Hedging Predictions in Machine Learning

Alexander Gammerman and Vladimir Vovk

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

{alex,vovk}@cs.rhul.ac.uk

January 19, 2007

Abstract

Recent advances in machine learning make it possible to design efficient prediction algorithms for data sets with huge numbers of parameters. This article describes a new technique for 'hedging' the predictions output by many such algorithms, including support vector machines, kernel ridge regression, kernel nearest neighbours, and by many other state-of-the-art methods. The hedged predictions for the labels of new objects include quantitative measures of their own accuracy and reliability. These measures are provably valid under the assumption of randomness, traditional in machine learning: the objects and their labels are assumed to be generated independently from the same probability distribution. In particular, it becomes possible to control (up to statistical fluctuations) the number of erroneous predictions by selecting a suitable confidence level. Validity being achieved automatically, the remaining goal of hedged prediction is efficiency: taking full account of the new objects' features and other available information to produce as accurate predictions as possible. This can be done successfully using the powerful machinery of modern machine learning.

1 Introduction

The two main varieties of the problem of prediction, classification and regression, are standard subjects in statistics and machine learning. The classical classification and regression techniques can deal successfully with conventional small-scale, low-dimensional data sets; however, attempts to apply these techniques to modern high-dimensional and high-throughput data sets encounter serious conceptual and computational difficulties. Several new techniques, first of all support vector machines [42, 43] and other kernel methods, have been

developed in machine learning recently with the explicit goal of dealing with high-dimensional data sets with large numbers of objects.

A typical drawback of the new techniques is the lack of useful measures of confidence in their predictions. For example, some of the tightest upper bounds of the popular theory of PAC (probably approximately correct) learning on the probability of error exceed 1 even for relatively clean data sets ([51], p. 249). This article describes an efficient way to 'hedge' the predictions produced by the new and traditional machine-learning methods, i.e., to complement them with measures of their accuracy and reliability. Appropriately chosen, not only are these measures valid and informative, but they also take full account of the special features of the object to be predicted.

We call our algorithms for producing hedged predictions conformal predictors; they are formally introduced in Section 3. Their most important property is the automatic validity under the randomness assumption (to be discussed shortly). Informally, validity means that conformal predictors never overrate the accuracy and reliability of their predictions. This property, stated in Sections 3 and 5, is formalized in terms of finite data sequences, without any recourse to asymptotics.

The claim of validity of conformal predictors depends on an assumption that is shared by many other algorithms in machine learning, which we call the assumption of randomness: the objects and their labels are assumed to be generated independently from the same probability distribution. Admittedly, this is a strong assumption, and areas of machine learning are emerging that rely on other assumptions (such as the Markovian assumption of reinforcement learning; see, e.g., [36]) or dispense with any stochastic assumptions altogether (competitive on-line learning; see, e.g., [6, 47]). It is, however, much weaker than assuming a parametric statistical model, sometimes complemented with a prior distribution on the parameter space, which is customary in the statistical theory of prediction. And taking into account the strength of the guarantees that can be proved under this assumption, it does not appear overly restrictive.

So we know that conformal predictors tell the truth. Clearly, this is not enough: truth can be uninformative and so useless. We will refer to various measures of informativeness of conformal predictors as their 'efficiency'. As conformal predictors are provably valid, efficiency is the only thing we need to worry about when designing conformal predictors for solving specific problems. Virtually any classification or regression algorithm can be transformed into a conformal predictor, and so most of the arsenal of methods of modern machine learning can be brought to bear on the design of efficient conformal predictors.

We start the main part of the article, in Section 2, with the description of an idealized predictor based on Kolmogorov's algorithmic theory of randomness. This 'universal predictor' produces the best possible hedged predictions but, unfortunately, is noncomputable. We can, however, set ourselves the task of approximating the universal predictor as well as possible.

In Section 3 we formally introduce the notion of conformal predictors and state a simple result about their validity. In that section we also briefly describe results of computer experiments demonstrating the methodology of conformal

prediction.

In Section 4 we consider an example demonstrating how conformal predictors react to the violation of our model of the stochastic mechanism generating the data (within the framework of the randomness assumption). If the model coincides with the actual stochastic mechanism, we can construct an optimal conformal predictor, which turns out to be almost as good as the Bayes-optimal confidence predictor (the formal definitions will be given later). When the stochastic mechanism significantly deviates from the model, conformal predictions remain valid but their efficiency inevitably suffers. The Bayes-optimal predictor starts producing very misleading results which superficially look as good as when the model is correct.

In Section 5 we describe the 'on-line' setting of the problem of prediction, and in Section 6 contrast it with the more standard 'batch' setting. The notion of validity introduced in Section 3 is applicable to both settings, but in the on-line setting it can be strengthened: we can now prove that the percentage of the erroneous predictions will be close, with high probability, to a chosen confidence level. For the batch setting, the stronger property of validity for conformal predictors remains an empirical fact. In Section 6 we also discuss limitations of the on-line setting and introduce new settings intermediate between on-line and batch. To a large degree, conformal predictors still enjoy the stronger property of validity for the intermediate settings.

Section 7 is devoted to the discussion of the difference between two kinds of inference from empirical data, induction and transduction (emphasized by Vladimir Vapnik [42, 43]). Conformal predictors belong to transduction, but combining them with elements of induction can lead to a significant improvement in their computational efficiency (Section 8).

We show how some popular methods of machine learning can be used as underlying algorithms for hedged prediction. We do not give the full description of these methods and refer the reader to the existing readily accessible descriptions. This article is, however, self-contained in the sense that we explain all features of the underlying algorithms that are used in hedging their predictions. We hope that the information we provide will enable the reader to apply our hedging techniques to their favourite machine-learning methods.

2 Ideal hedged predictions

The most basic problem of machine learning is perhaps the following. We are given a *training set* of *examples*

$$(x_1, y_1), \ldots, (x_l, y_l), \tag{1}$$

each example (x_i, y_i), $i = 1, \ldots, l$, consisting of an *object* x_i (typically, a vector of attributes) and its *label* y_i; the problem is to predict the label y_{l+1} of a new object x_{l+1}. Two important special cases are where the labels are known *a priori* to belong to a relatively small finite set (the problem of classification) and where the labels are allowed to be any real numbers (the problem of regression).

The usual goal of classification is to produce a prediction $\hat{y}_{l+1}$ that is likely to coincide with the true label y_{l+1}, and the usual goal of regression is to produce a prediction $\hat{y}_{l+1}$ that is likely to be close to the true label y_{l+1}. In the case of classification, our goal will be to complement the prediction $\hat{y}_{l+1}$ with some measure of its reliability. In the case of regression, we would like to have some measure of accuracy and reliability of our prediction. There is a clear trade-off between accuracy and reliability: we can improve the former by relaxing the latter and vice versa. We are looking for algorithms that achieve the best possible trade-off and for a measure that would quantify the achieved trade-off.

Let us start from the case of classification. The idea is to try every possible label Y as a candidate for x_{l+1}'s label and see how well the resulting sequence

$$(x_1, y_1), \dots, (x_l, y_l), (x_{l+1}, Y) \tag{2}$$

conforms to the randomness assumption (if it does conform to this assumption, we will say that it is 'random'; this will be formalized later in this section). The ideal case is where all Ys but one lead to sequences (2) that are not random; we can then use the remaining Y as a confident prediction for y_{l+1}.

In the case of regression, we can output the set of all Ys that lead to a random sequence (2) as our 'prediction set'. An obvious obstacle is that the set of all possible Ys is infinite and so we cannot go through all the Ys explicitly, but we will see in the next section that there are ways to overcome this difficulty.

We can see that the problem of hedged prediction is intimately connected with the problem of testing randomness. Different versions of the universal notion of randomness were defined by Kolmogorov, Martin-Löf and Levin (see, e.g., [24]) based on the existence of universal Turing machines. Adapted to our current setting, Martin-Löf's definition is as follows. Let $\mathbf{Z}$ be the set of all possible examples (assumed to be a measurable space); as each example consists of an object and a label, $\mathbf{Z} = \mathbf{X} \times \mathbf{Y}$, where $\mathbf{X}$ is the set of all possible objects and $\mathbf{Y}$, $|\mathbf{Y}| > 1$, is the set of all possible labels. We will use $\mathbf{Z}^*$ as the notation for all finite sequences of examples. A function $t : \mathbf{Z}^* \to [0, 1]$ is a *randomness test* if

1. for all $\epsilon \in (0, 1)$, all $n \in \{1, 2, \dots\}$ and all probability distributions P on $\mathbf{Z}$,
$$P^n \{z \in \mathbf{Z}^n : t(z) \le \epsilon\} \le \epsilon; \tag{3}$$
2. t is upper semicomputable.

The first condition means that the randomness test is required to be valid: if, for example, we observe $t(z) \le 1\%$ for our data set z, then either the data set was not generated independently from the same probability distribution P or a rare (of probability at most 1%, under any P) event has occurred. The second condition means that we should be able to compute the test, in a weak sense (we cannot require computability in the usual sense, since the universal test can only be upper semicomputable: it can work forever to discover *all* patterns in the data sequence that make it non-random). Martin-Löf (developing Kolmogorov's

earlier ideas) proved that there exists a smallest, to within a constant factor, randomness test.

Let us fix a smallest randomness test, call it the *universal test*, and call the value it takes on a data sequence the *randomness level* of this sequence. A random sequence is one whose randomness level is not small; this is rather informal, but it is clear that for finite data sequences we cannot have a clear-cut division of all sequences into random and non-random (like the one defined by Martin-Löf [25] for infinite sequences). If t is a randomness test, not necessarily universal, the value that it takes on a data sequence will be called the *randomness level detected by* t.

Remark The word 'random' is used in (at least) two different senses in the existing literature. In this article we need both but, luckily, the difference does not matter within our current framework. First, randomness can refer to the assumption that the examples are generated independently from the same distribution; this is the origin of our 'assumption of randomness'. Second, a data sequence is said to be random with respect to a statistical model if the universal test (a generalization of the notion of universal test as defined above) does not detect any lack of conformity between the two. Since the only statistical model we are interested in in this article is the one embodying the assumption of randomness, we have a perfect agreement between the two senses.

Prediction with confidence and credibility

Once we have a randomness test t, universal or not, we can use it for hedged prediction. There are two natural ways to package the results of such predictions: in this subsection we will describe the way that can only be used in classification problems. If the randomness test is not computable, we can imagine an oracle answering questions about its values.

Given the training set (1) and the test object x_{l+1}, we can act as follows:

- consider all possible values $Y \in \mathbf{Y}$ for the label y_{l+1};
- find the randomness level detected by t for every possible completion (2);
- predict the label Y corresponding to a completion with the largest randomness level detected by t;
- output as the *confidence* in this prediction one minus the second largest randomness level detected by t;
- output as the *credibility* of this prediction the randomness level detected by t of the output prediction Y (i.e., the largest randomness level detected by t over all possible labels).

To understand the intuition behind confidence, let us tentatively choose a conventional 'significance level', say 1%. (In the terminology of this article, this corresponds to a 'confidence level' of 99%, i.e., 100% minus 1%.) If the confidence in our prediction is 99% or more and the prediction is wrong, the actual

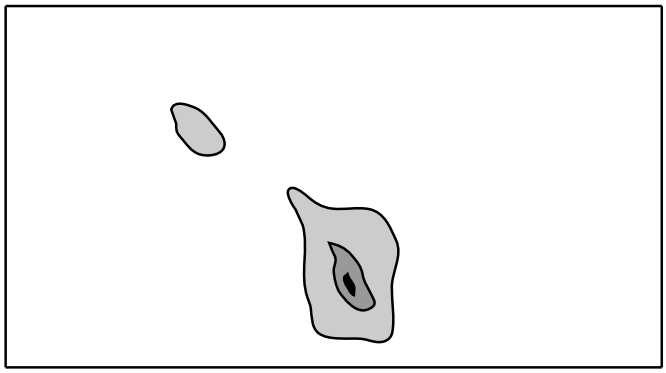

Figure 1: An example of a nested family of prediction sets (casual prediction in black, confident prediction in dark grey, and highly confident prediction in light grey).

data sequence belongs to an *a priori* chosen set of probability at most 1% (the set of all data sequences with randomness level detected by t not exceeding 1%).

