Inductive Venn-Abers Predictive Distribution

Ilia Nouretdinov I.R.NOURETDINOV@RHUL.AC.UK
Information Security Group; Computer Learning Research Center, Department of Computer Science, Royal Holloway, University of London, London, UK

Denis Volkhonskiy DENIS.VOLKHONSKIY@SKOLTECH.RU
Skolkovo Institute of Science and Technology, Skolkovo, Moscow Region, Russia

Pitt Lim PITT.LIM@STGEORGES.NHS.UK
St.Georges Hospital, London, UK

Paolo Toccaceli PAOLO.TOCCACELI@RHUL.AC.UK
Computer Learning Research Center, Department of Computer Science, Royal Holloway, University of London, London, UK

Alexander Gammerman A.GAMMERMAN@RHUL.AC.UK
Computer Learning Research Center, Department of Computer Science, Royal Holloway, University of London, London, UK

Editor: Alex Gammerman, Vladimir Vovk, Zhiyuan Luo, Evgueni Smirnov and Ralf Peeters

Abstract

Venn predictors are a distribution-free probabilistic prediction framework that transforms the output of a scoring classifier into a (multi-)probabilistic prediction that has calibration guarantees, with the only requirement of an i.i.d. assumption for calibration and test data.

In this paper, we extend the framework from classification (where probabilities are predicted for a discrete number of labels) to regression (where labels form a continuum). We show how Venn Predictors can be applied on top of any regression method to obtain calibrated predictive distributions, without requiring assumptions beyond i.i.d. of calibration and test sets. This is contrasted with methods such as Bayesian Linear Regression, for which the calibration guarantee instead relies on specific probabilistic assumptions on the distribution of the data.

The adaptation of Venn Machine to regression required a theoretical analysis of the transductive and inductive forms of the predictor. We identify potential consistency problems and provide solutions for them.

Finally, to illustrate their advantages, we apply regression Venn Predictors to the medical problem of predicting the survival time after Percutaneous Coronary Intervention, a potentially risky procedure that improves blood flow to a patient's heart. The predictive distributions obtained with this method allow a variety of interpretations that include probability of survival time exceeding a chosen threshold or the shortest survival time guaranteed with a given probability.

Keywords: reliable prediction, Venn machine, regression.

1. Introduction

The classical Machine Learning problems of Classification and Regression are concerned with producing "point" predictions in the sense that, given a test object, they choose

a single label from a discrete or a continuous set, respectively, that minimizes a given loss function. It can be argued that for various applications (e.g. decision making) the point prediction alone is inadequate and that a richer supply of information regarding the prediction is preferable.

In has been argued that the most informative form of output for a prediction problem is a probability distribution on the label space. Especially for decision making, the availability of predictive densities or predictive cumulative distribution functions has the fundamental advantage of decoupling the modeling and learning stage from the specific choice of loss function. Such choice, with its critical importance for the specific decision that ensues, can be deferred and changed at will with no impact on the preceding stages.

Venn prediction is a framework for distribution-free (multi-)probabilistic prediction. The distinguishing advantage of Venn Predictors is that they have a calibration guarantee, for which the only requirement is that data be drawn independently from the same distribution (i.i.d. assumption). By calibration, we refer to the property that:

$$\mathbb{P}\left[Y = a \mid P_a\right] = P_a \qquad \text{almost surely} \tag{1}$$

that is, the probability that the label is a given that the predicted probability is P_a is indeed P_a. Practically, this corresponds to the relative frequencies of label being a tending to P_a when computed only on the objects for which the probability of label a is P_a.

In their general form described in Vovk et al. (2005), Venn Predictors output discrete probability distributions over the set of possible labels. Venn Predictors are described as multi-probability predictors because for each test object they output as many probability distributions as possible labels. Each of these distributions is obtained by assuming that the test object x has a given hypothetical label y and adding the example $z = (x, y)$ to the training set. Strictly speaking, the calibration property applies to one of such distributions, but exactly which one of the distributions cannot be predicted. The differences across various distributions however can provide an indication of how uncertain the probabilistic prediction itself is. The key component of Venn Predictors is the Venn Taxonomy, a partition of the space of the examples in which the elements (referred to as categories) are sets grouping together examples that can be considered similar for the purposes of calculating relative frequencies. The predicted distribution for a test object is the set of relative frequencies (one for each possible label) calculated on the category in which the hypothetical test example falls. The choice of the taxonomy is critical: a coarse taxonomy is going to have categories with many elements, leading to more robust and representative relative frequencies, but its predictions are going to be not very specific (or sharp, if one follows the terminology in (Gneiting et al.)). In the Venn Prediction framework, the taxonomy is obtained by means of the underlying Machine Learning algorithm. Note that once the taxonomy is established, the probabilistic predictions are determined (as relative frequencies of the labels) without further recourse to the underlying Machine Learning algorithm.

In the rest of the paper, we'll restrict our attention to binary Venn Predictors, i.e. Venn Predictors for two labels, which we'll denote arbitrarily as 0 and 1. In this context, the Venn Predictor outputs 2 probability distributions, one assuming label 0 for the test object and the other assuming label 1. It is customary (although potentially confusing) to refer to p_0 and p_1 to the probability of label 1 in the two distributions, respectively. In a rather broad sense, the p_0 and p_1 can be viewed as lower and upper probabilistic estimates.

Among binary Venn Predictors, *Venn-Abers* predictors (Vovk and Petej, 2014) can be used to calibrate a score produced by a scoring classifier, whenever such score is supposed to be directly related to the probability of the positive label. The score $s(x)$ (which could have any arbitrary domain, not restricted to [0,1]) is transformed into a multi-probabilistic calibrated prediction $(p_0(x), p_1(x))$, with $p_0(x) = g_0(s(x))$, $p_1(x) = g_1(s(x))$, where $g_0(s)$ and $g_1(s)$ are monotonic functions of s. In addition to being calibrated in the sense mentioned earlier, the probabilities $(p_0(x), p_1(x))$ also maximize the likelihood over the "augmented" training set (i.e. the training set plus the test object with a hypothetical label).

1.1. Outline of the approach

The aim of this work is to extend Venn Prediction to regression problems, i.e. with continuous labels as opposed to discrete labels. The approach we propose is to map the original regression setting onto a binary classification setting on which we can apply Venn-Abers prediction. This is achieved by choosing a threshold t and applying Venn Predictors to produce upper and lower estimates for $\mathbb{P}[\leq t]$. By repeating this for different thresholds t, we obtain the values at t of an upper and a lower predictive distribution $\hat{P}_1(t)$ and $\hat{P}_0(t)$.

The principal challenge this approach presents is the potential for inconsistencies. Inconsistencies here mean contradictions between probabilistic predictions at different threshold levels, i.e. $P_0(t_i) > P_0(t_j)$ for $t_i < t_j$. The Appendix of this paper includes a theoretical analysis of the problem and the justification of the framework. We will show that inconsistency is not avoidable in the transductive form of Venn-Abers scheme, but disappears for the *inductive* version of Venn machine developed in (Lambrou et al., 2015).

The method described here can be seen as complementary to the one proposed for conformal predictive distributions (Vovk et al., 2017), where a similar form of output (upper and lower bounds for the distribution) is produced by calculating conformal predictive regions for different significance levels.

