Transductive conformal predictors

Vladimir Vovk

практические выводы
теории вероятностей
могут быть обоснованы
в качестве следствий
гипотез о *предельной*
при данных ограничениях
сложности изучаемых явлений

On-line Compression Modelling Project (New Series)

Working Paper #8

First posted April 24, 2013. Last revised September 21, 2013.

Project web site:
http://alrw.net

Abstract

This paper discusses a transductive version of conformal predictors. This version is computationally inefficient for big test sets, but it turns out that apparently crude "Bonferroni predictors" are about as good in their information efficiency and vastly superior in computational efficiency.

Contents

1 Introduction

The most standard learning problems are inductive: given a training set of labelled objects, the task is to come up with a predictor with a reasonable performance on unknown test objects. In typical transductive problems ([7], Chapter VI, Sections 10–13, [6], Chapter 8) we are given both a training set of labelled objects and a test set of unlabelled objects; the task is to come up with a predictor, which may depend on both sets, with a reasonable performance on the test set. Conformal predictors ([9], Chapter 2) are not transductive in this sense, although they do have a transductive flavour (see, e.g., [9], pp. 6–7).

The goal of this paper is to introduce a fully transductive version of conformal predictors. The basic definitions are given in Section 2. Section 3 introduces Bonferroni predictors, a simple and computationally efficient modification of conformal predictors adapted to the transductive framework. Sections 4 and 5 contain simple theoretical results about transductive conformal predictors and Bonferroni predictors. Section 6 reports on experimental results. Finally, Section 7 concludes.

The expression "transductive conformal predictors" has been used before (see, e.g., [3]) to refer to what is called "conformal predictors" in [9] and this paper. This agrees with our terminology, since conformal predictors are a special case of our transductive conformal predictors corresponding to a test set of size 1.

2 Transductive Conformal Predictors

Let $z_1 = (x_1, y_1), \dots, z_l = (x_l, y_l)$ be a training set and $x_{l+1}, \dots, x_{l+k}$ be a test set. The test set consists of *objects* $x_j \in \mathbf{X}$ and the training set consists of labelled objects, or *examples*, $z_i = (x_i, y_i) \in \mathbf{Z} := \mathbf{X} \times \mathbf{Y}$. The *object space* $\mathbf{X}$, *label space* $\mathbf{Y}$, and *example space* $\mathbf{Z} := \mathbf{X} \times \mathbf{Y}$ are fixed throughout the paper; they are assumed to be measurable spaces.

Transductive conformal predictors are determined by their transductive nonconformity measures, which are defined as follows. A *transductive nonconformity measure* is a measurable function $A : \mathbf{Z}^* \times \mathbf{Z}^* \to \mathbb{R}$ such that $A(\zeta_1, \zeta_2)$ does not depend on the ordering of ζ_1. (For the specific transductive nonconformity measures used in this paper $A(\zeta_1, \zeta_2)$ will not depend on the ordering of ζ_2 either.) The intuition is that $A(\zeta_1, \zeta_2)$ (the *transductive nonconformity score*) measures the lack of conformity of the "test set" ζ_2 to the "training set" ζ_1.

The *transductive conformal predictor* (*TCP*) corresponding to A finds the prediction region for the test set $x_{l+1}, \dots, x_{l+k}$ at a *significance level* $\epsilon \in (0, 1)$ as follows:

- For each possible set of labels $(\upsilon_1, \dots, \upsilon_k) \in \mathbf{Y}^k$:
  - set $y_j := \upsilon_{j-l}$ and $z_j := (x_j, y_j)$ for $j = l+1, \dots, l+k$;
  - compute the transductive nonconformity scores

$$\alpha_S := A(z_{\{1,\dots,l+k\} \setminus S}, z_S),$$

where S ranges over all $(l+k)!/l!$ ordered subsets $(s_1,\dots,s_k)$ of $\{1,\dots,l+k\}$ of size k, z_S stands for the sequence $(z_{s_1},\dots,z_{s_k})$ (when $S=(s_1,\dots,s_k)$), and $z_{\{1,\dots,l+k\}\setminus S}$ stands for z_B, B being any ordering of $\{1,\dots,l+k\}\setminus S'$ and S' being the set of all elements of S (it does not matter which ordering is chosen, by the definition of a transductive nonconformity measure);

– compute the p-value

$$p(\upsilon_1,\dots,\upsilon_k) := \frac{\left|\{S \,|\, \alpha_S \geq \alpha_{(l+1,\dots,l+k)}\}\right|}{(l+k)!/k!}\,, \tag{1}$$

where S ranges, as before, over all $(l+k)!/l!$ ordered subsets of $\{1,\dots,l+k\}$ of size k, and $|\dots|$ stands for the size of a set.

- Output the prediction region

$$\Gamma^\epsilon(z_1,\dots,z_l,x_{l+1},\dots,x_{l+k}) := \left\{(\upsilon_1,\dots,\upsilon_k)\in\mathbf{Y}^k \,|\, p(\upsilon_1,\dots,\upsilon_k)>\epsilon\right\}. \tag{2}$$

Smoothed TCPs are defined in the same way except that (1) is replaced by

$$p(\upsilon_1,\dots,\upsilon_k) := \frac{\left|\{S \,|\, \alpha_S > \alpha_{(l+1,\dots,l+k)}\}\right| + \theta\left|\{S \,|\, \alpha_S = \alpha_{(l+1,\dots,l+k)}\}\right|}{(l+k)!/k!}\,,$$

where θ are random variables distributed uniformly on $[0,1]$ (no independence between different sets of postulated labels $\upsilon_1,\dots,\upsilon_k$ is required, but later on when we consider the online prediction protocol we will assume that θ are independent between different trials).