Intuitively, low credibility means that either the training set is non-random or the test object is not representative of the training set (say, in the training set we have images of digits and the test object is that of a letter).

Confidence predictors

In regression problems, confidence, as defined in the previous subsection, is not a useful quantity: it will typically be equal to 0. A better approach is to choose a range of confidence levels $1-\epsilon$, and for each of them specify a *prediction set* $\Gamma^\epsilon \subseteq \mathbf{Y}$, the set of labels deemed possible at the confidence level $1-\epsilon$. We will always consider nested prediction sets: $\Gamma^{\epsilon_1} \subseteq \Gamma^{\epsilon_2}$ when $\epsilon_1 \geq \epsilon_2$. A *confidence predictor* is a function that maps each training set, each new object, and each confidence level $1-\epsilon$ (formally, we allow ϵ to take any value in $(0,1)$) to the corresponding prediction set Γ^ϵ. For the confidence predictor to be *valid* the probability that the true label will fall outside the prediction set Γ^ϵ should not exceed ϵ, for each ϵ.

We might, for example, choose the confidence levels 99%, 95% and 80%, and refer to the 99% prediction set $\Gamma^{1\%}$ as the highly confident prediction, to the 95% prediction set $\Gamma^{5\%}$ as the confident prediction, and to the 80% prediction set $\Gamma^{20\%}$ as the casual prediction. Figure 1 shows how such a family of prediction sets might look in the case of a rectangular label space $\mathbf{Y}$. The casual prediction pinpoints the target quite well, but we know that this kind of prediction can be wrong with probability 20%. The confident prediction is much bigger. If we want to be highly confident (make a mistake only with probability 1%), we must accept an even lower accuracy; there is even a completely different location that we cannot rule out at this level of confidence.

Given a randomness test, again universal or not, we can define the corresponding confidence predictor as follows: for any confidence level $1 - \epsilon$, the corresponding prediction set consists of the Ys such that the randomness level of the completion (2) detected by the test is greater than ϵ. The condition (3) of validity for statistical tests implies that a confidence predictor defined in this way is always valid.

The confidence predictor based on the universal test (the *universal confidence predictor*) is an interesting object for mathematical investigation (see, e.g., [50], Section 4), but it is not computable and so cannot be used in practice. Our goal in the following sections will be to find computable approximations to it.

3 Conformal prediction

In the previous section we explained how randomness tests can be used for prediction. The connection between testing and prediction is, of course, well understood and have been discussed at length by philosophers [32] and statisticians (see, e.g., the textbook [9], Section 7.5). In this section we will see how some popular prediction algorithms can be transformed into randomness tests and, therefore, be used for producing hedged predictions.

Let us start with the most successful recent development in machine learning, support vector machines ([42, 43], with a key idea going back to the generalized portrait method [44]). Suppose the label space is $\mathbf{Y} = \{-1, 1\}$ (we are dealing with the binary classification problem). With each set of examples

$$(x_1, y_1), \dots, (x_n, y_n) \tag{4}$$

one associates an optimization problem whose solution produces nonnegative numbers $\alpha_1, \dots, \alpha_n$ ('Lagrange multipliers'). These numbers determine the prediction rule used by the support vector machine (see [43], Chapter 10, for details), but they also are interesting objects in their own right. Each α_i, $i = 1, \dots, n$, tells us how strange an element of the set (4) the corresponding example (x_i, y_i) is. If $\alpha_i = 0$, (x_i, y_i) fits set (4) very well (in fact so well that such examples are uninformative, and the support vector machine ignores them when making predictions). The elements with $\alpha_i > 0$ are called support vectors, and the large value of α_i indicates that the corresponding (x_i, y_i) is an outlier.

Applying this procedure to the completion (2) in the role of set (4) (so that $n = l + 1$), we can find the corresponding $\alpha_1, \dots, \alpha_{l+1}$. If Y is different from the actual label y_{l+1}, we expect (x_{l+1}, Y) to be an outlier in the set (2) and so α_{l+1} be large as compared with $\alpha_1, \dots, \alpha_l$. A natural way to compare α_{l+1} to the other αs is to look at the ratio

$$p_Y := \frac{|\{i = 1, \dots, l+1 : \alpha_i \geq \alpha_{l+1}\}|}{l+1}, \tag{5}$$

which we call the *p-value* associated with the possible label Y for x_{l+1}. In words, the p-value is the proportion of the αs which are at least as large as the last α.

Table 1: Selected test examples from the USPS data set: the p-values of digits (0–9), true and predicted labels, and confidence and credibility values.

0	1	2	3	4	5	6	7	8	9	true label	pre-diction	confi-dence	credi-bility
0.01%	0.11%	0.01%	0.01%	0.07%	0.01%	100%	0.01%	0.01%	0.01%	6	6	99.89%	100%
0.32%	0.38%	1.07%	0.67%	1.43%	0.67%	0.38%	0.33%	0.73%	0.78%	6	4	98.93%	1.43%
0.01%	0.27%	0.03%	0.04%	0.18%	0.01%	0.04%	0.01%	0.12%	100%	9	9	99.73%	100%

The methodology of support vector machines (as described in [42, 43]) is directly applicable only to the binary classification problems, but the general case can be reduced to the binary case by the standard 'one-against-one' or 'one-against-the-rest' procedures. This allows us to define the strangeness values $\alpha_1, \dots, \alpha_{l+1}$ for general classification problems (see [51], p. 59, for details), which in turn determine the p-values (5).

The function that assigns to each sequence (2) the corresponding p-value, defined by expression (5), is a randomness test (this will follow from Theorem 1 stated in Section 5 below). Therefore, the p-values, which are our approximations to the corresponding randomness levels, can be used for hedged prediction as described in the previous section. For example, in the case of binary classification, if the p-value p_{-1} is small while p_1 is not small, we can predict 1 with confidence $1 - p_{-1}$ and credibility p_1. Typical credibility will be 1: for most data sets the percentage of support vectors is small ([43], Chapter 12), and so we can expect $\alpha_{l+1} = 0$ when $Y = y_{l+1}$.

Remark When the order of examples is irrelevant, we refer to the data set (4) as a set, although as a mathematical object it is a multiset rather than a set since it can contain several copies of the same example. We will continue to use this informal terminology (to be completely accurate, we would have to say 'data multiset' instead of 'data set'!)

Table 1 illustrates the results of hedged prediction for a popular data set of hand-written digits called the USPS data set [23]. The data set contains 9298 digits represented as a 16×16 matrix of pixels; it is divided into a training set of size 7291 and a test set of size 2007. For several test examples the table shows the p-values for each possible label, the actual label, the predicted label, confidence, and credibility, computed using the support vector method with the polynomial kernel of degree 5. To interpret the numbers in this table, remember that high (i.e., close to 100%) confidence means that all labels except the predicted one are unlikely. If, say, the first example were predicted wrongly, this would mean that a rare event (of probability less than 1%) had occurred; therefore, we expect the prediction to be correct (which it is). In the case of the second example, confidence is also quite high (more than 95%), but we can see that the credibility is low (less than 5%). From the confidence we can conclude that the labels other than 4 are excluded at level 5%, but the label 4 itself is also excluded at the level 5%. This shows that the prediction algorithm was unable to extract from the training set enough information to allow us to confidently classify this example: the strangeness of the labels different from 4 may be due

to the fact that the object itself is strange; perhaps the test example is very different from all examples in the training set. Unsurprisingly, the prediction for the second example is wrong.

In general, high confidence shows that all alternatives to the predicted label are unlikely. Low credibility means that the whole situation is suspect; as we have already mentioned, we will obtain a very low credibility if the new example is a letter (whereas all training examples are digits). Credibility will also be low if the new example is a digit written in an unusual way. Notice that typically credibility will not be low provided the data set was generated independently from the same distribution: the probability that credibility will not exceed some threshold ϵ (such as 1%) is at most ϵ. In summary, we can trust a prediction if (1) the confidence is close to 100% and (2) the credibility is not low (say, is not less than 5%).

Many other prediction algorithms can be used as underlying algorithms for hedged prediction. For example, we can use the nearest neighbours technique to associate

$$\alpha_i := \frac{\sum_{j=1}^{k} d_{ij}^{+}}{\sum_{j=1}^{k} d_{ij}^{-}}, \quad i = 1, \dots, n, \tag{6}$$

with the elements (x_i, y_i) of the set (4), where d_{ij}^{+} is the jth shortest distance from x_i to other objects labelled in the same way as x_i, and d_{ij}^{-} is the jth shortest distance from x_i to the objects labelled differently from x_i; the parameter $k \in \{1, 2, \dots\}$ in Equation (6) is the number of nearest neighbours taken into account. The distances can be computed in a feature space (that is, the distance between $x \in \mathbf{X}$ and $x' \in \mathbf{X}$ can be understood as $\|F(x) - F(x')\|$, F mapping the object space $\mathbf{X}$ into a feature, typically Hilbert, space), and so definition (6) can also be used with the kernel nearest neighbours.

The intuition behind Equation (6) is as follows: a typical object x_i labelled by, say, y will tend to be surrounded by other objects labelled by y; and if this is the case, the corresponding α_i will be small. In the untypical case that there are objects whose labels are different from y nearer than objects labelled y, α_i will become larger. Therefore, the αs reflect the strangeness of examples.

The p-values computed from Equation (6) can again be used for hedged prediction. It is a general empirical fact that the accuracy and reliability of the hedged predictions are in line with the error rate of the underlying algorithm. For example, in the case of the USPS data set, the 1-nearest neighbour algorithm (i.e., the one with $k = 1$) achieves the error rate of 2.2%, and the hedged predictions based on Equation (6) are highly confident (achieve confidence of at least 99%) for more than 95% of the test examples.

General definition

The general notion of conformal predictor can be defined as follows. A *nonconformity measure* is a function that assigns to every data sequence (4) a sequence of numbers $\alpha_1, \dots, \alpha_n$, called *nonconformity scores*, in such a way that interchanging any two examples (x_i, y_i) and (x_j, y_j) leads to the interchange of the

corresponding nonconformity scores α_i and α_j (with all other nonconformity scores unaffected). The corresponding *conformal predictor* maps each data set (1), $l = 0, 1, \ldots$, each new object x_{l+1}, and each confidence level $1 - \epsilon \in (0, 1)$ to the prediction set

$$\Gamma^\epsilon (x_1, y_1, \ldots, x_l, y_l, x_{l+1}) := \{Y \in \mathbf{Y} : p_Y > \epsilon\}, \tag{7}$$

where p_Y are defined by Equation (5) with $\alpha_1, \ldots, \alpha_{l+1}$ being the nonconformity scores corresponding to the data sequence (2).

We have already remarked that associating with each completion (2) the p-value (5) gives a randomness test; this is true in general. This implies that for each l the probability of the event

$$y_{l+1} \in \Gamma^\epsilon (x_1, y_1, \ldots, x_l, y_l, x_{l+1})$$

is at least $1 - \epsilon$.

