eq. 2.6 prescribes that the test object should become part of the bag which represents the training set. In Transductive CP, given a hypothetical completion $(x_{\ell+1}, \bar{y}_{\ell+1}$, the underlying ML model is retrained $\ell + 1$ times to compute each α_i. For each $i = 1, \ldots, \ell + 1$, the underlying ML is trained from scratch on $\langle z_1, \ldots, z_\ell, z_{\ell+1} \rangle / (x_i, y_i))$, that is, a training set in which the i-th example has been removed. So, for one test object, the underlying ML model is trained $|\mathbf{Y}| \cdot (\ell + 1)$ times. In practice, the resulting computational cost becomes prohibitive for all but the simplest practical applications.

A different form of CP has been proposed (Papadopoulos et al., 2002) which prescribes a different way of computing the NCM and retains the validity property. This different form is referred to as Inductive CP or, by some authors (see, for instance, (Lei et al., 2018)), as Split CP.

As illustrated schematically in Figure 2.5, the training set is partitioned into two sets, called proper training set and calibration set[8]. The proper training set is used to train the underlying ML method. The training of the underlying ML method

[8]It could be argued that Inductive CPs require more data because they need a calibration set in addition to a proper training set. While intuitively justifiable, there is no theoretical result (as far as

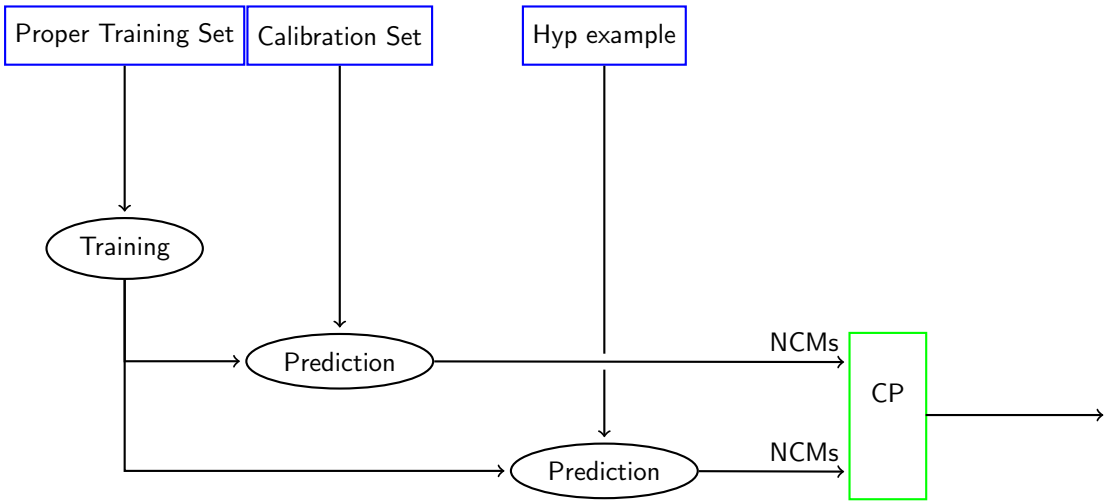


FIGURE 2.5: A schematic representation of Inductive CP

is performed once only. The same fitted model is used to compute the α_i for the examples of the calibration set and for the $\alpha_{\ell+1}$ on the hypothetical examples formed by trying out in turn every possible label.

Assuming that the first m examples constitute the calibration set and the remaining $k = \ell - m$ examples the proper training set, the α_i can be formally expressed as:

$$\alpha_i = A((x_i, y_i), \{z_1, \ldots, z_k\}) \qquad i = 1, \ldots, m$$
$$\alpha_{m+1} = A((x_{\ell+1}, y_{\ell+1}), \{z_1, \ldots, z_k\})$$

The (smoothed) p-value for a hypothesis $y_{\ell+1} = \bar{y}$ about the label of test object $x_{\ell+1}$ is defined as follows:

$$p_{\bar{y}} = \frac{|\{i = 1, \ldots, m+1 : \alpha_i > \alpha_{m+1}\}| + \tau|\{i = 1, \ldots, m+1 : \alpha_i = \alpha_{m+1}\}|}{m+1}$$

It is interesting to note that, for Inductive CP, the assumption of exchangeability is required only for the calibration set and the test set. The validity property is in fact independent of the proper training set, which can then be chosen at will. In fact, it can also be observed that there is no inherent reason for the NCM to have to be learned on a proper training set. It can be chosen at will, possibly based on some a priori knowledge, as long as it satisfies the requirements stated in Section 2.4.1.

the author was able to ascertain) that prescribes the partition into calibration and training set. Cross-conformal predictors (Vovk, 2015) can mitigate this issue but the averaging that is recommended in the paper results in loss of validity.

2.4.6 Label-conditional (Mondrian) CP

Finally, the validity property as stated above guarantees an error rate over all possible label values, not on per-label value basis. The latter can be achieved with a variant of CP, called *label-conditional CP* (or also Mondrian[9] CP). The label-conditional CP is one form of conditional CPs, which are discussed in more general terms in (Vovk, 2013). The only change is in the calculation of the p-value: we restrict the α_i only to those that are associated with examples with the same label as the hypothetical label that we are assigning at the test object. So, the p-value for a hypothesis $y_{\ell+1} = \bar{y}$ about the label of test object $x_{\ell+1}$ is defined as follows:

$$p(\bar{y}) = \frac{|\{i = 1, \ldots, (m+1) : y_i = \bar{y}, \alpha_i \geq \alpha_{m+1}\}|}{|\{i = 1, \ldots, (m+1) : y_i = \bar{y}\}|} \tag{2.14}$$

The property of label-conditional validity is essential in practice when the CP is applied to an "imbalanced" data set, that is, a data set in which the proportions of labels are significantly different. Empirically, one can observe that with the plain validity property, the overall error rates tend within statistical fluctuation to the chosen significance level, but the minority class(es) are disproportionally affected by errors (see, for instance, (Löfström et al., 2015)). This property ensures that, even for the minority class, the long-term error rate will tend to the chosen significance level.

2.4.7 An example on synthetic data

In this section, we will walk through an example of Transductive CP on a binary classification problem using k Nearest Neighbours as the underlying ML method. A synthetic training data set is generated so that the examples form two roughly semicircular interlocking clouds of points in the plane (commonly referred to as "moons"), as illustrated in Figure 2.6.

The NCM is chosen as:

$$\alpha := \frac{\sum^{(k)}_{j \neq i : y_j = y_i} d(x_j, x_i)}{\sum^{(k)}_{j \neq i : y_j \neq y_i} d(x_j, x_i)}$$

where by $\sum^{(k)}$ we denote the sum of only the k smallest terms. In this example, k was set to 3. So, given a training set $\wr z_1, \ldots, z_\ell \int$ and a hypothetical test example $z_{\ell+1}$, the i-th α (with $i = 1, \ldots, \ell = 1$) is calculated by removing z_i from the bag $\wr z_1, \ldots, z_{\ell+1} \int$ and calculating the ratio between the sum of the distances of the $k = 3$ closest examples with the same label as z_i and the sum of the $k = 3$ closest examples with a different label than z_i.

[9] Conditional CPs are formally defined by introducing a notion of taxonomy on the space of the examples, on the basis of which the training set (or the calibration set in the Inductive CP case) is partitioned into categories. The graphical representation of this partitioning on bivariate examples gives rise to images that can remind one of the distinctive style associated with the Dutch-French artist Piet Mondrian.

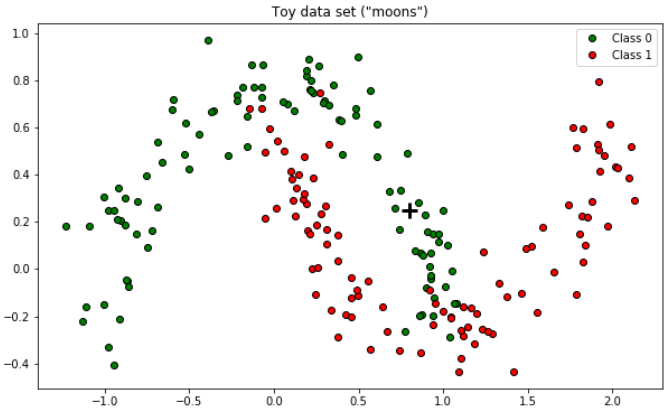


FIGURE 2.6: A synthetic data set (moons) for the binary classification example. For the test object represented by the black cross at coordinates (0.8,0.25), the p-values are $p_0 = 0.5622$ and $p_1 = 0.00995$. If we chose $\epsilon = 0.01$, the prediction set would be $\{0\}$, label 1 being rejected because $p_1 < 0.01$.

For any test point there are four possibilities as to the prediction:

	Prediction set
$p_0 \leq \epsilon, p_1 \leq \epsilon$	$\varnothing$
$p_0 > \epsilon, p_1 \leq \epsilon$	$\{0\}$
$p_0 \leq \epsilon, p_1 > \epsilon$	$\{1\}$
$p_0 > \epsilon, p_1 > \epsilon$	$\{0, 1\}$

Figure 2.7 shows the predictions for every every test object in a rectangular region, using a different colour for each of the four possible outcomes listed above. As the target error rate ϵ is reduced, the areas where the CP makes single predictions shrink. The CP outputs more uncertain predictions as it cannot reject at that significance level any of the labels.

One basic technique to assess the performance of classification models is to use a confusion matrix, i.e. a form of contingency table in which rows correspond to the actual labels of test examples, columns to the predicted labels, and the cells (i, j) contain the counts of examples of label i predicted as j.

In the case of CP, the confusion matrix takes a slightly different form than usual as a consequence of the different nature of the predictions, which are sets rather than single values. Considering only the binary classification case, for each actual label, we may be interested in how many examples were correctly predicted (with the prediction set containing only the correct label), how many examples were incorrectly predicted (that is, the prediction set contains only the incorrect label) how many examples were predicted inconclusively (that is, the prediction set contains both labels), and finally how many examples were given an empty prediction set.

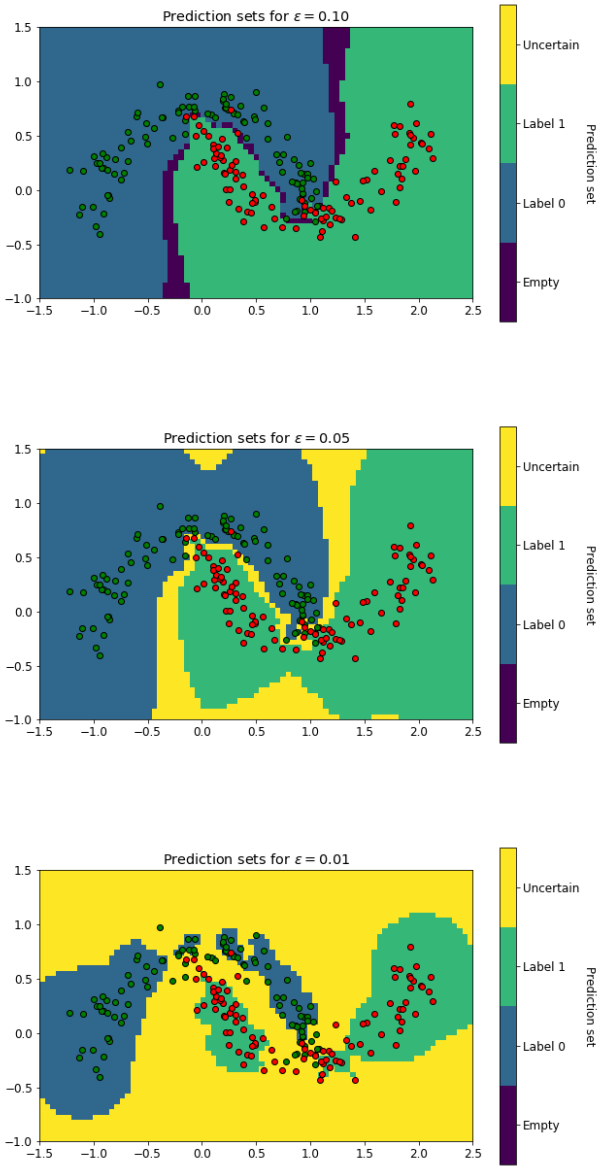


FIGURE 2.7: Each plot shows the prediction sets at a given significance level for a grid of test points. The color at each point codes whether the prediction set for the test object at that point is empty (dark blue), contains only label 0 (light blue), contains only label 1 (green), or contains both labels and hence is an uncertain prediction (yellow). The plots correspond to decreasing significance levels, starting from the top to the bottom. For significance level $\epsilon = 0.10$, a narrow blue area, where the prediction set is empty, divides the domains where the prediction set contains one label. The blue area corresponds to objects for which whichever label assignment resulted in (hypothetical) examples that looked too non-conform for the chosen significance level. As the significance level is decreased (i.e. we demand a lower error rate) the blue areas shrink and eventually yellow areas (where the prediction sets contain both labels) take their place. The eventual prevalence of the yellow areas arises because the label hypotheses can no longer be rejected at such low significance levels.

TABLE 2.1: Example of CP predictions for various values of the significance level ϵ. The error rate is reported next to the significance level ϵ to facilitate the verification of the validity property. By the comparison of the two columns, one can confirm that the validity property does hold, within statistical fluctuation. The entries of the confusion matrices are presented (arranged linearly) in the central group of 8 columns. The test data set had 1200 class 0 examples and 400 class 1 examples.

ϵ	Error rate	Uncertain fraction	1 pred 1	1 pred 0	0 pred 0	0 pred 1	1 pred $\varnothing$	0 pred $\varnothing$	1 pred $\{0,1\}$	0 pred $\{0,1\}$	1 error rate	0 error rate
0.05	0.056	0.011	329	62	1163	28	0	0	9	9	0.155	0.023
0.10	0.111	0.000	310	33	1113	9	57	78	0	0	0.225	0.072
0.15	0.165	0.000	294	15	1042	1	91	157	0	0	0.265	0.132
0.20	0.204	0.000	281	8	992	0	111	208	0	0	0.297	0.173
0.25	0.251	0.000	259	4	939	0	137	261	0	0	0.352	0.217
0.50	0.497	0.000	109	0	695	0	291	505	0	0	0.728	0.421
0.75	0.744	0.000	27	0	383	0	373	817	0	0	0.932	0.681
0.80	0.783	0.000	21	0	326	0	379	874	0	0	0.948	0.728
0.85	0.829	0.000	14	0	259	0	386	941	0	0	0.965	0.784
0.90	0.892	0.000	7	0	166	0	393	1034	0	0	0.983	0.862
0.95	0.943	0.000	1	0	91	0	399	1109	0	0	0.998	0.924

A summary of the predictions for different significance levels is shown in Table 2.1. The error rate is within statistical fluctuation of the significance level, consistently with the validity property of CP. The data set in this example is imbalanced, with one third of the examples of class 1 and the remaining two thirds of class 0.

As discussed in Section 2.4.6, when the data set is imbalanced, the validity guarantee of CP does not apply to each class separately. Indeed, one can observe that in Table 2.1 the error rate for class 1 is markedly greater than the significance level. To remedy this, one can use label-conditional CP. In Table 2.2, label-conditional CP is applied to the same training and test data as in Table 2.1. The resulting error rate for class 0 and error rate for class 1 are both close to the significance level, as graphically illustrated in Figure 2.8

2.4.8 Confidence and Credibility

Restricting now our attention to classification problems, in some cases, it may be desirable to focus the hedged forecast on a single value referred to as point prediction, rather than a set or an interval. The most straightforward choice is to take the label that is associated with the largest p-value[10]. The hedging of the prediction can then be expressed by complementing the point prediction with quantities that characterize the uncertainty. For example, (Saunders, Gammerman, and Vovk, 1999; Gammerman and Vovk, 2007) recommend using *confidence* and *credibility*. Confidence is defined as:

$$\sup\{1-\epsilon : |\Gamma_\epsilon| \leq 1\}$$

[10]This is not necessarily equivalent to taking the highest scoring label according to the underlying ML method. In the case of Mondrian CP, for instance, the p-value for label $\bar{y}$ is calculated with respect to the calibration set examples with label $\bar{y}$

TABLE 2.2: Example of label-conditional CP predictions for various values of the significance level ϵ. This uses the same data set (in fact, the same α_i) as Figure 2.1, but computes the p-values using the label-conditional method of Eq. 2.14. The validity property holds at the level of each label, as the two rightmost columns show.

ϵ	Error rate	Uncertain fraction	1 pred 1	1 pred 0	0 pred 0	0 pred 1	1 pred $\varnothing$	0 pred $\varnothing$	1 pred $\{0,1\}$	0 pred $\{0,1\}$	1 error rate	0 error rate
0.05	0.043	0.094	359	11	1021	58	0	0	30	121	0.028	0.048
0.10	0.106	0.000	364	19	1067	72	17	61	0	0	0.090	0.111
0.15	0.160	0.000	335	9	1009	35	56	156	0	0	0.163	0.159
0.20	0.222	0.000	308	4	937	8	88	255	0	0	0.230	0.219
0.25	0.284	0.000	291	2	855	1	107	344	0	0	0.273	0.287
0.50	0.507	0.000	225	0	564	0	175	636	0	0	0.438	0.530
0.75	0.719	0.000	123	0	326	0	277	874	0	0	0.693	0.728
0.80	0.768	0.000	84	0	287	0	316	913	0	0	0.790	0.761
0.85	0.830	0.000	69	0	203	0	331	997	0	0	0.828	0.831
0.90	0.893	0.000	44	0	127	0	356	1073	0	0	0.890	0.894
0.95	0.958	0.000	16	0	52	0	384	1148	0	0	0.960	0.957

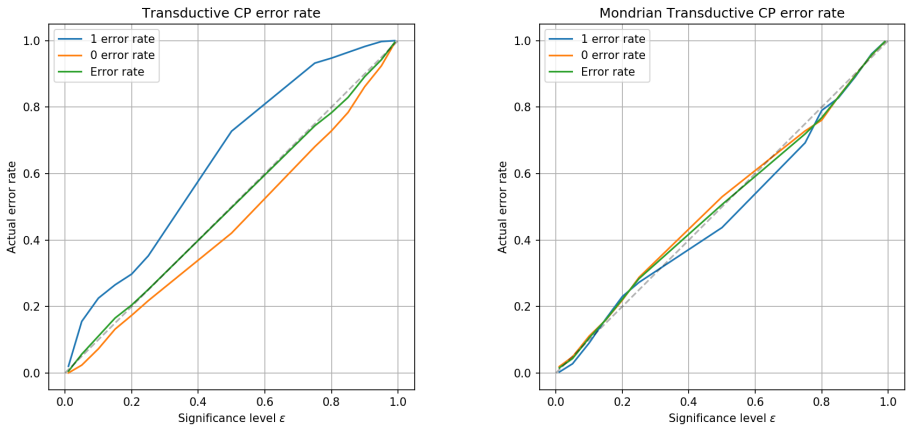


FIGURE 2.8: Label-conditional validity. The plot on the left shows that the plain CP exhibits validity overall (green line), but deviates significantly from it on the minority class. This issues does not occur in the plot on the right, where label-conditional CP is used.

that is, the largest "confidence" (in sense defined in section 2.4.1) for which the prediction set contains only one label. It can be computed as $1 -$ second largest p_y.
Credibility is:

$$\inf \{\epsilon : |\Gamma_\epsilon| = 0\}$$

which can also be expressed more simply as the largest p-value.
It has to be noted that there are no theoretical guarantees on these two quantities.

2.4.9 CP and Statistical Hypothesis Testing

The framework of CP can be interpreted as an application of the methods of "traditional" Statistical Hypothesis Testing (SHT) to Machine Learning. Indeed the p-value can be viewed as the probability of drawing from the same distribution $\mathbf{F}$ that generated the training set an example that is as or more contrary to the hypothesis of randomness than the one in hand. The prediction set for an object $x_{\ell+1}$ is then formed by all the labels $\bar{y}$ for which the Null Hypothesis that the (hypothetical) example $(x_{\ell+1}, \bar{y})$ comes from $\mathbf{F}$ cannot be rejected at the chosen significance level ϵ. The NCM plays the role of test statistic, i.e. of a value that is larger the more contrary to the Null Hypothesis a sample is. This explains why, instead of a Conformity Measure, the possibly less straightforward notion of Non Conformity Measure is used in defining CP.

Given the p-value defined with Eq. 2.10 can be viewed as cognate of the p-value in SHT, one may be warned that it then carries the baggage of controversy of the latter. While virtually ubiquitous, the notion is often misused to an extent that some scientific journals have explicit editorial guidelines that reject out-of-hand manuscript that only provide p-values without further information. Also, every few years open letters are signed by groups of scientists and statisticians — see for instance (Benjamin et al., 2017) — advocating the deprecation of p-values for confirming a scientific hypothesis or, at the very least, the use of tighter thresholds.

Some of the shortcoming of SHT p-values are perhaps not so relevant in this specific application. One main source of issues with the notion of p-value can perhaps be traced to the fact that the p-value can very easily be misinterpreted as the posterior probability of the Null Hypothesis. This pitfall is so widespread and insidious that some authors (for example, (Sellke, Bayarri, and Berger, 2001)) refer to it as the "p-value fallacy".

To illustrate the issue in an informal empirical way, Figure 2.9 shows a typical histogram of the p-values produced by a Mondrian Inductive CP that uses a relatively accurate NCM. The histogram counts separately the examples of class 0 and class 1 whose p_0 falls in each one of the bins $[0, 0.1], \ldots, [0.9, 1]$. One can observe that the examples of class 0 have p_0 values that are, within reason, uniformly distributed. This is indeed the manifestation of the p-value validity property: under the Null Hypothesis H_0, if one rejects H_0 for $p < \epsilon$ with $\epsilon \in [0, 1]$, the (Type I) error rate is going

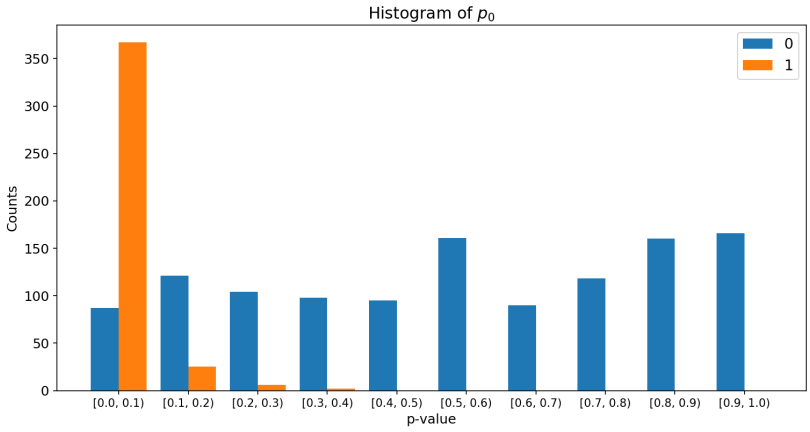


FIGURE 2.9: Histogram of p-values. Test examples are binned by their p_0. In each bin, class 0 examples and class 1 examples are counted separately. The p-values are taken from the example discussed in detail later in section 2.4.7. The data set is imbalanced: there were 1200 class 0 examples and 400 class 1 examples.

to be ϵ in the long run. The p_0 for the class 1 examples presents instead a distribution that appears to peak sharply at 0. This reflects the fact that the hypothetical assignment of the label 0 to those test objects results in an example so incompatible with the training set (their NCM is higher than most of the training set) that only a small percentage of the examples in the training set is judged more non-conform. The (posterior) probability of the Null Hypothesis (here, that the assignment of label 0 to the test object in hand creates an example that comes from the same distribution as the training set) can be empirically estimated roughly in each bin as the ratio of the label 0 examples to the total number of examples in the bin.

So, in the $[0, 0.1]$ bin there are 367 examples from class 1 and 87 for class 0. So we may empirically estimate that the probability of a test object being class 0 conditional on $p_0 \in [0, 0.1]$ is $\frac{87}{87+367} = \approx 0.19$.

This should illustrate very clearly the different nature of the two concepts, p-value and conditional probability of the H_0. The former has to do with the guarantee on the incorrect rejections of the Null Hypothesis. The latter depends on two factors: (a) the quality of the NCM and (b) the imbalance of data set. When the NCM discriminates less accurately the conformity of an example, the p-values for the "correct" hypothesis will still tend to be uniformly distributed, but the p-values for the "incorrect" hypothesis tend to concentrate less around zero. The conditional probability of H_0 as estimated by the ratio of the examples in each bin discussed above will then vary (for the same value of the p-value). The ratio will also vary with the prevalence of one class over the other.

(Vovk, 1993) and (Sellke, Bayarri, and Berger, 2001) later have shown that it is possible under relatively reasonable assumptions to calibrate a p-value into some valid measure of empirical evidence. (Sellke, Bayarri, and Berger, 2001) derive a

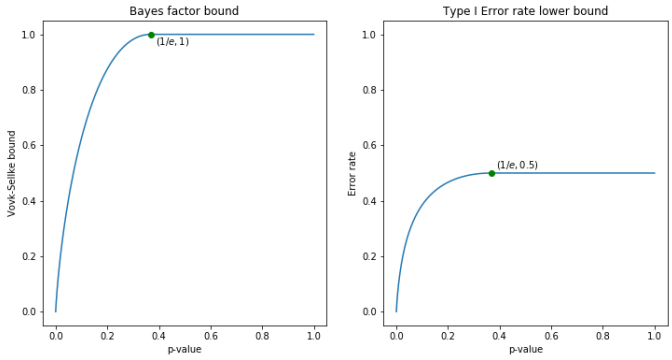


FIGURE 2.10: p-value calibration.

lower bound for the Bayes factor[11] for H_0 against $\bar{H}_0$ or a frequentist probability of error. The lower bound for the Bayes factor is:

$$B(p) = \begin{cases} -e\, p \log(p) & p < 1/e \\ 1 & p \geq 1/e \end{cases}$$

and also a lower bound on the conditional probability of rejecting incorrectly H_0:

$$\alpha(p) = \left(1 + (-e\, p \log(p))^{-1}\right)^{-1}$$

Figure 2.10 depicts the bounds as a function of p. The main assumption behind this derivation is that the p-values under the $\bar{H}_0$ follow a Beta$(\xi, 1)$ distribution i.e. with density $\xi p^{\xi-1}$ where $0 < \xi \leq 1$

2.5 Survey

We conclude this introduction to Conformal Predictors with a survey of the field. Rather than providing a long list of bibliographic references which could be easily gleaned from a query on general-purpose search engine or on a specialised academic site (e.g. ResearchGate), we attempt here to provide a reasoned overview of themes and publications.

[11] By Bayes factor (Jeffreys, 1998, sec. 5)we denote

$$K = \frac{\frac{\Pr(H_0|D)}{\Pr(\bar{H}_0|D)}}{\frac{\Pr(H_0)}{\Pr(\bar{H}_0)}} = \frac{\Pr(D|H_0)}{\Pr(D|\bar{H}_0)}$$

This is also known as likelihood ratio. The odds of H_0 to $\bar{H}_0$ can be obtained by multiplying the Bayes factor by the ratio of the priors $\frac{\Pr(D|H_0)}{\Pr(D|\bar{H}_0)}$. A Bayes factor of 1 indicates that the data supports equally H_0 and H_1; the actual odds in that case corresponds to those determined by the priors.

Conformal Prediction can be viewed as part of the long tradition of nonparametric estimation (Wasserman, 2010a; Tsybakov, 2008). The reference text is the monograph "Algorithmic Learning in a Random World" published in 2005 (Vovk, Gammerman, and Shafer, 2005), which presents in a coherent whole the fundamental theoretical results gradually established by the authors in the previous decade. Among the papers that followed, (Gammerman and Vovk, 2007) summarized succinctly the key aspects of CP and was complemented by a very interesting discussion in which leading scholars provided their perspectives. In more recent times, (Balasubramanian, Ho, and Vovk, 2014) collected contributions from key researchers in the field and provided a thorough overview of the state-of-the-art as of 2015 for both CP theory and CP applications. To avoid repetitions of material and references covered in that book, this section will focus on the developments that occurred after 2014.

An increasing number of academic groups around the world appear to be involved in research on CP. The one with the longest tradition is arguably the Centre for Reliable Machine Learning[12], at Royal Holloway, University of London, where the founders of the field of Conformal Prediction, Prof. Alexander Gammerman and Prof. Vladimir Vovk, continue to lead the investigations, opening new avenues, such as the Conformal Predictive Distributions (Vovk et al., 2019). A significant body of research with acknowledged connections to CP have been made by scholars at Carnegie-Mellon University, often in collaboration with researchers from the University of Chicago and Stanford University. The focus is more on the batch mode (as opposed to the online mode) of operation and on regression. Among the many contributions, we highlight a method to "conformalize" the LASSO (Lei, 2019), a method to compensate for a form of deviation at test time from the i.i.d. assumption (namely, covariate shift) (Tibshirani et al., 2019), a way to exploit CP to make deep learning image classifiers more robust (Hechtlinger, Póczos, and Wasserman, 2018), a modification of Cross Conformal Predictors (Barber et al., 2019), a method to apply CP to quantile regression algorithms[13] algorithms (Romano, Patterson, and Candès, 2019) and investigations on the limits of distribution-free inference(Barber, 2020).

Another prolific group of researchers operates in Sweden, gravitating around the KTH Royal Institute of Technology in Stockholm, the Stockholm University, the Jönköping University, Uppsala University, and the University of Borås, often in connection with the Discovery Science department of the pharmaceutical company AstraZeneca. Restricting our survey to the last 5 years, the studies that communiity contributed focus on combination of CP (Linusson et al., 2017; Linusson, Johansson, and Boström, 2019), on the use of random forests as underlying ML method (Johansson et al., 2018; Johansson et al., 2019b; Vasiloudis, de Francisci Morales, and Boström, 2019), on interpretable conformal regression (Johansson et al., 2019a), and on several applications listed further down.

[12] https://cml.rhul.ac.uk/, known as Computer Learning Research Centre (CLRC) up to 2019

[13] quantile regression refers to conditional quantile functions, where quantile function appear to be another name for predictive distribution

Other centres of intense research activity include Fredrik University in Cyprus (under the guidance of Harris Papdopoulos) and Maastricht University in the Netherlands (Evgueni Smirnov). It is also surprising how many papers appear from disparate research institutions.

The main forum for the CP researchers is the yearly Symposium on "Conformal and Probabilistic Prediction with Applications" (COPA) which started in 2012. The papers presented at the symposium are published as Proceedings of Machine Learning Research[14]. The latest at the time of writing is PMLR Vol. 128 (COPA2020), whereas for earlier editions see for instance (Gammerman et al., 2016). CP has also been the focus of conferences, such as the 2015 "DST-EPSRC Indo-UK Workshop on Conformal Prediction and Applications" in Hyderabad and the 2015 "Statistical Learning and Data Sciences" Symposium in Egham, UK.

A number of journals have featured special issues on CP. These include: Annals of Mathematics and Artificial Intelligence (Papadopoulos, Vovk, and Gammerman, 2015; Gammerman and Vovk, 2017), Journal of Cheminformatics (Spjuth, 2018), Machine Learning (Gammerman et al., 2019b), Neurocomputing (Gammerman et al., 2019a), Pattern Recognition (to be published in 2020).

In recent years, CP has seen a plethora of applications. A large number of those can be grouped under the banner of life sciences. CP has been applied in neuropsychology to predict the progress of Alzheimer's Disease (Pereira et al., 2017; Pereira et al., 2018; Pereira et al., 2020), in a biomedical setting to predict lung cancer survival (Qaddoum, 2020) or breast cancer survivability (Alnemer, Rajab, and Aljarah, 2016) or detecting seizures (Eliades and Papadopoulos, 2018) or detecting lung cancer using a electronic nose (Zhan et al., 2020) in ecology to predict aquatic toxicity (Svensson and Norinder, 2020) but perhaps the lion's share of applications is in chemoinformatics (Spjuth, 2018) and, more specifically, drug discovery (Eklund et al., 2015; Ahlberg et al., 2017; Cortés-Ciriano and Bender, 2021; Bosc et al., 2019) and development, where CP is used in high-throughput screening (Toccaceli, Nouretdinov, and Gammerman, 2016; Svensson, Norinder, and Bender, 2017a; Sun et al., 2017a; Svensson et al., 2018a; Ahmed et al., 2018; Svensson et al., 2018b), toxicity studies (Svensson, Norinder, and Bender, 2017b; Ji et al., 2018; Morger et al., 2020), animal testing alternatives (Forreryd et al., 2018), proteochemometric studies (Cortés-Ciriano, Bender, and Malliavin, 2015).

There are in addition a variety of applications in surprisingly disparate fields, including nuclear fusion (Shabbir et al., 2015; Moreno et al., 2016), identification of maggots in forensics (Beyramysoltan et al., 2020), malware detection (Cherubin et al., 2015; Dash et al., 2016; Zhi et al., 2017), fairness in the justice system (Romano et al., 2019), classification of Chinese liquors using RAMAN spectra (Gu et al., 2019), recommender systems (Kagita et al., 2017; Ayyaz, Qamar, and Nawaz, 2018; Himabindu, Padmanabhan, and Pujari, 2018; Morsomme and Smirnov, 2019),

[14] http://proceedings.mlr.press/

detection of anomalous trajectories (Laxhammar and Falkman, 2015), classification of herbal medicines with an electronic nose (Zhan et al., 2018), predictive maintenance (Nouretdinov et al., 2019) and predictive monitoring of Hybrid Automata[15] (Bortolussi et al., 2019).

Applications of CP are not confined to research settings. Several companies are known to employ CP techniques in software that is used in "production" systems. Among them are AstraZeneca in Sweden, Janssen in Belgium, and Centrica in the UK.

[15]Hybrid Automata can be seen mathematical descriptions of devices that combine digital logic with analog processes (e.g. a cardiac pacemaker). They can be described as a finite state machine with a finite set of continuous variables, possibly governed by a system of differential equations.

Chapter 3

Application of CP to Compound Activity Prediction

3.1 Application to Compound Activity Prediction

To evaluate the performance of CP for Compound Activity Prediction in a realistic scenario, we sourced the data sets from a public-domain repository of High Throughput assays, PubChem BioAssay (Kim et al., 2019).

The data sets on PubChem identify a compound with its CID (a unique compound identifier that can be used to access the chemical data of the compound in another PubChem database) and provide the result of the assay as Active/Inactive as well as providing the actual measurements on which the result was derived, e.g. viability (percentage of cells alive) of the sample after exposure to the compound.

To apply machine learning techniques to this problem, the compounds must be described in terms of a number of numerical attributes. There are several approaches to do this. The approach that was followed in this study is to compute *signature descriptors* (Faulon, Visco, and Pophale, 2003). Each signature corresponds a given labelled subgraph in the molecule graph, with subgraphs limited to those with a given depth. In this exercise the signatures had at most height 3[1]. The signature descriptor for a molecule consists of the signatures present in the molecule along with their counts, i.e. the number of times the labelled subgraph of a signature occur in the graph of the molecule. An example of the signature descriptor for ascorbic acid (also known as vitamin C) is provided in Figure 3.1, Table 3.1, and Table 3.2.

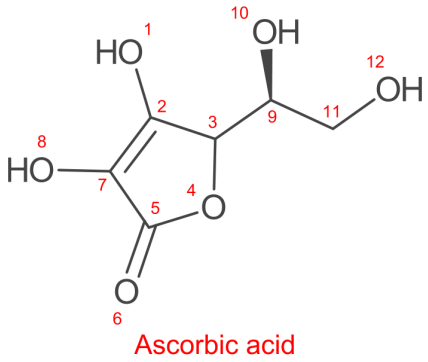


FIGURE 3.1: Chemical structure of *l*-ascorbic acid, commonly known as vitamin C. Consistently with convention in organic chemistry, carbon atoms and hydrogen atoms are not indicated as their presence can be easily inferred (carbon atoms are at every unlabelled vertex, hydrogen atoms are present wherever needed to saturate the valence of an atom). The numbering of the atoms in in this example is arbitrary.

To create a data set from a number of compounds, all the signatures in all compounds are first enumerated and the set of all signatures is obtained. Each unique signature corresponds to one attribute, hence one dimension of the data set. To build the matrix of training examples, each signature in the set is attributed (arbitrarily) a column index and each compound a row index. Each cell of the matrix contains the

[1]The signature descriptors and other types of descriptors (e.g. circular descriptors) can be computed with the CDK Java package or any of its adaptations such as the RCDK package for the R statistical software.

TABLE 3.1: Signatures for ascorbic acid.
For every "heavy" atom (i.e. every non-hydrogen atom), a labelled Directed Acyclic graph of given depth is computed and a string-based representation is provided. For instance, the string at row 7 and column 1 represents the DAG of depth 1 from atom 7, i.e. a carbon atom with a bond to a carbon atom, a double bond to another carbon atom and a bond to an oxygen atom.

	0	1	2	3
1	[O]	[O]([C])	[O]([C]([C]=[C]))	[O]([C]([C]([C][O])=[C]([C][O])))
2	[C]	[C]([C]=[C][O])	[C]([C]([C][O])=[C]([C][O])[O])	[C](=[C]([C](=[O][O,0])[O])[C]([C]([C][O])[O,0...
3	[C]	[C]([C][C][O])	[C]([C]([C][O])[C](=[C][O])[O]([C]))	[C]([C]([C]([O])[O])[C](=[C]([C,0][O])[O])[O](...
4	[O]	[O]([C][C])	[O]([C]([C][C])[C]([C]=[O]))	[O]([C]([C](=[C,0][O])=[O])[C]([C]([C][O])[C,0...
5	[C]	[C]([C][O]=[O])	[C]([C](=[C][O])=[O][O]([C]))	[C]([C](=[C]([C,0][O])[O])=[O][O]([C,0]([C])))
6	[O]	[O](=[C])	[O](=[C]([C][O]))	[O](=[C]([C](=[C][O])[O]([C])))
7	[C]	[C]([C]=[C][O])	[C](=[C]([C][O])[C]([O]=[O])[O])	[C]([C](=[O][O]([C,0]))=[C]([C,0]([C])[O])[O])
8	[O]	[O]([C])	[O]([C]([C]=[C]))	[O]([C](=[C]([C][O])[C]([O]=[O])))
9	[C]	[C]([C][C][O])	[C]([C]([O])[C]([C][O])[O])	[C]([C]([O])[C]([C](=[C][O])[O]([C]))[O])
10	[O]	[O]([C])	[O]([C]([C][C]))	[O]([C]([C]([O])[C]([C][O])))
11	[C]	[C]([C][O])	[C]([C]([C][O])[O])	[C]([C]([C]([C][O])[O])[O])
12	[O]	[O]([C])	[O]([C]([C]))	[O]([C]([C]([C][O])))

TABLE 3.2: Signature descriptor for ascorbic acid.
Each signature is used as a feature (a dimension); the number of occurrences is the value of the feature.

Counts	Signature
6	[C]
6	[O]
4	[O]([C])
2	[C]([C]=[C][O])
2	[C]([C][C][O])
2	[O]([C]([C]=[C]))
1	[C](=[C]([C](=[O][O,0])[O])[C]([C]([C][O])[O,0])[O])
1	[C](=[C]([C][O])[C]([O]=[O])[O])
1	[C]([C](=[C]([C,0][O])[O])=[O][O]([C,0]([C])))
1	[C]([C](=[C][O])=[O][O]([C]))
1	[C]([C](=[O][O]([C,0]))=[C]([C,0]([C])[O])[O])
1	[C]([C]([C]([C][O])[O])[O])
1	[C]([C]([C]([O])[O])[C](=[C]([C,0][O])[O])[O]([C,0](=[O])))
1	[C]([C]([C][O])=[C]([C][O])[O])
1	[C]([C]([C][O])[C](=[C][O])[O]([C]))
1	[C]([C]([C][O])[O])
1	[C]([C]([O])[C]([C](=[C][O])[O]([C]))[O])
1	[C]([C]([O])[C]([C][O])[O])
1	[C]([C][O])
1	[C]([C][O]=[O])
1	[O](=[C]([C](=[C][O])[O]([C])))
1	[O](=[C]([C][O]))
1	[O](=[C])
1	[O]([C](=[C]([C][O])[C]([O]=[O])))
1	[O]([C]([C](=[C,0][O])=[O])[C]([C]([C][O])[C,0]([O])))
1	[O]([C]([C]([C][O]))))
1	[O]([C]([C]([C][O])=[C]([C][O])))
1	[O]([C]([C]([O])[C]([C][O])))
1	[O]([C]([C]))
1	[O]([C]([C][C]))
1	[O]([C]([C][C])[C]([C]=[O]))
1	[O]([C][C])

count of the occurrences of the signature corresponding to the column in the compound corresponding to the row. The resulting matrix can be very large but it is also highly sparse, as detailed further on (see Tab. 3.5).
In summary, the problem of Compound Activity Prediction is framed here as a classification problem where the examples are pairs of

- a label y taking values in $Y = \{-1, +1\}$
- an object described by a vector of non-negative integers $(x_1, \ldots, x_k) \in \mathbb{N}_0^k$, where k is the number of all the signatures encountered in the data set and x_i is the count of occurrences in the given compound of the i-th signature

and we are interested in predicting the label of a test object given its vector of descriptor counts.

As a final introductory consideration, it is important to mention that compound activity prediction is a hard ML problem. The data sets can have several hundred thousand examples (compounds for which a measurement is available), but only a tiny minority (as low as 0.1%) exhibits activity. Each compound is described by a large, sparse set of features (each compound generally possesses only a fraction (30-300) of the total number of features, which could be a few hundred thousand). More importantly, biological data is inherently noisy and often censored (values are clipped to a fixed range) because of experimental limitations. Measurements could also come from different laboratories with different procedures and consequently different error distributions. As a result, the predictive performance is very often not as good as for many other common ML applications. As a reference, (Sturm et al., 2020) seems to consider ROC AUC [2] greater than 0.7 or F1-score [3] greater than 0.4 as a good result.

3.1.1 Underlying algorithms

As a first step in the study, we set out to extract relevant non-conformity measures from different underlying algorithms: Support Vector Machines (SVM) (Cortes and Vapnik, 1995), Nearest Neighbours, Naïve Bayes. The Non Conformity Measures for each of the three underlying algorithms are listed in Tab. 3.3.

There are a number of considerations arising from the application of each of these algorithms to Compound Activity Prediction.