A *nonconformity measure* can now be defined as the restriction of a transductive nonconformity measure to the domain $\mathbf{Z}^*\times\mathbf{Z}$ (we identify a 1-element sequence with its only element). Nonconformity measures are well studied and there are many useful examples of them (see, e.g., [9]). For example, a natural choice of a nonconformity measure is

$$A(\zeta,(x,y)) := \Delta(y,f(x))\,, \tag{3}$$

where $f:\mathbf{X}\to\mathbf{Y}'$ is a prediction rule found from ζ as the training set and $\Delta:\mathbf{Y}\times\mathbf{Y}'\to\mathbb{R}$ is a distance between a label and a prediction. (Usually $\mathbf{Y}'\supseteq\mathbf{Y}$, such as $\mathbf{Y}'=[0,1]\supseteq\{0,1\}=\mathbf{Y}$.)

An interesting class of transductive nonconformity measures can be obtained from nonconformity measures. Let $\mathbb{R}$ be the set of real numbers. A *simple nonconformity aggregator* is a function $M:\mathbb{R}^*\to\mathbb{R}$ that is symmetric and increasing in each argument. (The requirement that M be symmetric, i.e., $M(\zeta)$ not depend on the ordering of ζ, is not necessary but convenient for the following discussion. The requirement that M be increasing in each argument is not necessary either but very natural.) With each nonconformity measure A and simple nonconformity aggregator M we can associate the transductive nonconformity measure

$$A_M((z_1,\dots,z_l),(z_{l+1},\dots,z_{l+k})) := M(\alpha_{l+1},\dots,\alpha_{l+k})\,,$$

where

$$\alpha_j := A((z_1,\dots,z_l,z_{l+1},\dots,z_{j-1},z_{j+1},\dots,z_{l+k}),z_j), \qquad j = l+1,\dots,l+k\,. \quad (4)$$

Our experiments in Section 6 use the *Nearest Neighbour nonconformity measure*

$$A\big(((x_1,y_1),\dots,(x_l,y_l)),(x,y)\big) := \frac{\min_{i=1,\dots,l:y_i=y} d(x,x_i)}{\min_{i=1,\dots,l:y_i\neq y} d(x,x_i)}\,, \quad (5)$$

where d is a distance, and the *max nonconformity aggregator*

$$M(\alpha_1,\dots,\alpha_k) := \max(\alpha_1,\dots,\alpha_k)\,. \quad (6)$$

Remark. Alternatively, we could set $\alpha_j := A((z_1,\dots,z_l),z_j)$ in (4) (cf. (13) below), but this would adversely affect the already low computational efficiency of TCPs in our experiments in Section 6.

Rank-based Transductive Conformal Predictors

The notion of a simple nonconformity aggregator can be generalized as follows. A *nonconformity aggregator* is a function $M : \mathbb{R}^* \times \mathbb{R}^* \to \mathbb{R}$ such that $M(\zeta_1,\zeta_2)$ depends neither on the ordering of ζ_1 nor on the ordering on ζ_2. (The most natural class of nonconformity aggregators is where $M(\zeta_1,\zeta_2)$ is decreasing in every element of ζ_1 and increasing in every element of ζ_2, but it is too narrow for our purposes.) With each nonconformity measure A and nonconformity aggregator M we associate the transductive nonconformity measure

$$A_M((z_1,\dots,z_l),(z_{l+1},\dots,z_{l+k})) := M\big((\alpha_1,\dots,\alpha_l),(\alpha_{l+1},\dots,\alpha_{l+k})\big)$$

where

$$\alpha_i := A((z_1,\dots,z_{i-1},z_{i+1},\dots,z_l,z_{l+1},\dots,z_{l+k}),z_i), \qquad i = 1,\dots,l, \quad (7)$$

and $\alpha_{l+1},\dots,\alpha_{l+k}$ are defined by (4). We identify each simple nonconformity aggregator M with the nonconformity aggregator

$$M^\dagger((\alpha_1,\dots,\alpha_l),(\alpha_{l+1},\dots,\alpha_{l+k})) := M(\alpha_{l+1},\dots,\alpha_{l+k})\,.$$

For transductive nonconformity measures obtained from nonconformity measures and nonconformity aggregators, the p-value (1) as function of the nonconformity scores α_i of individual examples reduces to the well-known notion of a one-sided permutation test (see, e.g., [1], Section 1.7.E). In classical nonparametric statistics, the most popular permutation tests are rank tests, and we will give corresponding definitions in our current context. Let $\mathbb{N} := \{1,2,\dots\}$. A (simple) *rank aggregator* is a function $M : \mathbb{N}^* \to \mathbb{N}$ that is symmetric and increasing in each argument. The corresponding nonconformity aggregator is

$$M'\big((\alpha_1,\dots,\alpha_l),(\alpha_{l+1},\dots,\alpha_{l+k})\big) := M(R_{l+1},\dots,R_{l+k})\,, \quad (8)$$

where $R_1, \ldots, R_{l+k}$ are the ranks of $\alpha_1, \ldots, \alpha_{l+k}$, respectively, in the multiset $\wr\alpha_1, \ldots, \alpha_{l+k}\wr$. Formally, R_i is defined as

$$R_i := |\{j = 1, \ldots, l+k \,|\, \alpha_j < \alpha_i\}| + 1 \,.$$

If there are no ties (i.e., equal elements in $\wr\alpha_1, \ldots, \alpha_{l+k}\wr$), this is the usual notion of a rank; in the presence of ties, our definition is somewhat non-standard giving each tie the smallest of the ranks that it spans. (And this definition causes a counterintuitive behaviour of the definition (8), where M' is not necessarily increasing in α_j, $j \in \{l+1, \ldots, l+k\}$, even in the case where M is the max nonconformity aggregator (6).)

The most popular rank aggregator in classical nonparametric statistics is the *ranksum aggregator*

$$M(R_1, \ldots, R_k) := R_1 + \cdots + R_k \,, \tag{9}$$

which is used in the Wilcoxon ranksum test (see [10] or [1], Section 1.2). Using the ranksum aggregator, however, produces very poor results (see Appendix B) when the efficiency of TCPs is measured by the number of multiple predictions that they produce, as in this paper (see Section 6 below).