This definition works for both classification and regression, but in the case of classification we can summarize the prediction sets (7) by two numbers: the confidence

$$\sup \{1 - \epsilon : |\Gamma^\epsilon| \le 1\} \tag{8}$$

and the credibility

$$\inf \{\epsilon : |\Gamma^\epsilon| = 0\}. \tag{9}$$

Computationally efficient regression

As we have already mentioned, the algorithms described so far cannot be applied directly in the case of regression, even if the randomness test is efficiently computable: now we cannot consider all possible values Y for y_{l+1} since there are infinitely many of them. However, there might still be computationally efficient ways to find the prediction sets Γ^ϵ. The idea is that if α_i are defined as the residuals

$$\alpha_i := |y_i - f_Y(x_i)| \tag{10}$$

where $f_Y : \mathbf{X} \to \mathbb{R}$ is a regression function fitted to the completed data set (2), then α_i may have a simple expression in terms of Y, leading to an efficient way of computing the prediction sets (via Equations (5) and (7)). This idea was implemented in [28] in the case where f_Y is found from the ridge regression, or kernel ridge regression, procedure, with the resulting algorithm of hedged prediction called the *ridge regression confidence machine*. For a much fuller description of the ridge regression confidence machine (and its modifications in the case where the simple residuals (10) are replaced by the fancier 'deleted' or 'studentized' residuals) see [51], Section 2.3.

4 Bayesian approach to conformal prediction

Bayesian methods have become very popular in both machine learning and statistics thanks to their power and versatility, and in this section we will see

how Bayesian ideas can be used for designing efficient conformal predictors. We will only describe results of computer experiments (following [26]) with artificial data sets, since for real-world data sets there is no way to make sure that the Bayesian assumption is satisfied.

Suppose $\mathbf{X} = \mathbb{R}^p$ (each object is a vector of p real-valued attributes) and our model of the data-generating mechanism is

$$y_i = w \cdot x_i + \xi_i, \quad i = 1, 2, \dots, \tag{11}$$

where ξ_i are independent standard Gaussian random variables and the weight vector $w \in \mathbb{R}^p$ is distributed as $N(0, (1/a)I_p)$ (we use the notation I_p for the unit $p \times p$ matrix and $N(0, A)$ for the p-dimensional Gaussian distribution with mean 0 and covariance matrix A); a is a positive constant. The actual data-generating mechanism used in our experiments will correspond to this model with a set to 1.

Under the model (11) the best (in the mean-square sense) fit to a data set (4) is provided by the ridge regression procedure with parameter a (for details, see, e.g., [51], Section 10.3). Using the residuals (10) with f_Y found by ridge regression with parameter a leads to an efficient conformal predictor which will be referred to as the ridge regression confidence machine with parameter a. Each prediction set output by the ridge regression confidence machine will be replaced by its convex hull, the corresponding *prediction interval*.

To test the validity and efficiency of the ridge regression confidence machine the following procedure was used. Ten times a vector $w \in \mathbb{R}^5$ was independently generated from the distribution $N(0, I_5)$. For each of the 10 values of w, 100 training objects and 100 test objects were independently generated from the uniform distribution on $[-10, 10]^5$ and for each object x its label y was generated as $w \cdot x + \xi$, with all the ξ standard Gaussian and independent. For each of the 1000 test objects and each confidence level $1 - \epsilon$ the prediction set Γ^ϵ for its label was found from the corresponding training set using the ridge regression confidence machine with parameter $a = 1$. The solid line in Figure 2 shows the confidence level against the percentage of test examples whose labels were not covered by the corresponding prediction intervals at that confidence level. Since conformal predictors are always valid, the percentage outside the prediction interval should never exceed 100 minus the confidence level, up to statistical fluctuations, and this is confirmed by the picture.

A natural measure of efficiency of confidence predictors is the mean width of their prediction intervals, at different confidence levels: the algorithm is the more efficient the narrower prediction intervals it produces. The solid line in Figure 3 shows the confidence level against the mean (over all test examples) width of the prediction intervals at that confidence level.

Since we know the data-generating mechanism, the approach via conformal prediction appears somewhat roundabout: for each test object we could instead find the conditional probability distribution of its label, which is Gaussian, and output as the prediction set Γ^ϵ the shortest (i.e., centred at the mean of the conditional distribution) interval of conditional probability $1 - \epsilon$. Figures 4

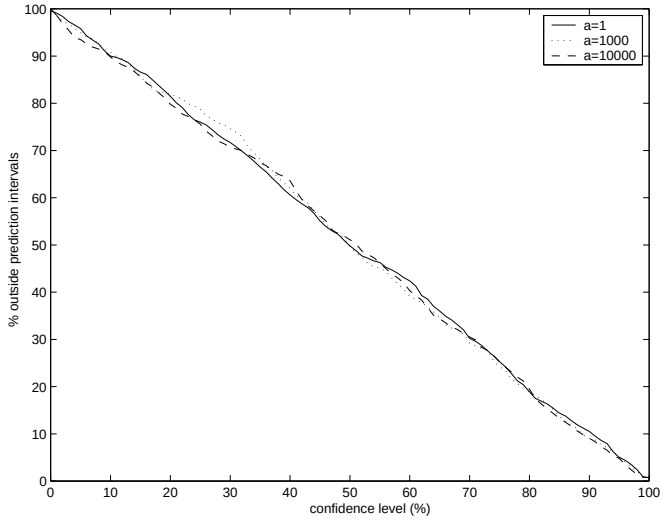

Figure 2: Validity for the ridge regression confidence machine.

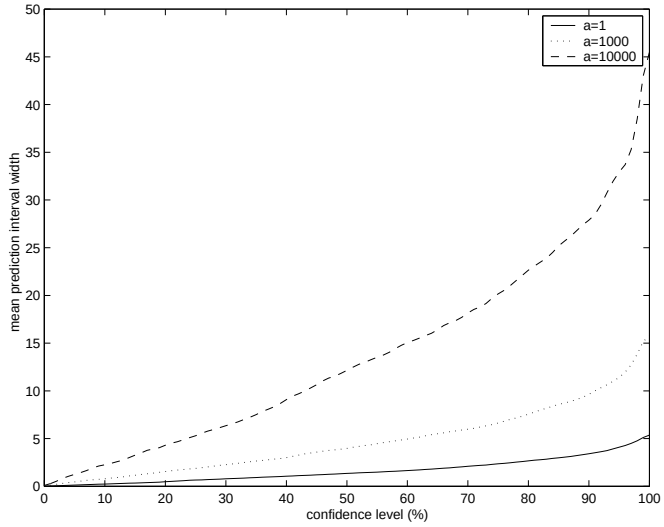

Figure 3: Efficiency for the ridge regression confidence machine.

and 5 are the analogues of Figures 2 and 3 for this *Bayes-optimal confidence predictor*. The solid line in Figure 4 demonstrates the validity of the Bayes-optimal confidence predictor.

What is interesting is that the solid lines in Figures 5 and 3 look exactly the same, taking account of the different scales of the vertical axes. The ridge regression confidence machine appears as good as the Bayes-optimal predictor. (This is a general phenomenon; it is also illustrated, in the case of classification, by the construction in Section 3.3 of [51] of a conformal predictor that is asymptotically as good as the Bayes-optimal confidence predictor.)

The similarity between the two algorithms disappears when they are given wrong values for a. For example, let us see what happens if we tell the algorithms that the expected value of $\|w\|$ is just 1% of what it really is (this corresponds to taking $a = 10000$). The ridge regression confidence machine stays valid (see the dashed line in Figure 2), but its efficiency deteriorates (the dashed line in Figure 3). The efficiency of the Bayes-optimal confidence predictor (the dashed line in Figure 5) is hardly affected, but its predictions become invalid (the dashed line in Figure 4 deviates significantly from the diagonal, especially for the most important large confidence levels: e.g., only about 15% of labels fall within the 90% prediction intervals). The worst that can happen to the ridge regression confidence machine is that its predictions will become useless (but at least harmless), whereas the Bayes-optimal predictions can become misleading.

Figures 2–5 also show the graphs for the intermediate value $a = 1000$. Similar results but for different data sets are also given in [51], Section 10.3. A general scheme of Bayes-type conformal prediction is described in [51], pp. 102–103.

5 On-line prediction

We know from Section 3 that conformal predictors are valid in the sense that the probability of error

$$y_{l+1} \notin \Gamma^\epsilon (x_1, y_1, \dots x_l, y_l, x_{l+1}) \tag{12}$$

at confidence level $1 - \epsilon$ never exceeds ϵ. The word 'probability' means 'unconditional probability' here: the frequentist meaning of the statement that the probability of event (12) does not exceed ϵ is that, if we repeatedly generate many sequences

$$x_1, y_1, \dots, x_l, y_l, x_{l+1}, y_{l+1},$$

the fraction of them satisfying Equation (12) will be at most ϵ, to within statistical fluctuations. To say that we are controlling the number of errors would be an exaggeration because of the artificial character of this scheme of repeatedly generating a new training set and a new test example. Can we say that the confidence level $1 - \epsilon$ translates into a bound on the number of errors for a natural learning protocol? In this section we show that the answer is 'yes' for the popular on-line learning protocol, and in the next section we will see to what degree this carries over to other protocols.

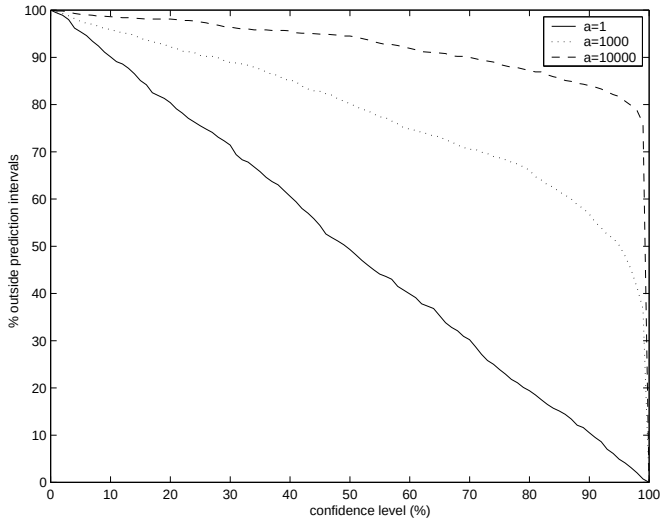

Figure 4: Validity for the Bayes-optimal confidence predictor.

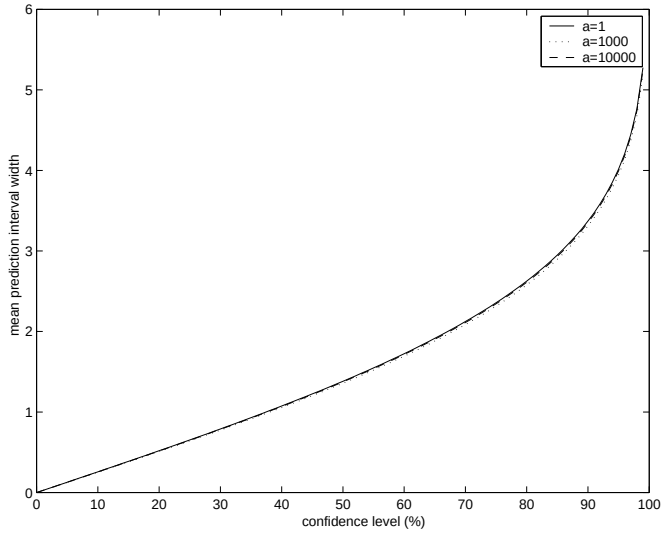

Figure 5: Efficiency for the Bayes-optimal confidence predictor.