SVM. The usage of SVM in this domain poses a number of challenges. First of all, the number of training examples was large enough to create a problem for our computational resources. The scaling of SVM (or kernel methods in general) to large data sets is indeed an active research area (Bottou et al., 2007; Chang, 2011; You et

[2] Receiver Operating Characteristic Area Under Curve
[3] The F1-score is the harmonic mean of Precision and Recall

TABLE 3.3: The Non Conformity Measures for the three underlying algorithms

Underlying	**Non Conformity Measure** α_i	**Comment**
SVM	$-y_i d(x_i)$	(signed) distance from separating hyperplane
kNN	$\frac{\sum_{j\neq i:y_j=y_i}^{(k)} d(x_j,x_i)}{\sum_{j\neq i:y_j\neq y_i}^{(k)} d(x_j,x_i)}$	here the summation is on the k smallest values of $d(x_j,x_i)$
Naïve Bayes	$-\log\ p(y_i=c\|x_i)$	p is the posterior probability estimated by Naïve Bayes

al., 2015), especially in the case of non-linear kernels[4]. We turned our attention to a simple approach proposed by V.Vapnik in (Graf et al., 2005), called Cascade SVM.

The sizes of the training sets considered here are too large to be handled comfortably by generally available SVM implementations, such as `libsvm` (Chang and Lin, 2011). The approach we follow could be construed as a form of *training set editing*. Vapnik proved formally that it is possible to decompose the training into an n-ary tree of SVM trainings. The first layer of SVMs is trained on training sets obtained as a partition of the overall training set. Each SVM in the first layer outputs its set of support vectors (SVs) which is generally smaller than the training set. In the second layer, each SVM takes as training set the merging of n of the SVs sets found in the first layer. Each layer requires fewer SVMs. The process is repeated until a layer requires only one SVM. The set of SVs emerging from the last layer is not necessarily the same that would be obtained by training on the whole set (but it is often a good approximation). If one wants to obtain that set, the whole training tree should be executed again, but this time the SVs obtained at the last layer would be merged into each of the initial training blocks. A new set of SVs would then be obtained at the end of the tree of SVMs. If this new set is the same as the one in the previous iteration, this is the desired set. If not, the process is repeated once more. In (Graf et al., 2005) it was proved that the process converges and that it converges to the same set of SVs that one would obtain by training on the whole training set in one go.

To give an intuitive justification, the fundamental observation is that the SVM decision function is entirely defined just by the Support Vectors. It is as if these examples contained all the information necessary for the classification. Moreover, if we had a training set composed only of the SVs, we would have obtained the same

[4]In the case of linear SVM, it is possible to tackle the formulation of the quadratic optimization problem at the heart of the SVM in the primal and solve it with techniques such as Stochastic Gradient Descent or L-BFGS, which lend themselves well to being distributed across an array of computational nodes.

decision function. So, one might as well remove the non-SVs altogether from the training set.

In experiments discussed here, we followed a simplified approach. Instead of a tree of SVMs, we opted for a linear arrangement as shown in Fig.3.2.

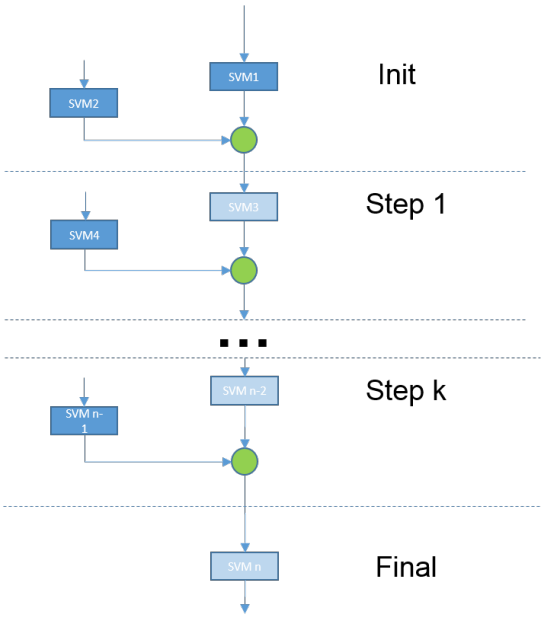


FIGURE 3.2: Linear Cascade SVM
At each step, the set of Support Vectors from the previous stage is merged with a block of training examples from the partition of the original training set. This is used as training set for an SVM, whose SVs are then fed to the next stage.

While we have no theoretical support for this semi-online variant of the Cascade SVM, the method appears to work satisfactorily in practice on the data sets we used.

The class imbalance was addressed with the use of per-class weighting of the C hyperparameter, which results in a different penalization of the margin violations. The per-class weight was set inversely proportional to the class representation in the training set.

Another problem is the choice of an appropriate kernel. While we appreciated the computational advantages of linear SVM, we also believed that it was not necessarily the best choice for the specific problem. It can easily be observed that the nature of the representation of the training objects (as discrete features) warranted approaches similar to those used in Information Retrieval, where objects are described in terms of occurrences of patterns (bags of words). The topic of similarity searching in chemistry is an active one and there are many alternative proposals (see (Monev, 2004)). We used as a kernel a notion called Tanimoto similarity[5]. The Tanimoto similarity extends the well-known Jaccard coefficient in the sense that whereas

[5]See (Gartner, 2008) for a proof that Tanimoto Similarity is a kernel.

the Jaccard coefficient considers only presence or absence of a pattern, the Tanimoto similarity takes into account the counts of the occurrences.

To explore further the benefits of non-linear kernels, we also tried out a kernel consisting of the composition the Tanimoto similarity with Gaussian RBF.

Tab. 3.4 provides the definitions of the kernels used in this study.

TABLE 3.4: SVM Kernels Definitions (where $A = (a_1, \ldots, a_d), B = (b_1, \ldots, b_d)$ are two objects, each described by a vector of d counts))

Tanimoto Coefficient	$T(A,B) = \frac{\sum_{i=1}^{d} \min(a_i,b_i)}{\sum_{i=1}^{d}(a_i+b_i) - \sum_{i=1}^{d} \min(a_i,b_i)}$
Tanimoto with Gaussian RBF	$TG(A,B) = e^{-\frac{\lvert T(A,A)+T(B,B)-2T(A,B)\rvert}{\gamma}}$

Naïve Bayes.

Naïve Bayes and more specifically Multinomial Naïve Bayes are widely regarded as effective classifiers when features are discrete (for instance, in text classification), despite their relative simplicity. This made Multinomial Naïve Bayes a natural choice for the problem at issue here. The probabilities were estimated using additive smoothing to avoid the problem of zero probabilities.

Nearest Neighbours

We chose Nearest Neighbours because of its good performance in a wide variety of domains. In principle, the performance of Nearest Neighbours could be severely affected by the high-dimensionality of the training set (Tab. 3.5 shows how in one of the data sets used in this study the number of attributes exceeds by $\approx$ 20% the number of examples), but in our experiments the "curse of dimensionality" did not manifest itself.

3.1.2 Tools and Computational Resources

The choice of the tools for these experiments was influenced primarily by the exploratory nature of this work. For this reason, programming languages and development environments were chosen for their ability to support interactivity and rapid prototyping, rather than optimal CPU utilization and memory efficiency.

The language adopted was Python 3.4 and the majority of programming was done using Jupyter Notebooks with the IPython kernel. The overall format turned out to be very effective for capturing results (and for their future reproducibility).

Several third-party libraries were used. Numerical processing and data management were performed with the help of `numpy/scipy` and `pandas`. SVM and other machine learning algorithms were provided by the `scikit-learn`(Pedregosa et al., 2011) package. Sections of Python code (e.g. the Tanimoto similarity) that were identified through profiling as critical for performance were re-implemented in `Cython`, a subset/dialect of Python that allows code to be compiled (via an intermediate C representation) rather than being interpreted. The computations were run initially on a local server (8 cores with 32GB of RAM, running OpenSuSE) and in later stages on a supercomputer (the IT4I Salomon cluster located in Ostrava, Czech Republic).The Salomon cluster is based on the SGI® ICE™ X system and comprises 1008 computational nodes (plus a number of login nodes), each with 24 cores (2 $\times$ 12-core Intel Xeon E5-2680v3 2.5GHz processors) and 128GB RAM, connected via high-speed 7D Enhanced hypercube InfiniBand FDR and Ethernet networks. Rated at 1,457.7 TFLOPS, at the time of its launch it ranked 14$^{\text{th}}$ in Europe[6].

Parallelization and computation distribution was accomplished with the help of the `ipyparallel`(Bussonnier, 2018) package, which is a high-level framework for the coordination of remote execution of Python functions on a generic collection of nodes (cores or separate servers). It provides a convenient environment for distributed computing which is well integrated with IPython and Jupyter and has a learning curve that is not as steep as that of alternative frameworks common in High Performance Computing (OpenMPI, for example), which offers a degree of fine-grained control that was not required in this application. In particular, `ipyparallel`, in addition to allowing the start-up and shut-down of a cluster comprising a controller and a number of engines where the actual processing (each is a separate process running a Python interpreter) is performed via integration with the job scheduling infrastructure present on Salomon (PBS, Portable Batch System), took care of the details such as data serialization/deserialization and transfer, load balancing, job tracking, exception propagation, etc. thereby hiding much of the complexity of parallelization. One key characteristic of `ipyparallel` is that, while it provides primitives for `map()` and `reduce()`, it does not constrain the choice to those two, leaving the implementer free to select the most appropriate parallel programming design patterns for the specific problem (see (McCool, Reinders, and Robison, 2012) for a reference on the subject).

In this work, parallelization was exploited to speed up the computation of the Gram matrix or of the decision function for the SVMs or the matrix of distances for kNN. In either case, the overall task was partitioned in smaller chunks that were then assigned to engines, which would then asynchronously return the result. Also, parallelization was used for SVM cross-validation, but at a coarser granularity, i.e. one engine per SVM training with a parameter. Data transfers were minimized by making use of shared memory where possible and appropriate. A key speed-up was

[6] in the latest top500.org global list of supercomputers (November 2020) it ranks at number 460.

achieved by using pre-computed kernels (computed once only) when performing Cross-Validation with respect to the regularization parameter C.

3.1.3 Results

To assess the relative merits of the different underlying algorithms, we applied Inductive Mondrian Conformal Predictors on the PubChem data set AID827[7]. The extraction of signature descriptors from the molecular structures were kindly performed on our behalf by Lars Carlsson and Ernst Ahlberg-Helgee at AstraZeneca. The characteristics of the resulting data set are listed in Tab. 3.5.

TABLE 3.5: Characteristics of the AID827 data set

Total number of examples	138,287	
Number of features	165,786	High dimensionality
Number of non-zero entries	7,711,571	
Density of the data set	0.034%	High sparsity
Active compounds	1,658	High imbalance (1.2%)
Inactive compounds	136,629	
Unique set of signatures	137,901	Low degeneracy

The test was articulated in 20 cycles of training and evaluation. In each cycle, a test set of 10,000 examples was extracted at random. The remaining examples were split randomly into a proper training set of 100,000 examples and a calibration set with the balance of the examples (28,387).

During the SVM training, 5-fold stratified Cross Validation was performed at every stage of the Cascade to select an optimal value for the hyperparameter C. Also, per-class weights were assigned to cater for the high class imbalance in the data, so that a higher penalization was applied to violators in the less represented class.

In Multinomial Naïve Bayes too, Cross Validation was used to choose an optimal value for the smoothing parameter.

The results are listed in Tab. 3.6, which presents the classification arising from the region predictor for $\epsilon = 0.01$. The numbers are averages over the 20 cycles of training and testing.

Note that a compound is classified as Active (resp. Inactive) if and only if Active (resp. Inactive) is the only label in the prediction set. When both labels are in the prediction, the prediction is considered Uncertain.

It has to be noted at this stage that there does not seem to be an established consensus on what the best performance criteria are in the domain of Compound Activity Prediction (see for instance (Jain and Nicholls, 2008)), although *Precision*

[7] Available at https://pubchem.ncbi.nlm.nih.gov/bioassay/827 – Last accessed January 3, 2021

TABLE 3.6: CP results for AID827 with significance $\epsilon = 0.01$. All results are averages over 20 runs, using the same test sets of 10,000 objects across the different underlying algorithms. "Active pred Active" is the (average) count of actually Active test examples that were predicted Active by Conformal Prediction. Uncertain predictions occur when both labels are output by the region predictor. Empty predictions occur when both labels can be rejected at the chosen significance level. For the specific significance level chosen here, there were never empty predictions.

Underlying	Active pred Active	Inactive pred Active	Inactive pred Inactive	Active pred Inactive	Empty pred	Uncertain
Naïve Bayes	38.20	104.30	183.30	1.10	0	9673.10
3NN	43.95	100.55	361.55	0.80	0	9493.15
Cascade SVM:						
- linear	34.20	99.00	591.85	1.20	0	9273.75
- RBF kernel	47.20	101.80	1126.75	1.80	0	8722.45
- Tanimoto kernel	48.45	97.65	986.85	0.80	0	8866.25
- Tanimoto-RBF kernel	47.65	94.10	1044.90	0.95	0	8812.40

(fraction of actual Actives among compounds predicted as Active) and *Recall* (fraction of all the Active compounds that are among those predicted as Active) seem to be generally relevant. In addition, it is worth pointing out that these (and many others) criteria of performance should be considered as generalisations of classical performance criteria since they include dependence of the results on the required confidence level.

TABLE 3.7: CP results for AID827 using SVM with Tanimoto+RBF kernel for different significance levels. The "Active Error Rate" is the ratio of "Active pred Inactive" to the total number of Active test examples. The "Inactive Error Rate" is the ratio of "Inactive predicted Active" to the total number of Inactive test examples.

Significance	Active pred Active	Inactive pred Active	Inactive pred Inactive	Active pred Inactive	Empty pred	Uncertain	Active Error Rate	Inactive Error Rate
1%	47.65	94.10	1044.90	0.95	0.0	8812.40	0.82%	0.95%
5%	67.20	490.40	3091.75	5.20	0.0	6345.45	4.52%	4.96%
10%	76.15	999.25	4703.75	10.60	0.0	4210.25	9.22%	10.11%
15%	82.10	1484.85	6021.80	17.30	0.0	2393.95	15.04%	15.02%
20%	86.55	1982.25	6928.95	22.80	0.0	979.45	19.83%	20.05%

At the shown significance level of $\epsilon = 0.01$, 34% of the compounds predicted as active by Inductive Mondrian Conformal Prediction using Tanimoto composed with Gaussian RBF were actually Active compared to a prevalence of Actives in the data set of just 1.2%. At the same time, the Recall was $\approx 41\%$ (ratio of Actives in the prediction to total Actives in the test set). Note that this is for a specific value of ϵ. In Section 3.1.5 we will discuss how a different trade-off between Precision and Recall

can chosen by varying ϵ[8].

We selected Cascade SVM with Tanimoto+RBF as the most promising underlying algorithm on the basis of the combination of its high Recall (for Actives) and high Precision (for Actives), assuming that the intended application is indeed to output a selection of compounds that has a high prevalence of Active compounds.

Note that in Tab. 3.6 the values similar to ones of confusion matrix are calculated only for certain predictions. In this representation, the concrete meaning of the property of class-based validity can be clearly illustrated as in Tab. 3.7: the two rightmost columns report the prediction error rate for each label, where by prediction error we mean the occurrence of "the actual label not being in the predictions set". When there are no Empty predictions, the Active Error rate is the ratio of the number of "Active predicted Inactive" to the number of Active examples in the test set (which was 115 on average).

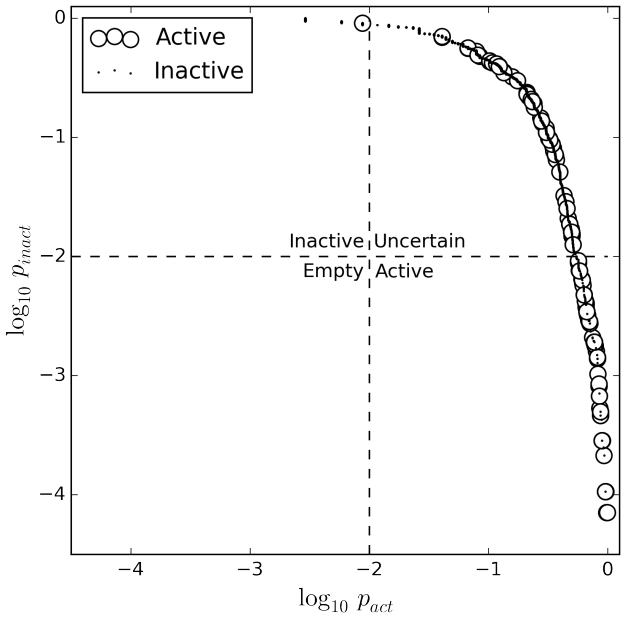
Fig. 3.3 shows the test objects according to the base-10 logarithm of their p_{active} and $p_{inactive}$. The dashed lines represent the thresholds for p-value set at 0.01, i.e. the significance value ϵ used in Tab. 3.6. The two dashed lines partition the plane in 4 regions, corresponding to the region prediction being Active ($p_{active} > \epsilon$ and $p_{inactive} \leq \epsilon$), Inactive ($p_{active} \leq \epsilon$ and $p_{inactive} > \epsilon$), Empty ($p_{active} \leq \epsilon$ and $p_{inactive} \leq \epsilon$), Uncertain ($p_{active} > \epsilon$ and $p_{inactive} > \epsilon$).

As suggested in Sec. 2.4.8, it is also possible to use the CP p-values to emit single-valued predictions. In this scheme, the prediction is the label with the highest p-value and it is complemented by *confidence* and *credibility*.

It can be argued that Conformal Prediction provides, in addition to the validity guarantees, a more informative output than the mere score produced by the underlying algorithm. We have seen that, in the binary classification case we're considering, the CP framework provides two p-values, one for the Active hypothesis and one for the Inactive hypothesis. Where the conventional interpretation of the score of Active vs. Inactive, with CP we obtain a estimate of how consistent with the training set each hypothesis is. Instead of framing the prediction as a choice between Active or Inactive, we are presented with the support for each label and this can provide more insight. Suppose, for instance, that a probabilistic predictor output the same probability for both labels and that, correspondingly, the CP might output the same p-value. While on the basis of the probabilistic prediction we would consider the two possibilities entirely equivalent, the CP output can reveal more: if both p-values are high, we could conclude that both labels are compatible with the training data, but if p-values are low, we would have an indication that both are hardly in line with the training data. This distinction is not possible by looking just at the output of conventional probabilistic predictors.

[8] In principle, the control of the trade-off can be achieved in scoring classifiers by varying the decision threshold. By using ϵ, the CP approach follows arguably a more principled approach, where ϵ has a definite statistical meaning.

FIGURE 3.3: Test objects plotted by the base-10 log of their their p_{active} and $p_{inactive}$. Note that many test objects are overlapping. Note that some of the examples may have identical p-values, so for example 1135 objects predicted as "Inactives" are presented as 4 points on this plot.



3.1.4 Application to different data sets

The previous section examined different underlying ML algorithms and obtained a variety of performance results, which seemed to suggest that SVM using Tanimoto+RBF kernel provides the best results. In this section, we apply Mondrian ICP with SVM with Tanimoto+RBF to different data sets to gain some insight on the range of performance that can be expected from this technique when applied to chemoinformatics data sets. Again, we are indebted to Lars Carlsson and Ernst Ahlberg-Helgee for assisting us in the choice of appropriate PubChem data sets and for extracting the signature descriptors for us. The main characteristics of the data sets are reported in Tab. 3.8.

In line with the previous set of experiments, we performed 20 iterations of training and testing, each with a different random partitioning of the overall data set into training and test sets. In each cycle, a test set of 10,000 examples was hold out and the rest was split between calibration set ($\approx 30,000$) and proper training set. The results, averaged over the 20 iterations, are reported in Tab. 3.9 for significance level $\epsilon = 0.01$. It can be observed that the proportion of errors (i.e. 'Inactive pred Active','Active pred Inactive', and 'Empty preds') is close to 1% for all data sets in line with the theory, whereas the proportion of correct, unambiguous predictions varies markedly.

TABLE 3.8: Data sets and their characteristics. Density refers to the percentage of non-zero entries in the full matrix of 'Number of Compounds × Number of Features' elements

Data Set	Assay Description	Number of Compounds	Number of Features	Actives (%)	Density (%)
827	High Throughput Screen to Identify Compounds that Suppress the Growth of Cells with a Deletion of the PTEN Tumor Suppressor.	138,287	165,786	1.2%	0.034%
1461	qHTS Assay for Antagonists of the Neuropeptide S Receptor: cAMP Signal Transduction.	208,069	211,474	1.11%	0.026%
1974	Fluorescence polarization-based counterscreen for RBBP9 inhibitors: primary biochemical high throughput screening assay to identify inhibitors of the oxidoreductase glutathione S-transferase omega 1(GSTO1).	302,310	237,837	1.05%	0.024%
2553	High throughput screening of inhibitors of transient receptor potential cation channel C6 (TRPC6)	305,308	236,508	1.06%	0.024%
2716	Luminescence Microorganism Primary HTS to Identify Inhibitors of the SUMOylation Pathway Using a Temperature Sensitive Growth Reversal Mutant Mot1-301	298,996	237,811	1.02%	0.024%

TABLE 3.9: Results of the application of Mondrian ICP with $\epsilon = 0.01$ using SVM with Tanimoto+RBF as underlying. Test set size: 10,000

DataSet	Active pred Active	Inactive pred Active	Inactive pred Inactive	Active pred Inactive	Empty preds	Uncertain
827	47.65	94.10	1044.90	0.95	0	8812.40
1461	29.45	101.30	1891.10	1.20	0	7976.95
1974	62.50	97.40	880.85	1.00	0	8958.25
2553	34.00	101.00	337.90	1.00	0	9526.10
2716	3.55	98.20	97.00	1.00	0	9800.25

3.1.5 Mondrian ICP with different ϵ_{active} and $\epsilon_{inactive}$

When applying class-conditional Mondrian ICP, there is no constraint to use the same significance ϵ for the two labels. There may be an advantage in allowing different "error" rates for the two labels given that the focus might be in identifying Actives rather than Inactives.

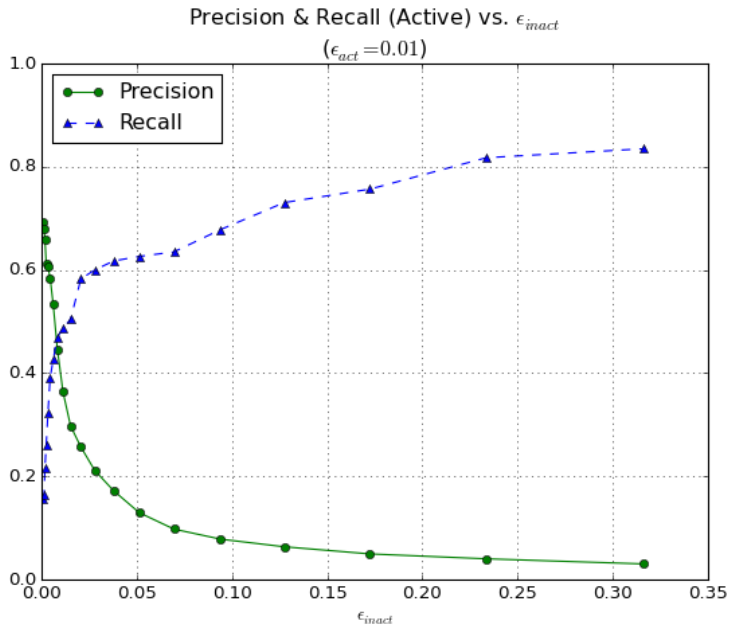
This allows to vary relative importance of the two kinds of errors. The validity of Mondrian machines implies that, for binary classification, the expected number of certain (i.e. just one label in the prediction set) predictions that turn out to be incorrect predictions is bounded by ϵ_{act} for (true) actives and by ϵ_{inact} for (true) non-actives. It is interesting to study the effect of varying the significance level on the precision and recall (within single-valued predictions).

Fig. 3.4 shows the trade-off between Precision and Recall that results from varying ϵ_{inact}. For very low values of the significance ϵ, a large number of test examples have $p_{act} > \epsilon_{act}$ as well as $p_{inact} > \epsilon_{inact}$. For these test examples, we have an 'Uncertain' prediction. As we increase ϵ_{inact}, fewer examples have a p_{inact} larger than ϵ_{inact}. So 'Inactive' is not chosen any longer as a label for those examples. If they happen to have a $p_{act} > \epsilon_{act}$, they switch from 'Uncertain' to being predicted as 'Active' (in the other case, they would become 'Empty' predictions).

3.2 Ranking of compounds by p-value

The p-value produced by a CP offers another way to rank test objects. In the case at hand, one might want to rank test compounds by how likely they are to exhibit the biological activity of interest. When using a scoring classifier as underlying model, one might observe that, if we denote the score by s (assuming higher score values are associated with higher probability of the object belonging to the Active class) the ordering will be the same when sorting by descending s as by descending p_{active}. The calculation of the p-value can in fact be seen as a isotonic mapping, i.e. a mapping that does not change the order, discounting ties. However, the p-values can

FIGURE 3.4: Trade-off between Precision and Recall by varying ϵ_{inact} — Data set AID827, with SVM with Tanimoto+RBF



convey an element of information that the bare scores do not provide. The validity property of class-conditional CP ensures that the objects with $p_{\text{active}} > \epsilon$ contain a fraction $1 - \epsilon$ of all the actives in the test set (barring, as usual, statistical fluctuation). An example of the ranking that can be obtained in this way is illustrated in Table 3.10. The table lists the 20 compounds that were assigned the largest p_{active} values in one of the runs for data set 827. In this example, if we choose $\epsilon = 0.90$, we would select the top 18 compounds. Among these 18 compounds, we would expect to find a fraction $1 - 0.90 = 0.10$ of all the actives in the test set. Given that the test set had 10,000 objects and that it was sampled from a data set with 1.2% of active compounds (see Table 3.8), among the 18 compounds we should see 12 active ones. In this specific instance, there are only 10, which is arguably within statistical fluctuation. In summary, the use of ranking by p-value can be of help in deciding which (and how many) compounds to test in order to find a chosen proportion of active compounds within a given library.

3.3 Conclusions

This chapter illustrated a methodology for applying conformal prediction to data sets from the chemoinformatics domain, characterized by large volumes, high imbalance, and sparseness. First, we verified the class-conditional validity guarantee of Mondrian Inductive Conformal Predictors using different underlying methods such as Nearest Neighbours, Naïve Bayes, and SVM with various kernels. In particular,

TABLE 3.10: The 20 candidate compounds (out of a held-out test set of 10,000) with the largest p_{active} values, from one of the runs for the data set AID827. Bold face denotes those compounds that are actually Active. The compound ID is just the index in the pre-processed data set and not the PubChem ID.

Ranking	Compound ID	p_{active}
1	**108198**	**0.997067**
2	103948	0.988270
3	62772	0.988270
4	129143	0.982405
5	**108632**	**0.961877**
6	**138051**	**0.961877**
7	**108920**	**0.941349**
8	**108877**	**0.938416**
9	**108783**	**0.932551**
10	**107957**	**0.932551**
11	5413	0.926686
12	4334	0.923754
13	**138177**	**0.923754**
14	71538	0.914956
15	54806	0.914956
16	16925	0.903226
17	**108026**	**0.903226**
18	**108584**	**0.900293**
19	**107943**	**0.894428**
20	**108032**	**0.894428**

we proposed a variation of the SVM method, namely Linear CascadeSVM, which overcomes the scalability limits of SVM at the cost of providing an approximate solution. The method appeared nonetheless to achieve the best efficiency in comparison to the other methods, when the Tanimoto+RBF kernel was used. The same method was then used in the application of MICP to a few other chemoinformatics sets to explore the range of performance that could be expected of the conformal approach in this particular domain. A discussion of the additional insight that can be gained from the p-values compared to bare scores concluded the chapter. As a final note, we would like to mention that the results presented here convinced the two pharmaceutical partners in the ExCAPE project (see Section 5.3) to integrate CP in their operational prediction systems. The improved quality of the predictions translated into savings (fewer lab tests) quantified in the range of tens of thousands of Euros (personal communication).

Chapter 4

Venn Predictors

4.1 Probabilistic Prediction

The previous chapters described Conformal Prediction and discussed some results of its application to BioAssay data. This chapter introduces Multi-probabilistic prediction, also referred to as Venn prediction.

In this section, we present the main concepts and their rationale. The reader is referred to the many publications for all the details. The treatment in this chapter follows (Vovk, Gammerman, and Shafer, 2005, Ch. 6).

As observed in Section 2.4.9, the p-value answers a different question from the one that is generally posed. What one is usually intuitively interested in is the probability of the label of a given test object taking a certain value. The $p_{\bar{y}}$ output by CP for a test object $x_{\ell+1}$ is instead the probability of drawing, from the same distribution as the calibration set, one example that is as or more contrary to hypothesis of randomness than $(x_{\ell+1}, \bar{y})$. Figure 2.9 illustrates in an empirical way the conceptual difference between p-value and the conditional probability of the label given the object. The latter is the subject of Probabilistic Prediction.

In this work Probabilistic prediction refers to the the task of producing a probability distribution over the label space $\mathbf{Y}$ for a given test object x_{n+1} on the basis of a training set $(x_1, y_1), \ldots, (x_n, y_n)$. In particular, we seek *valid* estimates of the actual probability distribution $\Pr(\mathbf{Y}|\mathbf{X})$: in this specific context, we consider probabilistic prediction valid is they perform well in statistical tests against the actual labels of the test examples.

In (Vovk, Gammerman, and Shafer, 2005) it is claimed that, under the assumption that the training set contains no repeated object, there cannot be a valid probability predictor. Venn predictors provide a way to circumvent this negative result.

4.1.1 Venn Predictors

It is perhaps fair to say that the more conventional and established methods for probabilistic prediction follow a parametric and possibly Bayesian approach. Such methods hinge on assumptions on the form of distribution and/or posit a prior distribution probability, which gets revised on the basis of the observations. By contrast, the approach followed by Venn Prediction is entirely non-parametric. Either approach has advantages and limitations. On one hand, Bayesian and parametric methods are particularly suited when the data generating mechanisms are known intimately enough for the assumptions to be justified. When the assumptions are warranted, such methods can produce accurate results with relatively small volumes of data. On the other hand, non-parametric methods seem to be more appropriate when data is abundant, but little can be reliably assumed on the form of the data distribution.

The concepts behind Venn Prediction can be traced all the way back to the foundations of the frequentist school, which views probability as the limit of a relative frequency. The practical application of this apparently elementary prescription leads very quickly to a difficulty, which is referred to as the reference class problem. John

Venn [1834–1923] was a logician and philosopher who, in addition to introducing the diagram that bears his name, is also credited for formulating explicitly and studying the issue (Venn, 1866, p.175). The "reference class" designates the class with respect to which we compute the relative frequency, which we then take as estimate of the probability. In his original treatment of the subject, Venn used the example of estimating the probability that John Smith, an Englishman, will die within one year. One estimate might simply to take the fraction of Englishmen that will die within a year, based on historical statistics. But John Smith is also a farmer, so we may get a more precise estimate by considering the fraction of English farmers that will die within a year. We are then told that he's also 30 years of age, so we could restrict further the reference class, and so forth. It is immediately evident that we cannot go on indefinitely as we would end up restricting the reference class to John Smith himself. There is therefore a trade-off between making the estimate specific and having a reference class with enough elements for the relative frequency to be a useful estimate of the probability. When we refine the reference class, we can only choose a finite number of attributes and this, in turn, creates the problem of selecting the most relevant ones.

Venn Prediction has three distinctive characteristics:

1. it clearly isolates the reference class problem within a notion of taxonomy to be defined as appropriate, by which the training set is partitioned into categories containing observations that can be considered similar for the purpose of computing probabilities as relative frequencies.

2. For each test object, it outputs a set of probability distributions, as opposed to one probability distribution.

3. It has a special form of a calibration property that applies for any choice of the taxonomy (under minimal conditions)

The taxonomy in item 1 above can be implemented by means of a ML algorithm, which would define an equivalence relation on a data set in a way that would minimize an appropriate loss function, thereby providing a principled way to tackle the reference class problem. Item 2 may be surprising and confusing and is introduced to avoid the technical issue of the impossibility of a valid probabilistic predictor mentioned at the end of the previous section. The Venn Predictor is a multi-probabilistic predictor and outputs as many probability distributions as possible label values. Each probability distribution provides the probability for each label. In (Vovk, Gammerman, and Shafer, 2005) it is proved that the Venn Predictor has a calibration property (by which the predicted probabilities correspond to the relative frequencies) that holds if one could choose the "right" distribution for each test object in turn.

4.1.2 Formal definition

In this section, we define the Venn predictors in general terms. We will restrict the scope to the case of classification, i.e.

- a *training set* made of ℓ *examples* $(x_i, y_i) \in \mathbf{Z}$, where x_i is an *object* generally represented as a vector of d *attributes* and $y_i \in \mathbf{Y}$ is a *label* taking a finite number k of values, say $\mathbf{Y} := \{c_1, \ldots, c_k\}$.
- a *test set* made of objects $x_{\ell+1}, x_{\ell+2}, \ldots$, whose labels we are asked to predict.

Venn taxonomy: A measurable function that assigns to each $n \in \{2, 3, \ldots\}$ and each sequence $(z_1, \ldots, z_n) \in \mathbf{Z}^n$ an equivalence relation $\sim$ on $\{1, \ldots, n\}$ which is equivariant in the sense that, for each n and each permutation π of $\{1, \ldots, n\}$,

$$\left(i \sim j \mid z_1, \ldots, z_n\right) \Rightarrow \left(\pi(i) \sim \pi(j) \mid z_{\pi(1)}, \ldots, z_{\pi(n)}\right)$$

The Venn taxonomy induces equivalence classes which we can define as:

$$A\left(j \mid z_1, \ldots, z_n\right) := \left\{i \in \{1, \ldots, n\} \mid \left(i \sim j \mid z_1, \ldots, z_n\right)\right\}$$

Intuitively, the equivalence classes of the taxonomy group objects that we consider sufficiently similar for the purposes of prediction. For instance, they could correspond to intervals of the value of the decision function, when using an SVM.

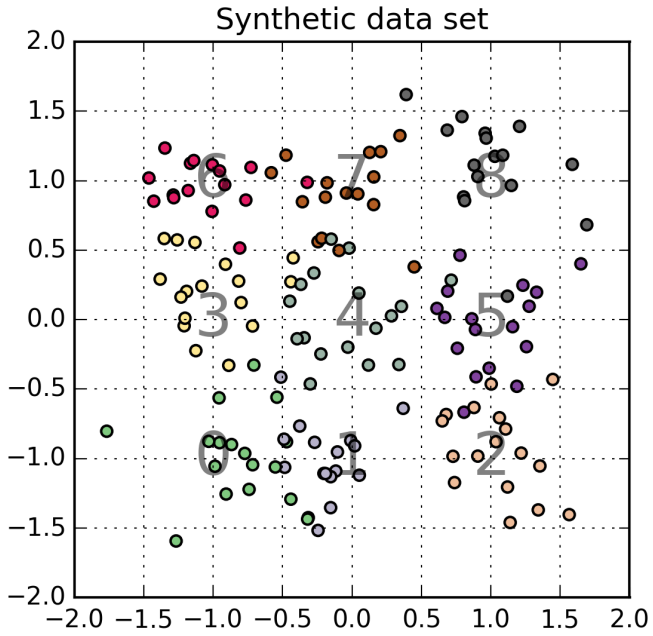
Venn predictor: Given a training sequence $(z_1, \ldots, z_\ell)$ and a test object $x_{\ell+1}$, the Venn predictor associated with a given Venn taxonomy outputs the probabilities $p_{i,j}$ as its prediction for $x_{\ell+1}$'s label, where

$$p_{i,j} = \frac{\left|\left\{k \in A(\ell+1 \mid z_1, \ldots, z_\ell, (x_{\ell+1}, c_i)) \mid y_i = c_j\right\}\right|}{\left|A(\ell+1 \mid z_1, \ldots, z_\ell, (x_{\ell+1}, c_i)\right|} \tag{4.1}$$

for $i, j \in \{1, \ldots, p\}$

In words, $p_{i,j}$ is the fraction of the examples with label c_j (including a hypothetical example made by assuming label c_i for the test object $x_{\ell+1}$) in the same equivalence class as the completion $(x_{\ell+1}, c_i)$. The approach described above can be viewed as *transductive* in the sense that we recompute the taxonomy from scratch for every test example (which implies retraining the underlying ML on which basis the specific taxonomy is defined). Figure 4.1 illustrates the case of Venn Predictor defined using a taxonomy inspired by the Nearest Neighbour algorithm.

The transductive approach is computationally infeasible in all but the simplest cases. However, similarly to what was discussed for Conformal Prediction, it is possible to have an *inductive* mode of operation with the same theoretical guarantees,



	0	1	2	3	4	5	6	7	8
0	0.6470	0.2941	0.0000	0.0588	0.0000	0.0000	0.0000	0.0000	0.0000
1	0.5625	0.3750	0.0000	0.0625	0.0000	0.0000	0.0000	0.0000	0.0000
2	0.5625	0.3125	0.0625	0.0625	0.0000	0.0000	0.0000	0.0000	0.0000
3	0.5625	0.3125	0.0000	0.1250	0.0000	0.0000	0.0000	0.0000	0.0000
4	0.5625	0.3125	0.0000	0.0625	0.0625	0.0000	0.0000	0.0000	0.0000
5	0.5625	0.3125	0.0000	0.0625	0.0000	0.0625	0.0000	0.0000	0.0000
6	0.5625	0.3125	0.0000	0.0625	0.0000	0.0000	0.0625	0.0000	0.0000
7	0.5625	0.3125	0.0000	0.0625	0.0000	0.0000	0.0000	0.0625	0.0000
8	0.5625	0.3125	0.0000	0.0625	0.0000	0.0000	0.0000	0.0000	0.0625

FIGURE 4.1: Example of Venn Predictor on synthetic data. The data set is made up of points in the plane belonging each to one of 9 classes (indicated by color). The examples are generated as a mixture of 9 bivariate Gaussians, one per class, with centers arranged on the $\{-1,0,1\} \times \{-1,0,1\}$ grid. The Venn Predictor uses the following rule to establish a taxonomy: "two examples are assigned to the same category if their nearest neighbours have the same label". The table shows the Venn Predictor output for a test object at $(-1,-1)$.

at the cost of larger training data sets. We will discuss it in the context of Venn-ABERS predictors (Vovk, Petej, and Fedorova, 2015).

We have introduced Venn Predictors in a generic way. The rest of the thesis will be restricted to the case of binary classification, as this is the setting in which it will be applied to chemoinformatics problems.

We will use p_0 and p_1 to denote the probability that the label y of the test example be 1, but under two different hypotheses, namely $y = 0$ and $y = 1$.

4.1.3 Validity of Venn Predictors

Let's say that a random variable P taking values in $[0, 1]$ is *calibrated* for a random variable Y taking values in $\{0, 1\}$ if

$$\forall p \in [0, 1] \quad \mathbb{E}(Y \mid P) = P \qquad \text{almost surely} \tag{4.2}$$

Intuitively, P is the prediction made by a probabilistic predictor for Y, and calibration means that the probabilistic predictor gets the probabilities right, at least on average, for each value of the prediction. We then say that the probabilistic predictor is *calibrated*[1].

Note that this is one special type of statistical test. Calibration is a less stringent property than validity (as defined at the end of Section 4.1) as it requires only that the condition in Equation 4.2 be satisfied. It can be proved that if training set and test set are i.i.d., the Venn Predictor is well calibrated in the sense that there exists an "oracle" that would choose between p_0 and p_1 for every test object so that the output probability would be calibrated.

One may wonder how to make use of a multi-probabilistic calibrator, especially one in which at each "turn" one of the two predicted probabilities is calibrated but we do not know which one. In fact, the difference between p_1 and p_0 conveys some useful information. If p_1 differs significantly from p_0, it means that for that test object the estimates are so uncertain that it suffices to change the label of one observation (in this case, the hypothetical observation) to affect significantly the prediction of the probability.

Finally, it is important to observe that calibration is certainly desirable, but it is not the only property that useful probabilistic predictions must exhibit. If we were asked to predict with what probability it will rain tomorrow, we could simply respond with the long-term "climatological" average probability of rain. It would certainly be a calibrated prediction[2], but it would hardly be regarded as a good prediction. It is easy to convince oneself that if we were to offer bets on the basis of the climatological odds, we would lose money against adversaries who incorporated slightly more specific information into their probability estimates. This suggests

[1] The word 'calibrated' in the present context has a specific meaning that bears no relation to that of Section 2.4.5 where we define a "calibration set".

[2] barring, that is, climate change...

that useful predictions need to be *specific* as well as calibrated. Venn Predictors offer a theoretically-backed framework in which we no longer have to worry about calibration; we can focus only on making predictions more specific.

4.2 Venn-ABERS Predictors

Many binary classifiers compute a numerical value, a decision function value, and emit a bare prediction by comparing such value to an appropriate threshold. Some of these methods claim to yield directly a probability estimate, but in fact this is true only under stringent assumptions; in practice, deviation from calibration are observed. Venn-ABERS Predictors (VAP) are a form of Venn Predictors that can be used to transform the score that a binary classifier outputs into a calibrated probability estimate under the usual, minimal i.i.d. assumption. VAP automatically optimizes (in a sense that will be defined shortly) the choice of taxonomy. Let's assume that $s(x) : \mathbf{X} \to \mathbb{R}$ is the scoring function learned by a scoring classifier on a set $(x_1, y_1), \ldots, (x_\ell, y_\ell)$ where the labels y_i take values in $\{0, 1\}$. The *isotonic calibrator*[3] g for $((s(x_1), y_1), (s(x_2), y_2), \ldots, (s(x_\ell), y_\ell))$ is the non-decreasing function on $s(x_1), s(x_2), \ldots, s(x_\ell)$ that maximizes the likelihood

$$\prod_{i=1,2,\ldots,\ell} l_i$$

where:

$$l_i = \begin{cases} g(s(x_i)) & \text{if } y_i = 1 \\ 1 - g(s(x_i)) & \text{if } y_i = 0 \end{cases}$$

Let $s_0(x)$ be the scoring function for $(z_1, z_2, \ldots, z_\ell, (x, 0))$, $s_1(x)$ be the scoring function for $(z_1, z_2, \ldots, z_\ell, (x, 1))$, $g_0(x)$ be the isotonic calibrator for

$$((s_0(x_1), y_1), (s_0(x_2), y_2), \ldots, (s_0(x_\ell), y_\ell), (s_0(x), 0))$$

and $g_1(x)$ be the isotonic calibrator for

$$((s_1(x_1), y_1), (s_1(x_2), y_2), \ldots, (s_1(x_\ell), y_\ell), (s_1(x), 1))$$

.