Notice that the nonconformity aggregator (6) is equivalent (in the sense of leading to the same TCP) to the rank aggregator $M'(R_1, \ldots, R_k) := \max(R_1, \ldots, R_k)$. The corresponding TCP will be called the *rankmax TCP* (and the TCP corresponding to (9) will be called the *ranksum TCP*).

It is easy to give an explicit representation of the rankmax TCP. Remember that the size of the training set is l and the size of the test set is k and suppose that the value of the *rankmax test statistic* $\max(R_{l+1}, \ldots, R_{l+k})$ is t. The probability that a random subset $\{s_1, \ldots, s_k\}$ of $\{1, \ldots, l+k\}$ of size k will lead to a value of the test statistic $\max(R_{s_1}, \ldots, R_{s_k})$ of at least t can be found as 1 minus the probability that a random subset of $\{1, \ldots, l+k\}$ of size k is covered by a fixed subset of $\{1, \ldots, l+k\}$ of size $t-1$ (namely, by the set of indices i with $R_i < t$). In other words, the p-value is

$$1 - \frac{\binom{t-1}{k}}{\binom{l+k}{k}} = 1 - \frac{(t-1)!\, l!}{(t-1-k)!\, (l+k)!} \tag{10}$$

(which is understood to be 1 when $t \le k$).

3 Bonferroni Predictors

Unfortunately, transductive conformal predictors are computationally inefficient, especially if we want to predict many test objects at once: we have to go over all $|\mathbf{Y}|^k$ combinations of labels for the test set. (Even if $A(\zeta_1, \zeta_2)$ does not depend on the ordering of ζ_2, there are no computational savings unless the test set contains many identical objects.) We next introduce a family of region predictors based on the idea of the Bonferroni adjustment of p-values. In brief,

a Bonferroni predictor computes a p-value for each test object separately and then combines the k p-values into one p-value using the Bonferroni formula

$$p := \min(kp_1, \dots, kp_k, 1)\,. \tag{11}$$

The full description of the *Bonferroni predictor* (*BP*) corresponding to a nonconformity measure A is as follows:

- For each object x_j, $j \in \{l+1, \dots, l+k\}$, in the test set and each possible label $\upsilon \in \mathbf{Y}$:
  - set $y_j := \upsilon$ and $z_j := (x_j, y_j)$;
  - compute the nonconformity scores

$$\alpha_i := A((z_1, \dots, z_{i-1}, z_{i+1}, \dots, z_l, z_j), z_i), \qquad i = 1, \dots, l, \tag{12}$$

$$\alpha_j := A((z_1, \dots, z_l), z_j)\,; \tag{13}$$

  - compute the p-value

$$p_{j-l}(\upsilon) := \frac{|\{i = 1, \dots, l \,|\, \alpha_i \geq \alpha_j\}| + 1}{l+1}\,. \tag{14}$$

- Output the prediction region

$$\Gamma^\epsilon(z_1, \dots, z_l, x_{l+1}, \dots, x_{l+k}) := \prod_{j=l+1}^{l+k} \{\upsilon \,|\, p_{j-l}(\upsilon) > \epsilon/k\}\,, \tag{15}$$

  where $\epsilon \in (0, 1)$ is the significance level.

Notice that the prediction region (15) output by the BP can be rewritten in the form (2) if we define

$$p(\upsilon_1, \dots, \upsilon_k) := \min(kp_1(\upsilon_1), \dots, kp_k(\upsilon_k), 1) \tag{16}$$

(cf. (11)).

It is difficult to compare the rankmax TCP and the corresponding BP theoretically, but the following intermediate notion facilitates a comparison. The *semi-Bonferroni predictor* (*SBP*) is defined as follows:

- For each possible set of labels $(\upsilon_1, \dots, \upsilon_k) \in \mathbf{Y}^k$ for the test set:
  - set $y_j := \upsilon_{j-l}$ and $z_j := (x_j, y_j)$ for $j = l+1, \dots, l+k$;
  - compute nonconformity scores α_i, $i = 1, \dots, l+k$, by

$$\alpha_i := A((z_1, \dots, z_{i-1}, z_{i+1}, \dots, z_{l+k}), z_i), \quad i = 1, \dots, l+k \tag{17}$$

    (cf. (7) and (4); the main difference of (17) from (7) and (4) is that (17) involves the true training examples and test objects whereas (7) and (4) involve arbitrary subsets of size l and k of the union of the training set and the test set with postulated labels);

– compute the p-value (14) for each $j = l+1, \dots, l+k$ and merge these p-values using (16).

- Output the prediction region (2).

Notice that the SBP becomes identical to the BP when (17) is replaced by (12) and (13) for $j = l+1, \dots, l+k$.

The following lemma shows that SBPs are usually weaker than the corresponding rankmax TCPs. (However, in Remark 3 and Section 6 we will see that the difference can be surprisingly small.)

Lemma 1. *Suppose all nonconformity scores (17) are different. The p-value (10) produced by a rankmax TCP does not exceed the p-value (16) produced by the corresponding SBP.*

Proof. Let t be the value of the rankmax test statistic, as defined at the end of Section 2. We are required to prove

$$1 - \frac{\binom{t-1}{k}}{\binom{l+k}{k}} \le k \frac{l+k-t+1}{l+1} . \tag{18}$$

Indeed, the left-hand side of (18) is identical to (10), and the ratio on the right-hand side of (18) is the smallest of the p-values (14) over j (cf. (16)). The statement that the ratio on the right-hand side of (18) is the smallest of the p-values (14) depends on (17) being all different (in fact, it is sufficient to assume that the maximum in the definition of the rankmax test statistic t is attained on only one test object). Notice, however, that the right-hand side of (18) is always an upper bound on the SBP p-value; this fact will be used in our discussions below.