Just as Venn-Abers is applicable to virtually any scoring classifier, the framework we propose produces valid predictive distributions on top of virtually any regression method. To illustrate its advantages, in Sec. 5 we provide a comparison with well-known Bayesian Linear Regressionshowing how the proposed method leads to narrower valid intervals on a synthetic data set. As a real-life case study, we apply the method to obtain a probabilistic prediction of patient survival time on the basis of historical data on patient characteristics and observed outcome, following Percutaneous Coronary Intervention (PCI), a non-surgical procedure used to treat narrowing (stenosis) of the coronary arteries of the heart.

2. Formal definition of the method

We start by recalling relevant notions from Vovk et al. (2005); Vovk and Petej (2014); Lambrou et al. (2015) related to Venn, Venn-Abers and Inductive Venn predictions for the binary classification task, i.e. when the predicted label can take one of 2 values, arbitrarily denoted here as 0 and 1.

2.1. Venn Predictor

Assume we are given training samples

$$(z_1, \ldots, z_n),\ z_i = (x_i, y_i),\ i = 1 \ldots n,$$

where $x_i \in R^d,\ y_i \in \{0, 1\},\ i = 1 \ldots n$.

Given a test object x_{n+1}, the Venn Predictor outputs a (multi-)probabilistic prediction in the form of a probability distribution over the possible values of the label.

For this we need to introduce the notion of *Venn taxonomy*. A Venn taxonomy A is a measurable function that assigns to each $n \in \{2, 3, \ldots\}$ and each sequence $(z_1, \ldots, z_n)$ an equivalence relation $\sim$ on $\{1, \ldots, n\}$. The relation has to be equivariant in the sense that, for each n and each permutation π of $\{1, \ldots, n\}$,

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

where $(i \sim j | z_1, \ldots, z_n)$ means that i is equivalent to j under the relation assigned by A to $(z_1, \ldots, z_n)$. Next we define a class of the equivalence of j as

$$A(j | z_1, \ldots, z_n) := \{i \in \{1, \ldots, n\} | (i \sim j | z_1, \ldots, z_n)\}.$$

A Venn predictor with a Venn taxonomy A outputs the pair (p_0, p_1), where

$$p_y = \frac{|\{i \in A(n+1 | z_1, \ldots, z_n, (x_{n+1}, y)) | y_i = 1\}|}{|A(n+1 | z_1, \ldots, z_n, (x_{n+1}, y))|} \tag{2}$$

Note that both p_0 and p_1 express the probability of the test object having 1 as label. $y \in \{0, 1\}$ will be referred to here as the *version* of Venn prediction.

Venn Predictors offer a calibration guarantee under the assumption that the observations are independently and identically distributed. A definition of calibration was given in eq. 1 in the Introduction; we refine it here to accommodate the multi-probabilistic nature of the prediction, according to (Vovk and Petej, 2014).

For a given discrete random variable Y, that takes values in $\{0, 1\}$ and Venn predictions $P_0, P_1 \in [0, 1]$, there exists a selector S (i.e. a random variable taking values 0 or 1) such that P_S is calibrated.

For further details, the reader is referred to (Vovk et al., 2005) where a more general but more complex game-theoretic form of Venn machine validity is discussed.

2.2. Venn-Abers Predictor

The Venn Prediction framework does not prescribe any method for constructing the taxonomy. Venn-Abers Predictors provide a method for establishing a taxonomy with desirable properties using a scoring classifier as underlying ML method. More specifically, the requirement posed by the Venn-Abers framework is that the score $s(x_{n+1})$ output by the underlying algorithm be directly related to the probability that label y_{n+1} is 1. In a nutshell, the higher the score, the higher the probability of the label being the positive one[1].

1. Note that a scoring classifier may not necessarily satisfy this property; in fact, for mere classification what is required is that the sign of the score be directly related to the probability of label.

The taxonomy constructed by Venn-Abers predictors (a partition of $\mathbb{R}$ into intervals) is designed so that the predicted probabilities are non-decreasing (as a function of s) and maximize the likelihood of the training set. This is achieved by a calibrator $g(s)$ maximizing:

$$\prod_{i=1,2,\ldots,n} p_i$$

where:

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

Based on results in (Ayer et al., 1955; Brunk, 1955; Zadrozny and Elkan, 2002), the sought $g(s)$ can be obtained as isotonic regression (i.e. order-preserving regression) on (s_i, y_i) defined as the non-decreasing function minimizing the sum of square residues:

$$\sum_{i=1}^{n} (g(s(x_i)) - y_i)^2$$

Then the multi-probabilistic prediction for x is the pair

$$(p_0, p_1) = (g_0(s_0(x)), g_1(s_1(x))).$$

Note that the isotonic regression is piecewise constant. The intervals of x where $(g(s(x))$ is constant are the categories of the Venn-Abers taxonomy.

2.3. Inductive Venn Predictor

The Venn machines described in the previous section are in general computationally demanding because they require that the underlying ML algorithm be retrained for every new test object and hypothetical label. They are said to be transductive (Vapnik, 1998; Gammerman et al., 1998); they are intended to provide predictions for one test object, as opposed to obtaining a model of general validity.

In the Inductive form, the underlying ML algorithm of a Venn machine is trained once only, on a subset of the training set which is referred to as the *proper training set*. The rest of the training set forms the *calibration set*. It is the calibration set that, along with the test object and its hypothetical labels, gets divided into the categories of the taxonomy and is used to compute the relative frequencies as defined in eq. 2.

In our notation, the proper training set is defined as:

$$T_P = \{z_{-1}, \ldots, z_{-r}\}.$$

and the calibration set as:

$$T_C = \{z_1, \ldots, z_h\}.$$

with $r + h = n$ and we denoted the test object as x_{h+1}.

Then $s(x)$ is defined as the score $S(x, T_P)$ assigned by the underlying method to x after training on the proper training set T_P. Since the proper training set is fixed, s can be considered as a function depending only on x.

The Inductive Venn Predictors enjoy the same type of calibration guarantee as their Transductive counterparts, whenever the scoring function can be represented in the form $S(x, T_P)$.

2.4. Inductive Venn-Abers Predictor

In the Inductive Venn-Abers Predictors, g is a monotonic function of s minimizing the sum of square residues over the calibration set T_C plus the test object with the hypothetical label:

$$\sum_{i=1}^{h+1}(g(s(x_{-i}, T_P)) - y_{-i})^2$$

Venn-Abers Predictors are known to satisfy calibration guarantees of Venn Predictors.

3. Inductive Venn-Abers Predictive Distributions

Our approach to Venn-Abers Predictive Distribution is a modification of Inductive Venn-Abers Predictor, therefore it is called Inductive Venn-Abers Predictive Distribution (IVAPD). A more straightforward transductive version is found to be inconsistent. Its description and inconsistency proof is located in Appendix A.

3.1. Methodology

3.1.1. From regression to classification

As in Sec. 2.3, we split the training set into a *proper training set* $T_P = \{z_{-1}, \ldots, z_{-r}\}$ and a *calibration set* $T_C = \{z_1, \ldots, z_h\}$, where $z_i = (x_i, y_i)$ is a pair of a feature vector $x_i \in \mathbb{R}^d$ and a label $y_i \in \mathbb{R}$. The difference in this setting is indeed that the labels y_i are real-valued, as opposed to binary.