The Venn-ABERS predictor outputs a multi-probabilistic prediction (p_0, p_1), where $p_0 = g_0(s_0(x))$ and $p_1 = g_1(s_1(x))$. It is customary to refer to p_0 and p_1 as lower and upper probability and indeed, in the case of Inductive VAP (Section 4.2.2), it is claimed (Vovk, Petej, and Fedorova, 2015, Section 2) that it is always the case that $p_0 < p_1$.

[3] Monotonic: "one ordering", either Isotonic ("order-preserving") or Antitonic ("against the order")

4.2.1 Observations

The definition of VAP may appear a bit disorientating. There is no mention of a taxonomy, which Section 4.1.2 put at the heart of Venn Predictors. It is not immediate to see the equivalence classes over which we compute the predicted probabilities as relative frequencies, according to Equation 4.1. This section provides some element of clarification and justification.

The key theoretical results on which VAP rests are in (Ayer et al., 1955) and in (Barlow and Brunk, 1972). The "ABERS" in Venn-ABERS is an acronym obtained from the names of the authors of the former paper.

The first key theoretical basis is that the isotonic calibrator defined in the previous section can be computed as Isotonic Regression (IR) on $(s(x_1), y_1), \ldots, (s(x_\ell), y_\ell)$.

A general definition of Isotonic Regression problem is to find the values g_i that:

$$\begin{aligned} &\text{minimize } \sum_{i=1}^{\ell} (g_i - y_i)^2 \\ &\text{subject to } g_i \leq g_j \text{ when } i \preceq j \end{aligned}$$

where $y_1, \ldots, y_\ell$ are given and $\preceq$ is a partial ordering on $\{1, \ldots, \ell\}$.

In our case, the $g_i = g(s(x_i))$ and the partial ordering $\preceq$ corresponds to the ordering on $s(x_i)$. While strictly speaking the Isotonic Regression is defined only on the $s(x_i)$, taking some liberty we can view it as a piecewise constant function of s. Figure 4.2 shows an example of Isotonic Regression.

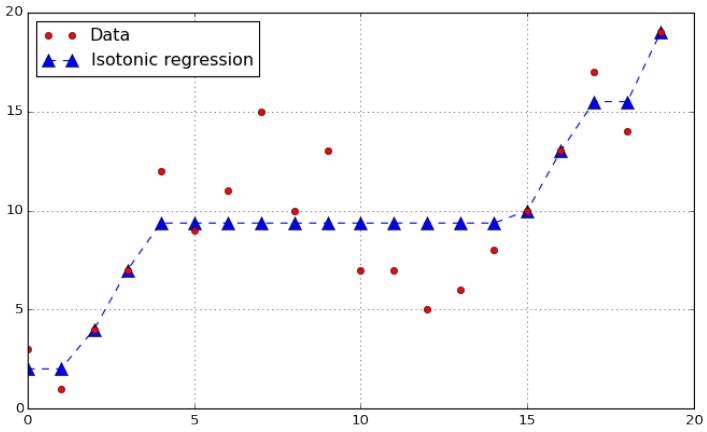


FIGURE 4.2: Isotonic Regression. An example of IR on a simple data set.

It turns out that the equivalences classes of the Venn Predictor are the intervals over which the IR is constant. The value of the IR is indeed the fraction of examples with label $y = 1$ among all the examples with score $s()$ in that interval.

4.2.2 Inductive VAP

The definition of VAP in the previous paragraphs requires that the underlying ML be re-trained and the scores as well as the IR be recalculated for each possible label value and for each test object. The resulting computational cost would become unaffordable even for relatively small data sets.

It is possible to reduce significantly the computational cost by adopting an inductive approach, similar in a broad sense to the procedure described for Inductive Conformal Predictors in Chapter 2.

The overall training set is split into a proper training set, which is used to train the underlying machine learning algorithm which produces scores, and a calibration set, which is used to "train" the isotonic calibrator which transforms scores into probability estimates.

The training of the underlying machine learning algorithm is performed only once, but the isotonic regression is still recalculated (ex novo) for each test object and for each possible value of the label.

4.2.3 Cross IVAP

A potential disadvantage of Inductive VAP is that, by separating a calibration set from the overall training set, it reduces the size of the set used for training the underlying algorithm. It may be of benefit to use a scheme inspired by cross-validation in which the overall training set is split into K sets; IVAP is then applied K times, each time using one of these sets K as calibration set and the rest as proper training set. The resulting K IVAPs can then be combined in a way that will become clearer in the next section.

4.2.4 Making probability predictions out of multi-probability ones

While there is a value in having multi-probability predictions, it is generally more intuitive to deal with a single-valued probability prediction. One way to construct a probability prediction from a multi-probability prediction is to derive the value that minimizes the regret[4] under a given loss function.

It is possible to prove (see (Vovk, 2012, Section 4)) that under log-loss function the minimax probabilistic prediction is:

$$p = \frac{p_1}{1 - p_0 + p_1}$$

Note that p is calculated as if $1 - p_0$ and p_1 were the unnormalized probabilities of $y = 0$ and $y = 1$, respectively.

[4] by regret, we denote the loss suffered by making a given choice instead of the best available option

If, instead of the log-loss function, we consider the square loss function (also known as Brier loss), the minimax probabilistic prediction is:

$$p = \bar{p} + (p_1 - p_0)\left(\frac{1}{2} - \bar{p}\right)$$

where $\bar{p} := (p_0 + p_1)/2$.

In the case of Cross VAP, the K multi-probabilistic predictions can be combined in a minimax way with respect to log-loss with:

$$p = \frac{\mathrm{GM}\left(p_1\right)}{\mathrm{GM}\left(1 - p_0\right) + \mathrm{GM}\left(p_1\right)}$$

where GM() stands for geometric mean over the K values (see (Vovk, Petej, and Fedorova, 2015, Section 4) for the proof).

4.2.5 Fast Venn-ABERS

While it reduces the computational load significantly (compared to the original "transductive" formulation), Inductive Venn-ABERS prediction still remains too complex to scale to large data sets, the recalculation of the Isotonic Regression for every label value and for every test object being the main limiting factor. (Vovk, Petej, and Fedorova, 2015) found, from a detailed analysis of the algorithm used to compute the IR and of the particular way in which it used in the Venn-ABERS application, that it was possible to come up with a new algorithm that produces exactly the same results, but requires one initial *pre-calculation* ($O(\ell \log \ell)$) and then requires for the actual *evaluation* of the probability of each test object only a very efficient look-up ($O(\log \ell)$). An illustration of the Venn-ABERS calibration on synthetic data sets is provided in Fig. 4.3.

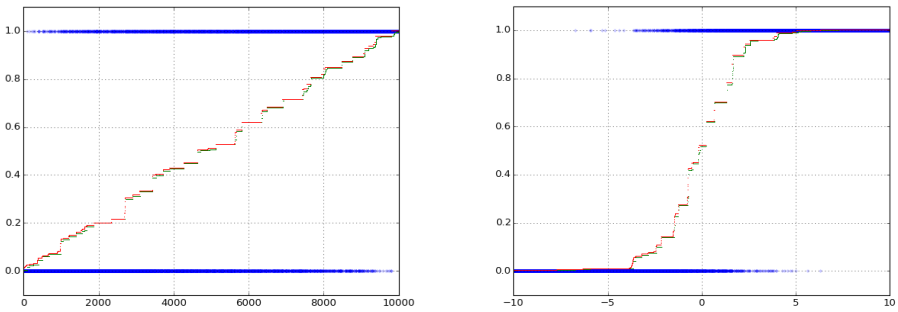


FIGURE 4.3: Multi-probabilistic predictions for two synthetic data sets. The blue dots represent the data, which can take values 0 or 1. The data sets were created so that the desired calibrator is a linear function in the left pane and a sigmoid in the right. This was by generating random variates from a uniform distribution between 0 and 1 and comparing them with a linear function (left pane) and a sigmoid (right pane). The Venn-ABERS predictors reconstruct the desired calibrator within an error inherent in the statistical fluctuation of the sample.

Tab. 4.1 gives an indication of the performance, on synthetic data sets, of a pure Python/Numpy implementation running on one core of a general-purpose server. (Implementing it in C/C++ is likely to reduce the quoted times by factor of 10 or more).

TABLE 4.1: Execution times of the pre-calculation and evaluation of the Fast Venn-ABERS IR. The number of examples quoted refers to the size of the calibration set as well as the test set.

Number of examples	CPU time for Pre-calculation on calibration set	CPU time for Evaluation on test set
10,000	3.62s	<0.01s
100,000	15.7s	0.01s
1,000,000	153s	0.13s

This enabled us to apply Venn-ABERS prediction to large data sets (hundreds of thousands of examples) without particular requirements in terms of computational resources.

4.3 Application of Venn-ABERS Prediction to BioAssay Data

We applied VAP to the same AID827 data set used in the previous chapter for the CP studies.

To recap the salient points, each tested compound is described by a variable number of *signature descriptors* (Faulon, Visco, and Pophale, 2003) derived from the chemical structure of the compounds itself. Each signature corresponds to the number of occurrences of a given labelled subgraph in the molecule graph. The resulting data set can be viewed as a relatively sparse matrix of attributes (the signatures on the columns) and examples (the compounds on the rows).

Total number of examples	=	138,287
Number of features	=	165,786
Number of non-zero entries	=	7,711,571
Density of the data set	=	0.00034
Active compounds	=	1,658 (1.2%)
Inactive compounds	=	136,629
Unique set of signatures	=	137,901

We reused the same values of the decision function calculated in that context for the calibration set and test set for the Inductive Conformal Prediction. As a side note, Fig. 4.4 shows the distribution of such values, with two separate histograms one for Active examples and the other Inactive Examples. The distribution for Active examples looks distinctly bimodal.

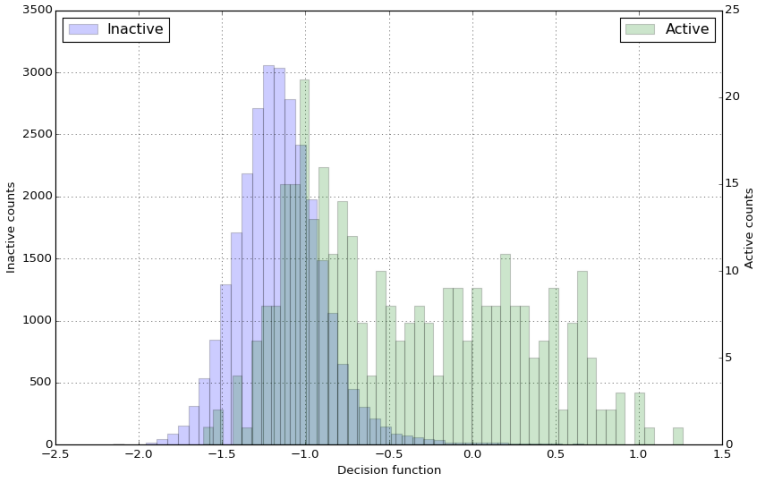


FIGURE 4.4: Distribution of Decision Function values on the calibration data set. Note the different y scale

.

The calibration set has 28,000 examples and the test set comprises 10,000 objects. Running the Fast Venn-ABERS predictors on these sets took a total time 4.84s (of course, this excludes the time to calculate the scores, given that we reused them, as explained above).

One illustration of the multi-probabilistic results is provided in Figure 4.5 which shows the cumulative sums of p_0, of p_1, and of the label as we sweep the test set.

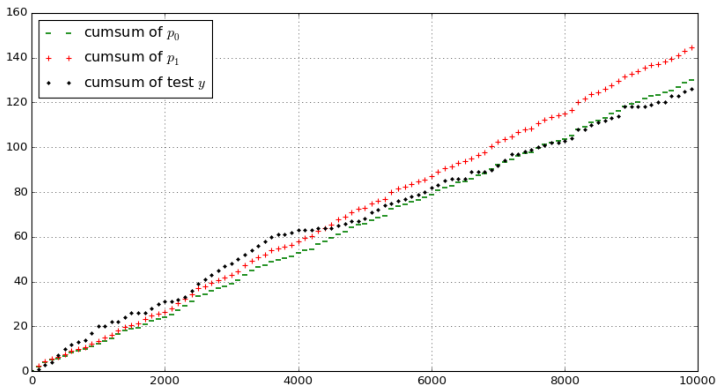


FIGURE 4.5: Cumulative sum of p_0, p_1, and of the label

.

It is perhaps useful to note that one might expect that as a consequence of the calibration property of the multiprobabilistic predictor, the black trace representing the counts of Active test examples should be eventually between the green and red traces. In fact, no such guarantee can be inferred from the calibration property.
To see this, let's consider an idealized case in which we have a probabilistic predictor always emits the same probability, namely the relative frequency of Actives in the

overall population, i.e. 1.2%. Such a predictor, while not very useful, is calibrated according to the definition stated in equation (4.2).
If we were to draw an analogous picture to Figure 4.5, instead of the green and red traces, we would draw just one trace, a straight line with slope 0.012 represented here in blue. The cumulative count of test examples of the Active class (the black trace) would weave its irregular path in the vicinity of the blue trace. The left plot in Figure 4.6 shows a few instances of the "path" of the counts. For each value k of the x-axis, the cumulative count of test examples can be considered a realization of a Binomial distribution of parameters $p = 0.012$ and $n = k$. The variance of the Binomial is $np(1-p)$. Note that the variance grows with n. Based on this observation, we can now appreciate that the paths will tend to diverge from the mean. The right plot in Figure 4.6 shows, for every value of x, the interval on the y-axis in which a path will be with probability 0.5. The size of the interval grows as $\sqrt{n}$ for $n \to +\infty$ (consider, for instance, the Gaussian approximation of the Binomial).
Note that this interval has nothing to do with lower and upper probability. It is a manifestation of the variability inherent in sampling.

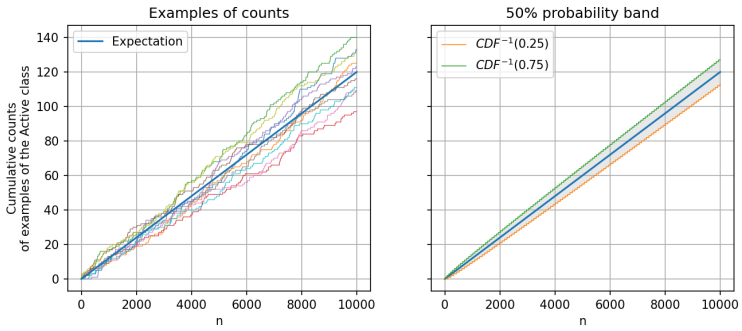


FIGURE 4.6: Cumulative sum of the label in an idealized case. The left plot shows 10 realizations (paths) of the cumulative sum of a Bernoulli variable. On the right, the shaded area is where 50% of the paths are expected to be.

We can now try to map these observations on Figure 4.5. The black trace corresponds to one of the possible instances represented in the left plot in Figure 4.6, whereas either one of the red and green traces corresponds to the blue line in the right plot in Figure 4.6. In fact, the calibration guarantee is for a specific but a priori unknown choice of the points on the green and red traces. It is the ensuing line, which we could view as an appropriate "hybrid" of the red and green lines, that corresponds to the blue line. Based on the previous discussion, we can appreciate that the actual path of the counts will tend to diverge with respect to this "hybrid" line. We can also convince ourselves that there is no reason to expect that the actual count will eventually be between the cumulative sums of lower and upper probabilities. The interval between these two cumulative sums has nothing to do with the phenomenon of the divergence of the actual count discussed earlier. That interval has to do with the sensitivity of the probabilistic predictions, that is, how much the isotonic regression changes (at the test point) when one varies the hypothetical label.

Getting now back to the main discussion, Fig. 4.7 has a plot of the "calibrators" $g_0(s)$ and $g_1(s)$ between the min and max of the decision function on the calibration set. The plot presents some aspects worth commenting. One interesting feature is the widening gap between $g_0(s)$ and $g_1(s)$ for $s > 0.5$. The gap between $g_0(s)$ and $g_1(s)$ could be taken as an indication of the uncertainty of the estimate of the probability. In fact, the gap expresses the sensitivity of the probability estimate to the hypothetical label assigned to the test object. It is only in this narrow sense that we can view the gap as uncertainty of the prediction. Intuitively, the sensitivity depends on the number of calibration examples that are in the same element of the Venn taxonomy as the test object. The Isotonic Regression identifies intervals over which the value of the regression is the average of the labels of the calibration examples in that interval. These intervals are the classes in the Venn taxonomy. When the label of the test object is changed, the average of the interval in which the test object falls will change more markedly the fewer the calibration examples in the interval (note that the change of the label can also change the interval itself). It could be argued that the uncertainty should be higher for probability predictions around 0.5 rather than at the two extremes 0 and 1. In reality, as just discussed, the uncertainty at a value s of the score is not determined by the value of the probability prediction itself, but by its sensitivity to changing the label of an object with score s. In the case of Figure 4.7, the data set had only 1.2% of examples belonging to the Active class. Figure 4.4 shows the histograms for the Active and the Inactive examples separately and with different scales (because otherwise the Inactives would have hardly registered). The chart supports the view that the gap between lower and upper probabilities at score s is inversely related to the "density" (in a broad sense) of calibration examples with score s.

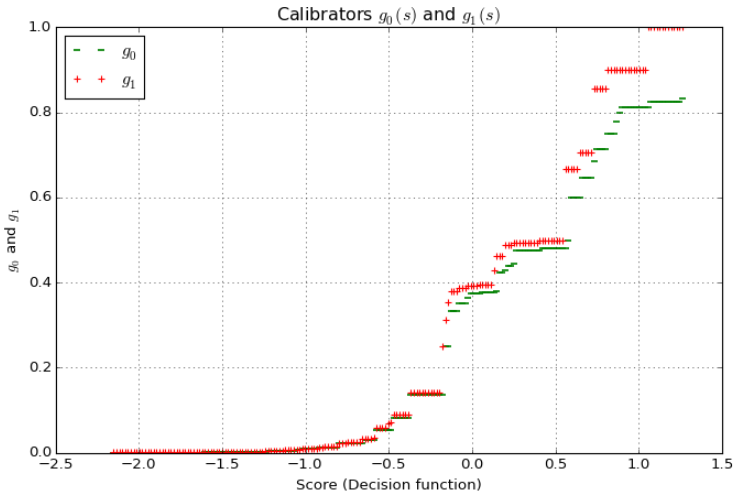


FIGURE 4.7: Calibrators $g_0(s)$ and $g_1(s)$

4.3.1 Comparison with Platt scaling

We set out to compare the relative merits of Platt scaling and Venn-ABERS predictors. We used the implementation of Platt scaling available as part of the Python `sklearn` package.

Fig. 4.8 shows the "calibrators" arising from the two methods (for one of 20 runs). In order to have a more meaningful comparison, we computed the log-loss with

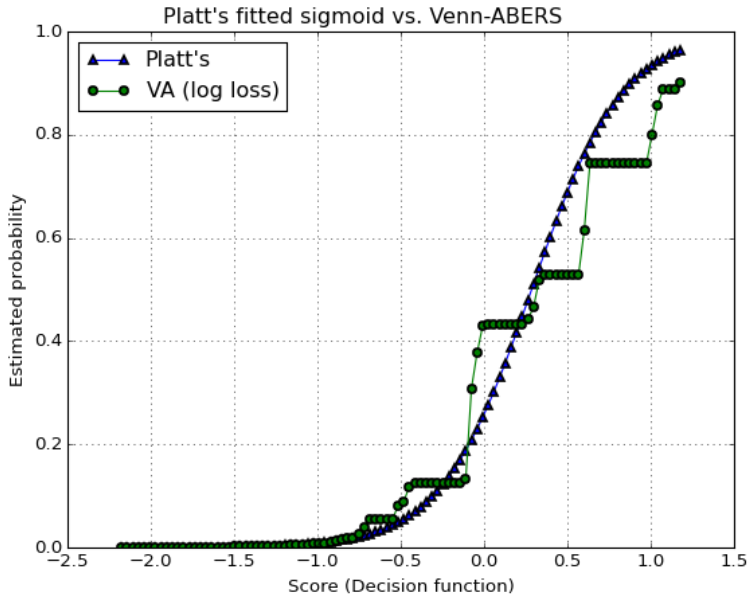


FIGURE 4.8: Comparison between Platt scaling and Venn-ABERS (combined according to minimax log-loss).

either method on the test set of each of the 20 runs. The resulting boxplots are shown in Fig. 4.9. The median is the same (VA is better by the tiniest of margins) and, if anything, Platt's method exhibits a bit more variance.

Given the difference between the two calibrators apparent in Fig. 4.8, one might have expected a difference also in the log-loss.

Imbalance plays a role in this apparent paradox. For values of the Decision Function (DF) less than -0.7, there is very little difference between the two calibrators. Consider that $\approx 95\%$ of the test samples happen to have a DF< -0.7. For those test samples the difference between Platt scaling probabilities and Venn-ABERS probabilities is going to be negligible.

It is questionable if the loss is "symmetrical" in our application, i.e. if the errors in predicting high probability have the same consequences and should be attributed the same cost as the errors of the same entity for low probabilities.

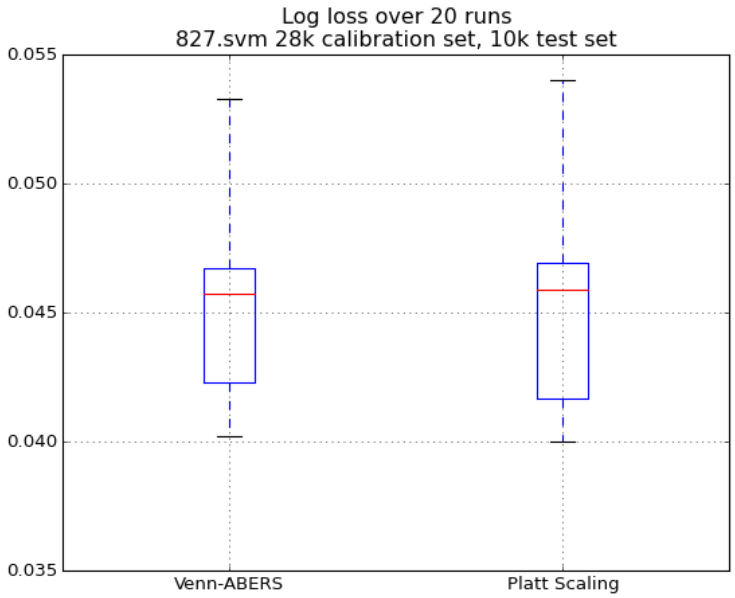


FIGURE 4.9: Log-loss on test sets, over 20 runs

.

One way to have an asymmetric log-loss is to assign different weights to the loss on Actives and on Inactives.

$$\text{AsymmLogLoss} = -C_{act} y_i \log p_i - C_{inact}(1 - y_i) \log(1 - p_i) \tag{4.3}$$

with $C_{act}, C_{inact} > 0$ and possibly $C_{act} + C_{inact} = 1$.
This has some consequences. One issue with this approach is that this formulation of asymmetric log-loss is not a proper loss function. A proper loss function $L(p, y)$ is such that the expectation $\mathbb{E}_{y \sim B(q)} L(p, y)$ is maximized (resp. minimized) when $p = q$. Log-loss as well as Brier loss are proper loss functions (Schervish, 1989). A proper scoring function is desirable because it keeps forecasters honest, in the sense that it rewards probabilistic predictions that reflect the actual probabilities (Winkler et al., 1996, Section 2). In general, if a proper loss function is of the form $L(y, p) = yL_1(1 - p) + (1 - y)L_0(p)$ with $L_0(p)$ and $L_1(p)$ monotone decreasing functions, the weighted loss function $L(y, p) = cyL_1(1 - p) + (1 - c)(1 - y)L_0(p)$ is in general no longer proper. A second issue is that the multi-probabilistic combination function for log loss discussed in Section 4.2.4 would have to be modified to account for the weighting. The combination is obtained in (Vovk, 2012, Section 4) by equating the regret in using the combined value p instead of the correct values (p_0 when the label is 0 and p_1 when the label is 1). Using the same approach, the equation that the combined p must satisfy is:

$$-C_{act} \log p + C_{act} \log p_1 = -C_{inact} \log(1 - p) + C_{inact} \log(1 - p_0)$$

This equation does not appear to admit a general closed-form solution for p. This creates obvious practical problems in the principled application of the asymmetrical log-loss.

In any case, notwithstanding the potential theoretical issues of an improper loss function, we believe that there is insight to be gained by using an asymmetrical log loss. It is useful to consider for instance only the contribution to log loss given by the cases in which the actual label is 'Active' (which equates to setting $C_{act} = 1$ and $C_{inact} = 0$ in equation(4.3)). It is important to remind ourselves that the 'Active' class represents a small minority and in the symmetrical log-loss any performance difference that might occur on that class would be diluted by the performance on the much more prevalent 'Inactive' class. Figure 4.10 shows that there is a more visible advantage, albeit still tiny and possibly not statistically significant, for Venn-ABERS over Platt scaling.

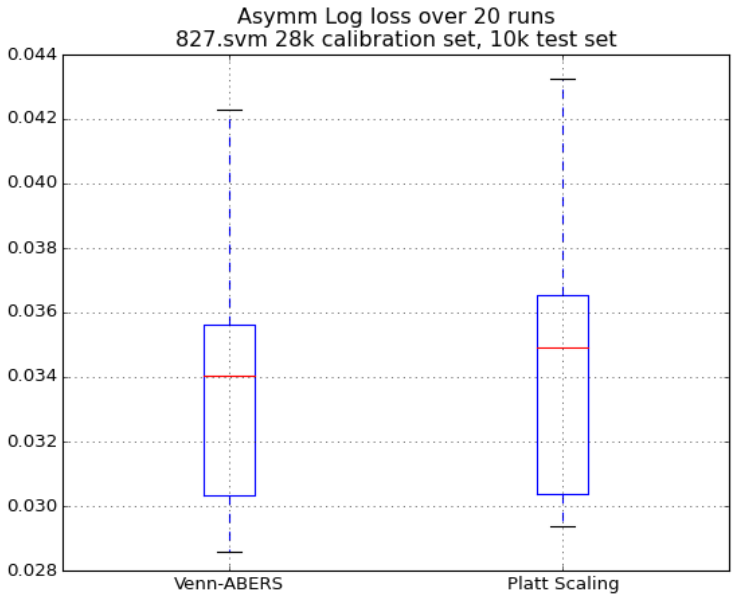


FIGURE 4.10: Asymmetric Log-loss on test sets, over 20 runs

.

4.4 Comparison between Conformal and Probabilistic results

Both Conformal and Probabilistic Predictors allow the user to rank the compounds for the purposes of screening. The two rankings are based on conceptually different ideas. The p-value of Conformal Prediction expresses the probability of encountering an Active compound that would appear as or more nonconform (among Active compounds) than the test object. The value output by the Probabilistic Prediction expresses the probability that the compound is Active (in the sense that if one keeps

taking compounds for which the predicted probability is 0.8, then one will find in the long term that 80% of such compounds are actually active).

Tab. 4.2 shows the 20 compounds (out of a test set of 10,000) that are ranked highest by Conformal Prediction and by Venn-ABERS Prediction. The compounds are identified by an ordinal that has no specific meaning outside of the specific data set we were supplied. The single probability was obtained from the multi-probabilistic prediction using the method discussed earlier for log-loss regret minimization.

Although the ranking may appear quite different at first glance, it is remarkably similar. The similarity is somewhat obscured by the fact that the p-values appear to have a finer granularity than the probabilities (there are more compounds sharing probabilities than sharing p-values). The rankings appear very similar if we consider that the probability ranking is arbitrary when the compounds have the same probability.

TABLE 4.2: The 20 candidate compounds (out of a test set of 10,000) that are ranked highest, according to Conformal Prediction and to Venn-ABERS Prediction. The Active compounds are in bold.

Ranking	best compounds by p-value	p value	best compounds by prob	prob
0	**108198**	**0.868**	**108198**	**0.800**
1	103948	0.798	129143	0.697
2	62772	0.774	62772	0.697
3	129143	0.716	103948	0.697
4	**138051**	**0.663**	**138051**	**0.695**
5	**108632**	**0.663**	**108632**	**0.695**
6	**107957**	**0.584**	**108877**	**0.683**
7	**108920**	**0.584**	**108920**	**0.683**
8	**108877**	**0.584**	**107957**	**0.651**
9	**108783**	**0.578**	**108783**	**0.651**
10	5413	0.557	5413	0.647
11	**138177**	**0.551**	54806	0.611
12	4334	0.537	16925	0.611
13	71538	0.513	**108032**	**0.611**
14	54806	0.513	**108026**	**0.611**
15	16925	0.493	4334	0.611
16	**108584**	**0.490**	71538	0.611
17	**108026**	**0.490**	**138177**	**0.611**
18	**107943**	**0.478**	**107943**	**0.611**
19	**108032**	**0.475**	**108584**	**0.611**

4.5 Summary and Conclusions

In this chapter we introduced Probabilistic Prediction and highlighted its conceptual differences with Conformal Prediction. We presented the notion of validity for

a probabilistic predictor and introduced Venn Predictors, multi-probabilistic predictors for which it is possible to prove the validity property. We then turned to Venn-ABERS predictors, which are Venn Predictors that calibrate classification scores into probabilities. Finally, we applied a fast method of computing Venn-ABERS probabilities to a bioassay data set and compared and contrasted its results with Platt scaling and Conformal Predictors.

Venn-ABERS predictors can improve on Platt scaling, in particular when the functional dependency of probabilities to scores departs significantly from a sigmoid. Their computational cost is perfectly affordable even on large data set sizes.

As a closing remark, we would like to point out that the qualities of the predictions produced by VAP are recognized also by independent industry researchers not only in pharmaceutical research (Mervin et al., 2020), but also in the agrochemical research[5].

[5] A researcher from an agrochemical industry research centre based in the UK contacted us for permission to use our code available at https://github.com/ptocca/VennABERS— and also subsequently found a minor bug!

Chapter 5

Combination of CP

5.1 Introduction

This chapter and the next one present some research results about the combination of Conformal Predictors. The motivation for combining arises naturally from the intuition that CPs with different underlying algorithms can complement each other, not only making the predictor more robust but also improving its performance in a synergistic way. In this study, we restrict our attention to methods that compute a *combined* p-value as a function of the p-values computed by *base* conformal predictors for a given test object. The advantage of combining p-values is that it solves in a principled way a potential complication that comes into play when ensembling scoring algorithms. The scores produced by different algorithms are expressed in different scales and with different laws. By applying Conformal Prediction, the output of the scoring classifier or any other ML method for which we can come up with a meaningful Non Conformity Measure is transformed into a p-value. In this way, the output of each base algorithm is expressed not only in the same $[0, 1]$ range, but also with same meaning and properties (e.g. Theorem 1 in Section 2.4.1) across ML methods. The problem of combination is therefore reduced to the problem of determining the optimal way of combining p-values. The following sections in this chapter explore combination methods whose implementations could be scaled to large data sets. Some results will be shown using chemoinformatics data sets from the ExCAPE project. The next chapter presents instead a method that has potentially optimal properties, but is computationally more demanding. Consequently, rather than the large data sets of this chapter, we use synthetic data sets of a manageble size. Also, the use of synthetic data set makes it possible to evaluate the methods under different levels of correlation among p-values.

5.2 Combination of p-values

The study of the problem of combining p-values to obtain a single test for a common hypothesis has a long history, originating very soon after the framework of statistical hypothesis testing was established (Fisher, 1932). A survey can be found in (Loughin, 2004). In its more general form, the problem raised a lot of attention for its application to meta-analysis, where the results of a number of independent studies, generally with different sample sizes and different procedures, are combined. The various methods that have been proposed over the years have tried to cater for the different ways in which the evidence manifests itself. In particular, some methods allow for weighting, thereby assigning more importance to some p-values (for instance, in the case of meta-analyses, those corresponding to studies with larger samples sizes). More importantly, each method is associated with a different shape of the rejection region (the portion of the k-dimensional space of the k p-values being combined for which the combined test of significance would reject the null hypothesis under a chosen significance level ϵ). The shape reflects the different way

in which evidence of different strength is incorporated into the aggregated p-value. It has been observed (see, for example, (Loughin, 2004, Section 3)) that there is no single combination method that outperforms all others in all applications. The combination of p-values from different Conformal predictors on the same test object is a very special form of the general problem outlined above.

A method for the combination of Conformal Predictors should aim to:

- **Preserve validity**: for the output of the combination method to be a Conformal Predictor, this is a necessary property.
- **Improve efficiency**: smaller prediction sets must result from a desirable method of combination.

In practice, one is interested in the two desiderata above if the resulting p-values are to be used to obtain prediction sets. There are domains of application where the p-values can be used in other ways. An example which will be developed further in the section 5.4.4 is in the context of Drug Discovery: the p-values can be used to rank candidate compounds (see (Toccaceli, Nouretdinov, and Gammerman, 2017)) in terms of the confidence in their activity (or lack of confidence in their inactivity), so that an informed decision can be made as to which candidate compounds to choose for a new batch of screenings.

There are two key observations that apply to p-values computed by Mondrian Inductive Conformal Predictors (MICP):

1. *The p-values from the same Conformal Predictor for the various test objects do not necessarily follow the uniform distribution.* The p-values in Statistical Hypothesis Testing are uniformly distributed by construction if the null hypothesis is true. Similarly, when one examines the MICP p-values for a set of test objects, it is apparent that only those for which the hypothetical label assignment is the correct one are uniformly distributed. The p-values for the objects for which the hypothetical label assignment is incorrect tend to have values towards 0.

2. *The p-values from different Conformal Predictors for the same test object are not independent.* One has to expect that, when testing the same hypothesis with different methods on the same object, the results will exhibit some degree of correlation. In other applications of p-value combination, the issue may be less of a concern. For instance, in meta-analyses of clinical trials, it is arguable that there is less correlation because the trials are not reusing the same patients in the same groups. However, the one considered is certainly not the only context in which dependent p-values are encountered and the issue has attracted some attention by statisticians (Pesarin, 2001; Brown, 1975; Alves and Yu, 2014; Poole et al., 2016).

5.2.1 Methods from "traditional" Statistical Hypothesis Testing

As outlined in (Loughin, 2004), the field of Statistical Hypothesis Testing has approached the problem of p-value combination as soon as the notion of p-value started to establish itself in the statistical community. One can identify, broadly speaking, two classes of p-value combination methods: quantile methods and order-statistic methods.

Order-statistic methods (Davidov, 2011) are mentioned here for completeness. Given k p-values coming from k experiments, the combining function is based on the order of the p-values. For instance, a combination method might simply consist in taking the smallest of the p-values; another method might take the second smallest, another the arithmetic average, another the maximum and so forth. Intuitively, a combination method that takes the smallest p-value "believes" the outcome that is most improbable under the Null Hypothesis, whereas a methods that takes the largest p-value would be stricter, in that it would take the outcome that is less contrary to the Null Hypothesis. In general, such methods appear to be inadequate for Conformal Predictors, because they result in the loss of validity.

On the other hand, quantile methods can satisfy this requirement. The quantile methods can be generally constructed by transforming the p-values using a function chosen as the inverse of a Cumulative Distribution Function (CDF) of a convenient distribution. The choice of the distribution is in principle arbitrary, but it is convenient to constrain it to those distributions for which the CDF of the sampling distribution of the sum of Random Variables (RVs) can be expressed with closed formulas or can be calculated with little computational effort (it has been noted (Zaykin et al., 2007) that "nowadays any form of CDF can be used with the aid of simple Monte Carlo evaluation"). Let's assume that $\mathbb{D}$ is a distribution with support $[a, b]$ and with invertible CDF $F_X(t) : [a, b] \to [0, 1]$. A quantile method would transform the p-values p_i (now considered Random Variables) into Random Variables $T_i = F_X^{-1}(p_i)$. By construction, these T_i are distributed according to $\mathbb{D}$. If we call $F_{T_1+T_2+\cdots+T_k}(t)$ the CDF of the sum of k $\mathbb{D}$-distributed RVs, the combined p-value is obtained as $p_{\text{comb}} = F_{T_1+T_2+\cdots+T_k}(t_1 + t_2 + \cdots + t_k)$. It is easy to see that P_{comb} is uniformly distributed, based on elementary property that will be proved in a later section

Here we consider one quantile method, namely Fisher's method (also known as chi-square method), although other quantile methods exist, such as Stouffer's method (Stouffer et al., 1949) (also known as z-transform test).

5.2.2 Fisher's method

Fisher's method (Fisher, 1932; Fisher, 1948) is among the earliest p-value combination methods. It relies on the key observation that if $p_1, p_2, \ldots, p_k$ are each the

realization of a uniformly distributed random variable,

$$h_i = -2 \log p_i \qquad \text{with} \quad i = 1, \ldots, k$$

is a random variable that follows a chi-squared distribution with 2 degrees of freedom.

The sum of k independent random variables each following a chi-squared distribution with 2 degrees of freedom is itself chi-squared distributed with $2k$ degrees of freedom, that is:

$$h = -2 \sum_{i=1}^{k} \log p_i$$

is a random variable that follows a chi-squared distribution with $2k$.

The combined p-value is:

$$p = \mathbb{P}\left\{ y \leq -2 \sum_{i=1}^{k} \log p_i \right\}$$

where y is a random variable following a chi-square distribution with $2k$ d.f. The integral required for calculating the probability above has a very simple closed form:

$$t \sum_{i=0}^{k-1} \frac{(-\log t)^i}{i!}$$

where $t = (p_1 \times p_1 \times \cdots \times p_k)$.

Interestingly, the formula above also arises as the probability of the product of independent uniform random variables (Zaykin et al., 2002). Fisher's method also exhibits a form of asymptotic optimality "among essentially all methods of combining independent tests" (Littell and Folks, 1973).

5.2.3 Validity Recovery

None of the methods discussed so far guarantees validity. Fisher's method guarantees valid p-values but only under the assumption of independence. The resulting combined p-value will exhibit a deviation from the uniform distribution that will be more pronounced the stronger the dependence among p-values. In our specific setting, the p-values are obtained by applying CP with different underlying algorithms on the same test object. It is therefore to be expected that the p-values will exhibit a substantial degree of correlation. Figure 5.1 illustrates the effect of correlation on combination.

The problem is well-known and there have been many attempts to introduce corrections based on some measure of the correlation or of the dependence. We propose a very simple calibration method[1], based on the following well-known elementary

[1] To avoid misunderstandings with previous occurrences of 'calibration" or "calibrated", let us clarify that in the present context "calibration" is to be understood as a transformation that ensures that a variable is uniformly distributed.

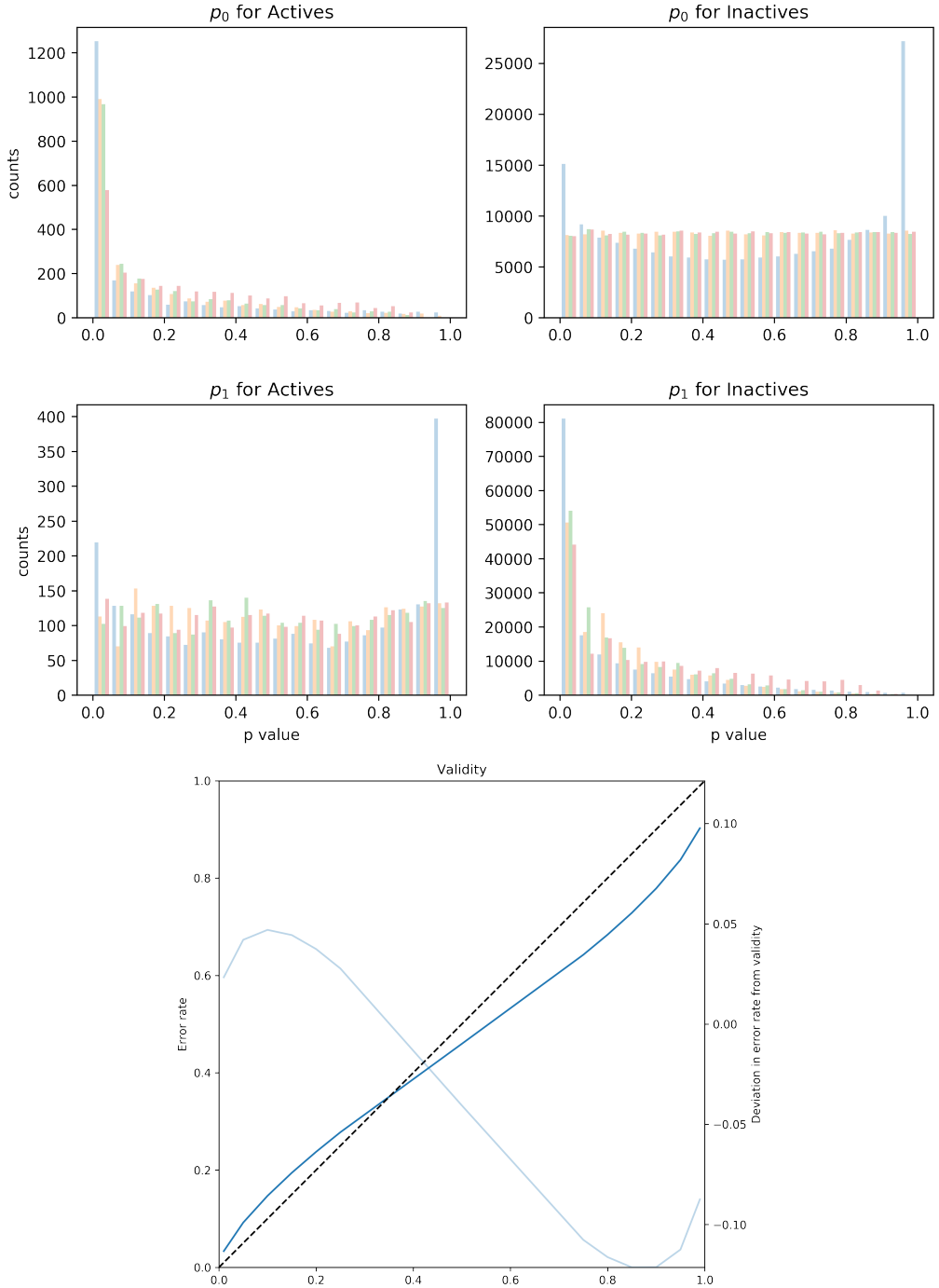


FIGURE 5.1: Deviation from validity when combining with Fisher's method. The four histograms at the top show the counts of p-values over 20 equally sized bins between 0 and 1 from one of the runs of real-world example discussed later. In each bin, the bars refer to the combined CP (the leftmost bar, in light blue) and three base CPs. The p-values $p_{\bar{y}}$ of test objects with label $\bar{y}$ (i.e. same label as the one for which we are computing the p-value) should be uniformly distributed, whereas they should be concentrated towards 0 for test examples with other labels. The uniform distribution is essential for the validity property. We should observe a uniform distribution for the upper right and lower left histograms. The correlation between p-values for the same object caused Fisher's method to deviate from uniformity. The Pearson correlation ranged between 0.44 and 0.77 for p_0 and between 0.36 and 0.78 for p_1. The effects on validity are shown in the bottom plot. The blue line should be overlapping the dashed line. The light blue line (whose y-axis is on the right) shows the difference between blue line and the dashed line.

result.