We will prove a slightly stronger inequality that (18) replacing the denominator $l+1$ by $l+k$. In principle, t can take any value in $\{1, \dots, l+k\}$, but we can assume, without loss of generality, that $t \in \{k+1, \dots, l+k\}$: if $t \le k$, the left-hand side of (18) is 1 by definition and the right-hand side is at least 1 (even when $l+1$ is replaced by $l+k$). Rewriting (18) (with $l+k$ in place of $l+1$) as

$$1 - \frac{(t-1)(t-2)\cdots(t-k)}{k!\binom{l+k}{k}} \le k \frac{l+k-t+1}{l+k} , \tag{19}$$

we can assume that $t \in [k+1, l+k]$. Since the fraction on the left-hand side of (19) is a convex function of t (the second derivative is obviously nonnegative) and for $t := k+l$ (19) holds (it becomes $k/(l+k) \le k/(l+k)$), it suffices to prove that the derivative in t of the left-hand side of (19) at the point $l+k$ is equal to or exceeds the derivative of the right-hand side:

$$-\frac{(\Gamma(t)/\Gamma(t-k))'_{t=l+k}}{k!\binom{l+k}{k}} \ge k \frac{-1}{l+k} ,$$

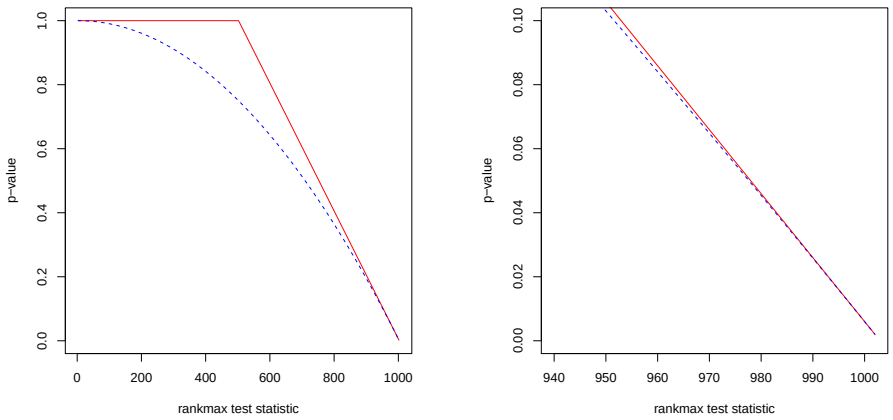

Figure 1: Left panel: SBP p-values (the solid red line) and rankmax TCP p-values (the dashed blue line) for $l = 1000$ and $k = 2$ as functions of t. Right panel: the lower right corner of the left panel.

where Γ is the gamma function, $\Gamma(n) = (n-1)!$ for $n \in \mathbb{N}$. By the definition of the digamma function ψ, the last inequality can be rewritten as

$$\frac{\Gamma(l+k)}{\Gamma(l)}\big(\psi(l+k) - \psi(l)\big) \le \frac{k}{l+k} k! \binom{l+k}{k},$$

which simplifies to

$$\psi(l+k) - \psi(l) \le \frac{k}{l}.$$

The well-known expression for ψ at the integer values of its argument (see, e.g., `http://dlmf.nist.gov/5.4.14`) allows us to rewrite the last inequality as

$$\frac{1}{l} + \frac{1}{l+1} + \cdots + \frac{1}{l+k-1} \le \frac{k}{l},$$

which is obviously true. □

Remark. The proof of Lemma 1 shows that the inequality (18) is strict whenever $k > 1$ (for $k = 1$ the two p-values coincide). Three factors contribute to its being strict: the SBP p-value is larger than the rankmax TCP p-value at $t = l + k$; as function of t, the SBP p-value has a steeper (negative) slope at $t = l + k$; besides, to the left of $t = l + k$ the SBP p-value goes in a straight line whereas the rankmax TCP p-value veers down. This is illustrated in Figure 1 for $l = 1000$ and $k = 2$ (typical values for our experiments reported in Section 6); the first two factors, however, are not noticeable.

It is plausible that a BP usually produces somewhat smaller p-values (and, therefore, somewhat smaller prediction regions) than the corresponding SBP: the only difference is that, when computing p-values, the SBP uses more test objects with arbitrarily assigned labels, and this may lead to a greater distortion of the nonconformity scores.

4 Validity

The strongest notion of validity for conformal and related predictors can be stated in the online mode. Suppose we are given a sequence of positive integer numbers $k_1, k_2, \ldots$ and the incoming sequence of examples is $z_1 = (x_1, y_1), z_2 = (x_2, y_2), \ldots$; set $l_n := \sum_{i=1}^{n-1} k_i$ (in particular, $l_1 := 0$). At trial $n = 1, 2, \ldots$ of the online prediction protocol, Predictor predicts the k_n labels $y_{l_n+1}, \ldots, y_{l_n+k_n}$ given the l_n examples $z_1, \ldots, z_{l_n}$ and k_n objects $x_{l_n+1}, \ldots, x_{l_n+k_n}$. The prediction is a subset Γ_n of $\mathbf{Y}^{k_n}$. It can be *multiple* ($|\Gamma_n| > 1$), *singleton* ($|\Gamma_n| = 1$), or *empty* ($|\Gamma_n| = 0$). Predictor *makes an error* if $(y_{l_n+1}, \ldots, y_{l_n+k_n}) \notin \Gamma_n$.

In this section we assume either that the sequence of examples $z_1, z_2, \ldots$ is infinite and the examples are produced independently from the same probability distribution on $\mathbf{Z}$, or that the sequence of examples is finite, $z_1, \ldots, z_N$, and produced from an exchangeable probability distribution on $\mathbf{Z}^N$.

The following simple result states the validity of TCPs in the online mode; its proof is standard (see, e.g., [8] or [9], Section 8.7) and is omitted.

Theorem 1. *In the online mode, a smoothed TCP makes errors with probability ϵ (the significance level) independently at different trials.*

A suitable version of validity in the absence of smoothing is *conservative validity*, i.e., being dominated by a sequence of independent Bernoulli trials with probability of success equal to the significance level; for details, see [9], p. 21. By Theorem 1, TCPs are conservatively valid:

Corollary 1. *In the online mode, each TCP is conservatively valid.*

Proof. Each TCP is conservatively valid since it can only make an error when the corresponding smoothed TCP (i.e., the smoothed TCP based on the same transductive nonconformity measure) makes an error. □

Lemma 1 suggests that SBPs can be regarded as conservatively valid for practical purposes, since an SBP can make an error only when the corresponding rankmax TCP makes an error, unless there are ties among nonconformity scores. (However, in general, it is not always true that an SBP can make an error only when the corresponding rankmax TCP makes an error. Consider, e.g., the case where $k = 2$ and the nonconformity scores of the two test examples are equal and exceed the nonconformity scores of all training examples; in this case, the SBP p-value will be smaller than the rankmax TCP p-value unless $l = 1$.)