In on-line learning the examples are presented one by one. Each time, we observe the object and predict its label. Then we observe the label and go on to the next example. We start by observing the first object x_1 and predicting its label y_1. Then we observe y_1 and the second object x_2, and predict its label y_2. And so on. At the nth step, we have observed the previous examples $(x_1, y_1), \dots, (x_{n-1}, y_{n-1})$ and the new object x_n, and our task is to predict y_n. The quality of our predictions should improve as we accumulate more and more old examples. This is the sense in which we are learning.

Our prediction for y_n is a nested family of prediction sets $\Gamma^\epsilon_n \subseteq \mathbf{Y}$, $\epsilon \in (0,1)$. The process of prediction can be summarized by the following protocol:

ON-LINE PREDICTION PROTOCOL

$\mathrm{Err}^\epsilon_0 := 0, \quad \epsilon \in (0,1)$;
$\mathrm{Mult}^\epsilon_0 := 0, \quad \epsilon \in (0,1)$;
$\mathrm{Emp}^\epsilon_0 := 0, \quad \epsilon \in (0,1)$;
FOR $n = 1, 2, \dots$:
  Reality outputs $x_n \in \mathbf{X}$;
  Predictor outputs $\Gamma^\epsilon_n \subseteq \mathbf{Y}$ for all $\epsilon \in (0,1)$;
  Reality outputs $y_n \in \mathbf{Y}$;
  $\mathrm{err}^\epsilon_n := \begin{cases} 1 & \text{if } y_n \notin \Gamma^\epsilon_n \\ 0 & \text{otherwise,} \end{cases} \quad \epsilon \in (0,1)$;
  $\mathrm{Err}^\epsilon_n := \mathrm{Err}^\epsilon_{n-1} + \mathrm{err}^\epsilon_n, \quad \epsilon \in (0,1)$;
  $\mathrm{mult}^\epsilon_n := \begin{cases} 1 & \text{if } |\Gamma^\epsilon_n| > 1 \\ 0 & \text{otherwise,} \end{cases} \quad \epsilon \in (0,1)$;
  $\mathrm{Mult}^\epsilon_n := \mathrm{Mult}^\epsilon_{n-1} + \mathrm{mult}^\epsilon_n, \quad \epsilon \in (0,1)$;
  $\mathrm{emp}^\epsilon_n := \begin{cases} 1 & \text{if } |\Gamma^\epsilon_n| = 0 \\ 0 & \text{otherwise,} \end{cases} \quad \epsilon \in (0,1)$;
  $\mathrm{Emp}^\epsilon_n := \mathrm{Emp}^\epsilon_{n-1} + \mathrm{emp}^\epsilon_n, \quad \epsilon \in (0,1)$
END FOR.

As we said, the family Γ^ϵ_n is assumed nested: $\Gamma^{\epsilon_1}_n \subseteq \Gamma^{\epsilon_2}_n$ when $\epsilon_1 \geq \epsilon_2$. In this protocol we also record the cumulative numbers Err^ϵ_n of erroneous prediction sets, Mult^ϵ_n of *multiple* prediction sets (i.e., prediction sets containing more than one label) and Emp^ϵ_n of empty prediction sets at each confidence level $1-\epsilon$. We will discuss the significance of each of these numbers in turn.

The number of erroneous predictions is a measure of validity of our confidence predictors: we would like to have $\mathrm{Err}^\epsilon_n \leq \epsilon n$, up to statistical fluctuations. In Figure 6 we can see the lines $n \mapsto \mathrm{Err}^\epsilon_n$ for one particular conformal predictor and for three confidence levels $1-\epsilon$: the solid line for 99%, the dash-dot line for 95%, and the dotted line for 80%. The number of errors made grows linearly, and the slope is approximately 20% for the confidence level 80%, 5% for the confidence level 95%, and 1% for the confidence level 99%. We will see below that this is not accidental.

The number of multiple predictions Mult^ϵ_n is a useful measure of efficiency in the case of classification: we would like as many as possible of our predictions to be singletons. Figure 7 shows the cumulative numbers of errors $n \mapsto \mathrm{Err}^{2.5\%}_n$

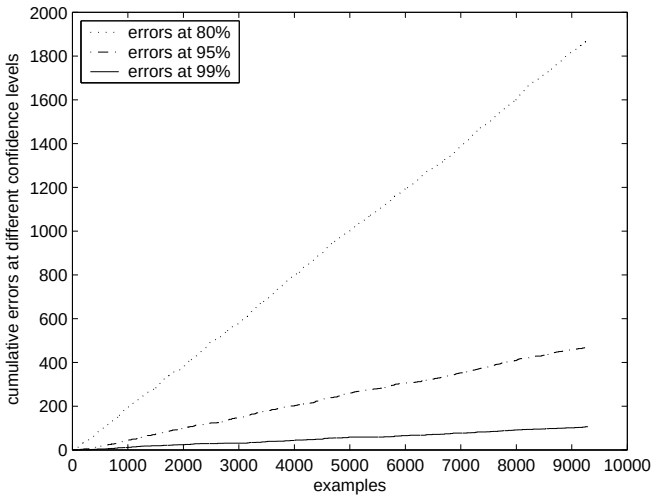

Figure 6: Cumulative numbers of errors for a conformal predictor (the 1-nearest neighbour conformal predictor) run in the on-line mode on the USPS data set (9298 hand-written digits, randomly permuted) at the confidence levels 80%, 95% and 99%.

(solid line) and multiple predictions $n \mapsto \mathrm{Mult}_n^{2.5\%}$ (dotted line) at the fixed confidence level 97.5%. We can see that out of approximately 10,000 predictions about 250 (approximately 2.5%) were errors and about 300 (approximately 3%) were multiple predictions.

We can see that by choosing ϵ we are able to control the number of errors. For small ϵ (relative to the difficulty of the data set) this might lead to the need sometimes to give multiple predictions. On the other hand, for larger ϵ this might lead to empty predictions at some steps, as can be seen from the bottom right corner of Figure 7: when the predictor ceases to make multiple predictions it starts making occasional empty predictions (the dash-dot line). An empty prediction is a warning that the object to be predicted is unusual (the credibility, as defined in Section 2, is ϵ or less).

It would be a mistake to concentrate exclusively on one confidence level $1 - \epsilon$. If the prediction Γ_n^ϵ is empty, this does not mean that we cannot make any prediction at all: we should just shift our attention to other confidence levels (perhaps look at the range of ϵ for which Γ_n^ϵ is a singleton). Likewise, Γ_n^ϵ being multiple does not mean that all labels in Γ_n^ϵ are equally likely: slightly increasing ϵ might lead to the removal of some labels. Of course, taking in the continuum of predictions sets, for all $\epsilon \in (0, 1)$, might be too difficult or tiresome for a human mind, and concentrating on a few conventional levels, as in Figure 1, might be a reasonable compromise.

For example, Table 2 gives the p-values for different kinds of abdominal pain obtained for a specific patient based on his symptoms. We can see that at the

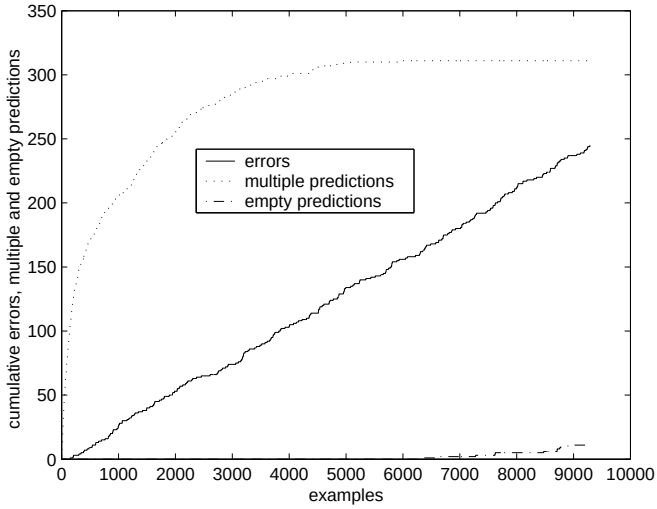

Figure 7: The on-line performance of the 1-nearest neighbour conformal predictor at the confidence level 97.5% on the USPS data set (randomly permuted).

Table 2: A selected test example from a data set of hospital records of patients who suffered acute abdominal pain [15]: the p-values for the nine possible diagnostic groups (appendicitis APP, diverticulitis DIV, perforated peptic ulcer PPU, non-specific abdominal pain NAP, cholecystitis CHO, intestinal obstruction INO, pancreatitis PAN, renal colic RCO, dyspepsia DYS) and the true label.

APP	DIV	PPU	NAP	CHO	INO	PAN	RCO	DYS	true label
1.23%	0.36%	0.16%	2.83%	5.72%	0.89%	1.37%	0.48%	80.56%	DYS

confidence level 95% the prediction set is multiple, {cholecystitis, dyspepsia}. When we relax the confidence level to 90%, the prediction set narrows down to {dyspepsia} (the singleton containing only the true label); on the other hand, at the confidence level 99% the prediction set widens to {appendicitis, non-specific abdominal pain, cholecystitis, pancreatitis, dyspepsia}. Such detailed confidence information, in combination with the property of validity, is especially valuable in medicine (and some of the first applications of conformal predictors have been to the fields of medicine and bioinformatics: see, e.g., [3, 35]).

In the case of regression, we will usually have $\mathrm{Mult}_n^\epsilon = n$ and $\mathrm{Emp}_n^\epsilon = 0$, and so these are not useful measures of efficiency. Better measures, such as the ones used in the previous section, would, for example, take into account the widths of the prediction intervals.

Theoretical analysis

Looking at Figures 6 and 7 we might be tempted to guess that the probability of error at each step of the on-line protocol is ϵ and that errors are made independently at different steps. This is not literally true, as a closer examination of the bottom left corner of Figure 7 reveals. It, however, becomes true (as noticed in [48]) if the p-values (5) are redefined as

$$p_Y := \frac{|\{i : \alpha_i > \alpha_{l+1}\}| + \eta\,|\{i : \alpha_i = \alpha_{l+1}\}|}{l+1}, \tag{13}$$

where i ranges over $\{1, \ldots, l+1\}$ and $\eta \in [0,1]$ is generated randomly from the uniform distribution on $[0,1]$ (the ηs should be independent between themselves and of everything else; in practice they are produced by pseudo-random number generators). The only difference between Equations (5) and (13) is that the expression (13) takes more care in breaking the ties $\alpha_i = \alpha_{l+1}$. Replacing Equation (5) by Equation (13) in the definition of conformal predictor we obtain the notion of *smoothed conformal predictor*.

The validity property for smoothed conformal predictors can now be stated as follows.

Theorem 1 *Suppose the examples*

$$(x_1, y_1), (x_2, y_2), \ldots$$

are generated independently from the same probability distribution. For any smoothed conformal predictor working in the on-line prediction protocol and any confidence level $1 - \epsilon$*, the random variables* $\mathrm{err}_1^\epsilon, \mathrm{err}_2^\epsilon, \ldots$ *are independent and take value 1 with probability* ϵ*.*

Combining Theorem 1 with the strong law of large numbers we can see that

$$\lim_{n\to\infty} \frac{\mathrm{Err}_n^\epsilon}{n} = \epsilon$$

holds with probability one for smoothed conformal predictors. (They are 'well calibrated'.) Since the number of errors made by a conformal predictor never

exceeds the number of errors made by the corresponding smoothed conformal predictor,

$$\limsup_{n\to\infty} \frac{\mathrm{Err}_n^\epsilon}{n} \le \epsilon$$

holds with probability one for conformal predictors. (They are 'conservatively well calibrated'.)