Given a random variable X and its CDF $F_X(t) \equiv \mathbb{P}[X \leq t]$ which we will assume invertible (but slightly less restrictive assumptions are possible), the random variable $Y \equiv F_X(X)$ follows the Uniform distribution.

This can be proved by showing that the CDF of Y is the identity function, i.e. $F_Y(y) = y$, along the following lines:

$$F_Y(y) \equiv \mathbb{P}\left[Y \leq y\right] = \mathbb{P}\left[F_X(X) \leq y\right] = \mathbb{P}\left[X \leq F_X^{-1}(y)\right] = F_X\left(F_X^{-1}(y)\right) = y$$

A similar approach has already been described in (Balasubramanian, Chakraborty, and Panchanathan, 2015) but when empirically evaluated in (Linusson et al., 2017) appeared to show limited effectiveness.

The method we propose consists in re-calibrating the combined p-value obtained with any of the methods above by using the CDF of the combined p-values. An estimate of such CDF can be obtained from the Empirical Cumulative Distribution Function (ECDF) observed on a Re-calibration Set (any set drawn from the same distribution, with the exclusion of the training set and the test set).

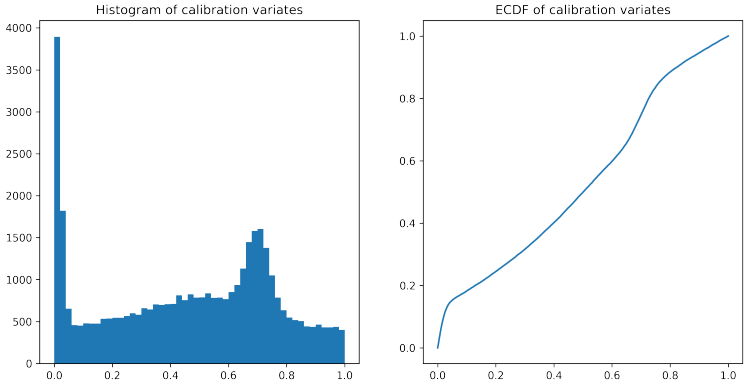


FIGURE 5.2: On the left, the plot is the histogram of about 40,000 variates of an arbitrary illustrative distribution $q(x)$ with values between 0 and 1. On the right, the plot shows the Empirical Cumulative Distribution Function $ECDF(x)$ obtained on the basis of the variates.

To clarify the procedure, we list in Algorithm 2 the steps involved in the computation of the ECDF-based re-calibration.

Algorithm 2: ECDF-based p-value re-calibration

Data:

Re-calibration set $\mathbf{R} := \{(x_i, y_i)\}$ from a partition of the training set

p-values $p^{(\bar{y})}_{\text{cal},i}$ for hypothetical label $\bar{y}$ for examples $(x_i, y_i) \in \mathbf{R}$

p-values $p^{(\bar{y})}_{\text{raw},j}$ for the test objects for hypothetical label $\bar{y}$

Result: calibrated p-values $\hat{p}^{(\bar{y})}_j$ for hypothetical label $\bar{y}$

1 Select p-values only of examples with $y = \bar{y}$: $\mathbf{P}_{\bar{y}} := \{p^{(\bar{y})}_{\text{cal},i} : y_i = \bar{y}\}$;

2 Compute ECDF $F_P(p)$ of p-values $p \in \mathbf{P}_{\bar{y}}$;

3 Apply ECDF to uncalibrated (test) p-values: $\hat{p}_j = F_P(p_{\text{raw},j})$;

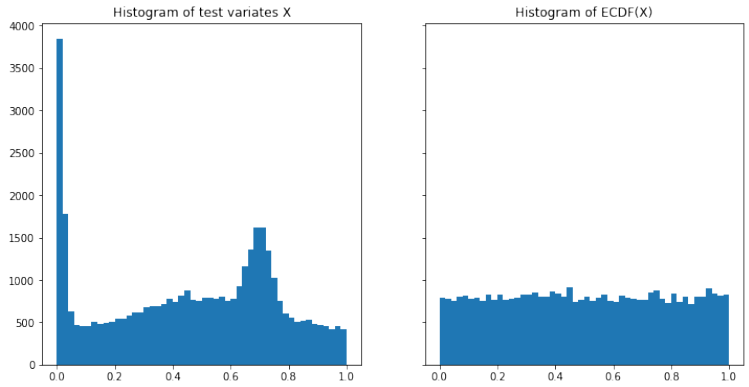


FIGURE 5.3: On the left, the plot shows the histogram of about 40,000 variates drawn from the same distribution $q(x)$ shown already in Figure 5.2. The right plot shows the histogram of the values $y = ECDF(x)$ obtained by applying the function $ECDF(x)$ shown in Figure 5.2 on the $\approx 40,000$ variates of the left plot.

Algorithm 2 makes it clear that when re-calibrating p-values for a label $\bar{y}$, the re-calibration set should contain the p-values of the re-calibration examples with label $\bar{y}$, because these p-values are supposed to be uniformly distributed.

Figures 5.2 and 5.3 illustrate how the ECDF-based re-calibration operates on variates from an example distribution. To recapitulate, we have shown that if by combining p-values with some arbitrary method we obtain p-values that are no longer valid, it is possible — provided we have some training data that we can use as recalibration data — to map them into p-value that exhibit the desired uniform distribution. One reviewer noticed that the ECDF-based re-calibration can be viewed as an application of CP in the special case in which the NCM is actually based on the combined p-value. In the next section, we propose a combination algorithm that indeed does not aim to produce valid p-values. The validity property will be recovered by applying the method described in this section to the output of the combination method.

5.2.4 Learning to combine

In the methods discussed so far, the combined p-value is a function only of the p-values from the individual CPs. The combination function does not take into account the object to which the p-value refers. Intuitively, it seems legitimate to wonder whether there are gains to be made by making the combination a function also of the object. Indeed, it may be argued that different underlying algorithms, especially when they are intrinsically different, might exhibit differences in relative performance on different objects: algorithm 1 might perform generally better than algorithms 2 and 3 in a certain region of the object space, whereas algorithm 2 might be better than the others in another region, and so on. The combination function could

be learned by means of an appropriate ML method. Although the idea of learning the combination function is not new — see (Balasubramanian, Chakraborty, and Panchanathan, 2015) — we believe that the approach we propose is novel.

Before setting up the learning problem, it may be useful to discuss what the ideal p-value combination should look like. Ultimately, the objective is to obtain a CP that outputs p-values for class c that are:

(a) uniformly distributed for objects belonging to the c class

(b) 0 for objects belonging to other classes.

In any practical application, the latter objective is really to obtain p-values as close to 0 as possible for objects belonging to the other classes. A CP that outputs such p-values would exhibit validity and maximum efficiency (where we define efficiency as the average size of the region prediction).

With these objectives in mind, we can formulate the problem of CP combination as the problem of predicting, given an object, which of the k base CPs will provide the p-value that best approximates requirements (a) and (b). In fact, a soft version of this formulation (as opposed to the 'hard' decision) might seek the weights with which to combine the individual CP predictions to best approximate requirements (a) and (b).

One possible setting of the problem is to use Logistic Regression with a training set constructed in the following manner. Suppose we have a set of objects $\mathbf{x}_i$, labels y_i, and p-values $p_{\text{act},i,j}$ and $p_{\text{inact},i,j}$. We will refer to this as combination training set. We use this to create a combiner training set, which is specific to the class c for which we are creating a combiner.
Let's consider the combiner for p_{Active}. For every example $(\mathbf{x}_i, y_i)$ in the combination training set, the combiner training set has k examples $(\mathbf{x}_i, 1), \ldots, (\mathbf{x}_i, k)$, i.e. one for each of the k base CPs. Note that in this combiner training set, the labels correspond to the base CPs. Each of these k examples that we create for each object $\mathbf{x}_i$, is assigned a weight $w_{i,j}$ (with $j = 1, \ldots, k$). The value of the weight $w_{i,j}$ is calculated as a function of the label y_i and of p-value $p_{\text{act},i}$. It is intended to express the desirability of following base CP j for predicting the p-value p_{act} for object $\mathbf{x}_i$. Note that the combiner training set is k times as big as the combination training set and that there are as many combiner training sets as possible labels (in the case we are discussing here, this number is 2, 'Active' and 'Inactive'). The predictions output by multinomial LR trained on the combiner training set for a test object $\mathbf{x}_{\ell+1}$ are k class "probabilities" $q_{\ell+1,j}$. Similarly to what stated for the weights, these "probabilities" are supposed to express numerically the desirability of following base CP j for object $\mathbf{x}_{\ell+1}$. The p-value for object $\mathbf{x}_{\ell+1}$ can be obtained by taking the p-value of the base CP with largest predicted "probability" (which we refer to as hard method) or by convex combination of base p-values using the probabilities as coefficients (soft method). It should be noted that this approach offers no guarantees that the resulting p-values

will be valid. Consequently, this combination method should be followed by the validity recovery discussed in Section 5.2.3.

It is evident that the success of this approach hinges on how we compute the weights $w_{i,j}$. We considered 2 different ways which are discussed next, but many more schemes are possible.

Before we discuss the ways of computing the weights, we hope to provide an element of further clarification in Figure 5.4 which shows an example of the application of the approach to a synthetic data set. In this admittedly contrived data set, the LR-based combiner manages to assign probabilities to each CPs that result in a combined CP with the best properties of each base CP (compare Figures 5.4c and 5.4d with Figure 5.4f.

Method 1: weighted

In this method the weight is higher for lower p-values when the example is not active and higher for higher p-values when the example is active (in the case of setting value of the p_{inactive} combiner, it would be the other way round).

$$w_{i,j} = \begin{cases} \frac{1-p_{\text{act},i,j}}{\sum_{j=1}^{k}(1-p_{\text{act},i,j})} & \text{when } y = \text{inact} \\ \frac{p_i}{\sum_{j=1}^{k} p_{\text{act},i,j}} & \text{when } y = \text{act} \end{cases} \tag{5.1}$$

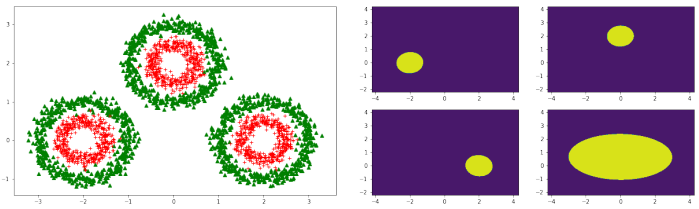
The weights are also normalized so that the weights for the same object add to 1. Also, per-class weighting is applied to compensate for imbalance. Specifically, the examples of the Active class are weighted by $N_{Inactive}/(N_{Inactive} + N_{Active})$ and those of the Inactive class by $N_{Active}/(N_{Inactive} + N_{Active})$. One issue with this method is that, while it may seem desirable to favour base CPs that produce higher p-values for Active examples, this does not appear to go in the direction of meeting requirement (a).

Method 2: reduced

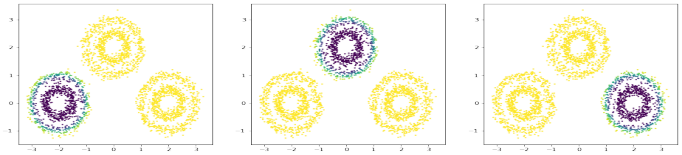
The 'reduced' method addresses the potential issue mentioned at the end of the previous paragraph by simply not including examples in the combiner training set if they have the Active (correct) label. In this way, we do not induce the combiner to favour higher p-values for Active examples.

$$w_{i,j} = \begin{cases} \frac{1-p_{\text{act},i,j}}{\sum_{j=1}^{k}(1-p_{\text{act},i,j})} & \text{when } y = \text{inact} \\ 0 & \text{when } y = \text{act} \end{cases} \tag{5.2}$$

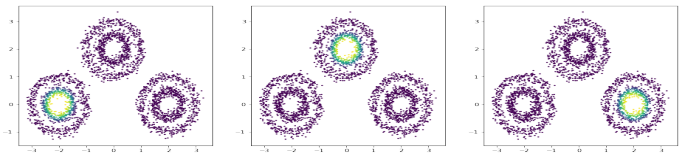
The examples that are assigned weight 0 can be discarded from the combiner training set. This is quite advantageous as it reduces the size of the training set, which is otherwise k as big as the combination set.



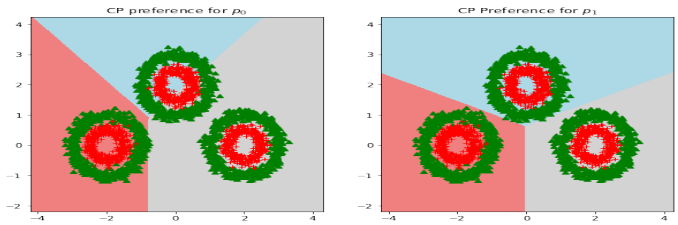
(A) (B)



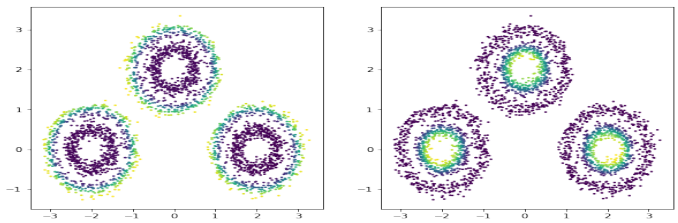
(C)



(D)



(E)



(F)

FIGURE 5.4: Figure 5.4a shows the synthetic data set, made up of three pairs of concentric circles, the inner containing positive examples and the outer negative examples. Three separate CPs are obtained by using three separate SVCs (polynomial kernel of degree 2) as underlying algorithms. Each SVC is trained on only one of the three clusters. The decision function for each of the SVCs is plotted in Figure 5.4b, with yellow denoting positive values and purple negative values. The lower bottom panel in Figure 5.4b refers to an SVC that was trained on the entire data set. Figures 5.4c and 5.4d show p_0 and p_1 from each of three CPs on a test set from the same distribution as the overall training set. Purple corresponds to low values, green to intermediate, and yellow to high values. In Figure 5.4e the color (coral, light blue, grey) corresponds to the CP with largest probability predicted by the LR combiner in that point. Finally, in 5.4f the values of p_0 and p_1 shown with the same color coding as in figures 5.4c and 5.4d.

5.3 An application: ExCAPE

In the sections that follow, we are going to present some results of the application of the combination methods to a real-world problem, which the authors encountered during their participation to a EU project called ExCAPE (Exascale for Compound Activity Prediction Engines) for the prediction of the activity of chemical compounds towards biological targets of interest.

Any advances in the ability to predict correctly such activity are of extreme interest to the pharmaceutical industry, as this can reduce substantially the number of lab assays required to identify new active compounds, thereby resulting directly in lower costs and a competitive advantage in the ability to innovate. While one could argue that this is not significantly novel as computer-aided drug discovery has been around for at least 30 years, the "Exascale" qualification in the name of the project alludes to one distinguishing feature of this research effort. ExCAPE explores methods that can be parallelized extensively, towards the goal of exploiting efficiently Exascale High Performance Computing (HPC) platforms, i.e. computing platforms capable of an aggregated 10^{18} FLOPS (Floating Point Operations per Second). This level of scalability is rapidly becoming relevant as it is expected that Exascale systems will become available in the 2020 timeframe[2].

5.3.1 Multiple Compound Activity Predictions

The specific problem tackled by ExCAPE is to predict the activity of a large number of compounds (several hundred thousands of compounds) towards a number of targets of interest (less than a thousand). The biological activity is known only for a fraction of the compound-target combinations. The challenge is to predict the activity in the large proportion of unknown compound-target cases. Different ML approaches are being pursued concurrently and separate heterogeneous models are being developed, namely Multi-task Deep Neural Networks and Bayesian Matrix Factorization. This created the need for a way to combine the predictions of these models (and possibly others) into one final set of predictions. Conformal Predictors can address this need by offering a solid framework for calibrating predictions expressed in different scales into p-values and then enabling their advantageous combination with the techniques described in this Chapter.

5.3.2 Chemoinformatics Data Sets

The data set used in the experiments in this Chapter was extracted from the latest version of the reference data base of the project, called ExCAPEDB (Sun et al., 2017b). The specific target, identified as IDH1, which was chosen because it's the

[2]Just for reference, at the time of start of the project (September 2015), the `top500.org` site was reporting that the most powerful supercomputer is the Sunway TaihuLight at the National Supercomputing Center in Wuxi, China, rated at $\approx 0.093 \times 10^{18}$ FLOPS, i.e. about 1/11 of what would be considered Exascale.

one with the largest number of tested compounds. The characteristics of the data set are reported in Table 5.1. The compounds are represented by means of features that capture structural properties of the molecule. From the standpoint of Machine Learning, chemoinformatics data sets can present the following challenges:

- Sparsity: the design matrix (the matrix that has a column for each feature and a row for each object) is often extremely sparse. In the case of the IDH1 data set, the fraction of non-zero entries is 0.01
- Imbalance: the prevalence of active compounds is often very small, 1% or less.

TABLE 5.1: Key statistics of the IDH1 data set. The lower part refers to the data sets used in each of the 50 runs.

Total number of examples	=	468,798	
Number of features	=	639,253	
Number of non-zero entries	=	31,523,836	
Density of the data set	=	0.0001	
Active compounds	=	2,194	(1.3%)
Proper training set size	=	200,000	
Calibration set size	=	100,000	
Test objects	=	168,798	

The three algorithms selected for this investigation are: linear SVC, Gradient Boosted Trees, and k Nearest Neighbours. The choice was driven by the intuition that ML algorithms based on inherently different approaches might have complementary strengths and weaknesses that a combination method could exploit to its advantage. Other choices might have been equally valid: for example, the same kernel method with different kernels, or the same regularized algorithm with different values of the regularization parameter, or distance-based algorithms with different types of distances.

5.4 Results and Discussion

5.4.1 Experimental setup

The experiments were primarily run on the computing facilities offered by the IT4Innovation National Supercomputing Center, Ostrava, Czech Republic. The Center operates two clusters, Anselm and Salomon, with 209 nodes and 1008 nodes respectively. The nodes are powerful servers, each equipped in case of Anselm with 16 cores (2 Intel Sandy Bridge E5-2665, 2.4 GHz) and 64 GB of RAM, and in the case of Salomon with 24 cores (2 Intel Xeon E5-2680v3, 2.5 GHz) and 132 GB of RAM.

The software was developed in Python, in large part using Jupyter Notebooks (Kluyver et al., 2016). The `scikit-learn` (Varoquaux et al., 2015) package

provided implementations for linear SVC, Gradient Boosted Trees and k Nearest Neighbours, whereas the preparation and handling of the data and of the results was carried out using the `numpy` (Walt, Colbert, and Varoquaux, 2011), `scipy` (Jones, Oliphant, Peterson, et al., 2001), and `pandas` (McKinney, 2010) packages. The distribution of the computation over the nodes obtained for a job relied on the `distributed` (Dask Development Team, 2016) package, which allows Python functions (or more in general Directed Acyclic Graphs (DAGs) of Python functions) to be submitted to a central "scheduler" for execution on a distributed cluster of workers.

Guaranteeing a full utilization of the nodes proved less straightforward than anticipated. Despite the computation consisting of fundamentally independent runs (a case of what is referred to as 'embarrassing parallelism'), it turned out that different algorithms had different CPU usage profiles and memory usage profiles, so the parameters governing the distribution (e.g. number of workers per node, maximum number of outstanding remote calls) had to be carefully tuned to avoid bottlenecks or memory overloads, especially on the node hosting the scheduler. The practical difficulty was compounded by the unexpected level of congestion on the Salomon and Anselm clusters, which meant that the number of nodes requested often had to be scaled back (all the way down to 6 or 8) to have a chance to be allowed to run.

The execution times for the LR-based Combination were dominated by the training times for the higher values of the regularization parameter C. For values larger than 1000, training time would be in the order of tens of minutes, whereas it would be of order of seconds for small values of C (heavy regularization). On a 16-node cluster on Anselm, one 10-fold CV over a range of 25 logarithmically-spaced values from 10^{-6} to 10^{6} required $\approx$ 1 hour.

5.4.2 Results

The original IDH1 data set was used to obtain 50 partitions into training, calibration, and testing sets. The training set size was chosen as 200,000 and the calibration set size as 100,000, leaving 168,798 examples for the test set. The splits were stratified, i.e. each set has the same proportion of the two classes as in the overall set. Linear SVC, Gradient Boosted Trees, and k Nearest Neighbour models were trained for each of the 50 splits and scores were obtained for the calibration and testing sets. Parameter optimization for was performed once for each of the algorithms. In all cases the reference metric was the F1 score. The potentially suboptimal performance deriving from a single setting of the parameters was not deemed to be a problem: the focus of the investigation is indeed on the combination of the Conformal Predictors, rather than on their individual quality. In fact, the variability of performance across splits might add a useful element of diversity in the relative merit of the predictors.

After obtaining p-values via Inductive Mondrian (Class-conditional) Conformal Predictors for each underlying algorithm using the NCM detailed in Table 5.2, we turned to the combination of p-values.

TABLE 5.2: The Non Conformity Measures for the three underlying algorithms

Underlying	**Non Conformity Measure α_i**	**Comment**
SVM	$-y_i f(x_i)$	where $f(x_i)$ is the SVM decision function
kNN	$\frac{\sum^{(k)}_{j \neq i: y_j = y_i} d(x_j, x_i)}{\sum^{(k)}_{j \neq i: y_j \neq y_i} d(x_j, x_i)}$	where $d(x_i, x_j)$ is the (Euclidean) distance; the summation is on the k smallest values of $d(x_j, x_i)$
XGB	$-y_i p(y_i = +1 \vert x_i)$	$p(y_i = +1 \vert x_i)$ is the probability of Activity estimated by the classifier

The application of the passive methods was obviously straightforward, whereas the LR-based combination required the majority of effort. For each of the runs, the LR classifier was obtained with a parameter optimization via 10-fold cross validation over 13 logarithmically-spaced values from 10^{-4} to 10^{2}. The calibration set (i.e. the set used for the Inductive CP) was also used as combination training set, from which the two combiner training sets (one for the combination of p_{active} and one for the combination of p_{inactive}) were derived. Note that, as explained in sec. 5.2.4, the combiner training sets are $k = 3$ times as large the combination set, hence their size is as large as 300,000.

The performance of the combined CP is examined on confusion matrices and ranking. The confusion matrix for CP region prediction is slightly different from the usual one for traditional classifiers as, in addition to a breakdown into correct and incorrect precise predictions, it includes a count of the empty predictions and a count of the uncertain predictions (the uncertain predictions occur when the region predictor contains more than one label). The metrics for CP confusion matrices include Precision and Recall. For reference, the definitions used in this study are summarized in Table 5.3. In order to have just one metric, we combine Precision and Recall into the F_1 score, which is their harmonic mean and is a special case of the F_β score, where β controls the "preference" of Precision vs. Recall.

The ranking performance is evaluated in terms of precision-at-k and average precision. The latter provides an overall view of how high in the ranking the examples belonging to the Positive class (here, the Active compounds) were placed. The precision-at-k offers an assessment of the ranking that is more focused on the top, which in many applications is what matters most. Precision-at-k is simply the fraction of Positive labels in the top k objects in the ranking. If, for instance, in a drug discovery setting only the top k compounds are chosen for actual lab testing,

then it is arguable that average precision is not relevant and we want to select the method that places the largest fraction of Actives in the top k. However, the situation might be different if the intended use is for test prioritization, in which case average precision might be relevant.

TABLE 5.3: Performance metrics used in this study

Metric	**Definition**	**Comment**
precision	$Pr = \frac{\lvert PP \cap TP \rvert}{\lvert PP \rvert}$	fraction of Positives among objects predicted as Positive
recall	$Re = \frac{\lvert PP \cap TP \rvert}{\lvert TP \rvert}$	fraction of all Positives included in the objects predicted as Positive
F_1 score	$F_1 = 2\frac{Pr \cdot Re}{Pr + Re}$	Harmonic mean of Precision and Recall
precision-at-k	$Pr@k = \frac{\lvert PP_k \cap TP \rvert}{\lvert PP_k \rvert}$	fraction of Positives in the top k ranked objects
average precision	$AP = \frac{1}{m}\sum_{j=1}^{m} Pr@k_j$	average over each Positive of the precision-at-k_j where k_j is its position in the ranking
TP: True Positive, PP: Predicted Positive, PP_k: Predicted Positive within the k top ranked objects, m: number of Positives, k_j: ranking position of j-th Positive		

Finally, statistical significance of the results is estimated. We use a non-parametric statistical test on paired observations, namely the Wilcoxon signed-rank test (Wilcoxon, 1945; Hollander and Wolfe, 1999). The null hypothesis of the Wilcoxon signed-rank test is that the distribution of the differences between elements of pairs is symmetrical around 0. However, in its basic form, the test does not apply to variables with discrete values such as counts but only to variables with continuous values, the reason being that the test was not designed to deal (*a*) with no differences in a pair and (*b*) with ties among the differences (occurrences of pairs with the same difference in absolute value). Variants have been proposed (by Wilcoxon himself, who suggested to disregard the observation pairs with no difference, and in (Pratt, 1959), who suggested a way to account for those) but the distribution of the statistic would change.

5.4.3 Region predictions

For the sake of brevity, we show only the counts for significance levels $\epsilon = 0.01$ (Table 5.4a) and $\epsilon = 0.05$ (Table 5.4b). We also report the error rate to provide a view on the validity deviation. If one compares error rates with and without it, ECDF-based calibration can be seen recovering validity very effectively.

TABLE 5.4: Confusion matrices for significance levels $\epsilon = 0.01$ (5.4a) and $\epsilon = 0.05$ (5.4b). Abbreviations: ApA - Active predicted Active, ApI - Active predicted Inactive, IpI - Inactive predicted Inactive, IpA - Inactive predicted Active. ApA, ApI, IpI, IpA refer to precise predictions, i.e cases in which the region prediction contained only one label; Empty refers to cases in which the prediction set was empty (both label could not be rejected); Uncertain refer to cases in which the region prediction contained more than one label. The Error rate allows to check whether the validity property is met. The Errors are the sum of ApI, IpI, and Empty. The number of test examples was 168,798. The values are averages over 50 runs.

(A)

Method	ApA	ApI	IpI	IpA	Empty	Uncertain	Error rate
SVC	598.06	22.58	17937.00	1648.08	0.00	148592.28	0.010
XGB	570.70	21.66	27929.04	1650.92	0.00	138625.68	0.010
kNN	339.12	20.64	24188.68	1666.54	0.00	142583.02	0.010
min	774.96	53.20	37467.94	3709.34	31.96	126760.60	0.022
max	217.02	1.52	9725.14	197.46	0.00	158656.86	0.001
mean	275.26	2.70	14459.18	314.64	0.00	153746.22	0.002
Fisher	941.06	83.00	50909.08	5542.32	0.06	111322.48	0.033
min ECDF	575.94	24.54	28212.10	1650.78	5.72	138328.92	0.010
max ECDF	468.72	20.98	27002.12	1645.46	0.00	139660.72	0.010
mean ECDF	515.78	21.38	28742.84	1656.98	0.00	137861.02	0.010
Fisher ECDF	626.96	22.86	30613.02	1655.20	0.00	135879.96	0.010
weighted soft	271.76	2.68	14572.52	306.80	0.00	153644.24	0.002
weighted hard	425.08	35.18	30855.70	874.06	0.66	136607.32	0.005
reduced soft	277.80	2.76	14587.00	317.54	0.00	153612.90	0.002
reduced hard	610.14	40.16	32519.74	2118.96	10.60	133498.40	0.013
weighted soft ECDF	524.30	37.86	35798.94	1753.50	0.00	130683.40	0.011
weighted hard ECDF	586.96	244.88	79351.32	2219.30	16.50	86379.04	0.015
reduced soft ECDF	525.06	21.70	28889.20	1658.02	0.00	137704.02	0.010
reduced hard ECDF	553.22	24.12	27090.12	1654.58	4.20	139471.76	0.010

(B)

Method	ApA	ApI	IpI	IpA	Empty	Uncertain	Error rate
SVC	1010.18	113.14	49788.14	8289.18	0.00	109597.36	0.050
XGB	992.04	109.82	55721.40	8286.66	0.00	103688.08	0.050
kNN	565.78	113.52	41779.04	8331.98	0.00	118007.68	0.050
min	1241.68	227.66	78018.16	16985.82	1650.70	70673.98	0.112
max	376.12	11.04	21675.18	840.58	0.00	145895.08	0.005
mean	531.12	27.92	31459.82	1795.16	0.00	134983.98	0.011
Fisher	1245.82	219.96	80820.78	15273.98	39.72	71197.74	0.092
min ECDF	998.04	110.14	52696.74	8100.72	215.38	106676.98	0.050
max ECDF	808.86	109.64	54665.34	8302.60	0.00	104911.56	0.050
mean ECDF	931.06	111.38	57696.06	8304.92	0.00	101754.58	0.050
Fisher ECDF	1054.82	114.66	59150.94	8294.36	0.72	100182.50	0.050
weighted soft	514.94	29.44	32349.68	1663.24	0.00	134240.70	0.010
weighted hard	694.66	172.30	66184.62	3776.18	51.72	97918.52	0.024
reduced soft	552.04	29.68	32329.56	1879.60	0.00	134007.12	0.011
reduced hard	1033.40	182.10	68065.44	10130.12	693.76	88693.18	0.065
weighted soft ECDF	934.64	195.72	77314.34	8692.52	0.00	81660.78	0.053
weighted hard ECDF	931.30	558.70	120985.68	9018.64	688.02	36615.66	0.061
reduced soft ECDF	969.38	111.82	58102.40	8304.52	0.00	101309.88	0.050
reduced hard ECDF	965.92	110.10	52332.94	8057.56	256.86	107074.62	0.050

The overall view of the strengths and weaknesses of the various methods across different significance levels is captured in Table 5.5, where we show the values of the F_1 scores for the Active class as well as for the Inactive class. The interpretation

TABLE 5.5: F1 score for precise predictions for various significance levels (averages over 50 runs). The best values are highlighted in bold.

	F_1 for the Active class				F_1 for the Inactive class			
epsilon	0.01	0.05	0.10	0.15	0.01	0.05	0.10	0.15
SVC	0.269	0.176	0.122	0.096	0.193	0.459	0.617	0.716
XGB	0.258	0.173	0.121	0.096	0.287	0.501	0.645	0.736
kNN	0.161	0.102	0.077	0.066	0.253	0.401	0.510	0.593
min	0.232	0.122	0.091	0.085	0.367	0.637	0.745	0.771
max	0.166	0.220	**0.209**	**0.177**	0.110	0.230	0.334	0.429
mean	0.198	0.235	0.187	0.142	0.159	0.317	0.477	0.603
Fisher	0.217	0.133	0.102	0.087	0.468	0.653	0.742	0.793
min ECDF	0.261	0.177	0.128	0.106	0.289	0.480	0.613	0.693
max ECDF	0.218	0.143	0.106	0.088	0.279	0.493	0.630	0.711
mean ECDF	0.236	0.163	0.120	0.097	0.294	0.514	0.656	0.743
Fisher ECDF	**0.280**	0.183	0.127	0.100	0.310	0.523	0.658	0.743
weighted soft	0.196	0.235	0.192	0.148	0.161	0.325	0.496	0.634
weighted hard	0.240	0.212	0.169	0.140	0.312	0.567	0.712	0.794
reduced soft	0.199	**0.239**	0.190	0.141	0.161	0.325	0.495	0.630
reduced hard	0.248	0.155	0.114	0.099	0.326	0.578	0.710	0.773
weighted soft ECDF	0.235	0.158	0.116	0.094	0.353	0.633	0.782	0.856
weighted hard ECDF	0.237	0.155	0.126	0.134	**0.643**	**0.839**	**0.897**	**0.896**
reduced soft ECDF	0.240	0.169	0.123	0.099	0.295	0.517	0.661	0.749
reduced hard ECDF	0.251	0.172	0.125	0.104	0.279	0.477	0.610	0.692

of the results is not straightforward. In the case of the Active class, there is no single method that outperforms consistently all the others in terms of the F_1 score. It is debatable whether non-valid methods should be considered, but they were reported to let the reader get a sense of how the ECDF-based re-calibration affects te performance. Among the simpler valid methods, "Fisher ECDF" improves over any base CP. To determine the statistical significance of the evidence against or in support of the equivalence of method A and method B, we computed the Wilcoxon statistic on the 50 pairs of observations, where one value in the pair comes from method A and the other from method B, both calculated on the same training/test dataset split which was constructed with the procedure described in Section 5.4.2. The statistical significance the Wilcoxon test attributes is at least at the level of $p < 1.3 \cdot 10^{-5}$. On the other hand, the LR-based combination methods fail to improve over the base CPs, especially in their valid variant.

In the case of the Inactive class, however, the "Weighted Hard ECDF-calibrated" method exhibits very good performance, with also "Weighted Soft ECDF-calibrated" scoring very high. The Wilcoxon test confirms the statistical significance: the hypothesis of no difference between "Weighted Hard ECDF-calibrated" and "Fisher" (the closest competitor among the simpler methods) is rejected at the level of $p < 8 \cdot 10^{-10}$.

One reviewer observed that, strictly speaking, one should account for the fact that multiple comparisons are performed here. One standard technique is to apply the Bonferroni correction (Wasserman, 2010b, Section 10.7): a threshold of α/k ensures that that the chance of at least one false Null Hypothesis rejection is less than or

equal to α when k tests are performed. In our case, $k = 21$ so for a statistical significance level of 5%, we should consider a threshold of $0.05/21 = 2.38 \cdot 10^{-3}$. The p-values mentioned above are well below this threshold.

As a final note, the difference in performance of LR-based methods between the two classes requires further investigation. One possibility is that per-class weighting mentioned in sec. 5.2.4 did not adequately compensate for the class imbalance.

5.4.4 Rankings

The second perspective under which we study the possible merits of CP combination is in terms of ranking. The test objects can be ranked according to their p-values in search of those that are most likely to be Active. In particular, we rank compounds by lowest $p_{inactive}$, i.e. by strength of the evidence against the Inactive hypothesis. We also rank compounds by highest p_{active}, although this may appear not to be justifiable within the framework of Statistical Hypothesis Testing. In fact, this appears empirically to provide good results. One justification might be that by ranking by highest p-value we are indeed ranking objects by how likely it would be to pick – from a set drawn from the same distribution as the training set and calibration set – an example that would be more contrary to the hypothesis of randomness.

The comparison of the ranking quality of the various methods is reported in Table 5.6, where we provide precision-at-k (we chose $k = 10, 25, 50, 100, 200$) and average precision. Note that, since the ECDF-based re-calibration is a monotone mapping, it does not affect the ranking, so there is no need to have separate cases for it. We report ranking precision for the Actives but not for the Inactives. Given the high imbalance (98.7% of the examples are Inactive), all methods managed to achieve the maximum score of 1 for all the 5 levels of Precision-at-k when ranking for inactivity. The Average Precision was also very high, exceeding 0.995 in all cases.

For the more challenging task of ranking for activity, the results indicate that combination in general improves significantly the precision across the board, compared to the base CPs. As a side note, it is surprising to see the kNN CP, which appeared to perform worse than SVC and XGB CPs in region prediction, achieve markedly higher precision-at-k (although the advantage disappeared for Average Precision).

LR-based combination, in particular in its "soft" variant, appears to be on a par with the simpler methods. While a simple visual inspection of Table 5.6 might suggest a tiny advantage for the "Weighted Soft, highest p_1" variant, the Wilcoxon test applied to the corresponding precision values for "Fisher, lowest p_0" and "weighted soft, highest p_1" reveals that any differences are of no statistical significance ($p >> 0.05$).

TABLE 5.6: Ranking precision for Actives expressed in terms of precision-at-k, for $k = 10, 25, 50, 100, 200$ and in terms of Average Precision. Best values in each column are highlighted in bold

CP type	Ranked by	k=10	k=25	k=50	k=100	k=200	Avg prec
SVC	Lowest p_0	0.652	0.647	0.638	0.618	0.580	0.180
	Highest p_1	0.632	0.648	0.636	0.617	0.579	0.180
XGB	Lowest p_0	0.614	0.580	0.578	0.545	0.509	0.165
	Highest p_1	0.604	0.601	0.575	0.547	0.511	0.165
kNN	Lowest p_0	0.698	0.703	0.691	0.657	0.603	0.106
	Highest p_1	0.720	0.714	0.690	0.656	0.603	0.106
min	Lowest p_0	0.644	0.653	0.652	0.623	0.583	0.177
	Highest p_1	0.754	0.749	0.719	0.687	0.627	0.168
max	Lowest p_0	0.752	0.752	0.718	0.684	0.627	0.152
	Highest p_1	0.616	0.656	0.648	0.613	0.574	0.156
mean	Lowest p_0	0.758	**0.759**	0.717	0.688	0.632	0.171
	Highest p_1	0.756	0.754	0.719	0.688	**0.636**	0.195
Fisher	Lowest p_0	0.754	0.746	0.719	0.685	**0.636**	**0.200**
	Highest p_1	0.764	0.756	0.718	0.688	0.635	0.189
weighted soft	Lowest p_0	0.764	0.756	0.718	0.689	0.631	0.170
	Highest p_1	0.760	0.754	**0.722**	**0.692**	**0.636**	**0.200**
weighted hard	Lowest p_0	0.694	0.699	0.675	0.636	0.587	0.165
	Highest p_1	0.656	0.688	0.678	0.647	0.605	0.180
reduced soft	Lowest p_0	0.756	0.756	0.716	0.691	0.631	0.176
	Highest p_1	**0.768**	0.753	0.717	0.689	0.633	0.190
reduced hard	Lowest p_0	0.650	0.646	0.626	0.599	0.558	0.166
	Highest p_1	0.710	0.714	0.681	0.645	0.595	0.165

5.4.5 Considerations and future directions

The LR-based p-value combination can be of benefit when the different base CPs exhibit different relative performance in regions of the object space that can be well separated by the combination classifier. It may be the case that the separation of the domains can be performed effectively with a function space of lower complexity than the ones that are required for the predictions themselves. In our example, the limited gains, if any, of the LR-based combination may be ascribed to a highly non-linear (hyper)surface of separation of the various domains, which the linear LR could not resolve. A future direction of research might be to incorporate non-linearity in the Classifier used for combination (for instance, with Kernel Logistic Regression). Another form of non-linearity to be experimented with is in the assignment of weights to the examples of the combiner training set, which eq. 5.1 and 5.2 set as linear function of the base CP p-values.

A further line of enquiry might be in approaching combination of CPs as a learning-to-rank problem, for which there is already a large body of research given its commercially valuable applications in Information retrieval (e.g. search engines). The p-value can in fact be interpreted as expressing a fractional rank (the p-value is the fraction of calibration set examples that are less conform than than hypothetical test example, so one can view this as the rank by non-conformity). With this approach, the combination would occur at the level of the NCM α_i.

Finally, on perhaps a more speculative note, combination opens new opportunities to assemble ML algorithms into classifiers capable of more complex tasks. We can envisage, for instance, a number of underlying algorithms with differing learning abilities providing predictions to a combiner which assesses their relative performance in the various regions of the problem space and learns how to best combine the individual predictions in a particular area. Underlying algorithms might be altogether different as in the real-world case discussed here or they might just have different level of regularization (to adapt to regions with different levels of noise) or they could be just trained each on a separate cluster of the overall data set, as in the synthetic data example, where their combination resolves regions of the data sets that one algorithm of the same class could not separate.

5.5 Conclusions

In this chapter we have discussed scalable methods for combining Conformal Predictors. The objective of CP combination was to improve efficiency, while preserving validity. We have applied established p-value combination methods from Statistical Hypothesis Testing as well as a novel method which in which the rule with which p-values are combined depends also on the object and is learned on a training set. Since all methods, in principle, impair validity, we have suggested a method to recover it with the use of a calibration set. A comparison of the methods was carried out on a challenging real-world chemoinformatics data set. Three base CPs were obtained as Inductive Mondrian CP with three different underlying ML algorithms on a strongly imbalanced data set (1.3% Active vs. 98.7% Inactive) with a total of almost half a million examples and over half a million features. The performance was assessed both in terms of region predictions and of ranking. We showed that combination methods such as Fisher's provide a statistically significant improvement over the individual CPs. In terms of precise predictions, ML-based combination methods showed no advantage for the Active class, but brought about significant improvements for the Inactive class; in terms of ranking, they improved on the base CPs, but not on the simpler combination methods.

Chapter 6

CP Combination with Neyman-Pearson Lemma

6.1 Introduction

In this chapter, we continue the investigation of CP combination methods started in the previous chapter, but, rather than evaluating performance directly on real-world data sets, we take a step back and conduct our study on a suitable synthetic data set. The main motivation for this choice is to control the statistical characteristics of p-values that we combine. In particular this will allow us to explore the effects of correlation among the p-values of the base CPs. This characterization of this effect is an important aspect because different methods are bound to behave differently in the face of correlation. We have to leave the convenience of the assumption of p-value independence and take stock of the fact that p-values from different base CPs for same object are going to exhibit some degree of correlation.

As in the previous chapter, the objective of the combination of Conformal Predictors is to increase efficiency, while preserving validity. In other words, we aim at reducing the average size of the prediction sets, while minimising any deviations of the error rate from the chosen significance level. We will restrict our scope to binary CPs, although the methods can be extended to more than two labels. In the context of binary classification, the maximisation of the efficiency corresponds to the minimisation of the occurrence of uncertain predictions (i.e. prediction sets that contain more than one label). For clarity, the setting for the CP combination is as follows: there are d CPs (which we will refer to as *base CPs*) and correspondingly d p-values $p^{(1)}, \ldots, p^{(d)}$ for a given label assignment to a test object. We are seeking a function $f(p^{(1)}, \ldots, p^{(d)})$ that computes a p-value that results in a valid and efficient CP, ideally for any joint distribution $P(p^{(1)}, \ldots, p^{(d)})$.