Theorem 2. *In the online mode, each BP is conservatively valid.*

Proof sketch. The proof follows the scheme of the proof in Appendix A.1 of [8]. Given the bag $\wr z_1, \dots, z_{l_n+k_n} \int$ and under the assumption of exchangeability, the probability that the BP will make an error at trial n for a given test example (e.g., for the second example in the test set) is at most ϵ/k_n. Therefore, the probability that it will make an error for some of the k_n test examples is at most ϵ. We can increase the indicator function of making an error to obtain a Bernoulli random variable with probability of success equal to ϵ (this might involve extending the probability space). It remains to notice that whether an error is made at trial n is determined by the bag $\wr z_1, \dots, z_{l_n} \int$ and examples $z_{l_n+1}, \dots, z_{l_n+k_n}$ (cf. Lemma 2 in [8]). □

5 Universality

A *transductive confidence predictor* is a measurable strategy for Predictor in the online prediction protocol (as described in the previous section) depending on a parameter $\epsilon \in (0, 1)$ (the significance level) in such a way that for each training set and each test set the prediction at a larger significance level is a subset of the prediction at a smaller significance level. We say that the transductive confidence predictor is conservatively valid if the sequence of errors that it makes at any significance level ϵ is dominated by a sequence of independent Bernoulli trials with probability of success ϵ. We say that it is *invariant* if, when fed with examples $z_1, \dots, z_{l_n}$ and objects $x_{l_n+1}, \dots, x_{l_n+k_n}$ at any trial n, it issues the same prediction regardless of the ordering of $z_1, \dots, z_{l_n}$. And we say that a transductive confidence predictor Γ' is *at least as good as* another transductive confidence predictor Γ'' if at any significance level ϵ the prediction region issued by Γ' is completely covered by the prediction region issued by Γ''. The following result, whose proof is omitted in this version of the paper, can be proved similarly to Theorem 2.6 in [9].

Theorem 3. *Suppose* $\mathbf{Z}$ *is a Borel space. For any invariant conservatively valid transductive confidence predictor* Γ *there exists a transductive conformal predictor* Γ' *that is at least as good as* Γ.

Theorem 3 says that TCPs are universal in the sense of dominating all invariant conservatively valid transductive confidence predictors. In particular, for any BP there is a TCP that is at least as good as that BP. However, in the next section we will see that the rankmax TCP corresponding to the same nonconformity measure does not always satisfy this property.

6 Experiments

In our experiments we will use the standard USPS data set of hand-written digits. The training set (7291 examples) is merged with the test set (2007 examples) and the resulting data set of 9298 examples is randomly permuted, to make sure the assumption of exchangeability is satisfied. The prediction protocol is online. In a typical scenario the digits might arrive in batches of $k = 5$

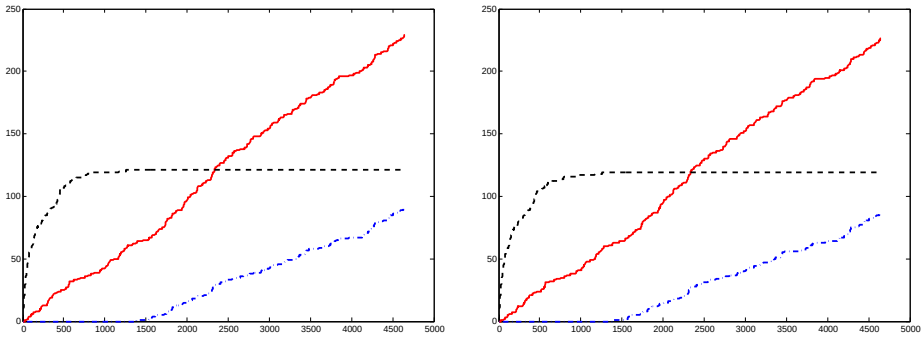

Figure 2: Left panel: the performance of the rankmax TCP based on Nearest Neighbour for tangent distance on the USPS data set (randomly permuted) for the size $k = 2$ of test sets and significance level 5%. The cumulative errors are shown with a solid red line, multiple predictions with a dashed black line, and empty predictions with a dash-dot blue line. Right panel: the analogous picture for the BP.

digits and represent American zip codes (in this case, however, the exchangeability assumption is only a crude approximation, since the digits within the same zip code are likely to be written by the same person). However, the TCP and SBP are too computationally inefficient to be applied in this case, and for comparing them with the BP we first consider online prediction of batches of $k = 2$ digits; intuitively, our task is to recognize a two-digit number.

We always use the Nearest Neighbour nonconformity measure (5), where d is tangent distance [5], and study empirically the corresponding rankmax TCP, SBP, and BP. As the significance level we always take 5%. The left panel of Figure 2 shows the performance of the rankmax TCP using three functions: the cumulative number of errors made over the trials $1, \dots, n$ as function of n, the cumulative number of multiple predictions made over the trials $1, \dots, n$ as function of n, and the cumulative number of empty predictions over the trials $1, \dots, n$ as function of n. The performance of the SBP and BP as measured by these functions is very similar; only the latter is shown in the right panel of Figure 2, but all three graphs are visually indistinguishable (cf. Figure 3). The BP even makes 2 fewer multiple predictions than the rankmax TCP, which confirms the claim made in Section 5 that the rankmax TCP corresponding to the same nonconformity measure as a given BP is not always at least as good as that BP. (It is not true in general that the BP always makes fewer multiple predictions than the corresponding rankmax TCP. It just happens to be true for tangent distance and seed 0 for the MATLAB pseudorandom number generator; e.g., the BP makes slightly more multiple predictions for Euclidean distance and seed 0.) The SBP makes one more multiple prediction than the rankmax TCP, which agrees with Lemma 1.