6 Slow teachers, lazy teachers, and the batch setting

In the pure on-line setting, considered in the previous section, we get an immediate feedback (the true label) for every example that we predict. This makes practical applications of this scenario questionable. Imagine, for example, a mail sorting centre using an on-line prediction algorithm for zip code recognition; suppose the feedback about the true label comes from a human 'teacher'. If the feedback is given for every object x_i, there is no point in having the prediction algorithm: we can just as well use the label provided by the teacher. It would help if the prediction algorithm could still work well, in particular be valid, if only every, say, tenth object were classified by a human teacher (the scenario of 'lazy' teachers). Alternatively, even if the prediction algorithm requires the knowledge of all labels, it might still be useful if the labels were allowed to be given not immediately but with a delay ('slow' teachers). In our mail sorting example, such a delay might make sure that we hear from local post offices about any mistakes made before giving a feedback to the algorithm.

In the pure on-line protocol we had validity in the strongest possible sense: at each confidence level $1-\epsilon$ each smoothed conformal predictor made errors independently with probability ϵ. In the case of weaker teachers (as usual, we are using the word 'teacher' in the general sense of the entity providing the feedback, called Reality in the previous section), we have to accept a weaker notion of validity. Suppose the predictor receives a feedback from the teacher at the end of steps $n_1, n_2, \ldots$, $n_1 < n_2 < \cdots$; the feedback is the label of one of the objects that the predictor has already seen (and predicted). This scheme [33] covers both slow and lazy teachers (as well as teachers who are both slow and lazy). It was proved in [29] (see also [51], Theorem 4.2) that the smoothed conformal predictors (using only the examples with known labels) remain valid in the sense

$$\forall \epsilon \in (0,1) : \mathrm{Err}_n^\epsilon / n \to \epsilon \text{ (as } n \to \infty) \text{ in probability}$$

if and only if $n_k/n_{k-1} \to 1$ as $k \to \infty$. In other words, the validity in the sense of convergence in probability holds if and only if the growth rate of n_k is subexponential. (This condition is amply satisfied for our example of a teacher giving feedback for every tenth object.)

The most standard *batch* setting of the problem of prediction is in one respect even more demanding than our scenarios of weak teachers. In this setting we

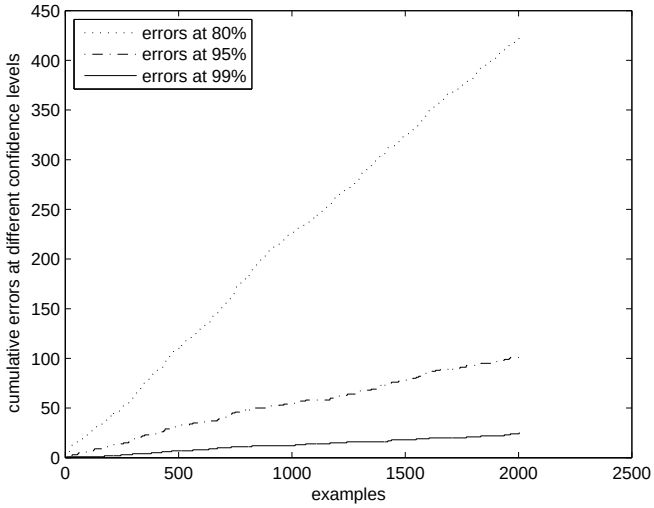

Figure 8: Cumulative numbers of errors made on the test set by the 1-nearest neighbour conformal predictor used in the batch mode on the USPS data set (randomly permuted and split into a training set of size 7291 and a test set of size 2007) at the confidence levels 80%, 95% and 99%.

are given a training set (1) and our goal is to predict the labels given the objects in the test set

$$(x_{l+1}, y_{l+1}), \ldots, (x_{l+k}, y_{l+k}). \tag{14}$$

This can be interpreted as a finite-horizon version of the lazy-teacher setting: no labels are returned after step l. Computer experiments (see, e.g., Figure 8) show that approximate validity still holds; for related theoretical results, see [51], Section 4.4.

7 Induction and transduction

Vapnik's [42, 43] distinction between induction and transduction, as applied to the problem of prediction, is depicted in Figure 9. In inductive prediction we first move from examples in hand to some more or less general rule, which we might call a prediction or decision rule, a model, or a theory; this is the inductive step. When presented with a new object, we derive a prediction from the general rule; this is the deductive step. In transductive prediction, we take a shortcut, moving from the old examples directly to the prediction about the new object.

Typical examples of the inductive step are estimating parameters in statistics and finding an approximating function in statistical learning theory. Examples of transductive prediction are estimation of future observations in statistics ([9], Section 7.5, [38]) and nearest neighbours algorithms in machine learning.

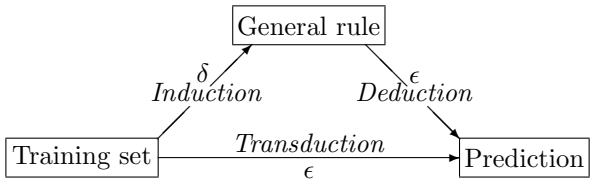

Figure 9: Inductive and transductive prediction.

In the case of simple (i.e., traditional, not hedged) predictions the distinction between induction and transduction is less than crisp. A method for doing transduction, in the simplest setting of predicting one label, is a method for predicting y_{l+1} from training set (1) and x_{l+1}. Such a method gives a prediction for any object that might be presented as x_{l+1}, and so it defines, at least implicitly, a rule, which might be extracted from the training set (1) (induction), stored, and then subsequently applied to x_{l+1} to predict y_{l+1} (deduction). So any real distinction is really at a practical and computational level: do we extract and store the general rule or not?

For hedged predictions the difference between induction and transduction goes deeper. We will typically want different notions of hedged prediction in the two frameworks. Mathematical results about induction usually involve two parameters, often denoted ϵ (the desired accuracy of the prediction rule) and δ (the probability of failing to achieve the accuracy of ϵ), whereas results about transduction involve only one parameter, which we denote ϵ in this article (the probability of error we are willing to tolerate); see Figure 9. For a review of inductive prediction from this point of view, see [51], Section 10.1.

8 Inductive conformal predictors

Our approach to prediction is thoroughly transductive, and this is what makes valid and efficient hedged prediction possible. In this section we will see, however, that there is also room for an element of induction in conformal prediction.

Let us take a closer look at the process of conformal prediction, as described in Section 3. Suppose we are given a training set (1) and the objects in a test set (14), and our goal is to predict the label of each test object. If we want to use the conformal predictor based on the support vector method, as described in Section 3, we will have to find the set of the Lagrange multipliers for each test object and for each potential label Y that can be assigned to it. This would involve solving $k\,|\mathbf{Y}|$ essentially independent optimization problems. Using the nearest neighbours approach is typically more computationally efficient, but even it is much slower than the following procedure, suggested in [30, 31].

Suppose we have an inductive algorithm which, given a training set (1) and a new object x outputs a prediction $\hat{y}$ for x's label y. Fix some measure $\Delta(y, \hat{y})$ of difference between y and $\hat{y}$. The procedure is:

1. Divide the original training set (1) into two subsets: the *proper training set* $(x_1, y_1), \ldots, (x_m, y_m)$ and the *calibration set* $(x_{m+1}, y_{m+1}), \ldots, (x_l, y_l)$.

2. Construct a prediction rule F from the proper training set.

3. Compute the nonconformity score

$$\alpha_i := \Delta(y_i, F(x_i)), \quad i = m+1, \ldots, l,$$

   for each example in the calibration set.

4. For every test object x_i, $i = l+1, \ldots, l+k$, do the following:

   (a) for every possible label $Y \in \mathbf{Y}$ compute the nonconformity score $\alpha_i := \Delta(Y, F(x_i))$ and the p-value

$$p_Y := \frac{\#\{j \in \{m+1, \ldots, l, i\} : \alpha_j \geq \alpha_i\}}{l - m + 1};$$

   (b) output the prediction set $\Gamma^\epsilon(x_1, y_1, \ldots, x_l, y_l, x_i)$ given by the right-hand side of Equation (7).

This is a special case of 'inductive conformal predictors', as defined in [51], Section 4.1. In the case of classification, of course, we could package the p-values as a simple prediction complemented with confidence (8) and credibility (9).

Inductive conformal predictors are valid in the sense that the probability of error

$$y_i \notin \Gamma^\epsilon(x_1, y_1, \ldots x_l, y_l, x_i)$$

($i = l+1, \ldots, l+k$, $\epsilon \in (0,1)$) never exceeds ϵ (cf. (12)). The on-line version of inductive conformal predictors, with a stronger notion of validity, is described in [48] and [51] (Section 4.1).

The main advantage of inductive conformal predictors is their computational efficiency: the bulk of the computations is performed only once, and what remains to do for each test object and each potential label is to apply the prediction rule found at the inductive step, to apply Δ to find the nonconformity score α for these object and label, and to find the position of α among the nonconformity scores of the calibration examples. The main disadvantage is a possible loss of the prediction efficiency: for conformal predictors, we can effectively use the whole training set as both the proper training set and the calibration set.

9 Conclusion

This article shows how many machine-learning techniques can be complemented with provably valid measures of accuracy and reliability. We explained briefly how this can be done for support vector machines, nearest neighbours algorithms

and the ridge regression procedure, but the principle is general: virtually any (we are not aware of exceptions) successful prediction technique designed to work under the randomness assumption can be used to produce equally successful hedged predictions. Further examples are given in our recent book [51] (joint with Glenn Shafer), where we construct conformal predictors and inductive conformal predictors based on nearest neighbours regression, logistic regression, bootstrap, decision trees, boosting, and neural networks; general schemes for constructing conformal predictors and inductive conformal predictors are given on pp. 28–29 and on pp. 99–100 of [51], respectively. Replacing the original simple predictions with hedged predictions enables us to control the number of errors made by appropriately choosing the confidence level.

Acknowledgements

This work is partially supported by MRC (grant 'Proteomic analysis of the human serum proteome') and the Royal Society (grant 'Efficient pseudo-random number generators').

A Discussion

Alexey Chervonenkis

Research Institute of Control Problems, Russian Academy of Sciences

Computer Learning Research Centre,
Royal Holloway, University of London

chervnks@ipu.rssi.ru

A large variety of machine-learning algorithms are now developed and applied in different areas of science and industry. This new technique has a typical drawback, that there is no confidence measure for prediction of output value for particular new objects. The main idea of the article is to look over all possible labellings of a new object and evaluate strangeness of each labelling in comparison to the labelling of objects presented in the training set. The problem is to find an appropriate measure of strangeness. Initially the authors try to apply the ideas of Kolmogorov complexity to estimate the strangeness of labelling. But firstly this complexity is not computable, then it is defined up to an additive constant, and finally it is applied to the total sequence of objects, but not to one particular object. So the authors came to another idea (still induced by Kolmogorov complexity). Based on particular machine-learning algorithm it is possible to find a reasonable measure of an object (with its labelling) strangeness. For regression (or ridge regression) it could be the absolute difference between regression result and real output value: the larger is the difference, the stranger is the object. In the SVM approach to pattern recognition it could be the weights of support vectors: the larger is the weight of a vector, the more doubtful seems its labelling, and similar measures of strangeness may be proposed for other algorithms. So the protocol is as follows: look through all possible labellings

of a new object. For each labelling add the object to the training set. Apply the machine-learning algorithm and rank the objects by their measure of strangeness. Estimate credibility of this labelling as (one minus) the ratio of the number of objects in the set stranger than the new one to the total number of objects in the set. This approach seems to be new and powerful. Its main advantage is that it is non-parametric and based only on the i.i.d. assumption. In comparison to the Bayesian approach, no prior distribution is used. The main theoretical result is the proof of validity of proposed conformal predictors. It means that on average conformal predictors never overrate the accuracy and reliability of their predictions. The second result is that asymptotically the relative number of cases when the real output value is within confidence interval converges to the average value of conformal predictors. Software implementing the proposed technique is now applied to a large variety of practical problems.