By introducing a threshold t, we can reduce the regression problem of predicting of a continuous label to a binary classifications problem (or, rather, to a multiplicity of binary classification problems). Specifically, each y_i is replaced by $y_i^t = 1$ if $y_i > t$ and by $y_i^t = 0$ otherwise. For the new example x_{h+1} we try to answer the question whether $y_{h+1}^t = 1$ i.e. $y_{h+1} > t$.

3.1.2. Apply Venn-Abers Prediction

The straightforward application of the Venn-Abers framework to this problem would be resort to a function $s(x)$ learned by an underlying ML algorithm (which would correspond to the scoring function) and to fit Isotonic Regressions to the sets:

$$T_{C_0}^t = \left\{ \left((x_1), y_1^{(t)} \right), \ldots, \left(s(x_h), y_h^{(t)} \right), (s(x_{h+1}), 0) \right\}$$

$$T_{C_1}^t = \left\{ \left((x_1), y_1^{(t)} \right), \ldots, \left(s(x_h), y_h^{(t)} \right), (s(x_{h+1}), 1) \right\}$$

obtaining the two Isotonic Calibrators $g_0(s)$ and $g_1(s)$, respectively, which define implicitly the two taxonomies $\mathcal{A}_0^{(t)}$ and $\mathcal{A}_1^{(t)}$.
The resulting multi-probabilistic estimate is then

$$(p_0(x_{h+1}), p_1(x_{h+1})) = (g_0(s(x_{h+1})), g_1(s(x_{h+1})))$$

p_0 and p_1 by construction are estimates of $\mathbb{P}\left[y_{h+1}^{(t)} = 1\right]$ and consequently of $\mathbb{P}\left[y_{h+1} > t\right]$.

By varying t, we obtain values of the predictive distribution for y as we set out to do[2].

3.1.3. Computational simplifications

The method stated above can be computationally onerous. We now introduce a variant that retains the desirable properties while requiring simpler calculations.

The first simplification affects how the Isotonic Regression is computed. In the method above the IR is fitted and evaluated for every test object and every chosen t twice. In this streamlined version we propose to compute it once only on the training set. So, we determine the non-decreasing $g(s)$ that minimizes the sum of square residues on T_P:

$$\sum_{i=1}^{r}(g(s(x_{-i}, T_P)) - y_{-i})^2$$

Note that the function g is defined at the points x_{-i}; on other points it is evaluated using the 1-nearest-neighbour method, i.e. $g(s) = g(s_i)$ where s_i is the nearest proper training set point to s.

The function $g(s)$ induces a (single) taxonomy $\mathcal{A}^{(t)}$, whose categories are the values of x for which $g(s(x))$ has the same value.

$$\mathcal{A}^{(t)} := \{i = 1, \ldots, h+1 : g(s(x_i)) = g(s(x_{h+1}))\}.$$

For each t the probability estimate is then calculated as:

$$\hat{P}\{y_{n+1} > t\} = \frac{|\{i \in \mathcal{A}^{(t)}(h+1) : y_i^t = 1\}|}{|A_t|}$$

where $\mathcal{A}^{(t)}(h+1) = \mathcal{A}(h+1|(x_1, y_1^t), \ldots, (x_h, y_h^t), (x_{h+1}, y_{h+1}^t))$ is the class of equivalence (or category) of the new example, so $i \in \mathcal{A}(h+1|U)$ means the (x_i, y_i) and (x_{h+1}, y_{h+1}) are from the same class of equivalence when the taxonomy is applied to the set U.

In this inductive scheme, the taxonomy does not depend on the test object nor on the hypothetical label assigned to it. This allows us to apply a second simplification: the upper and lower predictive distribution can be expressed directly as the empirical distribution of y_i for $i \in \mathcal{A}_t$:

$$\hat{P}\{y_{h+1} \leq t\} = \frac{|\{i \in \mathcal{A}^{(t)} \setminus \{h+1\} : y_i \leq t\}| + q}{|\mathcal{A}^{(t)}|}$$

where $q \in \{0, 1\}$ is the hypothetical label assigned to the test object.

This scheme is summarised in Algorithm 1. The output is made in the form of two (lower and upper) cumulative distribution functions.

2. A predictive distribution is generally expressed as $\mathbb{P}[y \leq t]$, as for a (cumulative) distribution function, whereas we estimate $\mathbb{P}[y > t]$, but the former can be obtained banally from the latter.

Algorithm 1 Inductive Venn-Abers Predictive Distribution

INPUT: proper training set $T_P = \{(x_{-1}, y_{-1}), \ldots, (x_{-r}, y_{-r})\}$.
INPUT: calibration set $T_C = \{(x_1, y_1), \ldots, (x_h, y_h)\}$.
INPUT: testing example x_{h+1}.
INPUT: underlying predictor $P : (x, T) \to s$
for $i := 1, \ldots, r$ **do**
  $s_{-i} := P\left(x_i, T \setminus \{(x_{-i}, y_{-i})\}\right)$
end for
find $(g_{-1}, \ldots, g_{-r})$ s.t. $\sum_{i=1}^{r}(g_{-i} - y_{-i})^2 \to \min$ wrt. $(s_{-i} \le s_{-j}) \Rightarrow (g_{-i} \le g_{-j})$
for $i := 1, \ldots, h+1$ **do**
  $s_i := P\left(x_i, T_P\right)$
  find s_{-j} which is the closest to s_i (may be not unique)
  $g_i := g_{-j}$ (take average if not unique)
end for
let $A := \{i = 1, \ldots, h : g_i = g_{h+1}\}$
let $\hat{Y} := \{y_i : i \in A\}$
OUTPUT:

$$\hat{P}_0\{y_{h+1} \le t\} := \frac{|\{\hat{y} \in \hat{Y} : \hat{y} \le t\}|}{|A| + 1}$$

$$\hat{P}_1\{y_{h+1} \le t\} := \frac{|\{\hat{y} \in \hat{Y} : \hat{y} \le t\}| + 1}{|A| + 1}$$

3.1.4. Validity and Consistency

Assume than we apply the computation shortcut, recomputing the function $g = g_{T_P}$ on the base of the proper training set T_P. The algorithm still may be considered as a valid form of Inductive Venn Predictor earlier described in Sec. 2.3 with

$$S(x, T_P) = g_{T_P}\left(s(x, T_P)\right).$$

For consistency of the distribution $\hat{P}$, we need to ensure that $t < t'$ yields

$$\hat{P}\{y_i > t\} \ge \hat{P}\{y_i > t'\}.$$

In Appendix B, the consistency is shown for a class of algorithms including this approach.

4. Calibration role of IVAPD

As shown in Zhou et al. (2011), one advantage of the Venn Prediction framework is that it achieves calibration under less restrictive assumptions than other methods. For instance, parametric methods are known to be sensitive to the assumption of the distribution that is supposed to generate the observations. While there are domains in which such assumptions are justified as they are known to be consistent with the nature of the process that generated the data, in other fields such assumptions have weak support or are dangerously unwarranted.

In this section, we show the advantages of the distribution-free approach of Venn-Abers Predictive Distributions compared to Bayesian Linear (Ridge) Regression.

Both methods are assessed in terms of the prediction intervals that can be extracted from their output. We consider two metrics:

validity: for a confidence value, we compute the rate with which the predicted interval contains the true value and we compare it with the chosen confidence value.

efficiency: the average size of the confidence intervals. If a method outputs consistently narrower intervals, it provides more specific, hence useful predictions. Other possible measures of efficiency will be considered later in Sec. 5.4.