The cause of the similarity between the two plots in Figure 3 is illustrated by

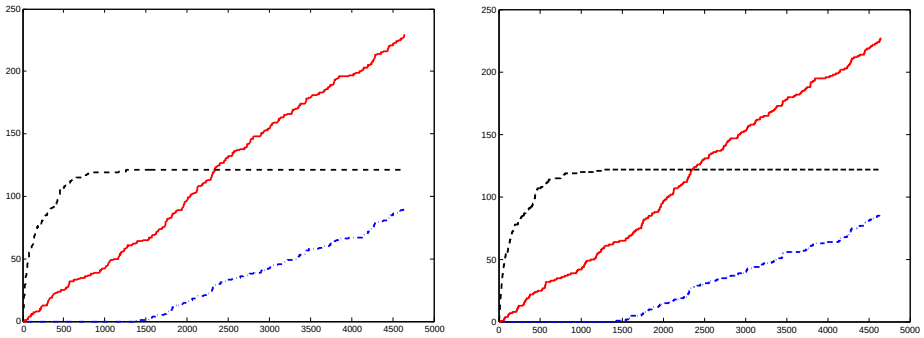

Figure 3: Left panel: reproduces the left panel of Figure 2 (the performance of the rankmax TCP). Right panel: the analogous picture for the SBP.

Figure 4 (essentially a version of Figure 1), which shows the p-values produced by the SBP plotted against the respective p-values produced by the corresponding rankmax TCP, assuming there are no ties among the nonconformity scores. When the p-values are small, they are remarkably close to each other. And even without making any assumptions, we can still see that the SBP p-values are never significantly worse than the respective rankmax TCP p-values, assuming the latter are not too large.

The main advantage of BPs is that they are much more computationally efficient than both TCPs and SBPs. Because of their computational efficiency, it is very easy to produce the analogue of the right panel of Figure 2 for $k = 5$ (as in American zip codes): see the left panel of Figure 5; but it is not clear at all how to make the computations for rankmax TCPs and SBPs feasible, even for moderately large k.

7 Conclusion

Based on our theoretical and empirical results, the preliminary recommendation is to use Bonferroni predictors in transductive problems: as compared to rankmax TCPs and SBPs, they enjoy the same theoretical validity guarantees, have comparable predictive performance empirically, but are much more computationally efficient.

The conclusion is preliminary since our empirical comparison in Section 6 only covers TCPs for a small size k of the test set, namely $k = 2$. The computational inefficiency of TCPs greatly complicates their empirical comparison with the BPs and SBPs for large values of k.

The comparison is much more straightforward in the case of transductive and Bonferroni extensions of inductive conformal predictors ([4]; [9], Section 4.1), and it can be shown that the two extensions produce similar p-values in practically important cases: see Appendix A for details.

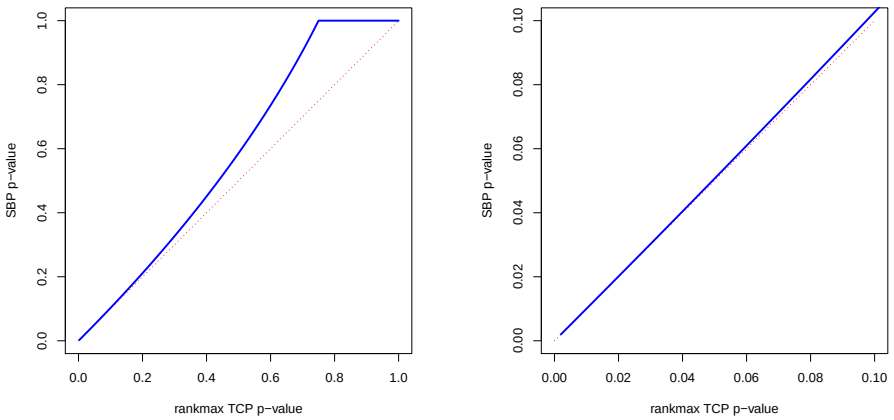

Figure 4: Left panel: the p-values produced by an SBP vs the p-values produced by the corresponding rankmax TCP (the solid blue line) for the size $l = 1000$ of the training set and $k = 2$ of the test set. Right panel: the lower left corner of the left panel.

Acknowledgments

I am grateful to Harris Papadopoulos for a discussion at COPA 2012 that rekindled my interest in transduction. Thanks to Wouter Koolen for illuminating discussions and for writing a MATLAB program for the Wilcoxon ranksum test (the standard programs in MATLAB and R produce unsatisfactory results in the default mode and are prohibitively slow in the exact mode). My thanks also go to the COPA 2013 reviewers, whose comments have helped me in improving the presentation and suggested new directions of research. In my experiments I used the C program for computing tangent distance written by Daniel Keysers and adapted to MATLAB by Aditi Krishn. This work was partially supported by the Cyprus Research Promotion Foundation (research contract TPE/ORIZO/0609(BIE)/24) and by EPSRC (grant EP/K033344/1).

References

[1] Erich L. Lehmann. *Nonparametrics: Statistical Methods Based on Ranks*. Springer, New York, revised first edition, 2006.

[2] National Institute of Standards and Technology. Digital library of mathematical functions. `http://dlmf.nist.gov/`, 6 May 2013.