6.1.1 Merging functions

The first requirement for the combination method is that validity be preserved. In the previous chapter, we applied some elementary combination rules, which resulted in a combined value that in general no longer enjoyed the validity property and therefore was a p-value just in name. It turns out that conservative validity can be achieved with simple changes to those elementary rules. A comprehensive analysis of a family of combination methods that ensure validity without requiring assumptions on the independence of the p-values can be found in (Vovk and Wang, 2012). Table 6.1 lists some of methods discussed in the study, namely minimum, maximum, arithmetic average, geometric average. The charts in the left column in Figure 6.1 illustrate the cumulative distribution of the p-values arising from the combination functions and the merging functions (assuming independence of the base CPs). For the combined CP to preserve the validity property, the distribution of the combined p-values must remain uniform. Consequently, in the charts the traces should follow the dashed diagonal; if the trace is below the diagonal, the predictors are conservative (i.e. leading to fewer incorrect predictions than what the significance level allows for) and vice versa. The charts in the left column show that the

merging functions would result in conservative CPs when the base CP are independent. While the absence of independence requirements bestows a wide applicability to the methods, this universal validity guarantee appears to come at the expense of efficiency.

TABLE 6.1: Some merging functions. These are special cases of the more general merging functions listed in Table 1 of (Vovk and Wang, 2012). The merging function for the Minimum is also known as Bonferroni method.

	Combination function	Merging function
Arithmetic average	$p_{arith\_avg} = \frac{1}{d}\sum_{i=1}^{d} p^{(i)}$	$2 \cdot p_{arith\_avg}$
Geometric average	$p_{geom\_avg} = \left(\prod_{i=1}^{d} p^{(i)}\right)^{\frac{1}{d}}$	$e \cdot p_{geom\_avg}$
Min	$p_{min} = \min(p^{(1)}, \ldots, p^{(d)})$	$d \cdot p_{min}$
Max	$p_{max} = \max(p^{(1)}, \ldots, p^{(d)})$	p_{max}

6.2 When the distribution is known

In Section 5.2.3 we observed that if we denote by $F_X(x)$ the (continuous) cumulative distribution of a random variable X, the random variable $F_X(X)$ is uniformly distributed.

$$F_X(x) = \mathbb{P}\{X \leq x\} \Rightarrow F_X(X) \sim U[0,1] \tag{6.1}$$

We calibrated a combination of p-values by using its ECDF in place of cumulative distribution. This method has two advantages: (a) it allows complete freedom in the choice of the law used to combine p-values, (b) it can account for the dependence in the base p-values. These advantages, however, come at the cost of having to dedicate part of the training set to the estimation of the ECDF.

Here instead we note that the cumulative distribution of a function of uniformly distributed RVs can be expressed in closed form in several interesting cases, in particular under the somewhat restrictive assumption of independence. Indeed, the distributions of minimum, maximum, arithmetic average and geometric average of d independent uniformly distributed RVs are known and are presented in Table 6.2. We can obtain a valid CP combination by combining the p-values and then applying the distribution function. We refer to this class of methods as CDF-calibrated. Figure 6.1 shows the actual error rate vs. significance level for the four methods. The plots confirm that the p-values combined as prescribed above result in valid CPs (within statistical fluctuation). The effect of dependence between p-values will be discussed in section 6.8.1

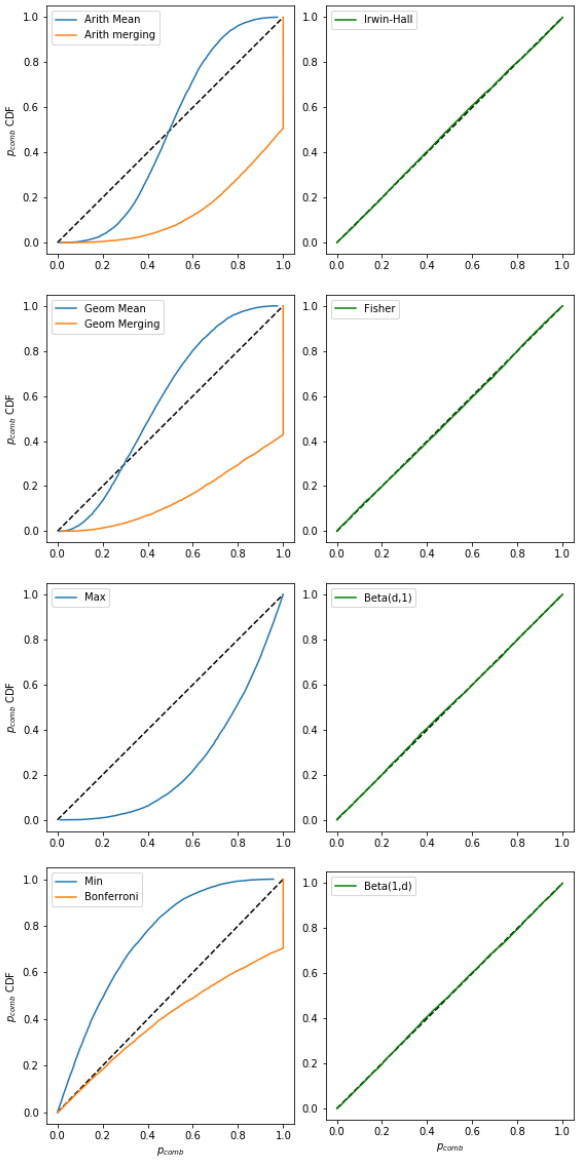


FIGURE 6.1: Comparison of validity of combination methods. Each plot shows the CDF of the combined p-value, when the base p-values are independent and uniformly distributed on $[0, 1]$. For the combined CP to be exactly valid, the trace should be the $(0, 0)$-$(1, 1)$ diagonal, indicated here with a dashed line. The left column shows the straightforward methods along with the merging variant that ensures (conservative) validity. The right column shows the CDF-calibrated versions.

TABLE 6.2: Some combination functions with known CDFs

Combination function	CDF	Comment
Arithmetic average (sum)	$\frac{1}{n!}\sum_{k=0}^{\lfloor t \rfloor}(-1)^k\binom{d}{k}(t-k)^d$	Irwin-Hall distribution
Geometric average (product)	$t\sum_{i=0}^{d-1}\frac{(-\log t)^i}{i!}$	Fisher formula(Fisher, 1948)
Min	`betainc()`	Beta(d, 1)
Max	`betainc()`	Beta(1,d)

6.3 Adaptive methods

The methods listed in Table 6.2 can be viewed as *a priori* methods, in the sense that the law with which the p-values are combined does not depend on the observed data. In this section we discuss a class of methods that adapt to the statistics of the observed data, albeit at the cost of having to set aside a fraction of the available observations for this purpose, thereby reducing the size of the training set for the underlying ML algorithms.

6.3.1 Multivariate ECDF

As stated in point (b) in the previous section, the ECDF calibration allows to recover validity after combining p-values with an arbitrary law. We illustrate this point further with an adaptive combination method, i.e. one in which the combination law varies with the observed data. The method we propose here combines p-values by computing the value of multivariate joint distribution of the p-values and then calibrates it to a $U[0,1]$ with the ECDF calibration discussed in section 6.2.
More formally, given RVs $X_1, \ldots, X_d$, the joint CDF is:

$$F_{X_1,\ldots,X_d}(x_1,\ldots,x_d) = \mathbb{P}\left\{X_1 \leq x_1, \ldots, X_d \leq x_d\right\}$$

The combination method discussed in this section can be expressed as:

$$p_{mecdf} = F_{P^{(1)},\ldots P^{(d)}}(p^{(1)}, \ldots p^{(d)}) \tag{6.2}$$

To perform the ECDF calibration, one must fit p_{mecdf} on a calibration set so that an ECDF of the p_{mecdf} can be computed. Then, the combined p-value is

$$p_{comb} = F_{P_{mecdf}}(p_{mecdf}) \tag{6.3}$$

Note that the calibration step above is needed to recover validity because the CDF property stated in eq. 6.1 for the univariate case does not hold in the multivariate case. That is, if $X^{(1)}, \dots X^{(d)}$ are independent uniformly distributed RVs, $F_{X^{(1)},\dots X^{(d)}}(X^{(1)}, \dots X^{(d)})$ is not distributed according to $U[0,1]$ [1]

Such CDF is unknown, but we can estimate it by computing the Multivariate ECDF on calibration data.

$$F_{\ell_{cal}}(x^{(1)}, \dots, x^{(d)}) = \frac{1}{\ell_{cal}} \sum_{i=1}^{\ell_{cal}} \prod_{k=1}^{d} \theta(x^{(k)} - x_i^{(k)})$$

6.4 Combination via Neyman-Pearson Lemma

The Neyman-Pearson Lemma (Neyman and Pearson, 1933) is a result in Statistical Hypothesis Testing on which basis it is possible to define a test statistic and a threshold so that the resulting significance test has Uniform Maximum Power (UMP). Here, *power* is defined as the probability to reject correctly the Null Hypothesis H_0.

This can be applied to CP by noticing that when we calculate, say, p_0, we assume as Null Hypothesis that the label is 0 and compute a p-value for the test object under this assumption. The p_0 can be interpreted as the probability of drawing from the same set as the calibration set an example that is as or more contrary to the hypothesis of randomness as the hypothetical test example.

The Neyman-Pearson Lemma (NPL) is particularly relevant to CP combination because it can optimise efficiency (i.e. results in smaller prediction sets). To see this, consider that with higher power one rejects more often H_0 when indeed it should be rejected. Consider also that the prediction set contains all the hypothetical label assignments that could not be rejected at the chosen significance level (as it contains all the labels y for which $p_y > \epsilon$). This means that the high the power of test, the less likely it will be that the prediction set contain incorrect labels. Not also that, in so for as validity is satisfied, the rate at which the correct label is in the prediction set is equal to the significance level. This approach to combination was discussed in (Toccaceli, 2019), but similar observations for CP in general appeared independently in (Sadinle, Lei, and Wasserman, 2019).

The NPL is based on the notion of likelihood (Fisher, 1932). For reference, we quote the definition of likelihood from (Edwards, 1984, Page 9): "The likelihood, $L(H|R)$, of the hypothesis H given data R, and a specific model, is proportional to $P(R|H)$, the constant of proportionality being arbitrary". The choice of defining likelihood up to an arbitrary constant (a constant, as Edwards points out, only within "any one application involving the same data and probability model") emphasizes that the notion makes sense not so much in itself but rather in comparison. Indeed, the Neyman-Pearson Lemma uses the ratio of likelihood of two hypotheses as test statistic.

[1] The distribution of $F_{X^{(1)},\dots X^{(d)}}(X^{(1)}, \dots X^{(d)})$ is referred to as Kendall distribution function (Genest and Rivest, 2001).

6.4.1 Statement of the Neyman-Pearson Lemma

Let's denote with $\mathbf{x}$ the data observed. Given a simple hypothesis H, we indicate with $\mathcal{L}(H \mid \mathbf{x})$ the likelihood. The most powerful test between two simple hypotheses H_0 and H_1 is the one that uses as test statistic the likelihood ratio:

$$\Lambda(\mathbf{x}) := \frac{\mathcal{L}(H_0 \mid \mathbf{x})}{\mathcal{L}(H_1 \mid \mathbf{x})} \tag{6.4}$$

and as threshold the value η that satisfies

$$\epsilon = \mathbb{P}\left[\Lambda(\mathbf{X}) \leq \eta \mid H_0\right] \tag{6.5}$$

where ϵ is the significance level.

6.4.2 Application to Combination of Conformal Predictors

As stated previously, we restrict our scope to binary Conformal Predictors. Let's assume that we have k separate CPs, each using some different underlying ML algorithm, producing for a test object the k p-values $p_{\bar{y}}^{(1)}, \ldots, p_{\bar{y}}^{(k)}$ for the hypothetical label assignment $y = \bar{y}$.

The k p-values are what was denoted as $\mathbf{x}$ in the statement of the NPL. The H_0 hypothesis is $y = \bar{y}$ and the H_1 hypothesis is $y \neq \bar{y}$.
Hence $\mathcal{L}(H_0 \mid \mathbf{x}) = \mathbb{P}\left[p_{\bar{y}}^{(1)}, \ldots, p_{\bar{y}}^{(k)} \,\middle|\, y = \bar{y}\right]$ and $\mathcal{L}(H_1 \mid \mathbf{x}) = \mathbb{P}\left[p_{\bar{y}}^{(1)}, \ldots, p_{\bar{y}}^{(k)} \,\middle|\, y \neq \bar{y}\right]$.
The likelihood ratio $\Lambda(p_{\bar{y}}^{(1)}, \ldots, p_{\bar{y}}^{(k)})$ can be then computed as:

$$\Lambda(p_{\bar{y}}^{(1)}, \ldots, p_{\bar{y}}^{(k)}) = \frac{\mathbb{P}\left[p_{\bar{y}}^{(1)}, \ldots, p_{\bar{y}}^{(k)} \,\middle|\, y = \bar{y}\right]}{\mathbb{P}\left[p_{\bar{y}}^{(1)}, \ldots, p_{\bar{y}}^{(k)} \,\middle|\, y \neq \bar{y}\right]} \tag{6.6}$$

If we denote by $F_\Lambda(\lambda)$ the (cumulative) distribution function of $\Lambda(\bar{y}^{(1)}, \ldots, p_{\bar{y}}^{(k)})$ given H_0, the p-value for the combination is then obtained as:

$$p_{\bar{y}}^{(\mathrm{NP})} = F_\Lambda(\Lambda(p_{\bar{y}}^{(1)}, \ldots, p_{\bar{y}}^{(k)})) \tag{6.7}$$

To justify the last equation, consider that eq. 6.5 can be expressed also as $\epsilon = F_\Lambda(\eta)$. In principle, there is no need to compute explicitly η. An alternative way of interpreting eq. 6.5 is saying is that the hypothesis should be rejected when the value of cumulative distribution function for the hypothetical example is less than or equal to ϵ. By computing $p_{\bar{y}}^{(\mathrm{NP})}$ according to eq. 6.7 we achieve precisely that. This procedure is performed twice, once for each possible value of the label of the test object.

The probabilities $\mathbb{P}\left[p_{\bar{y}}^{(1)}, \ldots, p_{\bar{y}}^{(k)} \,\middle|\, y = \bar{y}\right]$ and $\mathbb{P}\left[p_{\bar{y}}^{(1)}, \ldots, p_{\bar{y}}^{(k)} \,\middle|\, y \neq \bar{y}\right]$ are to be estimated using a calibration set extracted from the overall training set. In fact, it is possible to estimate directly the ratio rather than the individual probabilities separately and then take their ratio.

6.5 Implementation of Neyman-Pearson Combination

The method just described promises optimal efficiency, with no assumptions on the absence of correlation or even dependence among CPs. In principle, this method should outperform any other combination method, at least in terms of efficiency. However, the method revolves around the ratio of the likelihoods under the null and under the alternative hypothesis.
The estimation of a density, let alone a density ratio (Sugiyama, Suzuki, and Kanamori, 2012), is an ill-posed problem. The reader is referred to (Vapnik and Izmailov, 2015b, Section 3.1) for further discussion of this important fact. The difficulty of this estimation is further compounded by its multivariate nature. It is therefore important to investigate the question of how the method actually performs in practice, especially when only limited amounts of noisy data are available. The performance of the methods chosen for the estimation of the density ratio is critical for this method to realise its full potential.

Two approaches are described below, namely Naïve Neyman-Pearson and V-Matrix.

6.5.1 Naïve Neyman-Pearson

To apply the method described in section 6.4.2, one needs to compute the likelihood $\mathcal{L}(H_1|\mathbf{x})$, that is, the density $\mathbb{P}\left[X|H_1\right]$ evaluated at $\mathbf{x}$. In particular, we are looking for the likelihood for the joint event of $p_1, p_2, \ldots, p_k$.

To make the estimation more tractable, one approach is to make the *naïve* assumption that the p-values are independent (this is analogous to the independence assumption made in Naïve Bayes). So the density of the joint event can be calculated as the product of the densities of each of the simple events.

Consequently, a method that we refer to here as Naïve Neyman-Pearson obtains first an estimate of the (marginal) density of each of the p-values and then simply calculates the likelihood for the joint event as product of those densities. If we assume that the $p_{\bar{y}}$ are computed by a valid CP, they are uniformly distributed so the likelihood $\mathcal{L}(\theta_0|p)$ for each of the k p-value is 1. So, the NPL statistic can be expressed as:

$$\Lambda(X) = \frac{1}{\prod_{i=1}^{k} f_1(p_i)} \qquad \text{where} \quad X = (P_1, P_2, \ldots, P_k)$$

To obtain the combined p-value, we start from recalling that the threshold η is chosen so that the significance level ϵ is:

$$\epsilon = \mathbb{P}\left[\Lambda(X) \leq \eta | H_0\right]$$

We can therefore transform the statistic value λ into a p-value by applying to it the CDF of the NPL statistic evaluated conditional on H_0.

$$p_{\text{comb}} = CDF_{H_0}(\lambda)$$

where

$$CDF_{H_0}(\lambda) = \mathbb{P}\left[\Lambda(X) \leq \lambda \,|\, H_0\right]$$

Note that this ensures that the p-value for the Null Hypothesis be uniformly distributed. One obvious limitation of this approach is that it is hardly ever the case that the p-values of the base CPs are independent.

6.5.2 V-Matrix

To account fully for an arbitrary dependence between p-values one has to attempt to estimate the multivariate joint density ratio. Density estimation is central to statistical inference and the problem has been studied for decades, resulting in a variety of methods. A rigorous approach was proposed first in (Vapnik, 1995), and then in (Vapnik, Braga, and Izmailov, 2015) and (Vapnik and Izmailov, 2015a). The version of the method considered here is referred to as V-Matrix method and is described in (Vapnik and Izmailov, 2015b). We will recap just the key points here and refer the reader to papers just cited for the full derivation and all the attendant details.

Direct Constructive Setting

Let's consider first the problem of density estimation. Let's assume that we are given ℓ d-dimensional samples $x_i = (x_i^{(1)}, \ldots, x_i^{(d)})$ from a (cumulative) distribution $F(x)$. We are seeking a density $f(x)$ such that:

$$\int_{-\infty}^{x} f(t)dt = F(x)$$

The distribution F(x) is unknown, but from the samples we can compute the empirical cumulative distribution

$$F_\ell(x^{(1)}, \ldots, x^{(d)}) = \frac{1}{\ell} \sum_{i=1}^{\ell} \prod_{k=1}^{d} \theta(x^{(k)} - x_i^{(k)})$$

where $\theta(x)$ denotes the step function. A key result in Vapnik-Chervonenkis theory (Vapnik and Izmailov, 2015b, Equation 4) guarantees that the uniform convergence of $F_\ell(x)$ to $F(x)$ as $\ell \to \infty$ is fast:

$$\mathbb{P}\left[\sup_x |F_\ell(x) - F(x)| \geq \epsilon\right] \leq \exp\left(-\left(\epsilon^2 \ell - \log \ell\right)\right)]$$

In other words, the cumulative distribution function can be estimated from a limited amount of samples with a relatively small error. The direct constructive setting consists in estimating the density $f()$ as solution of the integral equation using the approximation given by empirical distribution function $F_\ell(x)$ in place of the actual but unknown $F(x)$.

Density Ratio

In the case of the density ratio estimation, we are given ℓ_{num} d-dimensional samples $x_i = (x_i^{(1)}, \ldots, x_i^{(d)})$ from a (cumulative) distribution $F_{num}(x)$ and ℓ_{den} d-dimensional samples $x_i = (x_i^{(1)}, \ldots, x_i^{(d)})$ from a (cumulative) distribution $F_{den}(x)$. We are seeking a density $r(x)$ such that:

$$\int_{-\infty}^{x} r(t)\, dF_{den}(t) = F_{num}(x) \tag{6.8}$$

Analogously to the density estimation case above, we estimate $r(x)$ by solving the integral equation after replacing $F_{num}(x)$ and $F_{den}(x)$ with their empirical counterparts, $F_{\ell_{num}}(x)$ and $F_{\ell_{den}}(x)$

Solution via regularization method

The integral equations arising from the direct constructive setting are *ill-posed*, in the sense that their solutions are not stable: informally stated, small changes to the right-hand side can result in significant changes to the solution. In the case of the density ratio problem, the difficulty is compounded by the fact that not only the right-side, but the left side are approximately defined. Problems of this nature are called stochastic ill-posed problems.

The method proposed in (Vapnik, 1995, Chapter 7) is to seek the function $r(x)$ that minimizes the sum of the L_2 distance (in a chosen metric space E) between $F_{\ell_{num}}()$ and the left-hand side of eq. 6.8 and a regularization term. The solution is sought in a Reproducing Kernel Hilbert Space of kernel $K(\cdot,\cdot)$ and has the form:

$$f(x) = \sum_{i=1}^{\ell} \alpha_i K(X_i, x) = A^T \mathcal{K}(x) \tag{6.9}$$

where $A = (\alpha_1, \ldots, \alpha_{\ell_{den}})^T$ and $\mathcal{K}(x)$ is a vector of $K(X_i, x), i = 1, \ldots, \ell_{den}$.
The functional to minimize is expressed as:

$$A^T K V K A - 2\left(\frac{\ell_{den}}{\ell_{num}}\right) A^T K V^* \mathbf{1}_{\ell_{num}} + \gamma A^T K A \tag{6.10}$$

where K is the $(\ell_{den} \times \ell_{den})$ matrix with elements $K(X_i, X_j)$, $i, j = 1, \ldots, \ell_{den}$ and V and V^* are matrices that reflect the geometry of the observed data. In addition, the solution should take non-negative values and should integrate to 1. These two

constraints are expressed in terms of the observed data as:

$$KA \geq \mathbf{0}_{\ell_{den}} \tag{6.11}$$

$$\frac{1}{\ell_{den}} A^T K V^* \mathbf{1}_{\ell_{num}} = 1 \tag{6.12}$$

V-Matrix

The $(\ell_{den} \times \ell_{den})$ V matrix and $(\ell_{num} \times \ell_{den})$ V^* matrix mentioned in eq. 6.10 have elements

$$V_{i,j} = \int \theta(x - X_i)\theta(x - X_j)\sigma(x)d\mu(x). \tag{6.13}$$

where $\sigma(x)$ and $\mu(x)$ are respectively a weighting function and a measure that arise in the definition of distance in the metric space E. $\sigma(x)$ and $\mu(x)$ allow to craft the definition of distance to suit the specific statistical inference problem. With the choice of $\sigma(x) = 1$ and μ the uniform measure, assuming that data belongs to the upper-bounded interval $[-\infty, u]$,

$$V_{i,j} = \prod_{k=1}^{d} \left(u - \max \left\{ X_i^{(k)}, X_j^{(k)} \right\} \right) \tag{6.14}$$

6.6 Experiments with synthetic data

In (Heard and Rubin-Delanchy, 2018), the Neyman-Pearson Lemma is used in combination with the common assumption that the distribution of p-value under the alternative hypothesis is of the form Beta(a, b) with $a \in (0, 1]$ and $b \in [1, +\infty)$. In particular, the paper claims that Fisher's method is the most powerful when the alternative hypothesis $p \sim$ Beta$(0.5, 1)$. One wonders how warranted this common Beta(a, b) assumption is (see also (Sellke, Bayarri, and Berger, 2001)), in particular in the specific context of Conformal Predictors. On a purely intuitive basis, it is not outside the realm of possibility that there may be some deeper relationship between the distribution of CP p-values, which can be seen as rank transformed scores, and the order statistics of the uniform distribution which indeed happen to be Beta-distributed random variables. However, in the present study it was felt that it would be more realistic to generate p-values computing them via CP on appropriate distributions of NCMs, rather than generating them from arbitrary distributions.

6.6.1 A realistic model of NCMs

The NCMs can in principle be obtained from a very wide variety of ML algorithms. One can model the distribution of NCMs as a mixture of two distributions, one for NCMs for examples of one class and the other for the NCMs of the other class. Figure 6.2 shows an example of the histogram of the distribution that arise in a real-life case. Of course, markedly different distributions can arise from different methods,

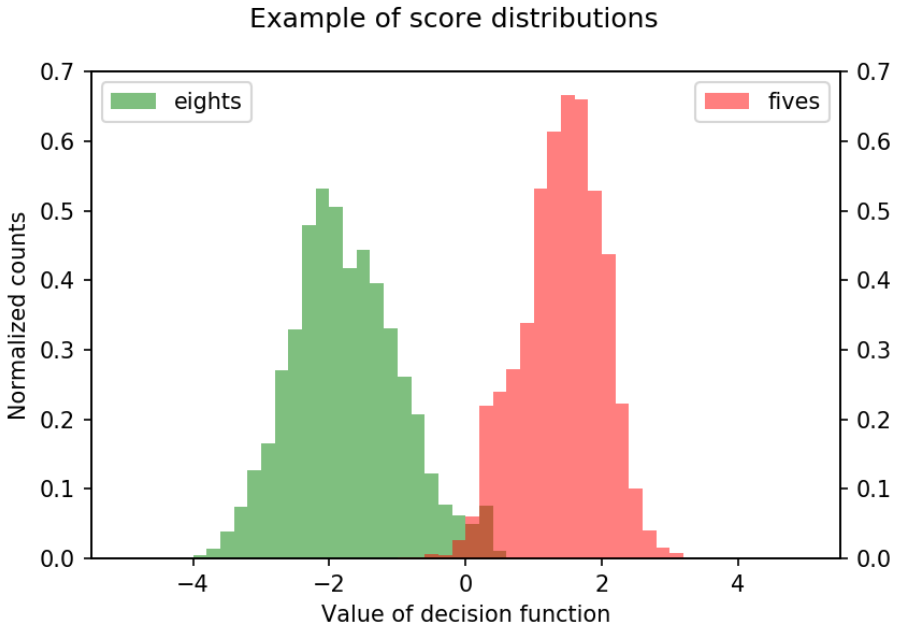


FIGURE 6.2: Example of score distribution from a real-life dataset. These scores were obtained as SVC decision function values. The SVC was trained to classify a dataset containing 28x28 images of handwritten "5" and "8" digits from the well-known MNIST dataset.

but the example suggests that it might be relevant to study the case in which the scores for the two classes are distributed as two Gaussians.

Throughout the rest of the Chapter, we assume that the NCMs are derived from the scores simply by a monotone transformation, e.g. changing the sign, as needed.

In Figure 6.3 four main cases are identified. In all four cases, the Gaussian distributions have mean -1 and +1. What differs is variance, which reflects the relative uncertainty of the prediction for each class. The four cases allow us to study the effect of larger and asymmetric overlaps.

6.7 The distribution of p-values under the Alternative Hypothesis

The distribution of p-values under the Null Hypothesis is uniform by construction. The distribution of p-values under the Alternative Hypothesis is determined by the distribution of the Nonconformity Measure. If we denote as $P_0(\alpha)$ the CDF of alphas under H_0 and $p_1(\alpha)$ the PDF for the NCM under H_1, the p-values can be viewed as Random Variables obtained as:

$$\mathbb{P}\left[\alpha_0 \geq \alpha_1\right] = 1 - P_0(A_1)$$

where A_1 is a random variable whose realisations are the NCM α_1 under the Alternative Hypothesis H_1.

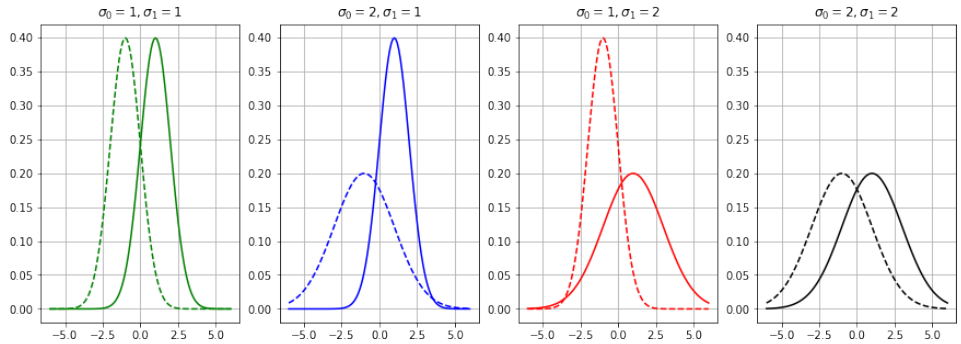
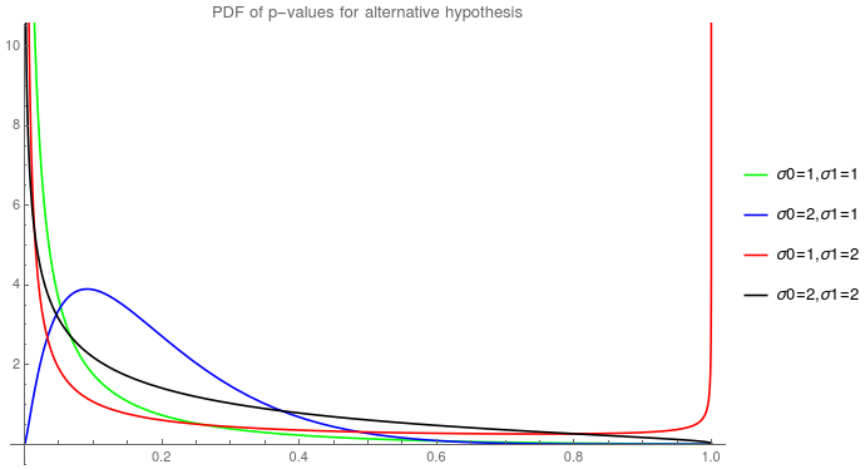


FIGURE 6.3: Cases of NCM distributions. The dashed lines correspond to H_0 and the solid lines to H_1. The cases at the extreme left and extreme right assume that the underlying method had the same uncertainty in classifying test examples of either label. The cases differ in the amount of "overlap". The plots in the middle refer to cases in which the classifier had more uncertainty for the Null Hypothesis label (blue) and less uncertainty for the Null Hypothesis label (red)



case	σ_0	σ_1	**PDF of p-values under** H_1
green	1	1	$\exp\left(-2\sqrt{2}\,\text{InvErfc}(2-2x)-2\right)$
blue	2	1	$2\exp\left(-3\,\text{InvErfc}^2(2-2x)-4\sqrt{2}\,\text{InvErfc}(2-2x)-2\right)$
red	1	2	$\frac{1}{2}\exp\left(\frac{1}{4}\left(3\,\text{InvErfc}^2(2-2x)-2\sqrt{2}\,\text{InvErfc}(2-2x)-2\right)\right)$
black	2	2	$\exp\left(\sqrt{2}\,(-\text{InvErfc}(2-2x))-\frac{1}{2}\right)$

FIGURE 6.4: The PDF of the p-values under H_1. InvErfc() is the inverse complementary error function. The complementary error function is $\text{Erfc}(x) = \frac{2}{\sqrt{\pi}}\int_x^{+\infty} e^{-t^2}\,dt$.

For the four cases shown in Figure 6.3 it is possible to express in closed form the PDF of the p-values under the Alternative Hypothesis. The equations are given in the table in Figure 6.4[2].

It is interesting to observe that while the black and green traces could be in qualitative agreement with the common assumption mentioned earlier in section that the Alternative Hypothesis p-values follow some form of Beta distribution, the blue and the red traces show a different behaviour. But it could be argued that the vertical asymptote at $p = 1$ for the red trace and the behaviour near $p = 0$ for the blue line have to do with the possibly unrealistic long tails of the wider Gaussian. Both occurrences can be explained by the fact that for sufficiently small and sufficiently large the PDF of the Gaussian of larger variance has larger values than that of the Gaussian of lower variance.

6.8 Experimental results

The CP combination methods discussed in the previous sections were applied to **two** base CPs, denoted here with CP_a and CP_b. Calibration sets and test sets had both 5,000 examples, with the two classes being represented in equal proportions. (Obviously, there is no proper training set as the NCMs are "simulated").

The code was entirely written in Python with the help of Jupyter Notebooks, using `numpy`, `scipy`, `numba` and `scikit-learn`. The V-Matrix implementation used the `cvxopt` package for the solution of the Quadratic Programming problem.

We assumed that in each CP the NCMs for examples of the two labels could be distributed in the one of four possible cases discussed in the previous section, namely:

- $\sigma_0^2 = 1, \sigma_1^2 = 1$
- $\sigma_0^2 = 1, \sigma_1^2 = 4$
- $\sigma_0^2 = 4, \sigma_1^2 = 1$
- $\sigma_0^2 = 4, \sigma_1^2 = 4$

The total number of pairings of cases, discounting symmetries, is $\frac{(n+1)n}{2} = \frac{5 \cdot 4}{2} = 10$. For each of these pairings, we then used 3 different settings of correlation between the NCMs of CP_a and CP_b. We generated NCM sets with covariance 0 (in fact, they were not only uncorrelated, but independent), covariance 0.8, and covariance -0.8. Figure 6.5 illustrates the NCMs and the resulting p-values for the 3 different covariance values in the case with $\sigma_0 = 2$, $\sigma_1 = 2$. From the NCMs, p-values for the test objects were computed according to the MICP framework. The p-values were then used to compute the prediction sets and the results, in turn, were summarised into confusion matrices, which provide counts of correct, incorrect, empty, and uncertain

[2]The symbolic expressions were computed using Mathematica®.

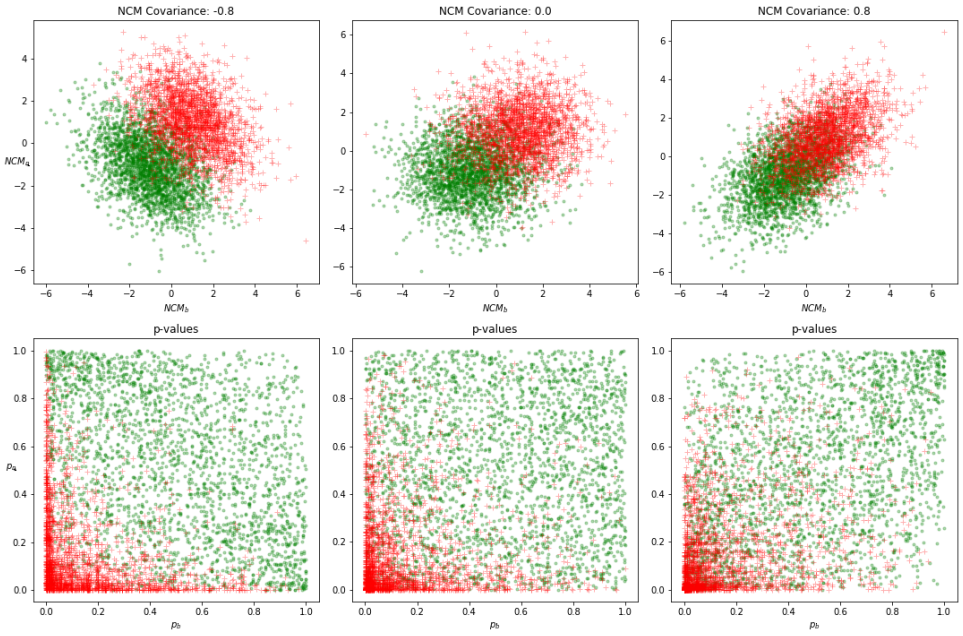


FIGURE 6.5: Example of NCM sets with same variance, but different covariance. The top row shows the NCMs, the bottom row the resulting MICP p-values p_0. The y-axis refers to CP a and the x-axis to CP b. The red crosses correspond to the data points with label 1 and the green dots to the data points with label 0. Note the variance and covariance referred to here are those of each component of the Gaussian mixture, i.e. between the NCMs for set a and set b for label 0 and the NCMs for set a and set b for label 1.

predictions. To assess validity, the confusion matrices were computed for different significance levels, namely 0.01, 0.05, 0.1, 0.2.

As stated in Section 6.1, the objective considered in this Chapter is to improve efficiency, while preserving validity. So, the analysis that follows will focus on these two properties. The results for the $10 \times 3 \times 4 = 120$ cases (each repeated 25 times) are summarized in Tables 6.3, 6.4, 6.5, 6.6. In Figure 6.6 we show one representative case out of the 120.

In the charts, the entries are grouped as follows:

base predictors: The base CPs, identified as "a" and "b"

reference: the theoretical optimal methods under the assumption of independence, listed as "Naive Neyman-Pearson Ideal"

basic methods: arithmetic average, geometric average, maximum, minimum

merging functions: methods discussed in sec. 6.1.1 which guarantee (conservative) validity

CDF calibrated methods: arithmetic average (CDF), geometric average (CDF), maximum (CDF), minimum (CDF)

ECDF calibrated methods: arithmetic average (ECDF), geometric average (ECDF), maximum (ECDF), minimum (ECDF)

adaptive methods: Multivariate ECDF, Naive Neyman-Pearson (histogram), V-Matrix

An orange background is applied to the groups as a visual reminder of their significant deviations from validity. Table 6.3 reports the average fraction of uncertain predictions (inversely related to efficiency) for one value of significance level, namely $\epsilon = 0.05$. In Tables 6.4,6.5,6.6, we present the rankings in terms of efficiency, averaged over the 25 repetitions, and disaggregated by significance level, correlation, and variances, respectively. Also, in these tables we removed the methods that deviate significantly from validity so that the ranking is fairer.

6.8.1 Findings

The analysis of the results confirms the observations made earlier while describing the methods. More specifically, taking Figure 6.6 as a representative case, we can in fact see that

1. all the basic methods (arithmetic average, geometric average, maximum, minimum) exhibit deviation from validity, i.e. in the top plot, their error rate averages are far from the horizontal dashed green line which corresponds to the significance level.

2. the merging functions are extremely conservative, perhaps with the exception of Bonferroni for low value of significance level.

3. The CDF calibrated methods are indeed valid when the base predictors are independent, but exhibit different forms of deviation in the presence of correlation.

4. ECDF calibrated methods exhibit small deviation form validity also in the presence of correlation.

The basic methods and the merging functions will not be discussed further as their deviation from exact validity defeats the purpose of CP combination considered in this study.

Turning now our attention to efficiency, the results shown in Tables 6.3, 6.4,6.5,6.6 support the following findings:

1. In the case of positive correlation, there is not much efficiency improvement in combining (refer to Table 6.3) This may be intuitively justified by observing that if the p-value are strongly correlated, they convey the same information. Bringing this to an extreme, we would not expect to see any improvement by combining a CP with itself. Conversely, negative correlation offers the best opportunities for efficiency gains.

2. The accuracy and robustness of density ratio estimation is critical to the success of the application of the Neyman-Pearson method. When a simple method

such as histogram is used, the N-P method often fails to improve CP efficiency. The improvements require the use of a more accurate and robust method such V-Matrix.

3. The superiority of V-Matrix method fails to manifest itself fully for very low values of the significance level (refer to Figure 6.4). This is indicative of inaccuracy in the low end of the prediction range (i.e. for values close to 0). This may be overcome with a better choice of kernel. In this study, the Gaussian RBF kernel was chosen after some experiments with Polynomial and INK-Spline Kernel failed to provide encouraging results. It is possible that a kernel on a [0,1] support and with a better suited functional form might perform better.

4. The Multivariate ECDF method performs well and it is competitive with respect to V-Matrix. This is particularly interesting given the simplicity of the method and the absence of any parameters that need optimisation (the V-Matrix method has a regularisation parameter and, possibly, also a kernel parameter).

Table 6.3 shows the fraction of uncertain predictions for each method at significance level $\epsilon = 0.05$ under the different scenarios.

6.9 Future directions

Several aspects of this study in this chapter could be developed further. First of all, we applied the combination methods only to 2 base CPs. It would be worthwhile to investigate how the performance varies when more than 2 CPs are combined. The likelihood ratio estimation, which is critical for the Neyman-Pearson method, becomes more challenging when more p-values are combined. It would be interesting to determine whether the difficulties in estimating it cancel the advantages that the Neyman-Pearson method theoretically guarantees. Also, a natural application of the combination techniques is in Cross-Conformal Predictors (Vovk, 2015), so it would be interesting to study how the methods perform in that context. More in general, a comparison should carried out on real-world data sets and a variety of underlying ML methods to gain a better understanding of their merits and limitations.

6.10 Conclusions

We propose to use a combination method based on the Neyman-Pearson Lemma, to achieve the CP combination objective of improving efficiency while preserving exact validity, at the cost of using part of the training set for calibration purposes. The critical component of the method is density ratio estimation and we showed on a realistic synthetic data set that an accurate and robust method such V-Matrix can be used successfully. We also showed that other approximate methods exist that provide, with much less complexity, only slightly inferior results.