4.1. Datasets

We evaluated the performance of IVAPD on four datasets. Two of them are benchmark data sets from the well-known UCI Machine Learning Repository, whereas the other two are synthetic data sets.

The data sets are:

1. Facebook Metrics (Moro et al., 2016) dataset contains information about 500 Facebook posts. The goal is to predict the number of Total Interactions depending on the user behavior in a social network.

2. Energy Efficiency (Tsanas and Xifara, 2012) dataset contains energy analysis using 12 different simulted building shapes (totally 768 instances). The predicted value is Heating Load.

3. Artificial dataset (Gaussian noise) with the following mechanism of generation. First, generate vector $w \in \mathbb{R}^{10}$ as Normally distributed vector, where each coordinate has zero mean and variance 1. Then, a vector $x \in \mathbb{R}^{10\times 2000}$ is generated from uniform distribution $\mathbb{U}_{[0,1]}$. Finally, $y = w \cdot x + \varepsilon$, where the noise term $\varepsilon \sim \mathcal{N}(0,1)$ has Gaussian distribution with mean 0 and variance 1. Also we made the noise term equal to zero for a random half of the generated data.

4. Artificial dataset (Exponential noise) with the same mechanism of generation, as for previous dataset, with different distribution for the noise. Instead of Gaussian noise, we use variates from an exponential distribution with parameter 1, i.e. $\varepsilon \sim \exp(1)$. As for the previous dataset, we also set zero noise for a random half of data.

4.2. Bayesian Linear Regression as underlying method

As an example of underlying prediction method, we use Bayesian Linear (Ridge) Regression, which is described in (Bishop, 2006). We recap briefly the key points of BLR and we also show how it can be used to provide a predictive distribution.

Assume the target t is given by linear function f of the features $\boldsymbol{x} \in \mathbb{R}^d$ plus some noise term:

$$t = f(\boldsymbol{x}, \boldsymbol{w}) + \varepsilon = \sum_{i=1}^{d} w_i x_i + \varepsilon,$$

where $\boldsymbol{w} \in \mathbb{R}^d$ is the vector of the weights and the noise term $\varepsilon \sim \mathbb{N}(0, \beta)$ is characterized by a normal distribution with mean 0 and variance β. Therefore we can say the probability of a value t is given by a normal distribution around the value of the function $f(\boldsymbol{x}, \boldsymbol{w})$ at that point:

$$p(t \mid \boldsymbol{x}, \boldsymbol{w}, \beta) = \mathcal{N}(t \mid f(\boldsymbol{x}, \boldsymbol{w}), \beta^{-1}), \tag{3}$$

where $\mathcal{N}(t \mid \mu, \sigma^2)$ is a normal distribution with mean μ and variance σ^2.

In Bayesian LR, it is also assumed that weights are distributed according to a prior that takes the form of an isotropic, zero-mean Gaussian:

$$p(\boldsymbol{w}|\alpha) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{0}, \alpha^{-1}\boldsymbol{I})$$

where α is the precision parameter.
The prediction of t for new values of $\boldsymbol{x}$ is done via the predictive distribution defined by:

$$p(t|\boldsymbol{t}, \alpha, \beta) = \int p(t|\boldsymbol{w}, \beta) p(\boldsymbol{w}|\boldsymbol{t}, \alpha, \beta) d\boldsymbol{w}, \tag{4}$$

where $\boldsymbol{t}$ is the vector of target values from the training set.
Denoting by $\boldsymbol{X}$ the design matrix, the posterior weight distribution $p(\boldsymbol{w}|\boldsymbol{t}, \alpha, \beta)$ is given by

$$p(\boldsymbol{w}|\boldsymbol{t}) = \mathcal{N}(\boldsymbol{m}_N, \boldsymbol{S}_N),$$

where $\boldsymbol{S}_N^{-1} = \alpha\boldsymbol{I} + \beta\boldsymbol{X}^T\boldsymbol{X}$ and $\boldsymbol{m}_N = \beta\boldsymbol{S}_N\boldsymbol{X}^T\boldsymbol{t}$.

Note that the Bayesian LR relies on restrictive assumptions on the distribution of the noise. If the distribution of the noise in the observed data departs significantly from Gaussian, the Bayesian predictive distribution, which by construction follows a Gaussian distribution, will not reflect the actual form of the noise distribution. This is bound to affect the validity of the predicted confidence intervals.

4.3. Confidence Intervals

In order to compare Bayesian LR with IVAPD on validity and efficiency we used confidence intervals. For each test object both algorithms predict the distribution of the label. Confidence intervals were constructed for these distributions and then compared using the validity and efficiency metrics earlier defined in Sec.4.

In case of Bayesian LR confidence intervals are constructed from the distribution they predict. As soon as we know the class of the predicted distribution — for our case it is normal, we could consider standard intervals for it. For each test object Bayesian LR predicts mean and variance of the target value, so the target distribution for each test object is completely defined. In that case for Gauss distribution confidence intervals will be s.t. that areas around the median are equal.

For IVAPD, each of the data set was divided into three parts of equal size: proper training, calibration and testing set. For Bayesian LR, the testing set is the same, while two first parts together are used as the training set.

In case of IVAPD, the confidence intervals are obtained in the following way. For each of the two (lower and upper) distributions, we find the shortest interval which has the

probability determined by the confidence level. Then the union of these two intervals is used as the output.

The results for validity testing are presented in Figure 1. The *coverage* on y axis means the percentage of test objects for which the predicted confidence interval covers the real value.