[3] Ilia Nouretdinov, Sergi G. Costafreda, Alex Gammerman, Alexey Chervonenkis, Vladimir Vovk, Vladimir Vapnik, and Cynthia H. Y. Fu. Machine learning classification with confidence: Application of transductive

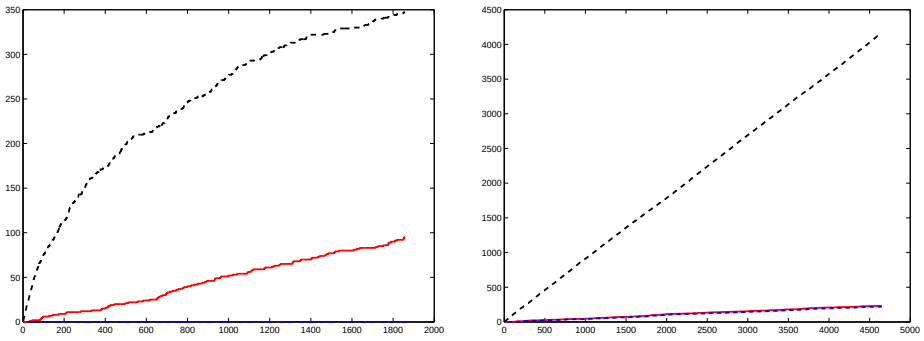

Figure 5: Left panel: the performance of the BP for the size $k = 5$ of test sets. Right panel: the performance of the ranksum TCP for $k = 2$ (very poor). The setting is as in Figure 2: the prediction algorithms are based on Nearest Neighbour and tangent distance; the cumulative errors are shown with a solid red line, multiple predictions with a dashed black line, and empty predictions with a dash-dot blue line; the significance level is 5%.

conformal predictors to MRI-based diagnostic and prognostic markers in depression. *Neuroimage*, 56:809–813, 2011.

[4] Harris Papadopoulos, Vladimir Vovk, and Alex Gammerman. Qualified predictions for large data sets in the case of pattern recognition. In *Proceedings of the First International Conference on Machine Learning and Applications*, pages 159–163, Las Vegas, NV, 2002. CSREA Press.

[5] Patrice Simard, Yann LeCun, and John Denker. Efficient pattern recognition using a new transformation distance. In S. Hanson, J. Cowan, and C. Giles, editors, *Advances in Neural Information Processing Systems*, volume 5, pages 50–58, San Mateo, CA, 1993. Morgan Kaufmann.

[6] Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

[7] Vladimir N. Vapnik and Alexey Y. Chervonenkis. Теория распознавания образов *(Theory of Pattern Recognition)*. Nauka, Moscow, 1974. German translation: *Theorie der Zeichenerkennung*, Akademie, Berlin, 1979.

[8] Vladimir Vovk. On-line Confidence Machines are well-calibrated. In *Proceedings of the Forty Third Annual Symposium on Foundations of Computer Science*, pages 187–196, Los Alamitos, CA, 2002. IEEE Computer Society.

[9] Vladimir Vovk, Alex Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer, New York, 2005.

[10] Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1:80–83, 1945.

A Transinductive conformal predictors

Even BPs are computationally inefficient when used for predicting a large number of test sets (of size k) from the same training set (of size l), since in general there is no way of reusing the computations carried out for the previous test sets when processing the current test set. This appendix combines the ideas of TCPs and inductive conformal predictors ([4]; [9], Section 4.1) to obtain a computationally efficient version of TCPs, which we will call transductive inductive, or transinductive, conformal predictors.

Split the training set $(z_1, \dots, z_l)$ into two parts: the *proper training set* $(z_1, \dots, z_m)$ of size $0 < m < l$ and the *calibration set* $(z_{m+1}, \dots, z_l)$ of size $l-m$; the test set $(x_{l+1}, \dots, x_{l+k})$ is as before. The *rankmax transinductive conformal predictor* (or *rankmax TICP*) corresponding to a nonconformity measure A is defined as follows:

- Compute the nonconformity scores

$$\alpha_i := A((z_1, \dots, z_m), z_i), \qquad i = m+1, \dots, l, \tag{20}$$

  for all calibration examples.

- For each possible set of labels $(\upsilon_1, \dots, \upsilon_k) \in \mathbf{Y}^k$:
  - set $y_j := \upsilon_{j-l}$ and $z_j := (x_j, y_j)$ for $j = l+1, \dots, l+k$;
  - compute the nonconformity scores

$$\alpha_j := A((z_1, \dots, z_m), z_j)\,, \tag{21}$$

    $j = l+1, \dots, l+k$, for all test examples;
  - compute the p-value

$$p(\upsilon_1, \dots, \upsilon_k) := \frac{|\{S \,|\max_{i \in S} \alpha_i \geq \max(\alpha_{l+1}, \dots, \alpha_{l+k})\}|}{\binom{l-m+k}{k}}\,, \tag{22}$$

    where S ranges over all $\binom{l-m+k}{k}$ subsets of $\{m+1, \dots, l+k\}$ of size k.

- Output the prediction region (2).

The *Bonferroni inductive predictor* (or *BIP*) corresponding to a nonconformity measure A is defined similarly:

- Compute the nonconformity scores (20) for all calibration examples.
- For each object x_j, $j \in \{l+1, \dots, l+k\}$, in the test set and each possible label $\upsilon \in \mathbf{Y}$:
  - set $y_j := \upsilon$ and $z_j := (x_j, y_j)$ for $j = l+1, \dots, l+k$;
  - compute the nonconformity score (21) for z_j;

– compute the p-value

$$p_{j-l}(\upsilon) := \frac{|\{i = m+1, \ldots, l \,|\, \alpha_i \geq \alpha_j\}| + 1}{l - m + 1} .$$

- Output the prediction region (15).

The rankmax TICP and BIP are especially computationally efficient for nonconformity measures of the form (3), since the prediction rule f can be precomputed.