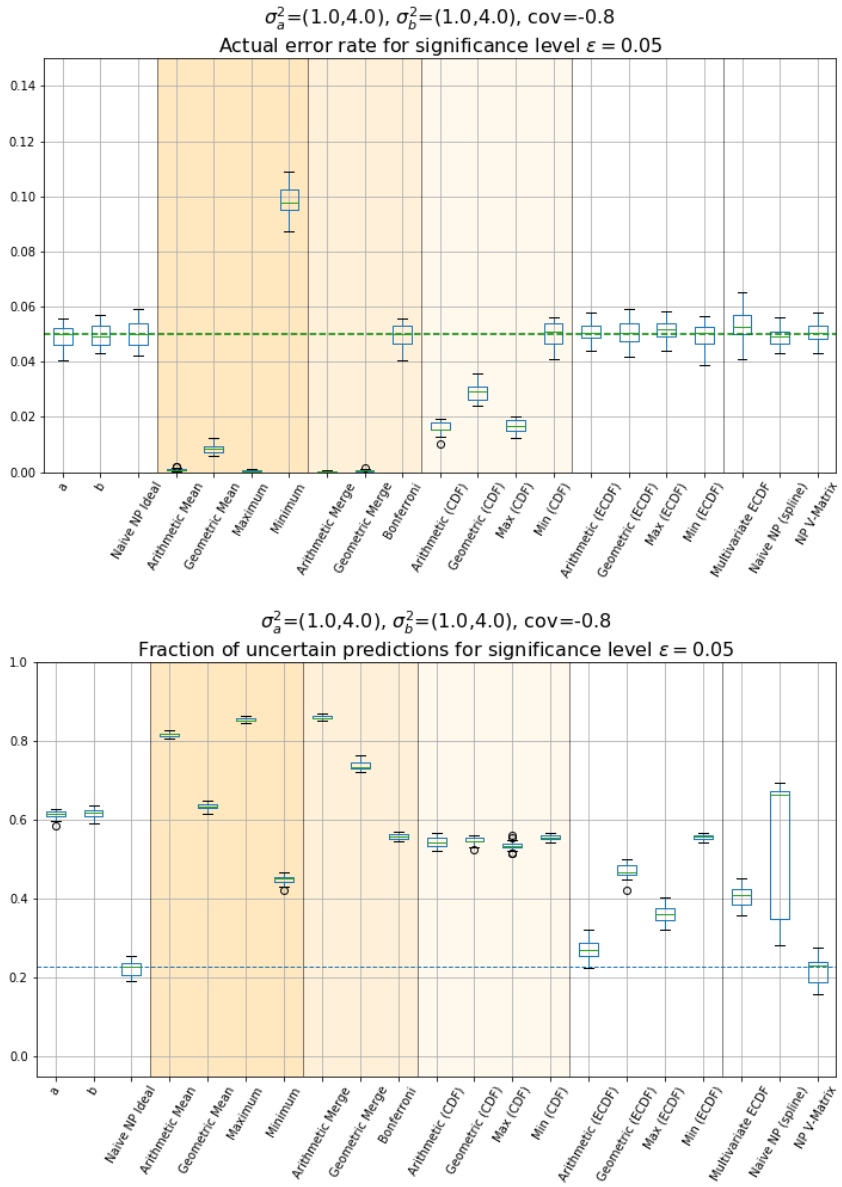


FIGURE 6.6: Boxplots for the error rate and for the fraction of uncertain predictions for one "representative" scenario. In top chart, which refers to error rate, the methods in the shaded areas show significant validity deviations (compare with dashed green line, which corresponds to the significance level). In the bottom chart, which refers to the fraction of uncertain predictions, we can see that NP V-Matrix outperform all the other methods. (The green line is the median rate for "Naïve NP" which we take here as reference.)

TABLE 6.3: Fraction of uncertain predictions for significance level $\epsilon = 0.05$. There are 10 scenarios in terms of the variances of the NCMs for the two labels and the 2 CPs. In the headings the two number for σ^2 are the variances for the NCMs for examples with label "0" and for examples with label "1". In each such scenario, different levels of correlations (-0.8, 0, +0.8) were injected between corresponding NCMs. The reported values are averages over 25 runs. The lower the fraction, the higher the CP efficiency.

	σ_a^2=(1.0,1.0), σ_b^2=(1.0,1.0)			σ_a^2=(1.0,1.0), σ_b^2=(1.0,4.0)			σ_a^2=(1.0,1.0), σ_b^2=(4.0,1.0)			σ_a^2=(1.0,1.0), σ_b^2=(4.0,4.0)			σ_a^2=(1.0,4.0), σ_b^2=(1.0,4.0)		
	-0.800	0.000	0.800	-0.800	0.000	0.800	-0.800	0.000	0.800	-0.800	0.000	0.800	-0.800	0.000	0.800
a	0.313	0.310	0.313	0.313	0.309	0.312	0.310	0.312	0.309	0.303	0.315	0.316	0.617	0.617	0.611
b	0.312	0.315	0.313	0.615	0.616	0.613	0.616	0.617	0.611	0.690	0.692	0.692	0.618	0.615	0.617
Naive NP Ideal	0.000	0.067	0.271	0.026	0.148	0.264	0.025	0.149	0.269	0.091	0.230	0.332	0.227	0.301	0.339
Arithmetic Mean	0.521	0.463	0.351	0.779	0.685	0.599	0.776	0.688	0.599	0.805	0.744	0.687	0.818	0.787	0.725
Geometric Mean	0.080	0.248	0.303	0.405	0.407	0.418	0.407	0.406	0.415	0.470	0.482	0.490	0.633	0.617	0.593
Maximum	0.704	0.591	0.433	0.846	0.783	0.720	0.845	0.784	0.719	0.886	0.831	0.780	0.853	0.830	0.779
Minimum	0.018	0.094	0.192	0.168	0.193	0.224	0.168	0.194	0.219	0.188	0.216	0.244	0.451	0.436	0.471
Arithmetic Merge	0.777	0.638	0.507	0.861	0.809	0.745	0.861	0.810	0.743	0.918	0.865	0.812	0.860	0.841	0.809
Geometric Merge	0.502	0.506	0.522	0.680	0.641	0.623	0.680	0.640	0.623	0.786	0.746	0.721	0.735	0.735	0.728
Bonferroni	0.092	0.214	0.344	0.306	0.328	0.363	0.303	0.328	0.357	0.347	0.371	0.409	0.558	0.560	0.583
Arithmetic (CDF)	0.131	0.112	0.024	0.277	0.257	0.175	0.280	0.257	0.176	0.407	0.362	0.303	0.544	0.445	0.354
Geometric (CDF)	0.007	0.095	0.147	0.184	0.226	0.241	0.183	0.226	0.242	0.244	0.276	0.297	0.547	0.467	0.435
Max (CDF)	0.215	0.170	0.038	0.385	0.320	0.199	0.390	0.320	0.206	0.479	0.424	0.351	0.535	0.451	0.313
Min (CDF)	0.090	0.210	0.338	0.301	0.323	0.359	0.300	0.324	0.355	0.338	0.368	0.404	0.557	0.559	0.580
Arithmetic (ECDF)	0.014	0.110	0.269	0.087	0.255	0.391	0.086	0.252	0.401	0.205	0.357	0.464	0.271	0.443	0.525
Geometric (ECDF)	0.000	0.095	0.270	0.082	0.222	0.334	0.081	0.225	0.345	0.158	0.273	0.374	0.469	0.465	0.504
Max (ECDF)	0.065	0.169	0.282	0.202	0.314	0.451	0.201	0.317	0.460	0.316	0.422	0.504	0.361	0.443	0.494
Min (ECDF)	0.090	0.208	0.288	0.305	0.322	0.322	0.303	0.324	0.327	0.344	0.369	0.390	0.558	0.555	0.559
Multivariate ECDF	0.000	0.090	0.267	0.024	0.216	0.314	0.025	0.221	0.325	0.109	0.266	0.368	0.409	0.456	0.493
Naive NP (histo)	0.479	0.533	0.630	0.533	0.576	0.675	0.528	0.574	0.684	0.133	0.453	0.361	0.665	0.613	0.499
NP V-Matrix	0.004	0.078	0.267	0.014	0.170	0.281	0.012	0.174	0.283	0.109	0.239	0.312	0.230	0.380	0.341

	σ_a^2=(1.0,4.0), σ_b^2=(4.0,1.0)			σ_a^2=(1.0,4.0), σ_b^2=(4.0,4.0)			σ_a^2=(4.0,1.0), σ_b^2=(4.0,1.0)			σ_a^2=(4.0,1.0), σ_b^2=(4.0,4.0)			σ_a^2=(4.0,4.0), σ_b^2=(4.0,4.0)		
	-0.800	0.000	0.800	-0.800	0.000	0.800	-0.800	0.000	0.800	-0.800	0.000	0.800	-0.800	0.000	0.800
a	0.612	0.616	0.615	0.612	0.616	0.616	0.615	0.615	0.616	0.617	0.617	0.615	0.691	0.692	0.689
b	0.618	0.617	0.616	0.694	0.694	0.689	0.618	0.613	0.615	0.692	0.690	0.691	0.690	0.693	0.691
Naive NP Ideal	0.196	0.283	0.343	0.333	0.404	0.456	0.227	0.296	0.340	0.329	0.405	0.455	0.471	0.544	0.587
Arithmetic Mean	0.905	0.846	0.779	0.892	0.854	0.809	0.814	0.789	0.725	0.890	0.853	0.811	0.915	0.887	0.853
Geometric Mean	0.518	0.502	0.493	0.673	0.650	0.632	0.634	0.619	0.592	0.671	0.651	0.634	0.780	0.759	0.739
Maximum	0.971	0.941	0.907	0.935	0.911	0.880	0.850	0.831	0.781	0.934	0.911	0.881	0.952	0.931	0.906
Minimum	0.318	0.324	0.334	0.420	0.429	0.441	0.452	0.436	0.472	0.426	0.427	0.443	0.470	0.480	0.487
Arithmetic Merge	0.981	0.954	0.922	0.946	0.927	0.902	0.858	0.842	0.809	0.946	0.927	0.903	0.969	0.952	0.930
Geometric Merge	0.681	0.664	0.649	0.817	0.801	0.790	0.736	0.735	0.726	0.816	0.801	0.786	0.941	0.919	0.902
Bonferroni	0.446	0.445	0.446	0.571	0.574	0.583	0.559	0.560	0.580	0.573	0.574	0.586	0.650	0.655	0.662
Arithmetic (CDF)	0.427	0.373	0.305	0.554	0.495	0.438	0.554	0.449	0.357	0.551	0.495	0.438	0.612	0.570	0.524
Geometric (CDF)	0.376	0.365	0.353	0.503	0.482	0.470	0.553	0.471	0.434	0.508	0.484	0.472	0.577	0.565	0.554
Max (CDF)	0.530	0.452	0.362	0.596	0.533	0.460	0.539	0.452	0.313	0.591	0.534	0.457	0.642	0.601	0.548
Min (CDF)	0.444	0.442	0.445	0.566	0.570	0.579	0.557	0.555	0.579	0.566	0.571	0.583	0.647	0.651	0.659
Arithmetic (ECDF)	0.218	0.369	0.484	0.392	0.492	0.557	0.272	0.445	0.522	0.392	0.490	0.556	0.508	0.568	0.602
Geometric (ECDF)	0.299	0.361	0.410	0.437	0.484	0.524	0.477	0.469	0.510	0.437	0.480	0.517	0.517	0.562	0.598
Max (ECDF)	0.323	0.449	0.562	0.452	0.526	0.580	0.358	0.452	0.495	0.457	0.529	0.575	0.560	0.596	0.622
Min (ECDF)	0.442	0.442	0.433	0.569	0.565	0.574	0.560	0.556	0.560	0.570	0.572	0.572	0.647	0.649	0.655
Multivariate ECDF	0.257	0.358	0.414	0.400	0.473	0.522	0.409	0.467	0.487	0.400	0.481	0.517	0.493	0.556	0.598
Naive NP (histo)	0.252	0.644	0.671	0.640	0.452	0.494	0.667	0.612	0.724	0.373	0.438	0.493	0.488	0.552	0.595
NP V-Matrix	0.249	0.320	0.346	0.396	0.442	0.453	0.203	0.376	0.330	0.399	0.434	0.451	0.484	0.547	0.592

TABLE 6.4: Average rank of the method when sorted by efficiency, as a function of significance level ϵ. Apart from the $\epsilon = 0.01$ case at the left, NP V-Matrix is consistently the best after the Naïve NP Ideal.

	0.010	0.050	0.100	0.150	0.200
Naive NP Ideal	1.433	1.400	1.400	1.167	1.000
Arithmetic (ECDF)	5.467	4.900	4.600	4.567	5.167
Geometric (ECDF)	4.100	4.733	4.767	4.500	3.700
Max (ECDF)	6.933	6.233	6.400	6.833	7.200
Min (ECDF)	5.800	6.933	7.200	7.300	7.233
Multivariate ECDF	2.100	3.733	4.000	4.267	4.000
Naive NP (histo)	6.733	6.100	5.600	5.033	4.500
NP V-Matrix	3.433	1.967	2.033	2.333	3.200

TABLE 6.5: Average rank of the method when sorted by efficiency, as a function of correlation. NP V-Matrix is consistently the best after the Naïve NP Ideal.

	-0.800	0.000	0.800
Naive NP Ideal	1.280	1.020	1.540
Arithmetic (ECDF)	4.660	4.880	5.280
Geometric (ECDF)	4.440	4.340	4.300
Max (ECDF)	6.740	6.760	6.660
Min (ECDF)	7.320	7.000	6.360
Multivariate ECDF	3.160	3.520	4.180
Naive NP (histo)	5.320	5.780	5.680
NP V-Matrix	3.080	2.700	2.000

TABLE 6.6: Average rank of the method when sorted by efficiency, for the various scenarios of σ_a and σ_b. With the exception of the two case at the left, NP V-Matrix is consistently the best after the Naïve NP Ideal.

	(1.0,1.0), (1.0,1.0)	(1.0,1.0), (1.0,4.0)	(1.0,1.0), (4.0,1.0)	(1.0,1.0), (4.0,4.0)	(1.0,4.0), (1.0,4.0)	(1.0,4.0), (4.0,1.0)	(1.0,4.0), (4.0,4.0)	(4.0,1.0), (4.0,1.0)	(4.0,1.0), (4.0,4.0)	(4.0,4.0), (4.0,4.0)
Naive NP Ideal	1.467	1.333	1.200	1.333	1.400	1.000	1.200	1.533	1.200	1.133
Arithmetic (ECDF)	5.067	5.467	5.400	6.000	4.533	4.400	4.667	4.400	5.000	4.467
Geometric (ECDF)	2.667	3.533	3.533	3.933	5.267	4.533	4.933	4.867	5.067	5.267
Max (ECDF)	6.867	7.000	7.000	7.600	5.400	6.800	7.133	5.200	7.200	7.000
Min (ECDF)	7.000	6.067	6.067	6.667	6.867	6.467	7.400	6.800	7.600	8.000
Multivariate ECDF	2.867	2.667	2.800	3.133	4.000	3.933	4.200	3.800	4.600	4.200
Naive NP (histo)	6.333	7.000	7.133	4.267	6.800	6.133	4.400	7.800	3.067	3.000
NP V-Matrix	3.733	2.933	2.867	3.067	1.733	2.733	2.067	1.600	2.267	2.933

Chapter 7

Conformal Predictive Distributions

7.1 Introduction

This chapter introduces a recent development in the probabilistic prediction of a continuous variable that takes advantage of the CP framework and the validity guarantees that it brings about.

In this chapter, we introduce the notion of Predictive Distribution in a frequentist framework and present Conformal Predictive Distributions (Vovk et al., 2019), a Machine Learning approach that computes estimates of probability distributions of a continuous variable without relying on prior assumptions on the mathematical form of the distributions. The method produces probability distributions for which it is possible to prove a statistical *validity* property (also known as *guaranteed coverage*), which ensures that the estimated probability corresponds to relative frequency. The only assumption is that the test data is sampled from the same (unknown) distribution as the training data.

7.1.1 Outline

Sections 7.2 and 7.3 delimit the setting and frame the problem of predictive distribution. Section 7.4 introduces formally the framework of Conformal Predictive Distributions following very closely (Vovk et al., 2019; Vovk et al., 2018) and in section 7.6 we will apply it to problems of prediction in a drug development setting. The method performance will be evaluated not only with the usual metrics for point predictions (such as R^2), but also with respect to probabilistic properties, such as validity and sharpness.

7.2 Generalities

Let's start by introducing our basic setting for the prediction problem. We have a training set consisting of n observations $z_i = (x_i, y_i) \in \mathbf{X} \times \mathbf{Y} = \mathbf{X} \times \mathbb{R}$, $i = 1, \ldots, n$. Each observation $z_i = (x_i, y_i)$, $i = 1, \ldots, n+1$, consists of two components, namely the object x_i belonging to a space $\mathbf{X}$ that we call the *object space* and the label y_i that belongs to a space $\mathbf{Y}$ that we call the *label space*. In the problem we are going to discuss, each of the x_i can be for instance a vector of reals or integers, encoding physicochemical properties or structural properties of a compound (as already discussed in Section 3.1), whereas the label correspond to the value of an assay of interest (e.g. a pharmacokinetic property such as hPPB, human Plasma Protein Binding) for the compound. The prediction problem is, given a test object x_{n+1}, to predict its label y_{n+1}. Here we are interested in the case of regression, where the label space is the real line, $\mathbf{Y} = \mathbb{R}$. In particular we are interested in the problem of probability forecasting, so our prediction takes the form of a probability distribution on the label space $\mathbf{Y}$.

7.3 Predictive Distributions

We assume that all the observations (in the training set as well as in the test set) are generated independently by a fixed but unknown distribution. In other words, the data is independent and identically distributed (i.i.d.) and no other assumption is made on the nature of the distribution. In particular, we do not assume that the distribution belongs to a family of which we want to estimate the parameters.

Intuitively, for every choice of training set $z_1, \ldots, z_n$ and for every test object x_{n+1}, we seek an estimate of the Cumulative Distribution Function of the label, i.e. a function $Q()$ such that:

$$\Pr(y_{n+1} < y) = Q(z_1, \ldots, z_n, x_{n+1}, y) \tag{7.1}$$

We can reformulate the equation above in terms of a validity property for predictive distributions.
Chosen a significance level $\epsilon \in [0, 1]$, we can define a prediction interval $\Gamma_\epsilon(z_1, \ldots, z_n, x_{n+1})$ (i.e. function of the training set and the test object) as:

$$\Gamma_\epsilon(z_1, \ldots, z_n, x_{n+1}) := \left\{ y \in \mathbf{Y} : \frac{\epsilon}{2} < Q(z_1, \ldots, z_n, x_{n+1}, y) < 1 - \frac{\epsilon}{2} \right\} \tag{7.2}$$

In words, the predicted interval contains the values of y for which $Q(z_1, \ldots, z_n, x_{n+1}, y)$ falls in a strip centred on $\frac{1}{2}$ with width $1 - \epsilon$ and is just one possible definition of the interval. In fact, any strip of width $1 - \epsilon$ would do. Figure 7.1 illustrates how the PD establishes relationships between intervals of y and their probabilities. The validity property of predictive distributions can then be

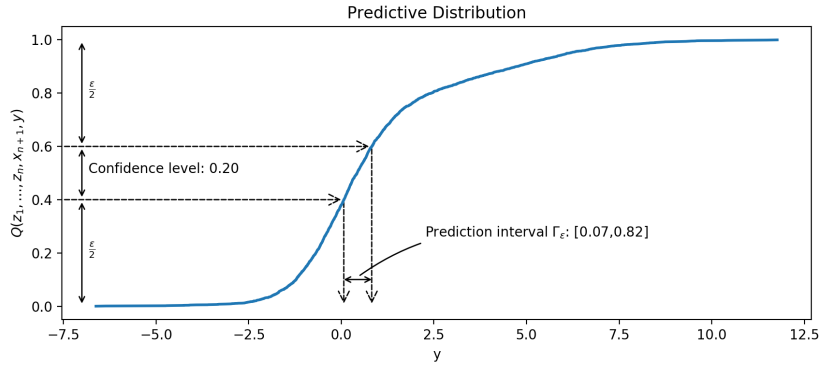


FIGURE 7.1: Example of Predictive Distribution. The chart shows a fictitious $Q(z_1, \ldots, z_n, x_{n+1}, y)$ and illustrates the construction of $\Gamma_\epsilon(z_1, \ldots, z_n, x_{n+1})$ defined by Eq. 7.2, assuming $\epsilon = 0.8$.

stated as:

$$\Pr\left(y_{n+1} \notin \Gamma_\epsilon(z_1, \ldots, z_n, x_{n+1})\right) = \epsilon \tag{7.3}$$

In words, the validity property is a guarantee that the probability of the actual label y_{n+1} being in the prediction interval Γ_ϵ is indeed ϵ. In practical terms, this means that given a test set of size m and for any choice of the significance level ϵ, if we compute

the $\Gamma_\epsilon(x_i)$ for each test object x_i, we will see that the actual labels y_i are in $\Gamma_\epsilon(x_i)$ with relative frequency approaching ϵ. This reformulation of Equation 7.1 highlights the close connection between Predictive Distributions and Conformal Regression.

The formal definition of the predictive distributions that we are going to use here can be best understood if one recalls the property of CDF stated in Section 5.2.3, i.e. given a random variable X and its CDF $F_X(t) \equiv \mathbb{P}[X \leq t]$ which we will assume continuous, the random variable $Y \equiv F_X(X)$ follows the Uniform distribution.

A function $Q : \mathbf{Z}^{n+1} \times [0,1] \to [0,1]$ is a *randomized predictive system* (RPS) if:

R1a For each training sequence $(z_1, \ldots, z_n) \in \mathbf{Z}^n$ and each test object $x_{n+1} \in \mathbf{X}$, the function $Q(z_1, \ldots, z_n, (x_{n+1}, y), \tau)$ is monotonically increasing in both y and τ.

R1b For each training sequence $(z_1, \ldots, z_n) \in \mathbf{Z}^n$ and each test object $x_{n+1} \in \mathbf{X}$,

$$\lim_{y \to -\infty} Q(z_1, \ldots, z_n, (x_{n+1}, y), 0) = 0,$$
$$\lim_{y \to +\infty} Q(z_1, \ldots, z_n, (x_{n+1}, y), 1) = 1.$$

R2 For any probability measure P on $\mathbf{Z}$, $Q(z_1, \ldots, z_n, z_{n+1}, \tau) \sim U$ when $(z_1, \ldots, z_{n+1}, \tau) \sim P^{n+1} \times U$.

where $U = U[0,1]$ be the uniform probability distribution on the interval $[0,1]$.

7.4 Conformal predictive distributions

This section describes how Predictive distributions can be obtained with a variant of Conformal Prediction.

Conformalized least square regression and kernel ridge regression have been studied by Burnaev and Nazarov (Burnaev and Nazarov, 2017) in the form of prediction intervals and was extended by (Vovk et al., 2019) to predictive distributions. Because of some technical subtleties, the formal definition of predictive distribution relies on a less intuitive property, rather than the validity guarantee stated in the previous section.

A *conformity measure* is a measurable function $A : \mathbf{Z}^{n+1} \to \mathbb{R}$ that is invariant with respect to permutations of the first n observations. A natural definition is

$$A(z_1, \ldots, z_{n+1}) := y_{n+1} - \hat{y}_{n+1}, \tag{7.4}$$

$\hat{y}_{n+1}$ being the prediction for y_{n+1} computed from x_{n+1} and $z_1, \ldots, z_{n+1}$ as training sequence.

The Conformal Predictive Distribution (or more formally the Conformal Transducer) determined by a conformity measure A is defined as

$$Q(z_1,\dots,z_n,(x_{n+1},y),\tau) := \frac{1}{n+1}\left(\left|\left\{i=1,\dots,n+1 \mid \alpha_i^y < \alpha_{n+1}^y\right\}\right| + \tau\left|\left\{i=1,\dots,n+1 \mid \alpha_i^y = \alpha_{n+1}^y\right\}\right|\right), \quad (7.5)$$

where $(z_1,\dots,z_n) \in \mathbf{Z}^n$ is a training sequence, $x_{n+1} \in \mathbf{X}$ is a test object, and for each $y \in \mathbb{R}$ the corresponding *conformity scores* α_i^y are defined by

$$\begin{aligned} \alpha_i^y &:= A(z_1,\dots,z_{i-1},z_{i+1},\dots,z_n,(x_{n+1},y),z_i), \qquad i=1,\dots,n, \\ \alpha_{n+1}^y &:= A(z_1,\dots,z_n,(x_{n+1},y)). \end{aligned} \quad (7.6)$$

Figure 7.2 shows a hypothetical CPD. Note that, while the randomization element might seem confusing, in practice it can be ignored. The $\frac{1}{n+1}$ width of the bands becomes negligible for any reasonably sized training set (also considering the errors inherent in the data sets).

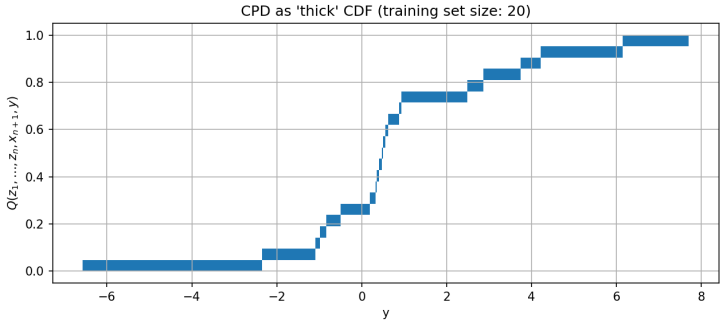


FIGURE 7.2: Example of Conformal Predictive Distribution. The CPD can be informally viewed as a "thick" discrete CDF. The CPD appears to be made of up to $n+1$ horizontal strips, each with width $\frac{1}{n+1}$ (or a multiple of it, in case of ties $\alpha_i^y == \alpha_{\ell+1}^y$) because of the random element introduced with the RV τ.

A *conformal predictive system* (CPS) is a function which is both a conformal transducer and a randomized predictive system. A *conformal predictive distribution* (CPD) is a function output by a conformal predictive system Q for a test object x_{n+1}.

7.4.1 Additional requirement

The equations in the previous sections are almost indistinguishable from those used in Transductive Conformal Regression (CR). There is, however, an important difference between CR and CPD: in CPD not all equivariant functions are acceptable Conformity Measures. For the resulting $Q()$ to have the properties of CDF, the conformity measure must be such that:

- $\alpha_{n+1}^y - \alpha_i^y$ is a monotonically increasing function of $y \in \mathbb{R}$

- $\lim_{y \to -\infty} (\alpha_{n+1}^{y} - \alpha_i^{y}) = -\infty$
- $\lim_{y \to +\infty} (\alpha_{n+1}^{y} - \alpha_i^{y}) = +\infty$

A simple example for which the properties above hold is the conformity measure $y - \hat{y}_{n+1}$ where $\hat{y}_{n+1}$ is the estimate obtained with K nearest neighbours regression, as shown in (Vovk et al., 2019, Section 2.2).

7.5 Kernel Ridge Regression Prediction Machine

The form of CPD that we applied to chemoinformatics data is the Kernel Ridge Regression Prediction Machine (KRRPM); it is the conformal transducer determined by a conformity measure computed using Kernel Ridge Regression (KRR). KRR seemed a particularly appropriate regression algorithm because it is a regularized method (which allows to control the "complexity" of the regression model and therefore combat overfitting) and can handle non-linearity (through the use of a kernel). KRRPM is described in (Vovk et al., 2018) and we refer the reader to that paper for all the details. We will just mention two points. First, to ensure that the additional requirements stated in section 7.4.1 are met, one cannot use as Conformity Measure the residual shown in (7.4) but has to turn to the so-called studentized residual, which takes into account the leverage of an object. Second, the algorithm admits an explicit form which also allows to perform a substantial amount of calculations once only, thereby mitigating the computational burden otherwise typical of transductive methods.

7.5.1 Advantages and Limitations of KRRPM

The key distinguishing feature of CPDs in general and KRRPM in particular is that the predictive distributions can take any form and are not constrained to belong to a given family of distributions (all too often Gaussian). It seems natural to view this as advantageous, especially in those cases in which there is scant evidence to support the application of the Central Limit Theorem, which generally is taken as justification for the use of Gaussian distributions. It is remarkable that CPDs allow this freedom, while achieving, at the same, validity, without requiring more than i.i.d training and test data. We should however keep in mind the limitations of the methods, some of which are pointed out in Section 7 of (Vovk et al., 2019). While it is true that KRRPM inherits the flexibility of Kernel Ridge Regression that comes from regularization and the use of nonlinear kernels, the distributions for the test objects are effectively computed on data sets that differ only by one observation, that is the hypothetical example consisting of the test object with the hypothetical label. Because of this, it is surmised that there will not be much of a difference among the predictive distributions for different test objects. This was discussed in the paper just mentioned on a simple data set, but our experience in high-dimensional settings

(see Figure 7.3) seems to suggest that lack of specificity is not as as bad as intimated, although it can be argued that the PDs appear to be stretched and shifted copies of one another. What is perhaps more troubling is the realization that this undesirable lack of specificity would become more pronounced as the size of the training set is increased (because changing one observation would affect the prediction relatively less). The paper just cited suggests what some practitioners call "local models", which consists in training models that are specific to the test objects. Instead of trying to create one model that applies to the entirety of a huge multidimensional space, the idea is to training a model on a set of training examples that are close in some sense to the test object.

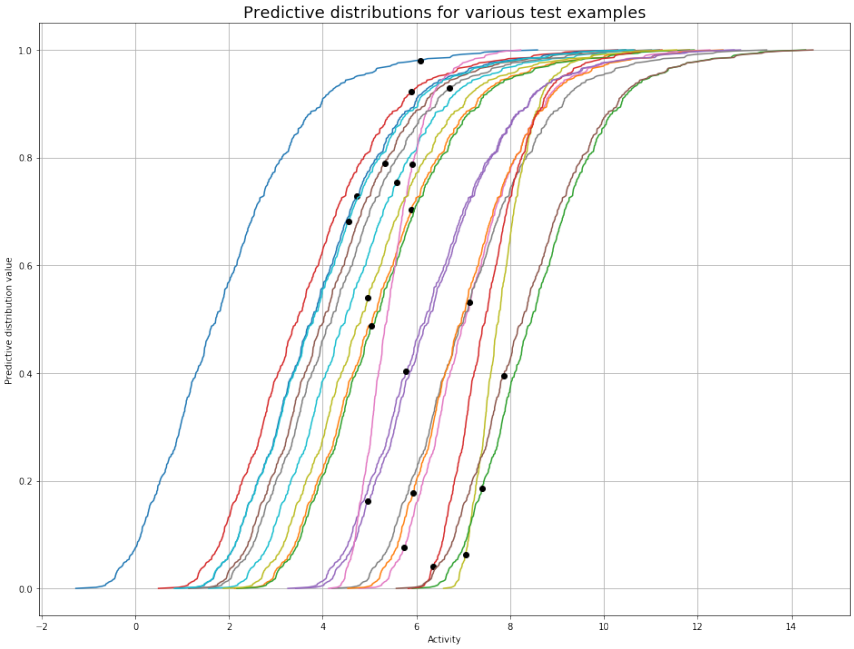


FIGURE 7.3: Some real-world Predictive Distributions. Predictive Distributions for 20 test objects, obtained with a KRRPM with Laplace kernel on a data set of 1000 training examples. The shape is very similar, but a few have markedly different slope and location. The black dots are the actual label values.

A recent further development of CPD is in the direction of universally consistent CPDs and Conformal Calibrators, which can theoretically output PDs that are object-conditional.

7.6 Application to Drug Development

In this section we provide some examples of the application of KRRPM to the prediction of pharmacokinetic and physicochemical properties. The objectives of this study were not so much to validate the KRRPM method in itself, but to evaluate its suitability to the kind of prediction tasks prevalent in drug development. In particular, KRRPM was compared against the existing prediction system in use at AstraZeneca (AZ). To get a meaningful comparison, four endpoints were chosen from

the repository of assay measurements at the company. We are grateful to Ioana Oprisiu in Gothenburg and Avid Afzal in Cambridge for assisting us in this choice and for preparing the data sets. The names of the endpoints and the characteristics of the data sets are shown in Table 7.1. The aim of the choice of targets was to cover a variety of domains (i.e. include pharmacokinetic as well as physicochemical properties), of data set sizes (from relative small to large number of compounds in the training set), and of predictive difficulty.

The numerical features for each compound in the data set were obtained by computing the signature descriptors (Faulon, Visco, and Pophale, 2003) of the compounds. The resulting training sets have a very large number of features (of the order of hundred of thousands), but they are also very sparse.

7.6.1 Implementation details

The KRRPM algorithm was implemented in Python, using the `numpy` and `sklearn` packages. The code was run on one node of the AZ high-performance computing platform, which was equipped with 132GB RAM and a 32-core server-grade processor. The scalability of the implementation was limited by the memory requirements, which are dominated by the n-by-n kernel matrix (which is generally a dense matrix) and other temporary matrices of the same size. In fact, it was possible to achieve a maximum training set size of 80,000 observations, only after implementing a custom version of the RBF kernel in Cython and rewriting the code so that temporary copies of the large matrices were avoided when performing the linear algebra calculations. The KRRPM algorithm as set out in (Vovk et al., 2018) requires a matrix inversion. This was performed in a memory efficient way using BLAS/LAPACK functions to implement the Cholesky method for symmetric matrices. In addition, the matrix-by-vector multiplication was implemented in-place as well (with an additional $O(n)$ requirement). This resulted in only 2 copies of the n-by-n matrix being needed. Given that the memory requirement for a 80,000-by-80,000 square matrix of floating numbers in double precision is 51.2GB, it was possible to accommodate the two copies within the nominal 132GB RAM available on a node. The execution time to train on 80,000 compounds and predict 5,000 was 1hr 35min.

7.6.2 Methodology

The objective of the study was to characterize the applicability of Conformal Predictive Distributions and specifically of KRRPM to prediction of candidate drug properties. In addition to the choice of suitable data sets already covered in Section 7.6, the evaluation of the method posed a number of methodological challenges, namely the preparation of training and test data, the choice of performance metrics, and the parameter optimization strategy.

Training and test sets

The primary objective of this investigation was to evaluate the methods under conditions that reproduce as closely as possible the deployment conditions. We acknowledged that the conventional shuffle-and-split partitioning of compounds into training sets and test sets was far from adequate in this respect. Shuffle-and-split results in test and training sets that can be viewed as independent and identically distributed and, therefore, meeting the assumptions under which the conformal methods make their guarantees. However, the choice of which compounds to test at a certain phase is definitely not random nor independent from the previous history. It is the product of the judgement of medicinal chemists who take into account the results obtained so far and explore just what they consider the more promising part of the chemical space for the specific target in hand. Indeed, it is possible to detect in all of the 4 data sets deviations from the i.i.d. assumption. In order to provide a more realistic view, the data sets were split into training and test sets in a way that preserved the temporal ordering of the compound data, without any element of randomness. This recreated the operating conditions that the method would encounter if deployed in the context of drug development processes. Of course, the prediction impairments attributable to deviation from i.i.d. in the data can be lessened by retraining periodically the models. A method to correct the models (avoiding a full retraining) in the presence of covariate shift is suggested in (Tibshirani et al., 2019).

The procedure is illustrated in Figure 7.4 and the resulting training and test sets are detailed in Table 7.2.

Target	Number of compounds	Number of features	"Density"	Description
hERG	37,801	80,108	0.000876	human Ether-à-go-go-Related Gene, cardiac toxicity
HLM	73,667	139,263	0.000514	Intrinsic Clearance (Clint) in Human Liver Microsomes, metabolic stability
hPPB	102,737	166,070	0.000431	human Plasma Protein Binding, drug distribution
LogD	163,186	221,415	0.000315	Log of distribution coefficient (D), hydrophilicity

TABLE 7.1: KRRPM Data sets stats. The "Density" column refers to the fraction of non-zero features. This very low density is typical of structural descriptor such the signature descriptors used in this study.

Evaluation criteria

The assessment of probabilistic forecasting remains largely an open and active research topic. The performance was evaluated with respect to a panel of metrics. In addition to metrics for conventional single-point predictions, we considered also metrics for probabilistic predictions. The former are well established and widely

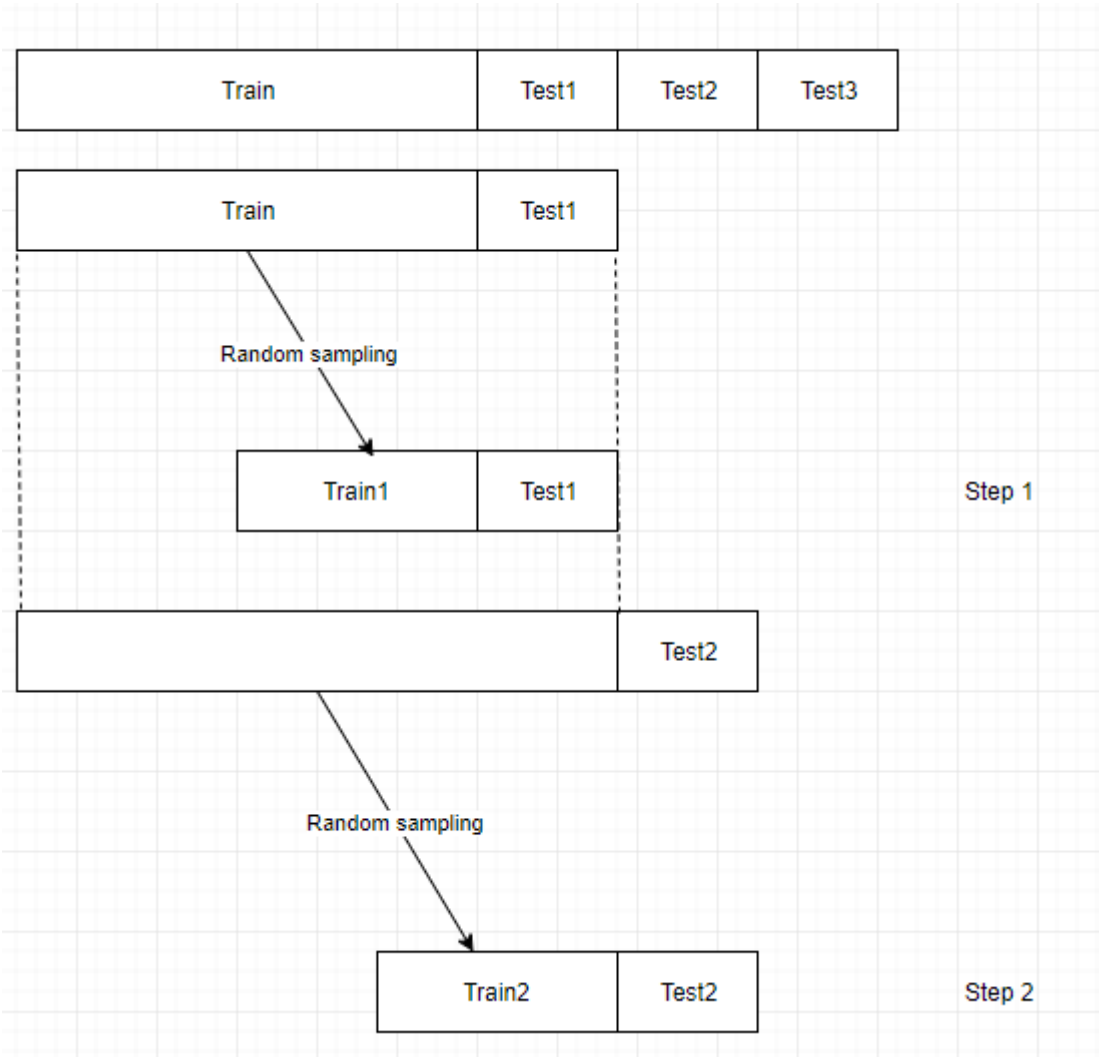


FIGURE 7.4: Creation of test and training sets. Test sets span 180 days. Calling T_{last} the most recent date in the overall data set, the first test set is made up of the compound measurements taken between $T_{\text{last}} - 3 * 180$days and $T_{\text{last}} - 2 * 180$days and the corresponding training set is made up of the compound measurements taken before $T_{\text{last}} - 3 * 180$days. In the diagram the earliest compound would be at the left end and the most recent at the right end. The random sampling is applied only when the training set size is larger than the maximum that KRRPM can handle.

Target	Split	Features	Training compounds	Training start date	Training end date	Test compounds	Test start date	Test end date
hERG	split_1	75543	35540	2004-07-30	2017-03-17	772	2017-03-18	2017-09-06
hERG	split_2	77283	36312	2004-07-30	2017-09-06	682	2017-09-21	2018-03-16
hERG	split_3	78646	36994	2004-07-30	2018-03-16	807	2018-03-22	2018-09-07
HLM	split_1	122025	61601	2010-01-05	2018-04-06	4218	2018-04-08	2018-10-04
HLM	split_2	127693	65819	2010-01-05	2018-10-04	3668	2018-10-09	2019-03-28
HLM	split_3	133158	69487	2010-01-05	2019-03-28	4180	2019-04-03	2019-09-26
hPPB	split_1	150762	90395	1997-08-04	2018-04-05	3832	2018-04-08	2018-10-04
hPPB	split_2	155348	94227	1997-08-04	2018-10-04	4259	2018-10-10	2019-03-28
hPPB	split_3	160598	98486	1997-08-04	2019-03-28	4251	2019-04-04	2019-09-26
LogD	split_1	204836	147187	1988-02-26	2018-04-09	5348	2018-04-12	2018-10-09
LogD	split_2	210040	152535	1988-02-26	2018-10-09	5302	2018-10-11	2019-04-04
LogD	split_3	215758	157837	1988-02-26	2019-04-04	5349	2019-04-11	2019-10-01

TABLE 7.2: Splits of the data sets. From each of the four data sets (hERG, HLM, hPPB, LogD), three splits were obtained so that the evaluation could be done on more cases. To ensure realistic conditions, the splits were not created in the conventional shuffle-and-split way, but as partitions that preserved the temporal ordering of the observations. The conventional random sampling creates training and test sets that are i.i.d. but do not reflect the actual conditions in which the method will operate in a real deployment.

understood, whereas the latter are still more specialistic and often harder to interpret intuitively. For example, the coefficient of determination R^2 has a clear intuitive interpretation as the fraction of variance that is explained by the single-point predictions of the model, whereas even if we consider a very common metric as log loss, we soon realize that does not have such a direct interpretation. Also, a metric as log loss is a function only of the probability assigned to the value that actually occurred and not of the entire predicted probability distribution. Log-loss or Brier loss are metrics that apply to tasks in which predictions must assign probabilities to a set of mutually exclusive *discrete* outcomes. They ignore, for instance, how concentrated or dispersed the probability mass was. A number of other metrics and diagnostic tools have been proposed. We considered the Probability Integral Transform (PIT) and Continuous Ranked Probability Score (CRPS).

Probability Integral Transform (PIT) It can be used to assess validity. It consists in evaluating the predictive distribution $F_i()$ on the actual label y_i. We observed that if the predictions $F_i(y)$ are ideal, $F_i(y_i)$ are variates from a $U(0,1)$ distribution. In the literature (Gneiting, Balabdaoui, and Raftery, 2007), the recommendation is to check that the histogram of the PIT should be as flat as possible. It can be argued, perhaps, that a better method would be to check that the ECDF of the PIT should be as close as possible to the (0,0)-(1,1) diagonal. One could as well use the Kolmogorov-Smirnov statistic to ascertain deviation from uniformity.

Continuous Ranked Probability Score (CRPS) The CRPS evaluates the predictive distribution in its entirety (see Figure 7.5). It can be viewed as the quadratic measure of discrepancy between the forecast CDF $F(y,x)$ and the "ideal forecast CDF" given the scalar observation y.

$$\text{CRPS}(F,x,y) = \int_{\mathbb{R}} \left[F(t,x) - \mathbb{I}(t \geq y)\right]^2 dt$$

where $\mathbb{I}()$ is the indicator function.