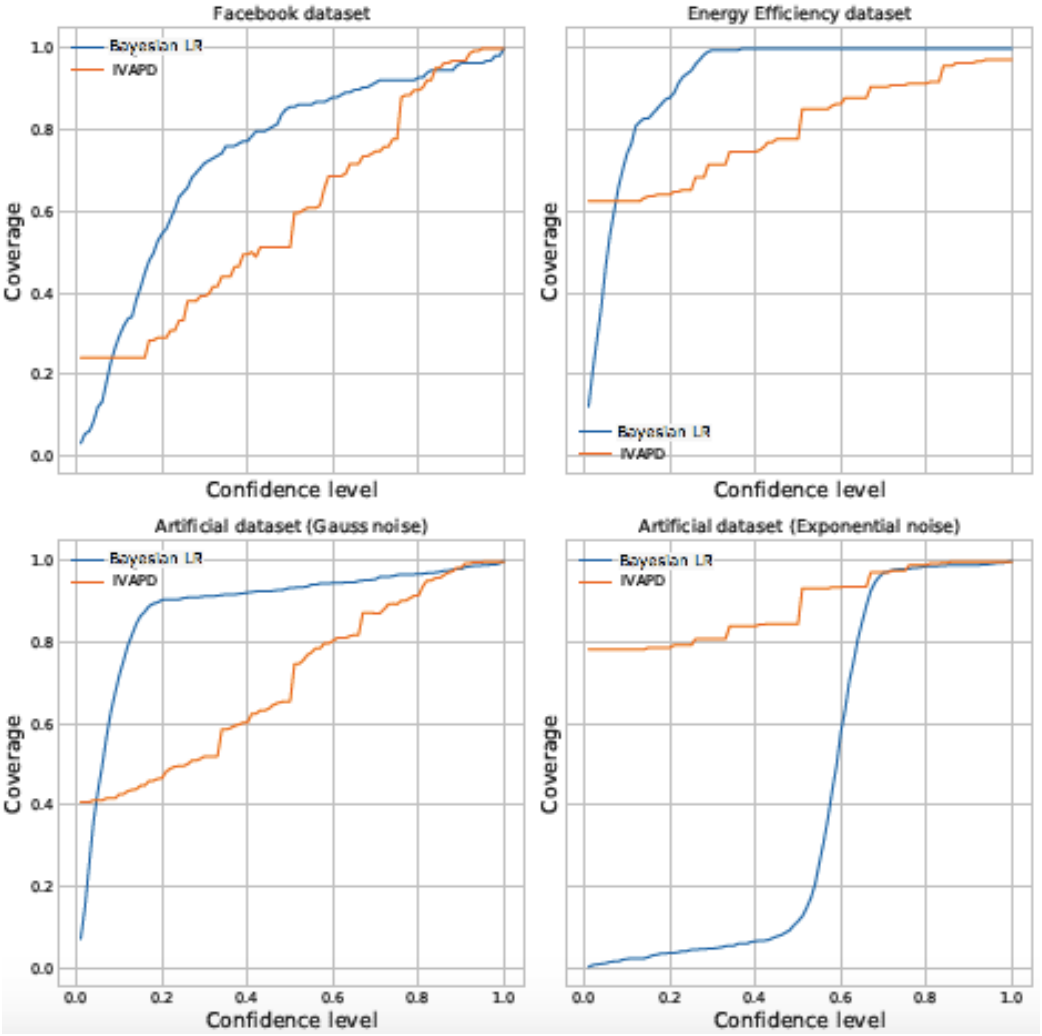

Figure 1: Coverage for different confidence levels: intervals produced by Bayesian LR directly vs. intervals produced by IVAPD with Bayesian LR underlying algorithm

The confidence level is the complement to 1 of the significance level and expresses the (chosen) probability that the confidence interval cover the true value of the label. Typical values of the confidence values used in practice are close to 1 (e.g. 95%). In Figure 1 a predictor with ideal validity would correspond to the diagonal. Points below the diagonal correspond to invalidity, while points above the diagonal are associated with conservative and potentially inefficient predictions.

The charts show that the results on real-world data sets are valid (above diagonal) for both approaches, although Bayesian LR typically shows larger deviation from the diagonal, indicating that the predictions are conservative and possibly leading to loss in efficiency.

When applied to the two artificial datasets, Bayesian LR exhibits validity only in one case, namely the one in which the noise is Gaussian (as assumed by the form of Bayesian LR used here). But if the noise is exponential, the confidence intervals depart from validity. On the other hand, IVAPD stays valid for both artificial datasets.

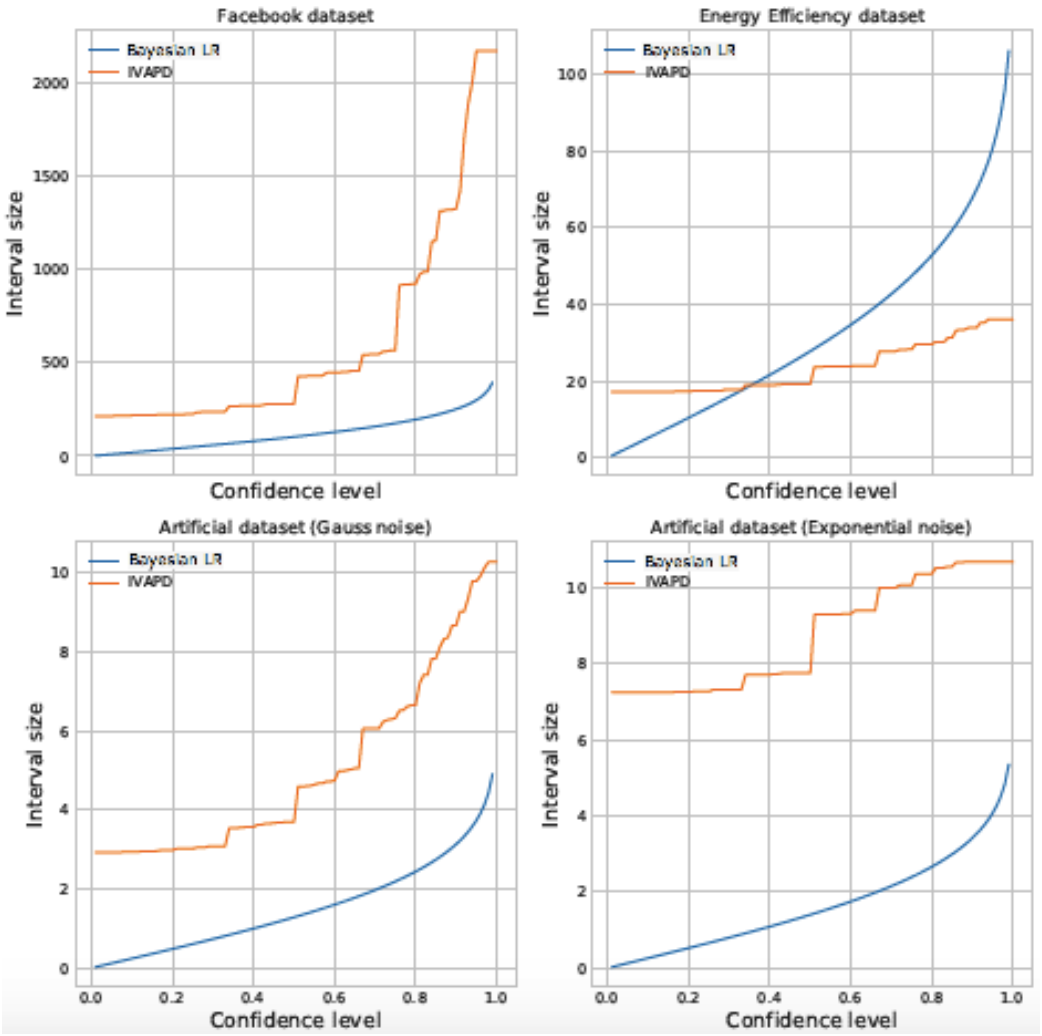

Figure 2: Interval width for different confidence levels: intervals produced by Bayesian LR directly vs. intervals produced by IVAPD with Bayesian LR underlying algorithm

Figure 2 presents efficiency chart in the form of interval width. The comparison has sense only when both methods show valid results, therefore it is not applicable to Artificial Dataset (Exponential noise) where Bayesian LR is sometimes far below the diagonal.

Validity of Facebook dataset is mostly correct but at risk at high values of confidence level (close to 1). The target value in case of Facebook is "Total Interactions" for each

user. If for confidence level α we obtain confidence interval of size d interactions, than with probability at least α the true value of Total Interactions will be inside the interval $[\mu - \frac{d}{2}; \mu + \frac{d}{2}]$, where μ — predicted mean value of Total Interactions for test object.

The same interpretation of interval width is for Energy Efficiency dataset. For confidence level α and interval width d the true Heating Load for a given test object will be inside $[\mu - \frac{d}{2}; \mu + \frac{d}{2}]$ with probability at least α.

Therefore the most essential example for comparison is Energy Efficiency as a real data example with doubtless validity. In this example Figure 2 shows better efficiency of IVAPD for high values of confidence level (which are commonly used).