Still I can mention two drawbacks of the article.

1. There is no theoretical discussion on the problem how far proposed confidence intervals are optimal for particular objects. In general it is possible that for some objects the interval is too large, for other it is too small, but on average validity in terms of the article is true. Optimality can be proved for the Bayesian approach, though it needs prior distribution. Experimental results of comparison of proposed conformal predictors with the Bayesian approach for particular problem is presented in the article, and it is shown that the results are quite close to the optimal ones, but some theoretical discussion seems to be useful.

2. In pattern recognition problems it is proposed to measure confidence as 'one minus the second largest randomness level detected'. It seems better to use as the measure the difference between the largest and the second largest value. For instance, in Table 1, line 3, we see that for true label 6, credibility is 1.43%, while confidence is 98.93%. If we take the difference between the largest and the second largest value, confidence becomes very low, and really in this case the prediction is false.

In total the article summarizes the whole cycle of works by the authors on conformal predictors and its presentation to the *Computer Journal* can be only greeted.

Philip M. Long

Google Inc.

plong@google.com

Conformal prediction is a beautiful and powerful idea. It enables the design of useful methods for assigning confidence to the predictions made by machine-learning algorithms, and also enables clean and relevant theoretical analyses.

It appears that conformal prediction may have a role to play in reinforcement learning, where an agent must learn from the consequences of its actions.

In reinforcement learning settings, the behaviour of the learner affects the information its receives, so there is a tension between taking actions to gather useful information (exploration), and taking actions that are profitable right now (exploitation). When an agent can be confident about how to behave, exploration is less advisable. A formalization of this idea has already been exploited to strengthen theoretical guarantees for some reinforcement learning problems [1]; it seems that conformal prediction might be a useful tool for analyses like this.

The authors advanced a view of conformal prediction methods as randomness tests. On the one hand, there is a *proof* that some conformal predictors are randomness tests. On the other hand, a procedure that satisfies the formal requirement of what is termed a randomness test might return scores that are most closely associated with some other property of the distribution governing all of the examples.

For example, suppose Equation (5) from the article is applied with support vector machines with the linear kernel, and the features are uniform random boolean variables. If the class designation is the parity of the features, the values of (5) should be expected to be less than if the class designation is the value of the first feature, even if the data is i.i.d. for both sources.

Very roughly speaking, in many applications, one expects randomness between examples and structure within them. A randomness test only detects the randomness between examples. It seems that much of the power of the conformal predictors is derived from their ability to exploit structure in the distribution generating the examples.

On the other hand, when a prospective class assignment is at odds with structure found in earlier examples, one possibility is to blame the apparent contradiction on the assertion the training examples were not representative.

Still, the parity example above suggests that effective conformal predictors must be more than good randomness tests, even if the formal notion of what has been termed a randomness test is useful for their analysis.

Whatever the source of the power, one thing that does seem clear is that conformal prediction is a powerful tool.

Xiaohui Liu

Brunel University

Impact of hedging predictions on applications with high-dimensional data

The authors are to be congratulated on their excellent discussions of the background in the area, their clear exposure of the inadequacies of current approaches to analysing high-dimensional data, and their introduction of ground-breaking methods for 'hedging' the predictions produced by existing machine-learning methods. In this response, I would like to argue that one of the key issues for widening the use of hedged predictions would be how to assist users with careful interpretation and utilisation of the two confidence measures in the predictions.

I shall use the classification of high-dimensional DNA microarray data as an example.

There has been a lot of work over the past few years on the use of various supervised learning methods to build systems that could classify subjects with or without a particular disease, or categorise genes exhibiting similar biological functions, using the expression levels of genes which are typically in the range of hundreds or thousands. Since algorithms for producing hedged predictions are capable of giving an indication of not only how accurate but also how reliable individual classifications are, they could provide biomedical scientists with a nice way of quickly homing in on a small set of genes with sufficiently high accuracy and reliability for further study.

But how should biologists choose the cut-off values for the two new measures to make that decision? If the values are set too high, we risk many false negatives—interesting genes may escape our attention. If they are too low, we may see many false positives—biologists may have to study many more genes than necessary, which can be costly since such a study may involve examining things such as the sequences of suspect genes, transcription factors, protein-protein interactions, related structural and functional information, etc., or even conducting further biological experiments [37]. Of course it is also challenging to address how to minimise the false positives and false negatives for any existing statistical confidence measure, but it would be crucial for practitioners to gain as much help as possible when any new measures are introduced.

Recently we have suggested a method for identifying highly predictive genes from a large number of prostate cancer and B-cell genes using a simple classifier coupled with a feature selection and global search method as well as applying data perturbation and cross-validation [45]. We will be keen to extend that approach using the proposed methods to produce hedged predictions, and then study the effects of using the two confidence measures for the same applications.

In short, the proposed methods for hedging predictions should provide practitioners with further information and confidence. Key issues in exploiting their full potentials in real-world applications include how one should effectively interpret the confidence measures and utilise them for decision making in a given situation, and how to build different types of conformal predicting tools to facilitate their use in diverse practical settings.

Sally McClean

University of Ulster

I would like to congratulate the authors on providing a very clear and insightful discussion of their approach to providing measures of reliability and accuracy for prediction in machine learning. This is undoubtedly an important area and the tools developed here should prove invaluable in a variety of contexts.

I was intrigued by the authors' concept of 'strangeness', as measured by the α_is. The examples given in the article seem very intuitive and also to perform well. However, I wondered if there were a more principled way of designing

good measures of strangeness or should we just look for measures that are high performing in terms of efficiency and computational complexity.

Zhiyuan Luo and Tony Bellotti

Computer Learning Research Centre,
Royal Holloway, University of London

This is a very stimulating article about the very important issue of making reliable decisions under uncertainty. We would like to discuss some applications of conformal predictors to microarray gene expression classification for cancer diagnosis and prognosis in our collaboration with Cancer Research UK Children's Cancer Group. Microarray technology allows us to take a sample of cells and measure the abundance of mRNA associated with each gene, giving a level of activity (expression) for each gene, expressed on a numeric scale. From the analysis of the microarray data, we can get insights into various diseases such as cancer. Typically machine-learning methods are used for microarray gene expression classification.

Most machine-learning algorithms such as the support vector machine [43] provide only bare predictions, in their basic form. However, not knowing the confidence of predictions makes it difficult to measure and control the risk of error using a decision rule. This issue has been discussed by several authors. Dawid [10] argues that many decisions can only be taken rationally when the uncertain nature of the problem domain is taken into consideration. An example of this is weather forecasting, where Probability of Precipitation forecasts are commonly used, instead of simple bare predictions of rain or no rain. Korb [21] argues that machine learning has traditionally emphasized performance measures that evaluate the amount of knowledge acquired, ignoring issues about confidence in decisions. It is important that decision rules also provide meta-knowledge regarding the limits of domain knowledge in order for us to use them effectively with an understanding of risk of outcome. This is possible if we provide a measure of confidence with predictions. In the medical domain, it is important to be able to measure the risk of misdiagnosis or disease misclassification, and if possible, to ensure low risk of error. Machine-learning algorithms have been used to make predictions from microarrays, but without information about the confidence in predictions. Confidence intervals can be given to estimate true accuracy, using classical statistical methods, but in practice the computed intervals are often too broad to be clear that the classification method is reliable. This is due to the typically low sample size and high-dimensionality of microarray data available for any one experiment. In particular, a study of cross-validation for microarray classification using bare prediction has shown high variance of results leading to inaccurate conclusions for small sample size [4]. The problem of sample size is exacerbated in the case of leukaemia by the large number of subtypes, which may mean that only a few samples are available for training for some subtypes. In such circumstances, bare predictions made by conventional algorithms must understandably be treated with caution. There-

fore, there is a need for a theoretical framework that will allow us to determine more accurately the reliability of classification based on microarray data.

The conformal predictors provide a framework for constructing learning algorithms that predict with confidence. Conformal predictors allow us to supplement such predictions with a confidence level, assuring reliability, even for small sample size. This approach is therefore particularly suitable for the classification of gene expression data. For traditional learning algorithms, usually given as simple predictors, the focus has naturally been on improved accuracy. For these algorithms, efficiency is fixed as all predictions are of one single class label. In contrast, with conformal predictors, accuracy is controlled by a preset confidence level and efficiency is variable and needs to be optimized. When evaluating the performance of a learning algorithm, it is important to measure error calibration as well as its accuracy. This has been a somewhat neglected aspect of evaluation. The main benefit of conformal predictors is that calibration is controlled by the *a priori* confidence level. The challenge is to design nonconformity measures for the underlying learning algorithms to maximize efficiency.

Another benefit of conformal predictors is that they can give a level of uncertainty regarding each individual prediction in the form of a hedged region prediction. In contrast the confidence interval supplies only a general estimate for true accuracy for single class label predictions, therefore supplying no information regarding uncertainty for individual predictions. For many learning problems, this may be important to distinguish those patients that are easier to diagnose from others, in order to control risk for individual patients.

David Bell

Department of Computer Science,
Queen's University Belfast, Belfast BT1 7NN

da.bell@qub.ac.uk

In data mining meaningful measures of validity and possible ways of using them are always welcome. They can supplement more naïve, readily accessible quantities. As in other aspects of computing, such as hashing or clustering, 'Horses for courses' is the rule when looking for mining algorithms and the same applies to measures of their 'goodness'. Now there are two types of thinker according to the philosopher A. Whitehead—'simple-minded' and 'muddle-headed'. Neither description is particularly flattering. Some abstract analysts looking for understanding and explanation tend to the first extreme, and some practical problem solvers looking for pay-offs are towards the other end of the spectrum.

In data mining exchanges of ideas between the two types are common. For example, Kolmogorov complexity is noncomputable, and some practitioners see it as conceptually so rarefied that it is of little use. However, due not least to the efforts of authors such as Alex Gammerman and Volodya Vovk, practical value can accrue from the concept. More muddle-headed activity can also be useful. Aeronautics has matured to a degree not yet registered in the emergence

of machine learning. Its pioneers had an interesting, muddle-headed way of working. In the early days, brash enthusiasts made 'wings' and jumped off cliffs. If something worked, the analysis/understanding/insights often came later, and led to real progress.

The BCS Machine Intelligence Prize is in this spirit. It is awarded annually for a live demonstration of Progress Towards Machine Intelligence—'can-do' system building by competitors—who might, incidentally, understand 'hedging' as something entirely more practical than its sense in our article, or at least something to do with programming language theory or XML. Full understanding often lags behind, but it would be better to have a nice balance between the simple-minded and muddle-headed inputs. Using the words of P. Dawid, experimentalist AI researchers who aim to produce programs with learning behaviour like that of animals make '... valuable contributions going beyond those likely to occur to a mindset constrained by probability theory or coding theory' [11], but progress will be held up if the foundations are not attended to.

Things are moving ahead in data mining. The simple-minded approach is becoming less simple. Increased scope is being introduced; e.g., in the training/learning sequences, test labels can be explicitly related, and dependent prediction can be beneficial even on i.i.d. data. Furthermore, M. Gell-Mann suggests using 'the length of the shortest message that will describe a system... employing language, knowledge, and understanding that both parties share' instead of Kolmogorov complexity [16]. Now some scientists resist, and 'share... a degree of embarrassment' at, including consciousness at the most fundamental levels—but, for example, it 'remains a logical possibility that it is the act of consciousness which is ultimately responsible for the reduction of the wave packet' in quantum mechanics [2].