For a number of predictions, one takes the average:

$$\overline{\text{CRPS}}(F) = \frac{1}{n}\sum_{i=1}^{n}\text{CRPS}(F, x_i, y_i)$$

The CRPS is affected by both deviations from validity (a property that depends on the actual observations) and variations in sharpness (the concentration of probability, which does not depend on the actual observations, but only on the forecasts). Note that the CRPS has the unit of the label, so it is highly inappropriate to average it across different data sets.

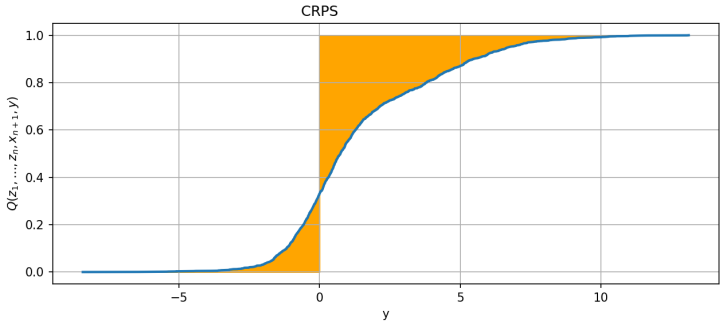


FIGURE 7.5: Continuous Ranked Probability Score. The CRPS is the integral of the square of the difference (orange) between predictive distribution and "ideal CDF". Here the actual value for the label y was assumed to be 0

While useful, these recommended metrics are not straightforward to interpret and in the author's experience for the application discussed in this chapter some more direct diagnostic tools were preferable, namely validity plot and interval boxplots.

Validity Plot Evaluates the actual coverage vs. the confidence. For all the confidence values of interest (e.g. $0.1, 0.2, \ldots, 0.9$), one computes the intervals for the objects in the validation set and then computes the relative frequency of "interval contains actual label" event. The relative frequency should be close to the confidence.

Interval Boxplots Provide a graphical representation of key descriptive statistics. For all the confidence values of interest (e.g. $0.1, 0.2, \ldots, 0.9$), inspect the boxplot showing median, interquartile range and outliers of the intervals of the test objects for the given confidence.

These two diagnostic tools should be examined in connection because of the presence of trade-off between validity and sharpness already discussed, in a slightly different form, in Section 2.4.4. The property of validity that KRRPM possesses, while beneficial, is not the only desirable property for probabilistic forecasting. To see this, consider that one could provide a valid PD by outputting the same distribution for

all test objects, namely the empirical distribution of the labels in the training set. It is obvious that this would not be a useful prediction. We also want the prediction to be *specific*. We want the prediction method to be able to take in account the features of a test object and produce a prediction that assigns probability in a less dispersed way. This would result for instance in narrower confidence intervals. We refer to methods producing narrower intervals for the same confidence as being more efficient. In general, when evaluating probabilistic predictions from different methods, one is faced with different mixes of degrees of validity and of efficiency. A dilemma arises when a model appears to be more efficient (i.e. produces narrower intervals) than another, but also exhibits worse validity (as the chance of the actual value being outside the interval, because narrower, increases). Unfortunately, as already observed in Section 2.4.4, there does not seem to be a principled, accepted way to combine in a single metric the trade-off between validity and efficiency. Conformal methods distinguish themselves in that they guarantee validity (at least, under the proviso of i.i.d. and within statistical fluctuation) and allow focusing only on improving efficiency.

Parameter optimisation strategy

KRRPM presents parameters that require optimisation, namely the regularisation parameter and, depending on the kernel, the kernel parameter. In our experiments, we performed 3-way cross validated parameter search on a grid of parameters. Parameter optimisation was performed separately on each split.

7.6.3 Results

The KRRPM algorithm produces a predictive distribution (a cumulative distribution) in the form of a vector of increasing label values. Each value corresponds to a $\frac{1}{n}$ step in the cumulative distribution. Figure 7.6 shows an example of the predictions for the hERG endpoint for three compounds.

A probabilistic prediction in the form of cumulative probability has the advantage of allowing a quick reading of the probability associated with an interval, as already illustrated in Figure 7.1. The same information is much harder to obtain on the graph of the probability density, because the probability is the definite integral of the density over the interval of interest. On the other hand, the probability density chart has the main advantage of showing clearly the modes (the maxima of the probability density), which are instead difficult to read on the cumulative probability graph because of the notorious perceptual limitation in judging slopes. But it can be argued that it is really the probability associated with an interval that carries meaning and the values of density plot may be misleading. While it is possible to estimate a probability density from the cumulative distribution, one has to keep in mind the fact that it is an ill-posed problem as discussed in Section6.5.2.

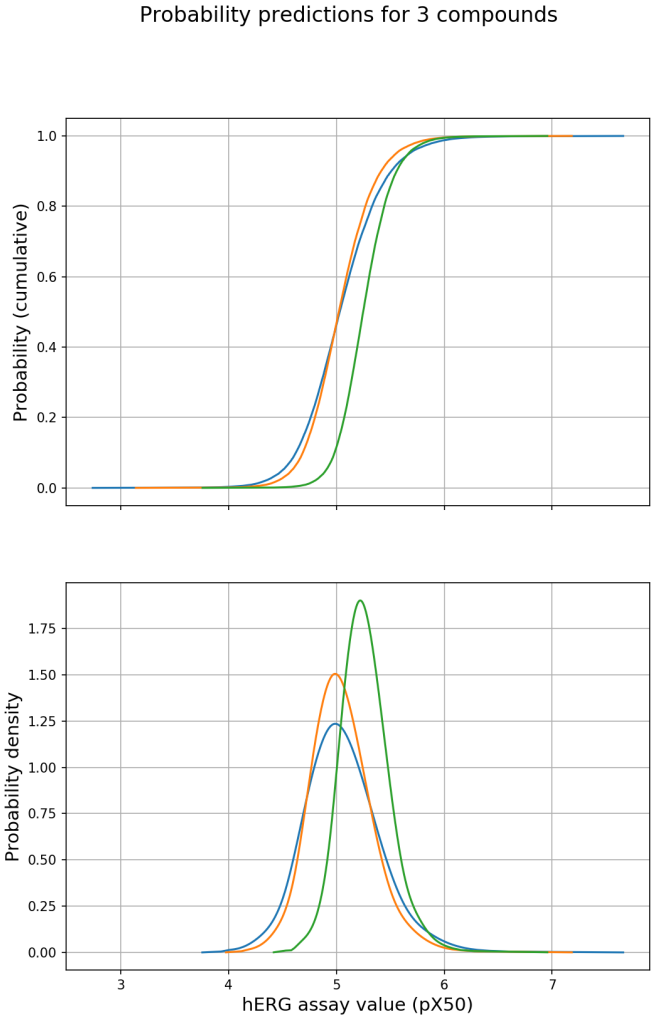


FIGURE 7.6: Predictive Distributions for the hERG target for three compounds. The top chart shows the predictions produced with KRRPM for three compounds. The bottom chart shows the corresponding density plots (which in general require special techniques to be derived, but were obtained here with numerical differentiation followed by smoothing with a spline).

The predictive distributions may be the most informative type of probabilistic prediction but they can be unwieldy and not immediately interpretable in a useful manner, especially by non-experts. Indeed, since this study aimed at providing prediction in a form that is immediately understandable by medicinal chemists, we extracted numerical metrics that can be of more direct interpretation. The first form of prediction we considered is a "point prediction" and for that we extracted the median from the distribution, which we favored over the mode as it can be more robustly computed from the PD. A point prediction carries little, if any, probabilistic information, but it was provided nonetheless to enable comparisons with conventional forms of prediction. The second form of estimates was that of conformal prediction regions: after choosing a significance level ϵ, for each test object, a subset of the label domain is computed. When the procedure is applied to a number of test objects, conformal predictions guarantee that the actual value belongs to the predicted interval a fraction $1 - \epsilon$ of the times (barring statistical fluctuation). The third form of estimates takes in a sense the opposite approach: instead of fixing a significance level ϵ and computing the corresponding intervals, it fixes an interval width and computes the probability of the actual label falling in the interval. It has to be noted that the CP validity guarantee does not imply a guarantee on this form of prediction.

The three different forms of prediction require three different forms of evaluation. The point predictions can be evaluated with any of the conventional metrics for regression. In our case, the coefficient of determination R^2 was used. Further insight can be gained by examining the relationship between point prediction and actual value. For the second form of predictions, i.e. the prediction regions for a given error rate, one should verify the validity of the predictions (which the method guarantees under the assumption of i.i.d.) and should assess the efficiency or sharpness of the predictor, which can be measured in terms of the average size of the prediction sets. Lastly, for the third form, one should evaluate the calibration of the predicted probability, that is, whether the predicted probability reflects the (long-term) relative frequency.

Figures 7.7 to 7.12 show the comparison of KRRPM versus CLAB, the existing prediction system. CLAB uses proprietary software, but we were led to believe that the algorithm that it uses is the one described in (Lapins et al., 2018). The comparison was possible for only two of the four targets (namely, hERG and HLM), because there were issues in running CLAB on the larger data sets of hPPB and LogD.

The 6 charts in each figure are organised by row: the top row deals with point predictions, the middle row with fixed interval predictions, and the bottom row with fixed error rate predictions.

The charts for the point predictions illustrate the trend of R^2 (on the y-axis) as more test objects are predicted (size of the test set on the x-axis). The test objects are submitted to the model in the temporal order in which they were entered in the assay repository. This gives an idea as to the variation in the performance of

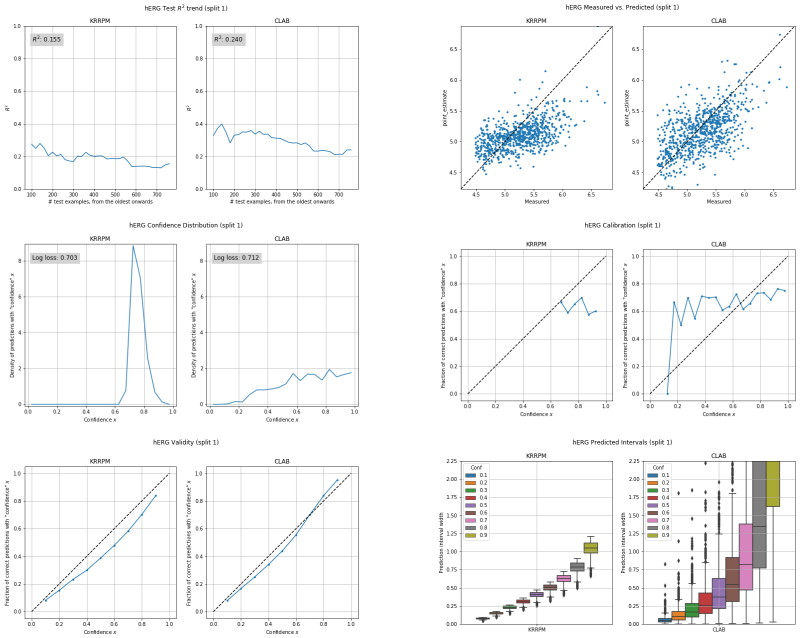


FIGURE 7.7: KRRPM vs. CLAB, hERG data set, split 1.

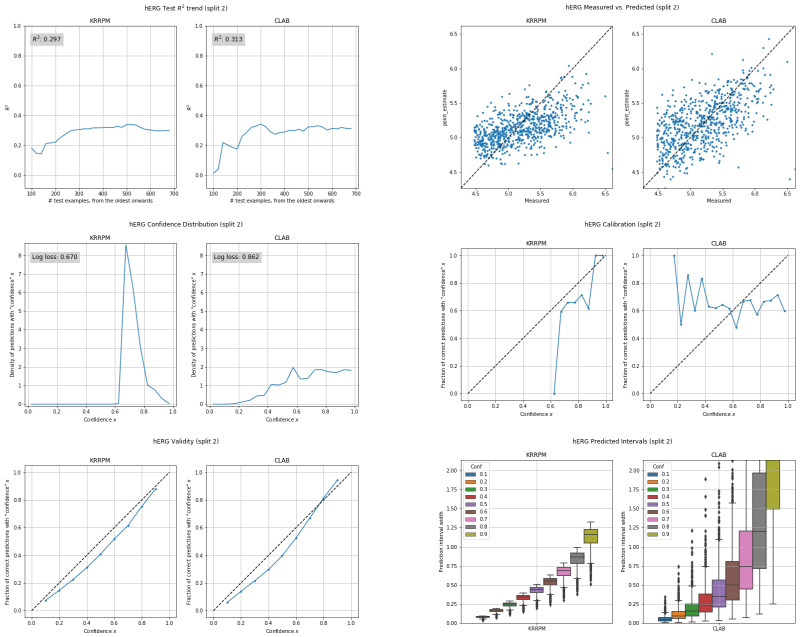


FIGURE 7.8: KRRPM vs. CLAB, hERG data set, split 2.

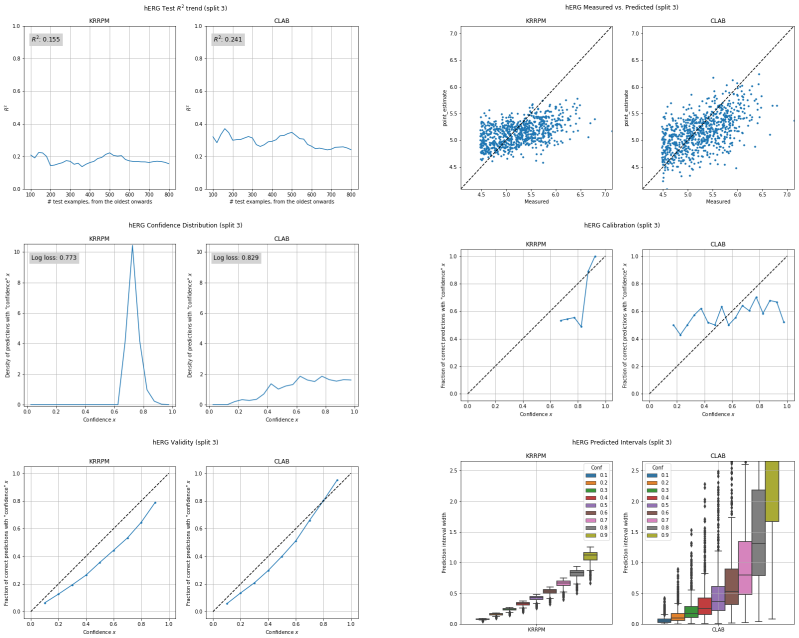


FIGURE 7.9: KRRPM vs. CLAB, hERG data set, split 3.

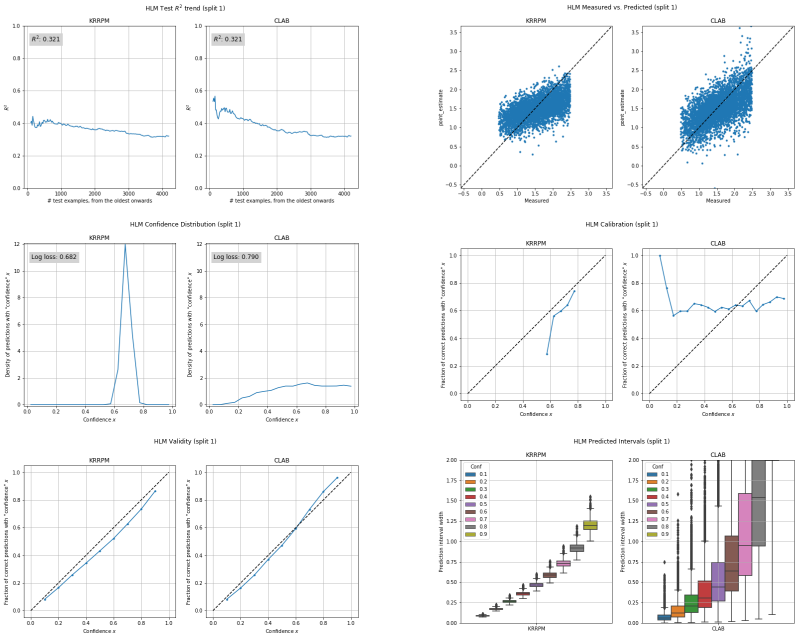


FIGURE 7.10: KRRPM vs. CLAB, HLM data set, split 1.

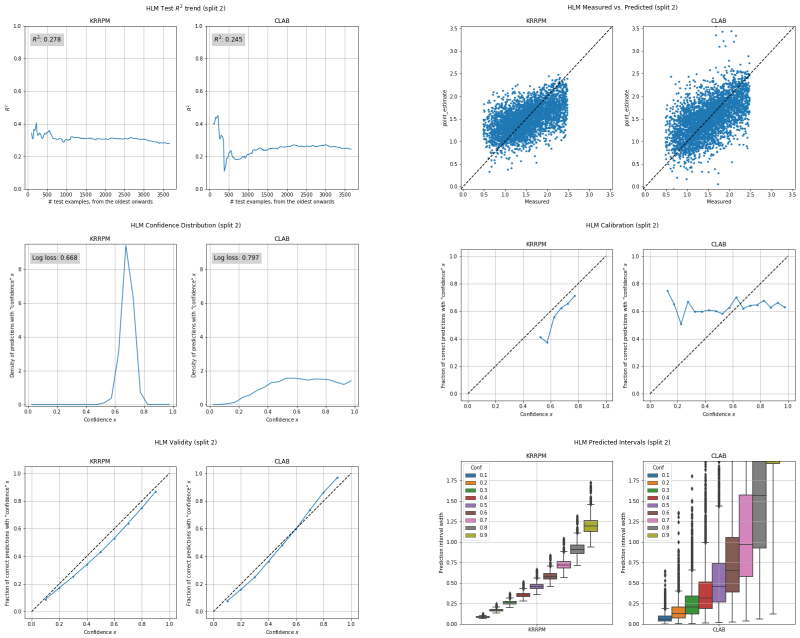


FIGURE 7.11: KRRPM vs. CLAB, HLM data set, split 2.

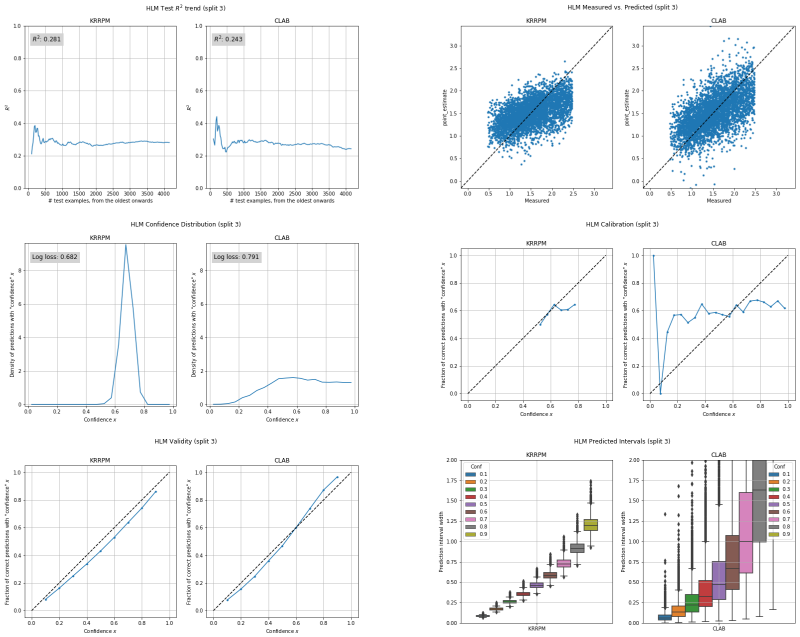


FIGURE 7.12: KRRPM vs. CLAB, HLM data set, split 3.

predictive model over time. The right chart shows the relationship between point predictions and actual values. Ideally the points should lie on the $y = x$ line (dashed line).

The middle row assesses the fixed-interval predictions. Calibration is the essential property. This is represented in the right chart, where the frequency of "actual value contained in interval" is plotted against the predicted "confidence". Again, ideally the trace should be along the $y = x$ line (dashed line). In itself the calibration plot, however, does not convey all the relevant information. A deviation from calibration matters in practice only if it happens for a large (or non-negligible, if one wants to be stricter) proportion of cases. The left plot shows a normalized count of the occurrences of values of predicted confidence.

The third row provides information from the perspective of the fixed-error perspective. Validity can be assessed with the left chart, where the relative frequency of correct predictions is plotted against the chosen confidence. Any deviation in validity can be seen as deviation from the $y = x$ line (dashed line). Different CPs should all have validity under i.i.d. (but may exhibit different degrees of sensitivity to deviation from i.i.d.) so validity is not so much a distinguishing feature. What we are seeking is in fact efficiency, i.e. as small prediction intervals as possible. The chart of the right shows box plots of interval widths for various confidence.

The diagrams seem to support the following observations:

- Both methods achieve very low R^2 (top left diagram). The relationship between measured and values and point predictions implied by the diagram in the top right has different characteristics in the two methods: KRRPM tends to exhibit lower variance than CLAB, but the average of CLAB predictions is more accurate (albeit with more "noise").

- In the fixed-interval case (middle row), KRRPM made probability predictions with a smaller range of values than CLAB, but CLAB predictions appeared to be badly calibrated (there are many cases of the actual probability of correct prediction differing from the predicted probability).

- in the bottom row (fixed confidence case), KRRPM appeared to be deviate from validity (left plots) slightly more than CLAB, possibly indicating a higher sensitivity to deviations from i.i.d. in the test data. However, the plots on the right clearly show that the intervals predicted by KRRPM are narrower and more consistent, especially for high values of confidence.

Judging only on the basis of Figures 7.7 to 7.12, one might as well remain unconvinced that the KKRPM does exhibit the validity property. We stated that the deviation from validity is attributable to the departure from the i.i.d. assumption in the test data. To provide an element of corroboration to our claim, Figure 7.13 shows the results obtained when the training and test are sampled randomly (without replacement) from one of the data sets.

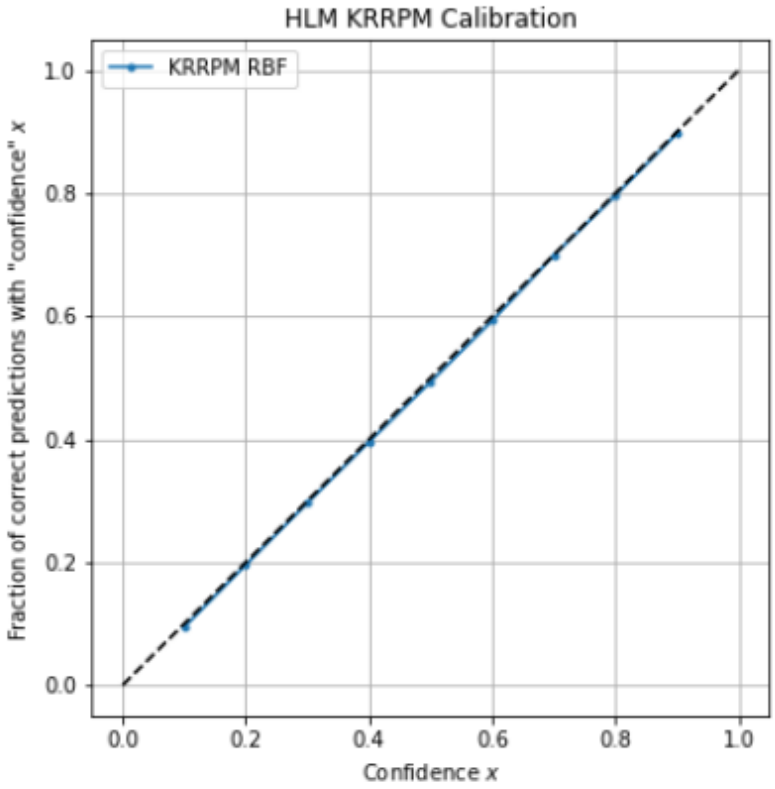


FIGURE 7.13: KRRPM Validity on i.i.d. data. The validity plot was obtained by applying KRRPM on the training and test sets the HLM data set by sampling without replacement. The difference with the analogous charts in Figures 7.7 to 7.12 is that in those charts the training-test split preserved the temporal ordering. The validity is verified to a high degree (the blue trace overlaps the dashed diagonal) when the data is randomly sampled.

7.7 Future directions and conclusions

Predictive distributions are the most informative form of probabilistic prediction. From them it is straightforward to derive other ways of presenting probabilistic information, such as point prediction as well as prediction intervals, but the real potential of PD is perhaps in their application in the field of decision-making (Vovk and Bendtsen, 2018). Another active line of research addresses the potential limitations alluded to in Section 7.5.1, by investigating methods designed to produce predictions that are more specific (Vovk, 2019; Vovk et al., 2020). Finally, a natural avenue of research might be to consider combination of predictive distibutions (from different algorithms, for the same object), using the techniques presented in Chapter 5 and 6. In summary, we showed that Conformal Predictive Distributions and in particular KRRPM can be successfully applied to chemoinformatics problems such as the prediction of pharmacokinetic and physicochemical properties. To achieve that, we implemented the method with a particular attention to using efficiently memory

and CPU resources. We also presented the predictions under various perspectives, highlighting the consequences of the violation of the i.i.d. assumption on validity and efficiency of the predictions.

Chapter 8

Conclusions

This thesis has presented the results of the research on the application of conformal techniques. We covered the three main methods currently available, namely Conformal Predictors, Venn Predictors, and Conformal Predictive Distributions. The domain of application considered in this study is to chemoinformatics and in particular the prediction of biological activity (framed as a binary classification), as well as of continuous pharmacokinetic or physicochemical properties. More specifically, we believe we addressed the goals set out in the Outline (Section 1.1) as follows:

1. *Conformal methods should cater for the peculiar characteristic of the data sets prevalent in chemoinformatics, namely size, imbalance, and sparseness.*
   We demonstrated the validity property of Conformal Predictors (Tables 3.7 and 3.9), the benefits of the Venn-ABERS predictors (Section 4.5), and the validity of Conformal Predictive Distributions (Section 7.7) on data sets with sizes ranging from $\approx$80,000 to $\approx$300,000, imbalance of 1%, and the high sparseness that derives from the use of structural descriptors. While the application to chemoinformatics was not novel for CP, we believe that this is the case for Venn-ABERS and Conformal Predictive Distributions.

2. *Conformal methods should be applicable at scale.*
   We showed that it is possible to implement the conformal methods in an efficient and scalable way that allows to make effective use of the processing power available in High Performance Computing systems as evidenced in Section 3.1.2, Table 4.1, Section 5.4.1, and Section 7.6.1.

3. *Conformal methods should take advantage of the benefits of ensembling.*
   We proposed various new methods for combining Conformal Predictors. We described a way to learn a function that combines p-values and applied it to a real-world data set (Section 5.4.2). We proposed a new combination method with theoretical support that maximises prediction efficiency (i.e. prediction sets as small as possible) while preserving validity and in Section 6.8.1 provided some evidence of its benefits on a realistic synthetic data set.

Appendix A

Demos and Software

A.1 Software Libraries

During the course of the research, several pieces of software were developed. Implementations of Inductive Conformal Predictors and Venn-ABERS Predictors were specifically coded in the context of the ExCAPE project and were applied with success to large chemoinformatics data sets (Toccaceli et al., 2015; Toccaceli et al., 2016; Toccaceli et al., 2017). The code and documentation were included among the RHUL deliverables of the project.

The implementation of the Venn-ABERS Predictors was also made publicly available on GitHub at `https://github.com/ptocca/VennABERS` and has been used for applied research (Mervin et al., 2020) and in the agrochemical industry (Marchese-Robinson 2020, personal communication).

The Mondrian Inductive CP implementation, although not packaged as a standalone library, is available as part of the source code of the demos described in the next section (stored in the repository `https://github.com/ptocca/`).

A library for Kernel Ridge Regression Predictive Machine was also developed in the context with the collaboration with AstraZeneca. The source code will be made available on GitHub as soon as permission is granted.

Finally, the author is proud to have contributed code (PR15049 in v.0.22.0, December 3, 2019) for a performance improvement to `scikit-learn`, possibly the most popular open-source Python library for Machine Learning.

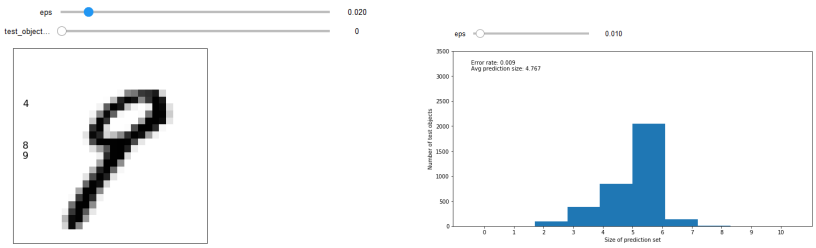
A.2 Demos

In addition, several demos were developed, mostly as Jupyter Notebooks, to illustrate some aspects of the predictors on concrete examples. The code for the demos is publicly available on GitHub at `https://github.com/ptocca`. The demos can be run on Binder (`https:\mybinder.org`), a site that hosts Jupyter Notebooks in the cloud[1]. In this section, we present briefly each demo and the specific aspect that each is meant to illustrate.

[1]Please note: launching a notebook on Binder can take a long time especially if the notebook has not been run in a while. Binder rebuilds the entire run-time environment (a Docker container) unless there is a cached copy for it. This can take several minutes.

A.2.1 CP MNIST Demo

The demo at `https://mybinder.org/v2/gh/ptocca/CP-MNIST-Demo/master` is a self-contained example of the application of Mondrian Inductive CP on the well-known MNIST handwritten digit classification problem. The NCMs are precomputed using a multiclass RBF SVM as underlying ML method, but the CP itself is computed on-the-fly. The `MNIST_CP.ipynb` notebook displays all the Python code that implements the demo, including the computation of the MICP. Two interactive cells allow the user to experiment with the significance level. One cell shows how the prediction set for a test object varies as the significance level is changed and another displays the distribution of prediction set sizes over the test objects for varying significance levels, as illustrated below.

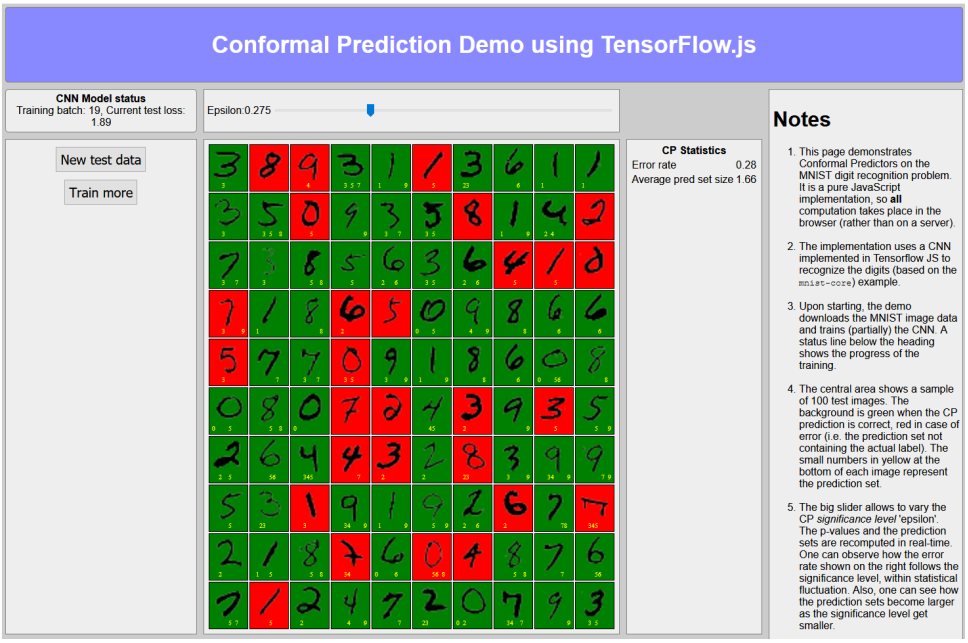


A.2.2 CP Demo using TF.js

The demo at `https://cml.rhul.ac.uk/people/ptocca/CPDemoTF/CPDemoTF.html` is an implementation of MICP in TF.js, i.e. Tensorflow for JavaScript. This demo runs locally in the client browser and this makes it much more responsive than the demos hosted remotely on Binder. The source code also shows how to implement CP in JavaScript within the Tensorflow framework.

This demo does everything from scratch. It downloads the MNIST image set and then trains a simple Convolutional Neural Network, which outputs scores for each of the 10 possible labels. The scores are used to compute p-values and the predictions sets, as usual. The page can take a few seconds to start because of the time required for downloading the MNIST set and the (partial) training of the CNN, but one can see the progress in a box on the left.

By dragging the slider in the middle one can vary the significance level and immediately see the effects on the predictions (the numbers in yellow for each displayed handwritten digit). It is also possible to perform a few more epochs of training by pressing the "Train more" button. This will illustrate that the efficiency (the average size of the prediction sets) improves. It is also possible to get a new test set by pressing the "New test data" button.



A.2.3 CP using ResNet50 on Imagenet

The demo at https://mybinder.org/v2/gh/ptocca/ILSVRC2012_CP/master?urlpath=%2Fapps%2FILSRVC_CP-Demo.ipynb applies MICP to the infamous ImageNet data set (ILSVR2012 Challenge) using a pre-trained ResNet50 neural network. It shows the probabilistic predictions emitted by ResNet50 alongside the prediction set emitted by CP. Some details are provided below, but also in the tab "Notes" in the app itself.

The demo is instructive also in letting the viewer appreciate the issues with the ImageNet data set itself. By exploring the test set, one realizes how debatable the choice of images and labels is[2]. Inevitably, one ends up wondering what the image classification task really tries to achieve here.

Notes

The user can choose an image out of a set of 2000 using a slider. The ImageNet label for the image is shown below the image itself. Another slider allows to choose the significance level ϵ (i.e. the target error rate) which can vary from 0 to 1. Below the sliders you can see two boxes containing the prediction set, one for ResNet50 and one for CP. It is also possible to choose between two forms of Non Conformity Measures, referred to here as NegProb and Ratio, explained further below. The plot shows the Empirical Cumulative Distribution Function (ECDF) of the Non Conformity Measure for the label selected in the CP prediction set shown on the right (or

[2]The artist Trevor Poglen was particularly active in bringing to the fore the serious problems in the ImageNet dataset, with his installations at the Barbican Centre in London ("From 'Apple' to 'Anomaly' ") and at the Fondazione Prada in Milan. Interestingly, the dataset is no longer available at its original repository.

the one with the largest p-value). The ECDF is calculated on the calibration examples (plus hypothetical completion).

Predictions

CP outputs sets of labels, whereas the ResNet50 model outputs a distribution of probability over the 1,000 possible labels defined for the ImageNet data set. In order to have a similar form of prediction for the two methods, we built a prediction set out of the ResNet50 probability distribution. Specifically, we want to build a prediction set with a validity property, i.e. a set of labels such that, if the probability estimates are calibrated (that is, if they correspond to long-term relative frequencies), the actual label is contained in the set with the relative frequency equal to the chosen confidence level $1 - \epsilon$. To do that, we output the smallest set of labels whose total probability (as estimated by ResNet50) exceeds $1 - \epsilon$. For CP, we simply show the prediction set for the chosen significance level. One should note that the ResNet50 sets constructed above are conservative, as the probability of hit equals or exceeds the targeted confidence.

Calibration set and test set

The CP calibration set and the test set are a random partition of the ILSVRC2012 Validation Set. The latter comprises 50,000 labelled images, evenly distributed over the 1,000 labels. For the purposes of this demo, the ILSVRC2012 Validation Set was partitioned into a calibration set with 48,000 images and test set with 2,000 images. The partitioning was done with shuffling and stratification, ensuring that each category has the same number of images.

NCMs

Two NCMs are used here. Of course, many other choices of NCMs are possible.
Some definitions:

- Let ℓ denote the number of observations (images in this case) in the calibration set and let's use the index $\ell + 1$ to denote a test object.
- Let $\{z_1, \ldots z_\ell\}$ the calibration set with observations $z_i = (x_i, y_i)$, where x_i is a 224-by-224 image and $y \in [1, 2, \ldots, 1000]$
- Let $(p_1, p_2, \ldots, p_{1000})$ the vector of 1,000 real numbers representing the probability distribution over the 1,000 labels estimated by ResNet50 for a test object $x_{\ell+1}$
- Let $\bar{y}$ be a hypothetical label for the test object.

NegProb

The NCM here referred to as NegProb is defined as:

$$\mathcal{A}(x_{\ell+1}, \bar{y}) = -p_{\bar{y}}$$

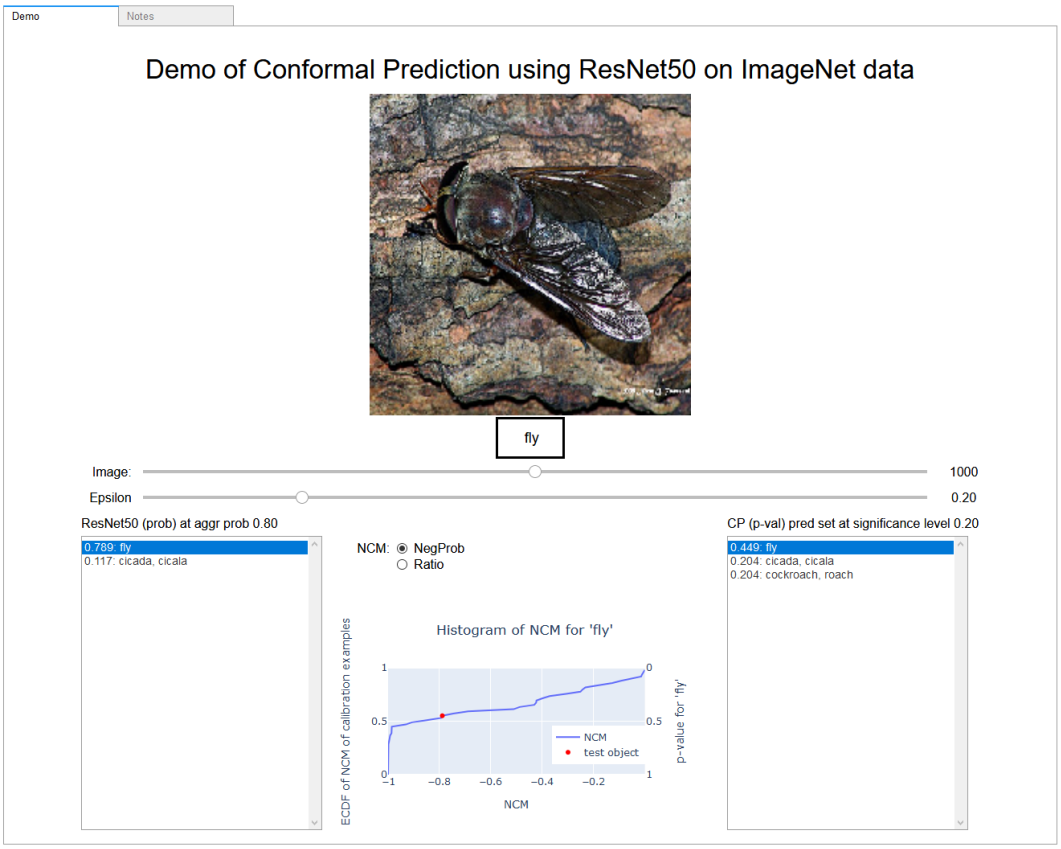
i.e. the probability estimated for the hypothetical label, with its sign changed.

Ratio

The NCM here referred to as Ratio is defined as:

$$\mathcal{A}(x_{\ell+1}, \bar{y}) = \frac{\max_{y \neq \bar{y}} p_y}{p_{\bar{y}}}$$

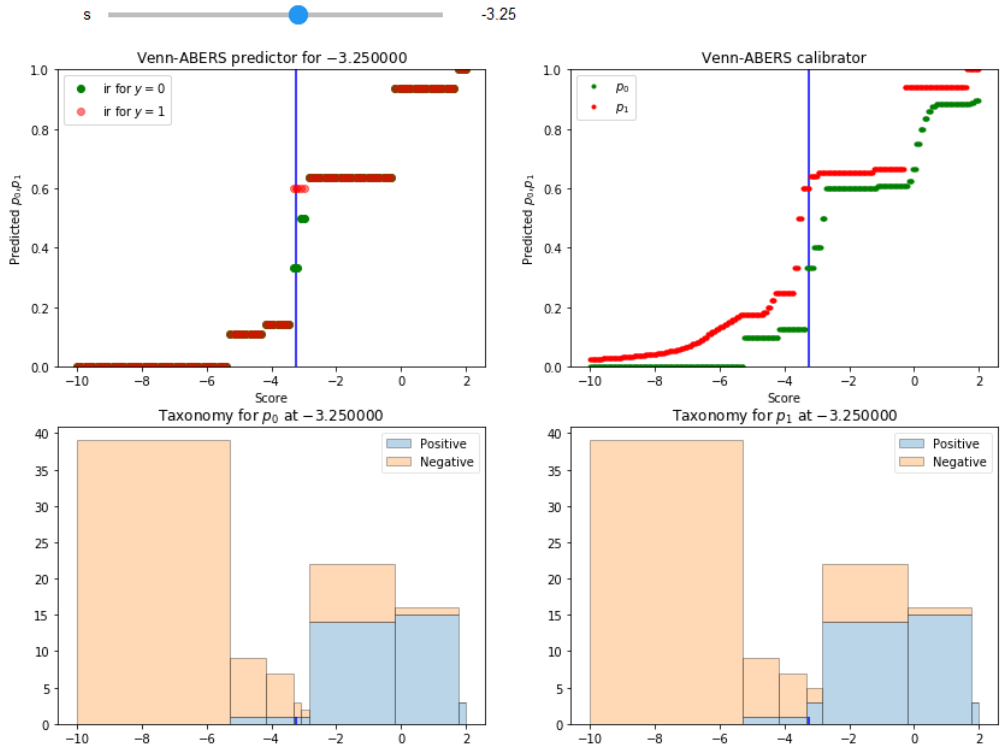
i.e. the ratio of the max probability estimated for labels other than the hypothetical one to the probability estimated for the hypothetical label.