In conclusion, the examples considered here confirm in practice the theoretically-backed validity of IVAPD for real and artificial data. In particular, the artificial dataset examples demonstrate practically that, for IVAPD, validity does not require assumptions on the data, but only that training and test observations be drawn independently from the same distribution.

5. Application

In this section, we apply IVAPD to an important medical problem: to estimate life expectancy of a patient following a Percutaneous Coronary Intervention procedure (PCI). Connecting the possible risk with the predictable survival time after the procedure leads to the decision whether the procedure can be performed or not. Providing the output is in the form of predictive distribution allows the doctors to make the choice.

Earlier, in medical applications, reliable methods of machine learning were mostly applied to classification (or diagnostic) problems. A review of some of them can be found in Nouretdinov et al. (2014). An example of Venn Machine application to a problem of early diagnostic can be found in Nouretdinov et al. (2015). An interesting example of regression estimation in medical field has been presented in Nouretdinov and Lebedev (2013).

In this work we consider the expected survival time of a patient after a risky PCI procedure.

The result of the calculation depends on what approach is used as the underlying method. For example, if the underlying method is over-fitted, this leads to larger difference between the upper and the lower estimates of probabilistic distribution, so the result is still valid and less efficient In this section we discuss and apply the criteria of evaluation, in order to show how different underlying method can be compared with each other.

5.1. Data description

Percutaneous Coronary Intervention life status dataset was collected in St.George Hospital of London. It contains 10,108 observations. Each observation represents a person who had the PCI procedure.

By recommendations of experts, the features listed in Table 1 were selected. The number (No.) in the first column refers to ID of the feature in the original data file. All the features are categorical, with exception of the age which is numerical. In our calculations the age is presented by a binary value: below/over 80.

All the observations have been randomised and only 4,212 were used when the features with missing values were excluded.

Table 1: Data features and labels

No.	Feature description	Range
-	Age (below/over 80)	30–97 (0–1)
1.07	Gender	1–2
2.03	Procedure Urgency	1–4
5.06	History of Renal Disease	0–4
2.04	Cardiogenic Shock Pre-Procedure	0–1
2.13	Previous MI	0–1
2.16	Diabetes	0–4
2.02	Indication for Intervention	1–10
5.07	Ventilated PreOp	0–1
5.15	Arterial Access	1–8

The life status ('Survived' on 'Not survived') was available for the patients taking the procedure, at the following 12 time points after the procedure: (1) 7 days; (2) 30 days; (3) 90 days; (4) 1 year; then (5-12) each 6 months until 5 years. The *survival time* for a patient is defined as the time point at which the patient was alive after the procedure, or 0 if there are no such points. We consider this time as a value predictable by regression. The observations were continued up to 5 years after the procedure and at that point the patient belongs to the category "survived".

5.2. Experimental settings

The randomised data examples were split into three parts of equal size (that is 1,403): proper training, calibration and testing sets. Algorithm 1 is applied to each example from the testing set as the testing example.

To include an underlying method into Algorithm 1 we need to define a function $s(x) = s(x, T_P)$ where $x \in R^d$ is an unlabelled feature vector and T_P is a set of labelled feature vectors $(x, y) \in R^d \times R$.

As the example of underlying method we use k Nearest Neighbours (kNN) with different values of k. Due to non-linearity of the problem we prefer it here to such methods as Linear Regression.

The underlying k-NN regression algorithm works as follows. For an example x, it finds its k nearest neighbours i.e. examples $(x_i, y_i) \in T_P$ with the smallest Hamming distance $H(x_i, x)$, and outputs the average of corresponding labels y_i of k nearest neighbours as $s(x)$.

5.3. Understanding individual predictions

An example of individual prediction is presented at Fig.3. The plot contains lower and upper estimates P_0 and P_1 for the predicted distribution P of the label y. The true value is 5 years (this patient in known to survive that time). The underlying method for NCM is k-NN with $k = 5$.

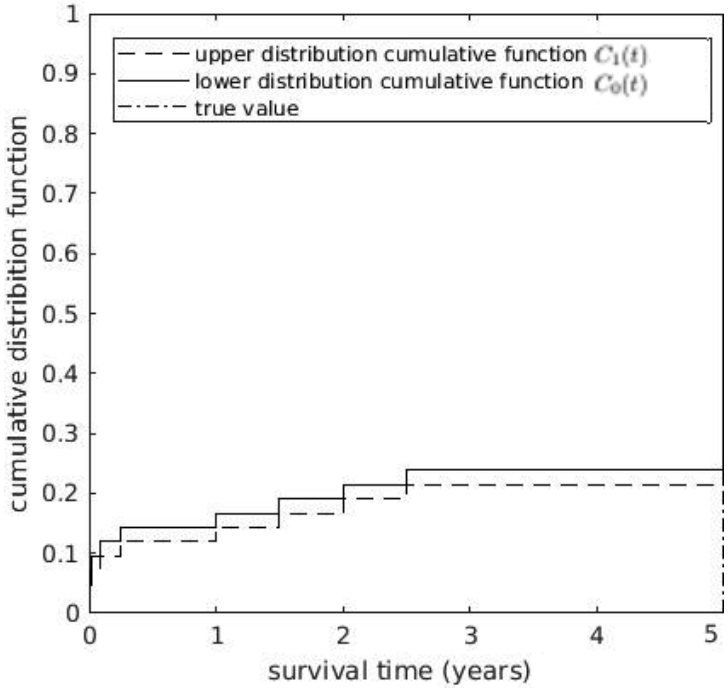

Figure 3: Examples of individual predictions: lower (pessimistic) and upper (optimistic) cumulative distribution functions $C_0(t)$ and $C_1(t)$ as functions of the survival time t.

One of ways to show a predictive distribution P on the plot is to present its non-decreasing cumulative distribution function (cdf), i.e. $P\{y : y < t\}$ as a function of t. We present distributions P_0 and P_1 as their cdf C_0 and C_1 where

$$C_i(t) = P_i\{y : y < t\}.$$

So, the plain line on Fig.3 is the lower (pessimistic) distribution cumulative function, and the dashed line the upper (optimistic) distribution cumulative function. The cdf reaches 1 at the maximal considered time point (5 years).

So for a time moment t, the prediction means that the probability of the survival time being below t - in other words, that the patient was not alive at time point t, - lies between $C_0(t)$ and $C_1(t)$. Therefore, there two values are considered as optimitic and pessimistic risk estimates.

Predictive distribution allows to answer the following questions about the risk of the procedure.

- **Question A.** What is the probability to survive after the procedure within a given time t? What is the risk (probability of non-survival)?

  Interpretation. What is the probability that the survival time is at least t / smaller than t?

  Answer. The survival probability lies between $1-C_0(t)$ and $1-C_1(t)$. The smallest of the two can be taken as the pessimistic estimate. The risk (understood as non-survival probability) is between $C_0(t)$ and $C_1(t)$.

Example. The procedure risk estimated for the first patient from Fig. 3 for 2-year survival time is about 20%, survival probability is estimated as 80%. If we look at 1-year survival time instead, the survival probability is estimated as 85%, the risk as 15%.

- **Question B.** What survival time after the procedure which can be guaranteed with probability p?

  Interpretation. What is the largest time point $\hat{t}$ with the property: the survival time is at least $\hat{t}$ will be wrong with probability at most $1-p$?