Another representation of the BIP is (2), where the p-values $p(\upsilon_1, \ldots, \upsilon_k)$ are defined by (16). Lemma 1 simplifies in the inductive case:

Lemma 2. *If the nonconformity scores (21) for the test examples are all different, the p-value (22) produced by a rankmax TICP never exceed the p-value (16) produced by the corresponding BIP.*

Proof. It suffices to apply (18) with $l - m$ (the size of the calibration set) in place of l to the value of the rankmax statistic $t := \max(R_{l+1}, \ldots, R_{l+k})$, where R_j is now the rank of α_j in the multiset $\{\!|\alpha_{m+1}, \ldots, \alpha_{l+k}|\!\}$. □

Section 6 was devoted to an empirical study of the difference between rankmax TCPs and the corresponding BPs. In the inductive case, such a study is, to a large degree, redundant. In the proof of Lemma 2 we saw that

$$1 - \frac{\binom{t-1}{k}}{\binom{l-m+k}{k}} \leq k \frac{l - m + k - t + 1}{l - m + 1} ,$$

where the left-hand side is the p-value produced by the rankmax TICP and the right-hand side is an upper bound on the p-value produced by the BIP (now we are not making any assumptions on the nonconformity scores). When all nonconformity scores are different, the left panel of Figure 4 is also the plot of BIP p-values vs rankmax TICP p-values for a calibration set of size $l - m = 1000$ and a test set of size $k = 2$; Figure 1 and the discussion in Remark 3 are also applicable in the inductive case. When no assumptions are made, the pair of a rankmax p-value and the corresponding BIP p-value always lies at or below the solid blue line in Figure 4. We can see that the BIP's results are never much worse in the interesting range of small p-values. Figure 6 gives analogous pictures for the sizes $k = 10$ and $k = 50$ of the test set, and we still observe the same phenomenon: the BIP's results are never much worse if we are interested in small p-values; but the figure also illustrates the increasing difficulty of obtaining small p-values as the size k of the test set increases: it is clear that the smallest achievable p-value is $k/(l-m+1)$ for the BIP and $k/(l-m+k)$ for the rankmax TICP.

B Ranksum TCP

This appendix briefly discusses ranksum TCPs, based on the ranksum aggregator (9). The results are shown in the right panel of Figure 5, in the same format

as before. They are very poor, and the following heuristic argument explains why.

Suppose the training set is very large, of size $l \gg 1$, and the test set contains two examples. The TCP assigns all possible labels to the two test objects, and we can expect the prediction to be a singleton whenever assigning a wrong label to either test object leads to a p-value not exceeding the significance level. Now suppose one of the test objects is assigned the correct label and the other a wrong label. Let us assume, optimistically, that the normalized rank $R/(l+2)$ of the latter test object (with a wrong label) is 1; the normalized rank x of the former test object (with the right label) will be, at best, uniformly distributed on $[0,1]$. In the limit of a very large training set and assuming the examples are exchangeable, the p-value corresponding to the normalized rank x of the former test object is at least

$$\mathbb{P}(\xi_1 + \xi_2 \geq 1 + x) = (1-x)^2/2\,,$$

where ξ_1 and ξ_2 are distributed uniformly on $[0,1]$, and so the expected p-value is at least $\int_0^1 \frac{(1-x)^2}{2} dx = 1/6 \approx 17\%$.

This shows that we can expect the bulk of our predictions to be singleton when using the ranksum TCP only when the significance level considerably exceeds 17%. For example, the ranksum TCP for the USPS data set at the 5% level will typically produce as its prediction the cross in $\{0,\dots,9\}^2$ centred on the pair of true labels, and this has been observed in our experiments.

It is interesting that using an unsuitable rank aggregator leads to the predictor sometimes issuing empty predictions before multiple predictions. Typically decent predictors start issuing empty predictions only after they stop issuing multiple predictions. For example, the rankmax TCP in the left panel of Figure 2 issues only singleton predictors in trials 1263–1385; before that it never issues empty predictions and after that it never issues multiple predictions.

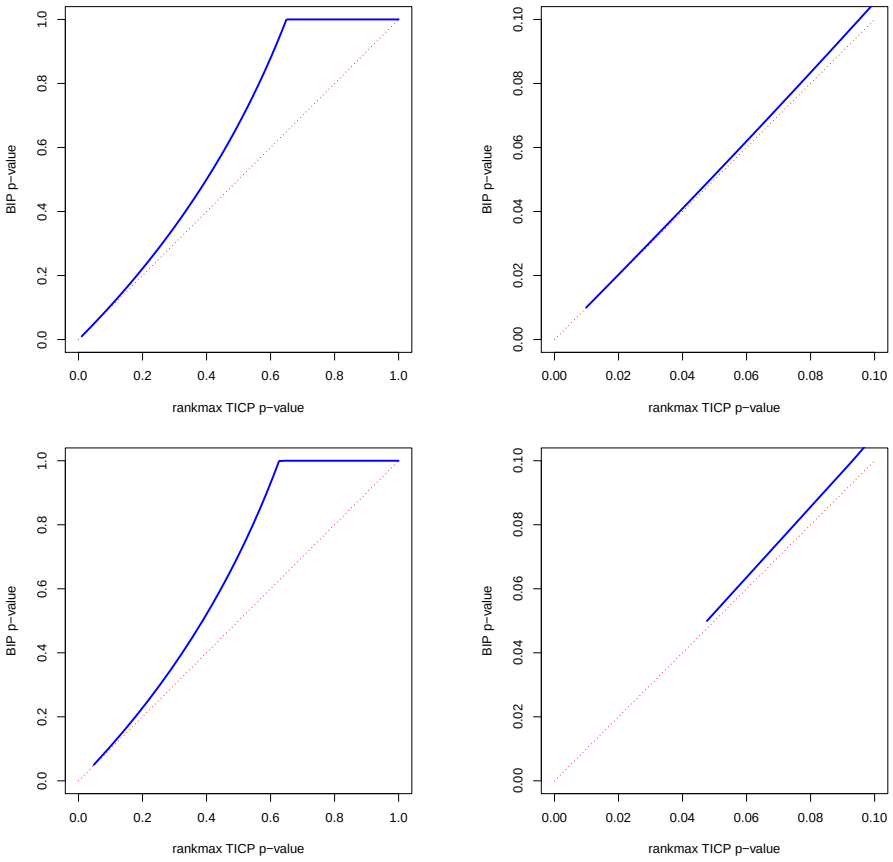

Figure 6: Left panels: the p-values produced by a BIP vs the p-values produced by the corresponding rankmax TICP (the solid blue lines). Right panels: the lower left corners of the corresponding left panels. The size of the calibration set is $l - m = 1000$ and the size of the test set is $k = 10$ for the top panels and $k = 50$ for the bottom panels.