A.2.4 Venn-ABERS Demo

The Notebook at https://mybinder.org/v2/gh/ptocca/VennABERS-demo/master?filepath=Venn-ABERS_Demo.ipynb illustrates some finer points about the Venn-ABERS predictor. The demo has been used in a tutorial at COPA2018. Among other things it has an interactive demonstration of

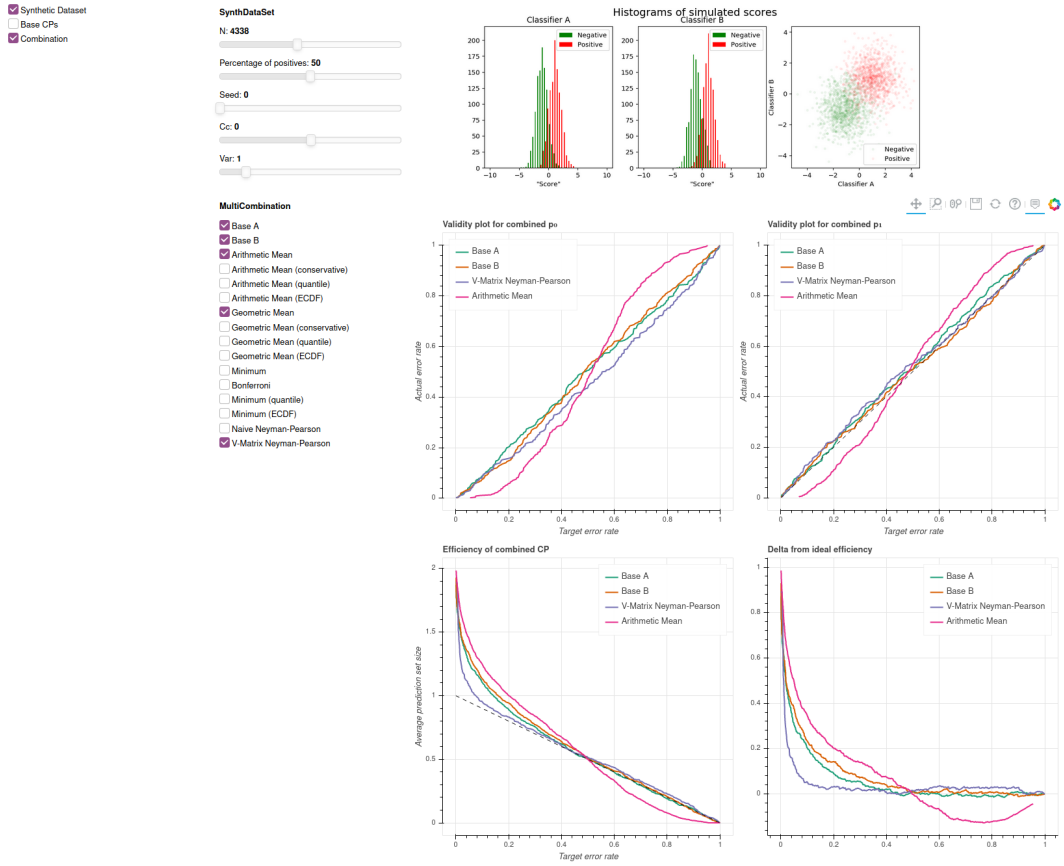
how the taxonomies are created for every test object and for the two possible label values, as illustrated in the screenshot below.



A.2.5 CP Combination Demo

The app at https://mybinder.org/v2/gh/ptocca/CP_Combination_Demo/Dynamic_layout?filepath=%2Fapps%2FCP_Combination_Demo.ipynb allows a user to experiment with the CP combination methods discussed in Chapters 5 and 6 using a synthetic data set of NCMs for two hypothetical base CPs A and B. The demo computes p-values from scores generated as variates of Gaussian distributions with means -1 and +1 (for observations of class 0 and class 1, respectively) and allows to vary the correlation between the two sets of scores (A and B).

The demo shows the validity plots as well as the average size of prediction sets for the various methods. The user can choose the size of the data sets, the variance and correlation of the scores, the initialization of the Pseudo-Random Number Generator, the imbalance, and which combination methods to apply. Note that the V-Matrix Neyman Pearson algorithm could be slow, in particular for larger data sets (the servers offered by the free MyBinder service on which the demo runs are not particularly powerful).


Synthetic Dataset
Base CPs
Combination
SynthDataSet
N: 4338
Percentage of positives: 50
Seed: 0
Cc: 0
Var: 1
MultiCombination
Base A
Base B
Arithmetic Mean
Arithmetic Mean (conservative)
Arithmetic Mean (quantile)
Arithmetic Mean (ECDF)
Geometric Mean
Geometric Mean (conservative)
Geometric Mean (quantile)
Geometric Mean (ECDF)
Minimum
Bonferroni
Minimum (quantile)
Minimum (ECDF)
Naive Neyman-Pearson
V-Matrix Neyman-Pearson
Histograms of simulated scores
Classifier A
Classifier B
Negative
Positive
"Score"
Validity plot for combined p0
Validity plot for combined p1
Base A
Base B
V-Matrix Neyman-Pearson
Arithmetic Mean
Actual error rate
Target error rate
Efficiency of combined CP
Delta from ideal efficiency
Average prediction set size

Bibliography

Ahlberg, Ernst et al. (2017). "Current Application of Conformal Prediction in Drug Discovery". In: *Annals of Mathematics and Artificial Intelligence* 81.1-2, pp. 145–154.

Ahmed, L. et al. (2018). "Efficient Iterative Virtual Screening with Apache Spark and Conformal Prediction". In: *Journal of Cheminformatics* 10.1. DOI: 10.1186/s13321-018-0265-z.

Alnemer, Loai M, Lama Rajab, and Ibrahim Aljarah (2016). "Conformal Prediction Technique to Predict Breast Cancer Survivability". In: *Int J Adv Sci Technol* 96, pp. 1–10.

Alves, Gelio and Yi-Kuo Yu (2014). "Accuracy Evaluation of the Unified P-Value from Combining Correlated P-Values". In: *PloS one* 9.3. DOI: 10.1371/journal.pone.0091225.

Ayer, Miriam et al. (1955). "An Empirical Distribution Function for Sampling with Incomplete Information". In: *The annals of mathematical statistics*, pp. 641–647.

Ayyaz, Sundus, Usman Qamar, and Raheel Nawaz (2018). "HCF-CRS: A Hybrid Content Based Fuzzy Conformal Recommender System for Providing Recommendations with Confidence". In: *PloS one* 13.10.

Balasubramanian, V., S.S. Ho, and V. Vovk (2014). *Conformal Prediction for Reliable Machine Learning: Theory, Adaptations and Applications*. Elsevier Science. ISBN: 978-0-12-401715-3.

Balasubramanian, Vineeth N., Shayok Chakraborty, and Sethuraman Panchanathan (June 2015). "Conformal Predictions for Information Fusion". In: *Annals of Mathematics and Artificial Intelligence* 74.1, pp. 45–65. ISSN: 1573-7470. DOI: 10.1007/s10472-013-9392-4.

Barber, Rina Foygel (2020). "Is Distribution-Free Inference Possible for Binary Regression?" In: arXiv: 2004.09477 [math.ST].

Barber, Rina Foygel et al. (2019). "Predictive Inference with the Jackknife+". In: arXiv: 1905.02928 [stat.ME].

Barlow, R. E. and H. D. Brunk (1972). "The Isotonic Regression Problem and Its Dual". In: *Journal of the American Statistical Association* 67.337, pp. 140–147. ISSN: 01621459. DOI: 10.2307/2284712.

Benjamin, Daniel J. et al. (2017). "Redefine Statistical Significance". In: *Nature Human Behaviour* 2, pp. 6–10.

Beyramysoltan, Samira et al. (Apr. 2020). "Identification of the Species Constituents of Maggot Populations Feeding on Decomposing Remains—Facilitation of the Determination of Post Mortem Interval and Time Since Tissue Infestation

through Application of Machine Learning and Direct Analysis in Real Time-Mass Spectrometry". In: *Analytical Chemistry* 92.7, pp. 5439–5446. ISSN: 0003-2700. DOI: 10.1021/acs.analchem.0c00199.

Bohacek, Regine S., Colin McMartin, and Wayne C. Guida (Jan. 1996). "The Art and Practice of Structure-Based Drug Design: A Molecular Modeling Perspective". In: *Medicinal Research Reviews* 16.1, pp. 3–50. ISSN: 0198-6325. DOI: 10.1002/(SICI)1098-1128(199601)16:1<3::AID-MED1>3.0.CO;2-6.

Bortolussi, L. et al. (2019). "Neural Predictive Monitoring". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 11757 LNCS, pp. 129–147. DOI: 10.1007/978-3-030-32079-9_8.

Bosc, Nicolas et al. (2019). "Large Scale Comparison of QSAR and Conformal Prediction Methods and Their Applications in Drug Discovery". In: *Journal of cheminformatics* 11.1, p. 4.

Bottou, Léon et al. (2007). *Large-Scale Kernel Machines (Neural Information Processing)*. The MIT Press. ISBN: 978-0-262-02625-3.

Brown, Morton B. (1975). "A Method for Combining Non-Independent, One-Sided Tests of Significance (Corr: V32 P955)". In: *Biometrics* 31.4, pp. 987–992. ISSN: 0006341X, 15410420.

Burnaev, E. and I. Nazarov (2017). "Conformalized Kernel Ridge Regression". In: Proceedings - 2016 15th IEEE International Conference on Machine Learning and Applications, ICMLA 2016, pp. 45–52. DOI: 10.1109/ICMLA.2016.65.

Bussonnier, Matthias (2018). *Using IPython for Parallel Computing*.

Capuccini, Marco et al. (Jan. 2015). *Conformal Prediction in Spark: Large-Scale Machine Learning with Confidence*, p. 67. DOI: 10.1109/BDC.2015.35.

Cesa-Bianchi, N. and G. Lugosi (2006). *Prediction, Learning, and Games*. Cambridge University Press. ISBN: 978-1-139-45482-7.

Chang, Chih-Chung and Chih-Jen Lin (2011). "LIBSVM: A Library for Support Vector Machines". In: *ACM transactions on intelligent systems and technology (TIST)* 2.3, pp. 1–27.

Chang, Edward Y (2011). "Psvm: Parallelizing Support Vector Machines on Distributed Computers". In: *Foundations of Large-Scale Multimedia Information Management and Retrieval*. Springer, pp. 213–230.

Cherubin, Giovanni et al. (2015). "Conformal Clustering and Its Application to Botnet Traffic". In: *Statistical Learning and Data Sciences*. Ed. by Alexander Gammerman, Vladimir Vovk, and Harris Papadopoulos. Cham: Springer International Publishing, pp. 313–322. ISBN: 978-3-319-17091-6.

Cortes, Corinna and Vladimir Vapnik (Sept. 1995). "Support-Vector Networks". In: *Machine Learning* 20.3, pp. 273–297. ISSN: 1573-0565. DOI: 10.1007/BF00994018.

Cortés-Ciriano, Isidro and Andreas Bender (2021). "Concepts and Applications of Conformal Prediction in Computational Drug Discovery". In: *Artificial Intelligence in Drug Discovery*. The Royal Society of Chemistry, pp. 63–101. ISBN: 978-1-78801-547-9. DOI: 10.1039/9781788016841-00063.

Cortés-Ciriano, Isidro, Andreas Bender, and Thérèse Malliavin (2015). "Prediction of PARP Inhibition with Proteochemometric Modelling and Conformal Prediction". In: *Molecular informatics* 34.6-7, pp. 357–366.

Dash, Santanu Kumar et al. (2016). "Droidscribe: Classifying Android Malware Based on Runtime Behavior". In: *2016 IEEE Security and Privacy Workshops (SPW)*. IEEE, pp. 252–261.

Dask Development Team (2016). *Dask: Library for Dynamic Task Scheduling*.

Davidov, Ori (2011). "Combining P-Values Using Order-Based Methods". In: *Computational Statistics & Data Analysis* 55.7, pp. 2433–2444. DOI: 10.1016/j.csda.2011.01.024.

Dearden, John C. (2017). "The History and Development of Quantitative Structure-Activity Relationships (QSARs)". In: *Oncology: Breakthroughs in Research and Practice*. Ed. by Information Resources Management Association. Hershey, PA, USA: IGI Global, pp. 67–117. ISBN: 978-1-5225-0549-5. DOI: 10.4018/978-1-5225-0549-5.ch003.

Edwards, Anthony William Fairbank (1984). *Likelihood*. Cambridge University Press Archive. ISBN: 978-0-521-31871-6.

Eklund, Martin et al. (2015). "The Application of Conformal Prediction to the Drug Discovery Process". In: *Annals of Mathematics and Artificial Intelligence* 74.1-2, pp. 117–132.

Eliades, Charalambos and Harris Papadopoulos (2018). "Detecting Seizures in EEG Recordings Using Conformal Prediction". In: *Conformal and Probabilistic Prediction and Applications*, pp. 171–186.

Faulon, Jean-Loup, Donald P Visco, and Ramdas S Pophale (2003). "The Signature Molecular Descriptor. 1. Using Extended Valence Sequences in QSAR and QSPR Studies". In: *Journal of chemical information and computer sciences* 43.3, pp. 707–720.

Fisher, R. A. (1948). "Question 14: Combining Independent Tests of Significance". In: *The American Statistician* 2.5, pp. 30–30.

Fisher, R.A. (1932). *Statistical Methods for Research Workers, 4th. Ed.* Edinburgh Oliver & Boyd.

Forreryd, A. et al. (2018). "Predicting Skin Sensitizers with Confidence — Using Conformal Prediction to Determine Applicability Domain of GARD". In: *Toxicology in Vitro* 48, pp. 179–187. DOI: 10.1016/j.tiv.2018.01.021.

Gammerman, A. et al. (2016). *Conformal and Probabilistic Prediction with Applications: 5th International Symposium, COPA 2016, Madrid, Spain, April 20-22, 2016, Proceedings*. Lecture Notes in Computer Science. Springer International Publishing. ISBN: 978-3-319-33395-3.

Gammerman, Alex et al. (Nov. 2019a). "Special Issue on Conformal and Probabilistic Prediction with Applications". In: *Neurocomputing*. ISSN: 0925-2312. DOI: 10.1016/j.neucom.2019.11.025.

Gammerman, Alexander and Vladimir Vovk (2007). "Hedging Predictions in Machine Learning (with Discussion)". In: *Comput. J.* 50.2, pp. 151–163. DOI: 10.1093/comjnl/bxl065.

— (Oct. 2017). "Foreword to This Special Issue: Conformal and Probabilistic Prediction with Applications". In: *Annals of Mathematics and Artificial Intelligence* 81.1, pp. 1–2. ISSN: 1573-7470. DOI: 10.1007/s10472-017-9557-7.

Gammerman, Alexander et al. (Mar. 2019b). "Conformal and Probabilistic Prediction with Applications: Editorial". In: *Machine Learning* 108.3, pp. 379–380. ISSN: 1573-0565. DOI: 10.1007/s10994-018-5761-x.

Gartner, Thomas (2008). *Kernels for Structured Data*. Vol. 72. World Scientific.

Genest, Christian and Louis-Paul Rivest (2001). "On the Multivariate Probability Integral Transformation". In: *Statistics & Probability Letters* 53.4, pp. 391–399.

Gneiting, Tilmann, Fadoua Balabdaoui, and Adrian E. Raftery (2007). "Probabilistic Forecasts, Calibration and Sharpness". In: *Journal of The Royal Statistical Society Series B-statistical Methodology* 69.2, pp. 243–268. DOI: 10.1111/j.1467-9868.2007.00587.x.

Graf, Hans P et al. (2005). "Parallel Support Vector Machines: The Cascade Svm". In: *Advances in Neural Information Processing Systems*, pp. 521–528.

Gu, Jiao et al. (2019). "Conformal Prediction Based on Raman Spectra for the Classification of Chinese Liquors". In: *Applied Spectroscopy* 73.7, pp. 759–766. DOI: 10.1177/0003702819831017. eprint: https://doi.org/10.1177/0003702819831017.

Guo, Chuan et al. (Aug. 2017). "On Calibration of Modern Neural Networks". In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. International Convention Centre, Sydney, Australia: PMLR, pp. 1321–1330.

Hastie, T., R. Tibshirani, and J. Friedman (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*. Springer Series in Statistics. Springer New York. ISBN: 978-0-387-84858-7.

Haugeland, John (1989). "Semantics". In: *Artificial Intelligence: The Very Idea P - 86*. MIT Press, p. 123. ISBN: 978-0-262-29114-9.

Heard, N. A. and P. Rubin-Delanchy (2018). "Choosing between Methods of Combining p-Values". In: *Biometrika* 105.1, pp. 239–246. DOI: 10.1093/biomet/asx076.

Hechtlinger, Yotam, Barnabás Póczos, and Larry Wasserman (2018). "Cautious Deep Learning". In: *arXiv preprint arXiv:1805.09460*. arXiv: 1805.09460.

Herbster, Mark and Manfred K Warmuth (1998). "Tracking the Best Expert". In: *Machine learning* 32.2, pp. 151–178.

Himabindu, T.V.R., V. Padmanabhan, and A.K. Pujari (2018). "Conformal Matrix Factorization Based Recommender System". In: *Information Sciences* 467, pp. 685–707. DOI: 10.1016/j.ins.2018.04.004.

Hollander, Myles and Douglas A Wolfe (1999). *Nonparametric Statistical Methods*. Wiley Series in Probability and Statistics. New York, NY: Wiley.

Jain, Ajay N and Anthony Nicholls (2008). "Recommendations for Evaluation of Computational Methods". In: *Journal of computer-aided molecular design* 22.3-4, pp. 133–139.

Jeffreys, H. (1998). *The Theory of Probability*. Oxford Classic Texts in the Physical Sciences. OUP Oxford. ISBN: 978-0-19-158967-6.

Ji, C. et al. (2018). "eMolTox: Prediction of Molecular Toxicity with Confidence". In: *Bioinformatics (Oxford, England)* 34.14, pp. 2508–2509. DOI: 10.1093/bioinformatics/bty135.

Johansson, U. et al. (2018). "Interpretable Regression Trees Using Conformal Prediction". In: *Expert Systems with Applications* 97, pp. 394–404. DOI: 10.1016/j.eswa.2017.12.041.

Johansson, U. et al. (2019a). "Customized Interpretable Conformal Regressors". In: Proceedings - 2019 IEEE International Conference on Data Science and Advanced Analytics, DSAA 2019, pp. 221–230. DOI: 10.1109/DSAA.2019.00037.

Johansson, U. et al. (2019b). "Efficient Venn Predictors Using Random Forests". In: *Machine Learning* 108.3, pp. 535–550. DOI: 10.1007/s10994-018-5753-x.

Jones, Eric, Travis Oliphant, Pearu Peterson, et al. (2001). *SciPy: Open Source Scientific Tools for Python*.

Kagita, V.R. et al. (2017). "Conformal Recommender System". In: *Information Sciences* 405, pp. 157–174. DOI: 10.1016/j.ins.2017.04.005.

Kahneman, Daniel (2011). *Thinking, Fast and Slow*. New York: Farrar, Straus and Giroux. ISBN: 978-0-374-27563-1 0-374-27563-7.

Kalnishkan, Y. et al. (2015). "Specialist Experts for Prediction with Side Information". In: *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, pp. 1470–1477.

Kim, Sunghwan et al. (2019). "PubChem 2019 Update: Improved Access to Chemical Data". In: *Nucleic acids research* 47.D1, pp. D1102–D1109.

Kluyver, Thomas et al. (2016). "Jupyter Notebooks – a Publishing Format for Reproducible Computational Workflows". In: *Positioning and Power in Academic Publishing: Players, Agents and Agendas*. Ed. by F. Loizides and B. Schmidt. IOS Press. IOS Press, pp. 87–90.

Korotin, Alexander, Vladimir V'yugin, and Evgeny Burnaev (Dec. 2019). *Integral Mixabilty: A Tool for Efficient Online Aggregation of Functional and Probabilistic Forecasts*.

Lapins, Maris et al. (Apr. 2018). "A Confidence Predictor for logD Using Conformal Regression and a Support-Vector Machine". In: *Journal of Cheminformatics* 10.1, p. 17. ISSN: 1758-2946. DOI: 10.1186/s13321-018-0271-1.

Lauritzen, Steffen (Apr. 2007). *Exchangeability and de Finetti's Theorem*.

Laxhammar, R. and G. Falkman (2015). "Inductive Conformal Anomaly Detection for Sequential Detection of Anomalous Sub-Trajectories". In: *Annals of Mathematics and Artificial Intelligence* 74.1-2, pp. 67–94. DOI: 10.1007/s10472-013-9381-7.

Lei, J. (2019). "Fast Exact Conformalization of the Lasso Using Piecewise Linear Homotopy". In: *Biometrika* 106.4, pp. 749–764. DOI: 10.1093/biomet/asz046.

Lei, Jing et al. (July 2018). "Distribution-Free Predictive Inference for Regression". In: *Journal of the American Statistical Association* 113.523, pp. 1094–1111. ISSN: 0162-1459. DOI: 10.1080/01621459.2017.1307116.

Linusson, H., U. Johansson, and H. Boström (2019). "Efficient Conformal Predictor Ensembles". In: *Neurocomputing*. DOI: 10.1016/j.neucom.2019.07.113.

Linusson, Henrik et al. (June 2017). "On the Calibration of Aggregated Conformal Predictors". In: *Proceedings of the Sixth Workshop on Conformal and Probabilistic Prediction and Applications*. Ed. by Alex Gammerman et al. Vol. 60. Proceedings of Machine Learning Research. Stockholm, Sweden: PMLR, pp. 154–173.

Littell, Ramon C. and J. Leroy Folks (1973). "Asymptotic Optimality of Fisher's Method of Combining Independent Tests II". In: *Journal of the American Statistical Association* 68.341, pp. 193–194. DOI: 10.1080/01621459.1973.10481362.

Löfström, Tuve et al. (2015). "Bias Reduction through Conditional Conformal Prediction". In: *Intelligent Data Analysis* 19.6, pp. 1355–1375. ISSN: 1571-4128. DOI: 10.3233/IDA-150786.

Loughin, Thomas M. (2004). "A Systematic Comparison of Methods for Combining P-Values from Independent Tests". In: *Computational Statistics & Data Analysis* 47.3, pp. 467–485.

Maggiora, Gerald M. (July 2006). "On Outliers and Activity CliffsWhy QSAR Often Disappoints". In: *Journal of Chemical Information and Modeling* 46.4, pp. 1535–1535. ISSN: 1549-9596. DOI: 10.1021/ci060117s.

McCool, Michael, James Reinders, and Arch Robison (2012). *Structured Parallel Programming: Patterns for Efficient Computation*. Elsevier.

McKinney, Wes (2010). "Data Structures for Statistical Computing in Python". In: *Proceedings of the 9th Python in Science Conference*. Ed. by Stéfan van der Walt and Jarrod Millman, pp. 51–56.

Mervin, Lewis et al. (Aug. 2020). "A Comparison Of Scaling Methods To Obtain Calibrated Probabilities Of Activity For Protein-Ligand Predictions". In: *Journal of Chemical Information and Modeling*. ISSN: 1549-9596. DOI: 10.1021/acs.jcim.0c00476.

Monev, Valentin (2004). "Introduction to Similarity Searching in Chemistry". In: *MATCH Commun. Math. Comput. Chem* 51, pp. 7–38.

Moreno, Raúl et al. (2016). "Conformal Prediction of Disruptions from Scratch: Application to an ITER Scenario". In: *Symposium on Conformal and Probabilistic Prediction with Applications*. Springer, pp. 67–74.

Morger, Andrea et al. (Apr. 2020). "KnowTox: Pipeline and Case Study for Confident Prediction of Potential Toxic Effects of Compounds in Early Phases of Development". In: *Journal of Cheminformatics* 12.1, p. 24. ISSN: 1758-2946. DOI: 10.1186/s13321-020-00422-x.

Morsomme, Raphaël and Evgueni Smirnov (2019). "Conformal Prediction for Students' Grades in a Course Recommender System". In: *Conformal and Probabilistic Prediction and Applications*, pp. 196–213.

Neyman, J. and E. S. Pearson (1933). "On the Problem of the Most Efficient Tests of Statistical Hypotheses". In: *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* 231, pp. 289–337. ISSN: 02643952.

Norinder, Ulf et al. (2014). "Introducing Conformal Prediction in Predictive Modeling. A Transparent and Flexible Alternative to Applicability Domain Determination". In: *Journal of chemical information and modeling* 54.6, pp. 1596–1603.

Nouretdinov, Ilia, Thomas Melluish, and Volodya Vovk (2001). "Ridge Regression Confidence Machine". In: *Proceedings of the Eighteenth International Conference on Machine Learning*. ICML '01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., pp. 385–392. ISBN: 1-55860-778-1.

Nouretdinov, Ilia et al. (2019). "Multi-Level Conformal Clustering: A Distribution-Free Technique for Clustering and Anomaly Detection". In: *Neurocomputing*. ISSN: 0925-2312. DOI: 10.1016/j.neucom.2019.07.114.

Papadopoulos, Harris, Vladimir Vovk, and Alexander Gammerman (June 2015). "Guest Editors' Preface to the Special Issue on Conformal Prediction and Its Applications". In: *Annals of Mathematics and Artificial Intelligence* 74.1, pp. 1–7. ISSN: 1573-7470. DOI: 10.1007/s10472-014-9429-3.

Papadopoulos, Harris et al. (Aug. 2002). *Inductive Confidence Machines for Regression*. Vol. 2430, p. 194. DOI: 10.1007/3-540-36755-1_29.

Pearl, Judea (2009). *Causality*. Cambridge: Cambridge University Press. ISBN: 978-0-521-89560-6. DOI: 10.1017/CBO9780511803161.

Pearl, Judea and Dana Mackenzie (2018). *The Book of Why: The New Science of Cause and Effect*. 1st. USA: Basic Books, Inc. ISBN: 0-465-09760-X.

Pedregosa, F. et al. (2011). "Scikit-Learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12, pp. 2825–2830.

Pereira, Telma et al. (2017). "Towards Trustworthy Predictions of Conversion from Mild Cognitive Impairment to Dementia: A Conformal Prediction Approach". In: *International Conference on Practical Applications of Computational Biology & Bioinformatics*. Springer, pp. 155–163.

Pereira, Telma et al. (2018). "Neuropsychological Predictors of Conversion from Mild Cognitive Impairment to Alzheimer's Disease: A Feature Selection Ensemble Combining Stability and Predictability". In: *BMC medical informatics and decision making* 18.1, p. 137.

Pereira, Telma et al. (2020). "Targeting the Uncertainty of Predictions at Patient-Level Using an Ensemble of Classifiers Coupled with Calibration Methods, Venn-ABERS, and Conformal Predictors: A Case Study in AD". In: *Journal of Biomedical Informatics* 101, p. 103350.

Pesarin, F. (2001). *Multivariate Permutation Tests: With Applications in Biostatistics*. Wiley. ISBN: 978-0-471-49670-0.

Polanski, J. (Jan. 2009). "4.14 - Chemoinformatics". In: *Comprehensive Chemometrics*. Ed. by Steven D. Brown, Romá Tauler, and Beata Walczak. Oxford: Elsevier, pp. 459–506. ISBN: 978-0-444-52701-1. DOI: 10.1016/B978-044452701-1.00006-5.

Polishchuk, P. G., T. I. Madzhidov, and A. Varnek (Aug. 2013). "Estimation of the Size of Drug-like Chemical Space Based on GDB-17 Data". In: *Journal of Computer-Aided Molecular Design* 27.8, pp. 675–679. ISSN: 1573-4951. DOI: 10.1007/s10822-013-9672-4.

Poole, William et al. (2016). "Combining Dependent P- Values with an Empirical Adaptation of Brown's Method". In: *Bioinformatics* 32.17, pp. i430–i436. DOI: 10.1093/bioinformatics/btw438.

Pratt, John W. (1959). "Remarks on Zeros and Ties in the Wilcoxon Signed Rank Procedure". In: *Journal of the American Statistical Association* 54, pp. 655–667.

Qaddoum, K. (2020). "Lung Cancer Patient's Survival Prediction Using GRNN-CP". In: *Communications in Computer and Information Science* 1187 CCIS, pp. 143–150. DOI: 10.1007/978-3-030-43364-2_13.

Romano, Yaniv, Evan Patterson, and Emmanuel J. Candès (2019). "Conformalized Quantile Regression". In: arXiv: 1905.03222 [stat.ME].

Romano, Yaniv et al. (2019). "With Malice towards None: Assessing Uncertainty via Equalized Coverage". In: arXiv: 1908.05428 [stat.ME].

Rosenhouse, J. (2009). *The Monty Hall Problem: The Remarkable Story of Math's Most Contentious Brain Teaser*. Oxford University Press. ISBN: 978-0-19-970990-8.

Sadinle, Mauricio, Jing Lei, and Larry Wasserman (2019). "Least Ambiguous Set-Valued Classifiers with Bounded Error Levels". In: *Journal of the American Statistical Association* 114.525, pp. 223–234. DOI: 10.1080/01621459.2017.1395341. eprint: https://doi.org/10.1080/01621459.2017.1395341.

Saunders, C., A. Gammerman, and V. Vovk (1999). "Transduction with Confidence and Credibility". In: vol. 2. IJCAI International Joint Conference on Artificial Intelligence, pp. 722–726.

Savant, M. (1997). *The Power of Logical Thinking: Easy Lessons in the Art of Reasoning...and Hard Facts about Its Absence in Our Lives*. St. Martin's Publishing Group. ISBN: 978-0-312-15627-5.

Schervish, Mark J. (1989). "A General Method for Comparing Probability Assessors". In: *The Annals of Statistics* 17.4, pp. 1856–1879. ISSN: 00905364.

Schneider, Petra et al. (May 2020). "Rethinking Drug Design in the Artificial Intelligence Era". In: *Nature Reviews Drug Discovery* 19.5, pp. 353–364. ISSN: 1474-1784. DOI: 10.1038/s41573-019-0050-3.

Sellke, Thomas, M. J. Bayarri, and James O. Berger (2001). "Calibration of p Values for Testing Precise Null Hypotheses". In: *The American Statistician* 55.1, pp. 62–71. DOI: 10.1198/000313001300339950.

Shabbir, A. et al. (Dec. 2015). "ELM Regime Classification by Conformal Prediction on an Information Manifold". In: *IEEE Transactions on Plasma Science* 43.12, pp. 4190–4199. ISSN: 0093-3813. DOI: 10.1109/TPS.2015.2489689.

Shafer, G. and V. Vovk (2019). *Game-Theoretic Foundations for Probability and Finance*. Wiley Series in Probability and Statistics. Wiley. ISBN: 978-0-470-90305-6.

Spjuth, Ola (Sept. 2018). "Novel Applications of Machine Learning in Cheminformatics". In: *Journal of Cheminformatics* 10.1, p. 46. ISSN: 1758-2946. DOI: 10.1186/s13321-018-0301-z.

Stouffer S.A.and Suchman, E.A. et al. (1949). *The American Soldier, Vol.1: Adjustment during Army Life.* Princeton University Press, Princeton.

Sturm, Noé et al. (Apr. 2020). "Industry-Scale Application and Evaluation of Deep Learning for Drug Target Prediction". In: *Journal of Cheminformatics* 12.1, p. 26. ISSN: 1758-2946. DOI: 10.1186/s13321-020-00428-5.

Sugiyama, M., T. Suzuki, and T. Kanamori (2012). *Density Ratio Estimation in Machine Learning*. Cambridge Books Online. Cambridge University Press. ISBN: 978-0-521-19017-6.

Sun, J. et al. (2017a). "Applying Mondrian Cross-Conformal Prediction to Estimate Prediction Confidence on Large Imbalanced Bioactivity Data Sets". In: *Journal of Chemical Information and Modeling* 57.7, pp. 1591–1598. DOI: 10.1021/acs.jcim.7b00159.

Sun, Jiangming et al. (Mar. 2017b). "ExCAPE-DB: An Integrated Large Scale Dataset Facilitating Big Data Analysis in Chemogenomics". In: *Journal of Cheminformatics* 9.1, p. 17. ISSN: 1758-2946. DOI: 10.1186/s13321-017-0203-5.

Svensson, F. and U. Norinder (2020). "Conformal Prediction for Ecotoxicology and Implications for Regulatory Decision-Making". In: *Methods in Pharmacology and Toxicology*, pp. 271–287. DOI: 10.1007/978-1-0716-0150-1_12.

Svensson, F., U. Norinder, and A. Bender (2017a). "Improving Screening Efficiency through Iterative Screening Using Docking and Conformal Prediction". In: *Journal of Chemical Information and Modeling* 57.3, pp. 439–444. DOI: 10.1021/acs.jcim.6b00532.

Svensson, Fredrik, Ulf Norinder, and Andreas Bender (2017b). "Modelling Compound Cytotoxicity Using Conformal Prediction and PubChem HTS Data". In: *Toxicology research* 6.1, pp. 73–80.

Svensson, Fredrik et al. (May 2018a). "Conformal Regression for Quantitative Structure-Activity Relationship Modeling—Quantifying Prediction Uncertainty". In: *Journal of Chemical Information and Modeling* 58.5, pp. 1132–1140. ISSN: 1549-9596. DOI: 10.1021/acs.jcim.8b00054.

Svensson, Fredrik et al. (Feb. 2018b). "Maximizing Gain in High-Throughput Screening Using Conformal Prediction". In: *Journal of Cheminformatics* 10.1, p. 7. ISSN: 1758-2946. DOI: 10.1186/s13321-018-0260-4.

Tibshirani, Ryan J et al. (2019). "Conformal Prediction under Covariate Shift". In: *Advances in Neural Information Processing Systems*, pp. 2526–2536.

Toccaceli, Paolo (Sept. 2019). "Conformal Predictor Combination Using Neyman–Pearson Lemma". In: *Proceedings of the Eighth Symposium on Conformal and Probabilistic Prediction and Applications*. Ed. by Alex Gammerman et al. Vol. 105. Proceedings of Machine Learning Research. Golden Sands, Bulgaria: PMLR, pp. 66–88.

Toccaceli, Paolo, Ilia Nouretdinov, and Alexander Gammerman (2016). "Conformal Predictors for Compound Activity Prediction". In: *Conformal and Probabilistic Prediction with Applications: 5th International Symposium, COPA 2016, Madrid, Spain, April 20-22, 2016, Proceedings*. Ed. by Alexander Gammerman et al. Cham: Springer International Publishing, pp. 51–66. ISBN: 978-3-319-33395-3. DOI: 10.1007/978-3-319-33395-3_4.

— (Oct. 2017). "Conformal Prediction of Biological Activity of Chemical Compounds". In: *Annals of Mathematics and Artificial Intelligence* 81.1, pp. 105–123. ISSN: 1573-7470. DOI: 10.1007/s10472-017-9556-8.

Toccaceli, Paolo et al. (2015). *ExCAPE WP1 - Conformal Predictors*.

Toccaceli, Paolo et al. (2016). *ExCAPE WP1 - Probabilistic Prediction*.

Toccaceli, Paolo et al. (2017). *ExCAPE WP1 - Integration of Conformal Prediction with ML Algorithms*.

Tsybakov, Alexandre B. (2008). *Introduction to Nonparametric Estimation*. 1st. Springer Publishing Company, Incorporated. ISBN: 0-387-79051-9.

Tukey, John W. (1986). "Sunset Salvo". In: *The American Statistician* 40.1, pp. 72–76. DOI: 10.1080/00031305.1986.10475361.

Vapnik, Vladimir, Igor Braga, and Rauf Izmailov (2015). "Constructive Setting for Problems of Density Ratio Estimation". In: *Statistical Analysis and Data Mining: The ASA Data Science Journal* 8.3, pp. 137–146. DOI: 10.1002/sam.11263.

Vapnik, Vladimir and Rauf Izmailov (2015a). "Statistical Inference Problems and Their Rigorous Solutions". In: *Statistical Learning and Data Sciences*. Ed. by Alexander Gammerman, Vladimir Vovk, and Harris Papadopoulos. Cham: Springer International Publishing, pp. 33–71. ISBN: 978-3-319-17091-6.

— (2015b). "V-Matrix Method of Solving Statistical Inference Problems". In: *Journal of Machine Learning Research* 16.51, pp. 1683–1730.

Vapnik, Vladimir N. (1995). *The Nature of Statistical Learning Theory*. Berlin, Heidelberg: Springer-Verlag. ISBN: 0-387-94559-8.

Varoquaux, Gaël et al. (2015). "Scikit-Learn: Machine Learning Without Learning the Machinery". In: *GetMobile* 19.1, pp. 29–33. DOI: 10.1145/2786984.2786995.

Vasiloudis, T., G. de Francisci Morales, and H. Boström (2019). "Quantifying Uncertainty in Online Regression Forests". In: *Journal of Machine Learning Research* 20.

Venn, John (1866). *The Logic of Chance. An Essay on the Foundations and Province of the Theory of Probability, with Especial Reference to Its Application to Moral and Social Science*. London: Macmillan.

Vovk, V. (2013). "Conditional Validity of Inductive Conformal Predictors". In: *Machine Learning* 92.2-3, pp. 349–376. DOI: 10.1007/s10994-013-5355-6.

Vovk, V. et al. (2019). "Nonparametric Predictive Distributions Based on Conformal Prediction". In: *Machine Learning* 108.3, pp. 445–474. DOI: 10.1007/s10994-018-5755-8.

Vovk, V. G. (1993). "A Logic of Probability, with Application to the Foundations of Statistics (Disc: P341-351)". In: *Journal of the Royal Statistical Society, Series B: Statistical Methodology* 55, pp. 317–341.

Vovk, Vladimir (2012). "Venn Predictors and Isotonic Regression". In: *CoRR abs/1211.0025*. arXiv: 1211.0025.

— (June 2015). "Cross-Conformal Predictors". In: *Annals of Mathematics and Artificial Intelligence* 74.1, pp. 9–28. ISSN: 1573-7470. DOI: 10.1007/s10472-013-9368-4.

— (Sept. 2019). "Universally Consistent Conformal Predictive Distributions". In: ed. by Alex Gammerman et al. Vol. 105. Proceedings of Machine Learning Research. Golden Sands, Bulgaria: PMLR, pp. 105–122.

Vovk, Vladimir and Claus Bendtsen (June 2018). "Conformal Predictive Decision Making". English. In: *Proceedings of Machine Learning Research*. Ed. by Alex Gammerman et al. Vol. 91, pp. 52–62.

Vovk, Vladimir, Alex Gammerman, and Glenn Shafer (2005). *Algorithmic Learning in a Random World*. Secaucus, NJ, USA: Springer-Verlag New York, Inc. ISBN: 0-387-00152-2.

Vovk, Vladimir, Ivan Petej, and Valentina Fedorova (2015). "Large-Scale Probabilistic Predictors with and without Guarantees of Validity". In: *Advances in Neural Information Processing Systems 28*. Ed. by C. Cortes et al. Curran Associates, Inc., pp. 892–900.

Vovk, Vladimir and Ruodu Wang (Dec. 2012). "Combining P-Values via Averaging". In: *arXiv e-prints*, arXiv:1212.4966.

Vovk, Vladimir et al. (2016). "Criteria of Efficiency for Conformal Prediction". In: *Conformal and Probabilistic Prediction with Applications*. Ed. by Alexander Gammerman et al. Cham: Springer International Publishing, pp. 23–39. ISBN: 978-3-319-33395-3.

Vovk, Vladimir et al. (Aug. 2018). "Conformal Predictive Distributions with Kernels". English. In: *Braverman Readings in Machine Learning. Key Ideas from Inception to Current State - International Conference Commemorating the 40th Anniversary of Emmanuil Bravermans Decease, Invited Talks*. Vol. 11100. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Springer-Verlag, pp. 103–121. ISBN: 978-3-319-99491-8. DOI: 10.1007/978-3-319-99492-5âĆŤ.

Vovk, Vladimir et al. (Sept. 2020). "Conformal Calibrators". In: ed. by Alexander Gammerman et al. Vol. 128. Proceedings of Machine Learning Research. PMLR, pp. 84–99.

Vovk, Volodya (2001). "Competitive On-Line Statistics". In: *International Statistical Review* 69.2, pp. 213–248. DOI: 10.1111/j.1751-5823.2001.tb00457.x.

Walt, Stéfan van der, S. Chris Colbert, and Gaël Varoquaux (2011). "The NumPy Array: A Structure for Efficient Numerical Computation". In: *Computing in Science & Engineering* 13.2, pp. 22–30. DOI: 10.1109/MCSE.2011.37.

Wasserman, L. (2010a). *All of Nonparametric Statistics*. Springer Texts in Statistics. Springer New York. ISBN: 978-1-4419-2044-7.

Wasserman, Larry (2010b). *All of Statistics: A Concise Course in Statistical Inference*. Springer Publishing Company, Incorporated. ISBN: 1-4419-2322-5.

Wilcoxon, F. (1945). "Individual Comparisons by Ranking Methods". In: *Biometrics Bulletin* 1.6, pp. 80–83.

Winkler, R. L. et al. (June 1996). "Scoring Rules and the Evaluation of Probabilities". In: *Test* 5.1, pp. 1–60. ISSN: 1863-8260. DOI: 10.1007/BF02562681.

Wouters, Olivier J., Martin McKee, and Jeroen Luyten (Mar. 2020). "Estimated Research and Development Investment Needed to Bring a New Medicine to Market, 2009-2018". In: *JAMA* 323.9, pp. 844–853. ISSN: 0098-7484. DOI: 10.1001/jama.2020.1166.

You, Yang et al. (2015). "Scaling Support Vector Machines on Modern HPC Platforms". In: *Journal of Parallel and Distributed Computing* 76, pp. 16–31.

Zaharia, Matei et al. (Oct. 2016). "Apache Spark: A Unified Engine for Big Data Processing". In: *Communications of the ACM* 59.11, pp. 56–65. ISSN: 0001-0782. DOI: 10.1145/2934664.

Zaykin, Dmitri V. et al. (2007). "Combining P-Values in Large-Scale Genomics Experiments". In: *Pharmaceutical Statistics* 6.3, pp. 217–226. ISSN: 1539-1612. DOI: 10.1002/pst.304.

Zaykin, D.V. et al. (2002). "Truncated Product Method for Combining P-Values". In: *Genetic Epidemiology* 22.2, pp. 170–185. ISSN: 1098-2272. DOI: 10.1002/gepi.0042.

Zhan, X. et al. (2018). "Online Conformal Prediction for Classifying Different Types of Herbal Medicines with Electronic Nose". In: vol. 2018. IET Conference Publications CP754. DOI: 10.1049/cp.2018.1730.

Zhan, X. et al. (2020). "An Electronic Nose-Based Assistive Diagnostic Prototype for Lung Cancer Detection with Conformal Prediction". In: *Measurement: Journal of the International Measurement Confederation* 158. DOI: 10.1016/j.measurement.2020.107588.

Zhi, WANG et al. (2017). "Fortifying Botnet Classification Based on Venn-Abers Prediction". In: *DEStech Transactions on Computer Science and Engineering* cst.