  Answer. It is $\min\{C_0^{-1}(1-p), C_1^{-1}(1-p)\}$.

  Example. If we return to the example from Fig. 3, for $p = 0.8$, the low risk time is 1.5 years; for $p = 0.9$, it is 3 months.

Both of these answers can be taken directly from the plot.

But the real advantage in using predictive distribution is that we can make a decision to use the procedure or not. Typically, the way of decision can be formulated as measuring the probabilistic expectation of a loss function that takes into account different factors (such as survival at more than one time point). Presentation of the output as predictive distribution allows to keep freedom of its choice. Once we have the full probability distribution and defined the loss function we can find an expected value.

5.4. Evaluation criteria

The algorithm produces valid predictions independently on its underlying method. However, different underlying methods can be evaluated and compared in terms of their relative efficiency (informativeness).

One of possible evaluation criteria (size of the prediction intervals) was earlier discussed in Sec. 4.3, but there are some other possibilities, because as we mentioned above, the output of Algorithm 1 can be interpreted in different ways according the preferred way of risk assessment. Table 2 shows several types of evaluation criteria.

First of all, it is possible to reduce the output to a question of classification: whether the example's labels lie within a given region U (such as a concrete ray or interval). The answer given by IVAPD is that this probability is either $P_0(U)$ or $P_1(U)$. A common way of measuring the performance of probabilistic predictions is using a loss function (such as logarithmic loss or Brier score) measuring the difference between the true label (yes or no) and the predicted probability. In order to compress two probabilistic predictions of a binary value to one, it is possible for example to use the formula that is given in Vovk and Petej (2014) and justified for logarithmic loss function:

$$P(U) = \frac{P_1(U)}{1 - P_0(U) + P_1(U)}.$$

Another criterion related to Venn Machine is the difference between $P_0(U)$ and $P_1(U)$ meaning precision of the probabilistic estimate. Typically, the loss and the precision complement each other. As a sort of intuitive approximation, it can be said, that the loss is

Table 2: Evaluation criteria for predictive distributions

No.	Inter-pretation	Criterion	Evaluated goal	Parameters
1	Prob.	Log-loss	Fitting	region U
2	Prob.	Precision	Overfitting	region U
3	Interval	Width	Inf-ness	significance level ε
4	Regression	Square loss	Accuracy	

the penalty for under-fitting of the underlying method, while high difference between upper and lower estimates reflects over-fitting, too complex underlying method.

An alternative direction is interpretation in the form of reliable regression, similar to one earlier used in Sec. 4.3. The question has a form: what interval covers the label with probability at least $(1-\varepsilon)$? Any interval U satisfying the property

$$\min\{P_0(U), P_1(U)\} \geq (1-\varepsilon)$$

is suitable. It is possible to select one of them which is optimal in some sense, for example having the smallest length. Its length may be the way of assessment: the smaller the interval is, the more informative is the prediction.

In addition, we mention the possibility to interpret the predictive distribution in terms of simple regression. This can be done by taking the average (expectation) of a random value distributed according to the mixture of lower and upper distributions. This criterion can be also used to compare the output of IVAPD with the straightforward output of the underlying method in terms of the accuracy.

5.5. Evaluation results

Here we compare the results of prediction with k-NN underlying method according evaluation criteria from Sec.5.4. For each of the criteria, the median over the testing set is taken.

The results are presented in Table 3. Three values of $k = 5, 25, 100$ are compared to each other.

Most of the criteria shows better quality of $k = 100$, although for some of the tasks $k = 5$ also works relatively well.

In addition, we show in Table 4 that the accuracy of the predictive distribution (reduced to its average for comparison) is not worse than one of the underlying method. The accuracy is measured as average square residuals. The shown effect is similar to one shown in Zhou et al. (2011) as a positive effect of calibration made by Venn-Abers framework. Observed in another dimension, this table also confirms the advantage of $k = 100$ parameter value.

6. Conclusion

In this paper, we presented a framework for reliable regression that gives output in its most complete form: lower and upper estimates for the whole predictive distribution. It allows

Table 3: Evaluation of the results

k	U	Log-loss	Prec.	ε	Width
5	$\{y : y > 15\}$	0.0296	0.0177	0.1	1026
	$\{y : y > 365\}$	0.0645	0.0177	0.2	423
25	$\{y : y > 15\}$	0.0458	0.0260	0.1	983
	$\{y : y > 365\}$	0.1001	0.0260	0.2	482
100	$\{y : y > 15\}$	0.0236	0.0177	0.1	1074
	$\{y : y > 365\}$	0.0708	0.0177	0.2	389

Table 4: Evaluation in terms of regression: average square residuals (in days) compared for the underlying method and IVAPD.

k	Underlying	IVAPD
5	579	565
25	554	547
100	570	544

interpretations both in terms of survival probability for a given threshold, and in terms of predictive regions, that cover true value with given probability. The framework allows to include any kind of standard regression method as its underlying algorithm. The choice of underlying algorithm may be evaluated by difference between lower and upper distributions, and by other criteria.

Acknowledgements

This work was supported by European Union Grant 671555 ("ExCAPE"), and by Technology Integrated Health Management (TIHM) project awarded to the School of Mathematics and Information Security at Royal Holloway as part of an initiative by NHS England supported by InnovateUK.

References

Miriam Ayer, H. D. Brunk, G. M. Ewing, W. T. Reid, and Edward Silverman. An empirical distribution function for sampling with incomplete information. *Ann. Math. Statist.*, 26 (4):641–647, 12 1955. doi: 10.1214/aoms/1177728423. URL https://doi.org/10.1214/aoms/1177728423.

Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. ISBN 0387310738.

H. D. Brunk. Maximum likelihood estimates of monotone parameters. *Ann. Math. Statist.*, 26(4):607–616, 12 1955. doi: 10.1214/aoms/1177728420. URL https://doi.org/10.1214/aoms/1177728420.

A. Gammerman, V. Vovk, and V. Vapnik. Learning by transduction. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, UAI'98, pages 148–155, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc. ISBN 1-55860-555-X. URL http://dl.acm.org/citation.cfm?id=2074094.2074112.

Tilmann Gneiting, Fadoua Balabdaoui, and Adrian E. Raftery. Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(2):243–268. doi: 10.1111/j.1467-9868.2007.00587.x. URL https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9868.2007.00587.x.

Antonis Lambrou, Ilia Nouretdinov, and Harris Papadopoulos. Inductive venn prediction. *Annals of Mathematics and Artificial Intelligence*, 74(1-2):181–201, 2015.

Sérgio Moro, Paulo Rita, and Bernardo Vala. Predicting social media performance metrics and evaluation of the impact on brand building: A data mining approach. *Journal of Business Research*, 69(9):3341–3351, 2016.

Ilia Nouretdinov and Alexander Lebedev. Defensive forecast for conformal bounded regression. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*, pages 384–393. Springer, 2013.

Ilia Nouretdinov, Tony Bellotti, and Alex Gammerman. *Biomedical Applications: Diagnostic and Prognostic*, pages 217–230. Elsevier, 2014. ISBN 978-0-12-398537-8.

Ilia Nouretdinov, Dmitry Devetyarov, Volodya Vovk, Brian Burford, Stephane Camuzeaux, Aleksandra Gentry-Maharaj, Ali Tiss, Celia Smith, Zhiyuan Luo, Alexey Chervonenkis, et al. Multiprobabilistic prediction in early medical diagnoses. *Annals of Mathematics and Artificial Intelligence*, 74(1-2):203–222, 2015.