In muddle-headed games of prediction, muddiness as defined by J. Weng [56] is prevalent, and they often have in-built structure. There are emerging paradigms of learning, e.g., in robotics and video mining. For example, second-order learning, or learning about learning, is evident when a predator watches a potential prey as it adapts, to try to get an advantage. Here, because of the inherent structuring in the data, we have both inductive and transductive learning. The inductive learning and inference approach is useful when an overview model of the problem is required. But such models are difficult to create and update, and they are often not needed. A long time ago, J. S. Mill [27] wrote 'An induction from particulars to generals, followed by a syllogistic process from those generals to other particulars... is not a form in which we must reason...'. (Muddle-headed?) transductive arguing from particulars to particulars is often better. To combine transductive and inductive reasoning for robotics, video mining and other applications, we focus on rough sets methods—for associative learning and multi-knowledge. Adaptability, representation and noise handling are key issues. Hopefully we can adopt some of the measures presented here.

David L. Dowe

Faculty of IT, Monash University, Clayton, Victoria 3800, Australia

dld@bruce.csse.monash.edu.au

Profs Gammerman and Vovk advocate a welcome preference for the generality of the (universal) Turing machine (TM) (and Kolmogorov complexity) approach over the conventional Bayesian approach (which usually assumes 'a parametric statistical model, sometimes complemented with a prior distribution on the parameter space') to (inference and) prediction. My comments below are based on my best understanding.

There are many parallels between the authors' approach (to prediction) and the Minimum Message Length (MML) approach (to inference) of Wallace et al. [53, 54, 55, 52], and also some apparent distinctions.

The authors mention randomness tests and that 'Martin-Löf (developing Kolmogorov's earlier ideas) proved that there exists a smallest, to within a constant factor, randomness test'. This parallels the formal relationship between Kolmogorov complexity, (universal) TMs and (Strict) MML [55] and the choice (within a small constant) ([52], Section 2.3.12) of a *simplest* UTM as a way of modelling prior ignorance.

In Section 2, the *confidence* in the prediction is one minus the second largest randomness level detected by t. For non-binary problems, this confidence seems too large—if all of the randomness levels were close in value to one another, the confidence should presumably be close to 1 divided by the number of classes. In Figure 6, perhaps relatedly, the three lines appear to have slightly larger gradients than their confidence levels should permit.

At the end of Section 2, because their universal confidence predictor is not computable, the authors set their goal to find computable approximations. In this case, there are both frequentist ([54], Section 3.3) and algorithmic complexity ([55], [52], Section 6.7.2, p. 275) Bayesian reasons for advocating Strict MML (SMML) as a form of inference. SMML can be further approximated ([52], Chapters 4–5, etc., [55], Section 6.1.2).

The choice of (universal or non-universal) TM and of randomness test, t, is still a Bayesian choice ([55], [52], Section 2.3) (even if not conventionally so ([52], Sections 2.3.11–2.3.13)), so in Section 4 when the authors find an improvement over the 'Bayes-optimal predictor' and talk of a conformal predictor being 'asymptotically as good as the Bayes-optimal', this might be because their underlying TM is more expressive than the original Bayesian prior and so has it as a special case.

In Table 1 and Section 4, which (non-universal) test is being used?

I would welcome log-loss scores reported with the error counts of Figures 6 and 7.

MML has dealt with problems where the amount of data per continuous-valued parameter is bounded above ([52], Section 6.9) and with 'inverse learning' problems where the best way to model the target attribute might be to model it jointly or to model other attributes in terms of it ([13], [7], [8], [40], Section 5).

Vapnik ([42], Section 4.6) discusses using MDL (or MML) to model SVMs. For a hybrid of both decision trees and SVMs using MML and allowing non-binary classifications (without requiring 'one-against-the-rest' procedures), see [40].

Inference and prediction are closely related ([55], Section 8), and we endorse the TM approach to both problems. Today's article has been a useful advance in this direction.

Glenn Shafer

Royal Holloway, University of London, and Rutgers University

This article provides an excellent explanation of the fundamentals of conformal prediction. I have already begun recommending it to those who want to master the method without wading into the more comprehensive and intricate exposition in [51].

Like all good ideas, conformal prediction has a complex ancestry. As Gammerman and Vovk explain, they invented the method as a result of their study of work by Chervonenkis, Vapnik, Kolmogorov, and Martin-Löf. But they subsequently discovered related ideas in earlier work by mathematical statisticians. As we explain on pp. 256–257 of [51], Sam Wilks, Abraham Wald, and John Tukey developed non-parametric tolerance regions based on permutation arguments in the 1940s, and Donald Fraser and J. H. B. Kemperman used the same idea to construct prediction regions in the 1950s. From our viewpoint, Fraser and Kemperman were doing conformal prediction in the special case where ys are predicted without the use of xs. It is easy (once you see it) to extend the method to the case where xs are used, and Kei Takeuchi has told us that he explained this in the early 1970s, first in lectures at Stanford and then in a book that appeared in Japanese in 1975 [38]. Takeuchi's idea was not taken up by others, however, and the rediscovery, thorough analysis, and extensions by Gammerman and Vovk are remarkable achievements.

Because it brings together methods well known to mathematical statisticians (permutation methods in non-parametrics) and a topic now central to machine learning (statistical learning theory), the article prompts me to ask how these two communities can be further unified. How can we make sure the next generation of mathematical statisticians and computer scientists will have full access to each other's experience and traditions?

Statistical learning theory is limited in one very important respect: it considers only the case where examples are independent and identically distributed, or at least exchangeable. The i.i.d. case has also been central to statistics ever since Jacob Bernoulli proved the law of large numbers at the end of the 17th century, but its inadequacy was always obvious. Leibniz made the point in his letters to Bernoulli: the world is in constant flux; causes do not remain constant, and so probabilities do not remain constant. Perhaps Leibniz's point is a counterexample to itself, for it is as topical in 2006 as it was in the 1690s. In the most recent issue of *Statistical Science*, David Hand gives as one of his

reasons for scepticism about apparent progress in classifier technology the fact that 'in many, perhaps most, real classification problems the data points in the design set are not, in fact, randomly drawn from the same distribution as the data points to which the classifier will be applied' [18].

It is revealing that Hand finds it necessary to say this three centuries after Leibniz. We can cite methods that have been developed to deal with non-i.i.d. data:

1. Starting at the end of the 18th century, probabilists used models in which the ys are independent only given the xs. To get results, they then made strong assumptions about the distribution of the ys. If we assume the ys are Gaussian with constant variance and means linear in the xs, we get the Gauss linear model, so called because Gauss used it to prove the optimality of least squares [51].

2. Starting with Markov at the end of the 19th century, probabilists have studied stochastic process models—probability models for successive examples that are not necessarily i.i.d.

3. Statisticians often take differences between successive observations, perhaps even higher-order differences, in attempt to get something that looks i.i.d.

4. A major topic in machine learning, prediction with expert advice, avoids making any probability assumptions at all. Instead, one specifies a class of prediction procedures that one is competing with [6].

But we have stayed so true to Bernoulli in our overview of what statistics is about that we seldom ask potential statisticians and data analysts to look at a list like this. A general course in statistical inference usually still studies the i.i.d. case, leaving each alternative to be taken up as something distinct, often in some specialized discipline, such as psychometrics, econometrics, or machine learning, whose special terminology makes its results inaccessible to others. Except perhaps in a course in 'consulting', we seldom ponder or teach how to compare and choose among the alternatives.

Reinforcing the centrality of the i.i.d. picture is the centrality of the Cartesian product as the central structure for relational databases. Neither in statistics nor in computer science have we built on Art Dempster's now classic (but unfortunately not seminal) article on alternatives to the Cartesian product as a data structure [12].

More than 15 years ago I urged that statistics departments embrace the insights of specialized disciplines such as econometrics and machine learning in order to regain the unifying educational role that they held in the mid-twentieth century [34]. It is now clear that this will not happen. Statistics is genetically imprinted with the Bernoulli code [5]. Perhaps the machine learning community, which has had the imagination to break out of the probabilistic mode altogether with its concept of prediction with expert advice, should pick up this leadership mantle.

Drago Indjic

London Business School

Are there any results in applying confidence and credibility estimates to active (statistical) experiment design?

Glenn Hawe

Vector Fields Ltd., Oxford

School of Electronics and Computer Science, Southampton

In cost-effective optimization, 'surrogate modelling' is the estimation of objective function values for unevaluated design vectors, based on a set of design vectors which have their objective function values known. In this sense, surrogate modelling is to an optimization researcher, what machine learning is to a computer scientist.

Surrogate model-assisted optimization algorithms may be divided into two main categories [19]: two-stage and one-stage varieties. Two-stage algorithms involve fitting a surface to the observed examples, and then selecting the next design vector (object, in machine-learning terminology) to evaluate, based on this prediction (the idea in optimization being to evaluate the design vector (object) with the lowest valued objective function value (label)). Usually it is just the object with the lowest valued label which is evaluated, but sometimes uncertainty considerations are taken into account too, e.g., [20].

One-stage algorithms differ significantly—they make a hypothesis about the position of the global minimum, both its position in design variable space (its object value), and its objective function value (its label—hypothesized to be lower than the current minimum label), and then calculate the credibility of the surface which passes through the hypothesized point and the observed points. The credibility of the surface is related to its 'bumpiness', with bumpier surfaces being deemed less credible. The design vector which is evaluated next is the one which has the most credible surface passing through it (i.e., the object which has its label observed next is the object which has the most credible surface passing through it, having hypothesized its label to be lower than the lowest valued label observed so far).

So, it appears that, in machine-learning terminology, 'inductive inference' is completely analogous to 'two-stage algorithms' and 'transductive inference' is completely analogous to 'one-stage algorithms'. The interesting thing for optimization is that there has only been one one-stage algorithm proposed so far in the literature: an algorithm known as `rbfsolve` [17], which uses radial basis functions to interpolate the points, and is one of the best performing (single-objective) optimization algorithms around. It would appear that the work done by Gammerman and Vovk allows the one-stage technique of selecting points to evaluate to be applied to a wider range of surrogate models (and in particular, support vector machines), as it proposes a quantitative measure of the reliability of a hypothesized prediction. I suspect that a greater range of

one-stage optimization algorithm will appear as a result of this work, and in the light of the results of [17], that they will perform extremely well.

Vladimir Vapnik

AT&T Bell Laboratories, Holmdel, NJ

Computer Learning Research Centre,
Royal Holloway, University of London

Vladimir.Vapnik@rhul.ac.uk

I would like to congratulate the authors with their interesting article and stimulating research that opens several new directions in predictive learning. The authors present a new methodology of *hedging predictions*, and have removed some of the *ad hoc* procedures that are often used in calculating the bounds and confidence of prediction. In fact they introduced a new paradigm in pattern recognition research based on the Kolmogorov concept of randomness and therefore have opened a way for many new methods and algorithms in classification and regression estimation. This new methodology makes reliable predictions and it is impressive to see its comparison with the Bayesian approach, where the conformal predictors give correct results while Bayesian predictions are wrong. The article is interesting also since it allows us to see how the conformal predictors have been applied to several real-world examples. The results can also be applied to the vast majority of well-known machine-learning algorithms and demonstrate the importance of the transductive mode of inference.

In the late 1960s, in order to overcome the *curse of dimensionality* for pattern recognition problems, Alexey Chervonenkis and I introduced a different approach (the VC theory) called *Predictive Statistics*. The VC theory for constructing predictive models was a continuation of the Glivenko–Cantelli–Kolmogorov line of analysis of induction. At the heart of this theory are new concepts that define the capacity of the set of functions (characterization of the diversity of the set of functions defined by a given number of points): the VC entropy of the set of functions, the Growth function, and the VC dimension.