Athanasios Tsanas and Angeliki Xifara. Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools. *Energy and Buildings*, 49:560–567, 2012.

Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998.

Vladimir Vovk and Ivan Petej. Venn-abers predictors. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, pages 829–838. AUAI Press, 2014.

Vladimir Vovk, Alex Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer Science & Business Media, 2005.

Vladimir Vovk, Jieli Shen, Valery Manokhin, and Min-ge Xie. Nonparametric predictive distributions based on conformal prediction. In *Conformal and Probabilistic Prediction and Applications*, pages 82–102, 2017.

Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, pages 694–699, New York, NY, USA, 2002. ACM. ISBN 1-58113-567-X. doi: 10.1145/775047.775151. URL http://doi.acm.org/10.1145/775047.775151.

Chenzhe Zhou, Ilia Nouretdinov, Zhiyuan Luo, Dmitry Adamskiy, Luke Randell, Nick Coldham, and Alex Gammerman. A comparison of venn machine with platts method in probabilistic outputs. In *Artificial Intelligence Applications and Innovations*, pages 483–490. Springer, 2011.

Appendix A. Transductive VAPD

Consider the training data with real-valued labels

$$(x_1, y_1), \ldots, (x_n, y_n)$$

and a testing example x_{n+1} with the label y_{n+1} for prediction.

For a given threshold t we are able to reduce the prediction problem to the binary one. We assume that there is a binary Venn taxonomy defined as an equivalence relation on the set of objects with binary labels. Each y_i is replaced by $y_i^t = 1$ if $y_1 > t$ and by $y_i^t = 0$ otherwise. For the new example x we try to answer the question whether $y_n^t = 1$ i.e. $y_n > t$.

In Venn algorithm both labels 0 and 1 are assigned to test object x for calculations, two versions $y_{n+1}^t \in \{0, 1\}$ are checked. Assume that this choice is already made that is the same for any t.

Then for each t we will get probability estimate:

$$\hat{P}\{y_{n+1} > t\} = \frac{|\{i \in A_t : y_i^t = 1\}|}{|A_t|}$$

where $A_t = A(n+1|(x_1, y_1^t), \ldots, (x_n, y_n^t), (x_{n+1}, y_{n+1}^t))$ is the class of equivalence of the new example, so $i \in A(n+1|U)$ means the (x_i, y_i) and (x_{n+1}, y_{n+1}) are from the same class of equivalence when the taxonomy is applied to the set U.

For consistency of the distribution $\hat{P}$, we need to ensure that $t < t'$ yields

$$\hat{P}\{y_i > t\} \geq \hat{P}\{y_i > t'\}.$$

Below we will show how this condition may be broken.

Inconsistency example

We have to study how $\hat{P}\{y_i > t\}$ behaves in dependence on t. Assume for convenience that all the real-valued labels of the training set are different, and the training example is ordered by label:

$$-\infty < y_1 < y_2 < y_3 < \cdots < y_n < +\infty$$

Whenever t is between y_k and y_{k+1}, examples the classification labels $y_i^t = 0$ for $i = 1, \ldots, k$ and $y_i^t = 1$ for $i = k+1, \ldots, n$. Denote the new example as $z = (x_{n+1}, y_{n+1})$. A_t is the class

Table 5: Inconsistency example for 1NN VACP

		t				2				1
i	x_i	y_i	y_i^2	s_i	g	A_2	y_i^1	s_i	g	A_1
3	11	2.5	1	0	0.25	+	1	1	1	
2	22	1.5	0	1	0.25	+	1	1	1	
1	44	0.2	0	0	0.25	+	0	0	0	+
4	55	0.3	0	0	0.25	+	0	0	0	+
		$\hat{P}_t$				0.25				0

of equivalence of the new example according to the classification taxonomy. Obviously, A_t is the same for any t from $y_k < t < y_{k+1}$.

Table 5 presents an example showing that this version of Venn-Abers method may be inconsistent. It shows an example when changing the threshold from $t = 2$ to $t = 1$ (with increasing the binary label of example 7) makes *decrease* of the estimate $\hat{P}_t = \hat{P}\{y > t\}$ for the predicted example 4.

For each of two values of t we start with reducing the label to its binary version (y_i^t). The considered version of probabilistic estimate is fixed as $y_{n+1}^2 = y_{n+1}^1 = 0$. Then the 1-nearest-neighbours prediction is made for each examples, s_i means the label of the example closest to x_i. $g(s_i)$ is the isotonic calibrator calculated from s_i and y_i^t. A_t is the corresponding class of equivalence: plus sign is assigned to all the examples having the same $g(s_i)$ as the new example $i = 14$. $\hat{P}_t$ is the proportion of positive labels ($y_i^t = 1$) within this class.

One can make the following explanation of inconsistency: changing the label of example 2 from negative (0) to positive (1) influence (increasing the prediction scores) some part of the new example's class of equivalence, that do not cover the new example itself. Therefore the new example's class of equivalence losses its 'progressive' part and becomes less 'positive' in average. This is actually a natural consistence of the uncalibrated process of adding examples one-by-one.

Appendix B. Sufficient conditions for consistency

Here we a proof of consistency for a more general class of Venn machines. The case of Inductive Venn Prediction can be interpreted in terms of this statement in the way discussed in Sec. 2.3. In this Appendix we omit the proper training set T_P as the argument of the taxonomy function, assuming it to be fixed.

Statement 1. *If the taxonomy function has the form $A_0(z_j)$, then Venn predictor is consistent i.e. $t < t'$ yields*

$$\hat{P}\{y_i > t\} \geq \hat{P}\{y_i > t'\}$$

for each of the versions $y \in \{0, 1\}$.

Proof: Remind that

$$\hat{P}\{y > t\} = \frac{|\{i \in A_t : y_i^t = 1\}|}{|A_t|}$$

where A_t is the class of equivalence of the new example. It is enough to check the statement for $y_{i-1} < t < y_i < t' < y_{i+1}$. The possible cases are:

1. *neither* $(x_i, 0)$ *nor* $(x_i, 1)$ *is equivalent to* z*:* $\hat{P}\{y_i > t\} = \hat{P}\{y_i > t'\}$ because x_i affects neither $\hat{P}\{y > t\}$ nor $\hat{P}\{y > t'\}$;

2. *both* $(x_i, 0)$ *and* $(x_i, 1)$ *are equivalent to* z*:* $\hat{P}\{y_i > t\} > \hat{P}\{y_i > t'\}$ because the taxonomy is the same, a negative example within the taxon just becomes positive when the threshold t' is changed to t, so the percentage of positive examples within this taxon increases;

3. $(x_i, 1)$ *is equivalent to* z*,* $(x_i, 0)$ *is not:* moving the threshold from t' to t makes the class of equivalence of the new example larger by adding one positive example $(x_i, 1)$ therefore $\hat{P}\{y_i > t\} > \hat{P}\{y_i > t'\}$;

4. $(x_i, 0)$ *is equivalent to* z*,* $(x_i, 1)$ *is not:* moving the threshold from t' to t makes the class of equivalence of the new example smaller by removing one negative example $(x_i, 1)$ therefore $\hat{P}\{y_i > t\} > \hat{P}\{y_i > t'\}$.