Until now, the traditional method of inference was the inductive-deductive method, where using available information one defines a general rule first, and then using this rule deduces the answer one needs. That is, first one goes from particular to general and then from general to particular. In the transductive mode one provides direct inference from particular to particular, avoiding the ill-posed part of the inference problem (inference from particular to general). The goal of transductive inference is to estimate the values of an unknown predictive function at a given point of interest (but not in the whole domain of its definition). By solving less demanding problems, one can achieve more accurate solutions. A general theory of transduction was developed where it was shown that the bounds of generalization for transductive inference are better than the corresponding bounds for inductive inference.

Transductive inference, in many respects, contradicts the main stream of the classical philosophy of science. The problem of the discovery of the general laws

of nature was considered in the philosophy of science to be the only scientific problem of inference because the discovered laws allow for *objective verification*. In transductive inference, *objective verification* is not straightforward. It would be interesting to know the authors' point of view on this subject.

Harris Papadopoulos

Frederick Institute of Technology, Nicosia, Cyprus

harrispa@cytanet.com.cy

I would like to congratulate the authors on this clearly written and detailed article. This article presents an excellent new technique for complementing the predictions produced by machine-learning algorithms with measures of confidence which are provably valid under the general i.i.d. assumption. One can easily appreciate the desirability of such measures in many real-world applications, as they can be used to determine the way in which each prediction should be treated. For instance, a filtering mechanism can be employed so that only predictions that satisfy a certain level of confidence will be taken into account, while the rest can be discarded or passed on to a human for judgment.

The most appealing feature of conformal prediction is that it can be applied to virtually any machine-learning method designed to work under the i.i.d. assumption without the need of any modification in order to achieve validity of the resulting confidence measures. Experimental results on a variety of conformal predictors (based on many different algorithms mentioned in the article) have shown that conformal predictors give high-quality confidence measures that are useful in practice, while their accuracy is, in almost all cases, exactly the same as that of their underlying algorithm. Consequently, conformal prediction does not have any undesirable effect on the accuracy of its base method, while it adds valuable information to its predictions.

The only drawback one can say that conformal predictors have, is their relative computational inefficiency, as they perform a much larger amount of computations than their underlying algorithms. Because of this, inductive conformal prediction (ICP), described in Section 8 of this article, was suggested in [30] for regression and in [31] for pattern recognition. We have successfully applied ICP to four widely used machine-learning techniques, namely ridge regression (described in [30]), nearest neighbours regression, nearest neighbours for pattern recognition (described in [30]) and neural networks for pattern recognition. The results obtained by applying these methods to benchmark data sets were almost as good as those produced by CPs. Undoubtedly ICPs suffer a small loss both in terms of accuracy and in terms of quality of their confidence measures; however, this loss is very small and tends to become even smaller as we go to larger data sets. In fact, for very large sets, such as the NIST and Shuttle data sets, this loss does not exist at all.

Furthermore, in the case of regression we have shown that by including

additional information, than just the error of our prediction rule

$$\alpha_i := |y_i - \hat{y}_i| \tag{15}$$

for each example i, in our nonconformity measure we can make it more precise. In [30] (for ridge regression), we have defined the nonconformity measure

$$\alpha_i := \left| \frac{y_i - \hat{y}_i}{\sigma_i} \right|, \tag{16}$$

where σ_i is an estimate of the accuracy of the decision rule f on x_i. More specifically, we take $\sigma_i := e^{\mu_i}$, where μ_i is the RR prediction of the value $\ln(|y_i - f(x_i)|)$ for the example x_i. The effect of using this nonconformity measure is that the prediction regions produced by ICP are smaller for points where the RR prediction is good and larger for points where it is bad.

Alan Hutchinson

Department of Computer Science, King's College London

The article by Gammerman and Vovk, presented to the BCS on Monday 12th June, is both novel and valuable. It outlines an approach for estimating the reliability of predictions made by machine-learning algorithms. Here are three short notes on it.

1: Intuitive interpretation The approach to learning via computability might be thought of as an attempt to discover a computable probability distribution P which seems to fit the training set well. (Professor Vovk points out that it isn't. It is designed to find the predictions which such a P might allow one to make, but it does so by means of a 'randomness test' t rather than directly through any P.)

Randomness seems to be a very strange approach. In machine learning, a seemingly random training set is the worst possible starting point. Learning is only practical if there is some non-randomness in the training set.

The answer to this quandary is that the training set should indeed have some non-random aspect, as viewed from the perspective of anyone living in ordinary space with its usual Euclidean metric and measure. The distribution P which might be learned is one according to which the training set is random. The more nearly the training data appear to be random according to P, the better P fits them. For instance, if the training set is a constant sequence $(z, z, \ldots, z)$ then the probability distribution which one might try to learn from it is the Dirac measure δ_z.

2: What is 'randomness'? The method depends on a function $t : \mathbf{Z}^* \to [0, 1]$ which is called a *randomness test*. The first condition on t is that

$$\forall \epsilon < 1 \; \forall n \; \forall P : \quad P^n \left(\{ s \in \mathbf{Z}^n : t(s) \leq \epsilon \} \right) \leq \epsilon.$$

Here, P ranges over all (computable) probability distributions on $\mathbf{Z}$. When P is the Dirac δ measure at z, this implies that

$$t(z, z, \dots, z) = 1 \text{ for any } z \in \mathbf{Z}.$$

My first reaction was that any such sequence $(z, z, \dots, z)$ appears to be as non-random as any training set could be, and perhaps t should be called a *non-randomness test*. However, this is not the right interpretation.

The point is, the condition on t is independent of any particular choice of P. According to such a test t, a sequence s should be random if there is any probability distribution P on $\mathbf{Z}$ under which s appears to be random. In this case, the constant sequence $(z, z, \dots, z)$ really is random under the distribution δ_z.

There are genuinely non-random sequences. Vovk gave the example '$101010\dots10$'.

3: Future research After the lecture by Gammerman and Vovk, I wondered if there may be learning situations in which there is a *computable* universal randomness test. In general, there are always universal randomness tests, and they are all not very different from each other, but all are only *upper semicomputable*. The class of machine-learning tasks with computable universal randomness tests may be interesting, unless it is empty.

Professor Vovk, who knows much more about it than me, says that any such machine-learning task must be exceedingly simple.

The subject can be developed in other directions, e.g., as by Peter Gács [14] and Vladimir Vovk [49].

B Rejoinder

We are very grateful to all discussants for their interest in our article and their comments. We will organize our response by major topics raised by them.

Efficiency of conformal predictors

As we say in the article, the two most important properties expected from confidence predictors are validity (they must tell the truth) and efficiency (the truth must be as informative as possible). Conformal predictors are automatically valid, so there is little to discuss here, but so far achieving efficiency has been an art, to a large degree, and Alexey Chervonenkis, Phil Long, and Sally McClean comment on this aspect of conformal prediction.

Indeed, as Prof. Chervonenkis notices, the article does not contain any theoretical results about efficiency. Such a result appears as Theorem 3.1 in our book [51]. We use a nonconformity measure based on the nearest neighbours procedure to obtain a conformal predictor whose efficiency asymptotically approaches that of the Bayes-optimal confidence predictor. (Remember that the

Bayes-optimal confidence predictor is optimized under the true probability distribution, which is unknown to Predictor.) This result only applies to the case of classification, and it is asymptotic. Nevertheless, it is our only step towards a 'more principled way of designing good measures of strangeness', as Prof. McClean puts it. Her question suggests the desirability of such more principled ways; we agree and would very much welcome further results in this direction.

An important aspect of efficiency is conditionality, discussed at length in [51] (see, e.g., p. 11). It would be ideal if we were able to learn the conditional probability distribution for the next label. Unfortunately, this is impossible under the unconstrained assumption of randomness, even in the case of binary classification ([51], Chapter 5). The definition of validity is given in terms of unconditional probability, and this appears unavoidable.

However, Prof. Chervonenkis's worry that for some objects the prediction interval might be too wide and for other too narrow has been addressed in [51]. If our objects are of several different types, the version of conformal predictors that we call 'attribute-conditional Mondrian conformal predictors' in [51] (Section 4.5) will make sure that we have separate validity for each type of objects. For example, in medical applications with patients as objects, we can always ensure separate validity for men and women.

Computational efficiency

We are concerned with two notions of efficiency in our article: efficiency in the sense of producing accurate predictions and computational efficiency (the latter being opposite to 'computational complexity', the term used by Prof. McClean). There is some scope for confusion, but the presence or absence of the adjective 'computational' always signals the intended meaning.

Harris Papadopoulos complements our brief description of inductive conformal predictors with an interesting discussion of experimental results. It was an unexpected and pleasing finding that the computationally efficient inductive conformal predictors do not suffer accuracy loss for even moderately large data sets. His two nonconformity measures for ridge regression, (15) and (16), illustrate the general fact that different nonconformity measures can involve different degrees of tuning to the data. Another finding of [30] and [31] was that more tuning (as in Equation (16), as compared to (15)) does not necessarily mean better accuracy: it can lead to overfitting when the available data are scarce.

Interpretation and packaging

The question of interpretation of p-values is a difficult one. In general, p-values are the values taken by a randomness test (they were also called 'the randomness level detected by a randomness test' in Section 2). They are not probabilities and we believe should not be criticized for not being probabilities; they satisfy condition (3) and this makes them valuable tools of prediction. They allow us to make probabilistic statements (such as 'at confidence level $1 - \epsilon$, smoothed

conformal predictors used in the on-line mode make mistakes with probability ϵ, independently for different examples').

Many of David Dowe's criticisms just remind us that a p-value, as well as a confidence, in our sense, is not a probability. He says that 'for non-binary problems, this confidence seems too large', with an argument endowing p-values with a property of probabilities (they are assumed to add to one). The fact that the three lines in Figure 6 have slightly larger gradients than the corresponding significance levels is accidental and not statistically significant. After all, we have a theorem (Theorem 1 on p. 18) that guarantees validity; the deviations are well within the double standard deviation of the number of errors. (To facilitate the comparison, the actual numbers of errors at the confidence levels 80%, 95% and 99% are 1873, 470 and 107, respectively; the expected numbers of errors are 1859.6, 464.9 and 92.98, respectively; the standard deviations are 38.57, 21.02 and 9.59, respectively. In this experiment the MATLAB generator of pseudo-random numbers was initialized to 0.) We could not report the log-loss scores for Figures 6 and 7 because the methods described in our article do not produce probability forecasts.

The problem of valid and efficient probabilistic prediction is considered in our book ([51], Chapters 6 and 9). We show that the 'Venn predictors' that we construct are automatically valid, but the notion of validity for probabilistic predictors is much subtler than that for confidence predictors in the practically interesting case of finite data sequences. (In the idealized case of infinite data sequences the asymptotic notion of validity is quite simple, and asymptotically valid probabilistic predictors are known as well-calibrated predictors.) Unfortunately, it was impossible to include this material in our talk and article.

To finish our reply to Dr Dowe's contribution, the randomness test used in Table 1 is given by formula (5) with the α_i computed using the support vector method with the polynomial kernel of degree 5 (as we say in the text); in Section 4 the randomness test is the one implemented by the ridge regression confidence machine (as we say both in the text and in the figure captions).

As Xiaohui Liu points out, a key issue for hedged prediction is how to assist users with the interpretation and utilization of our measures of confidence. The full information about the uncertainty in the value of the label to be observed, as given by a conformal predictor, is provided by the full set of p-values p_Y, $Y \in \mathbf{Y}$. Even in the case of classification, this set has to be somehow summarized when the set $\mathbf{Y}$ of potential labels is large. Our preferred way of summarizing the set $\{p_Y : Y \in \mathbf{Y}\}$ is to report two numbers: the confidence (defined by (8) or, equivalently, as one minus the second largest p-value) and credibility (9) (equivalently, the largest p-value). Prof. Chervonenkis suggests replacing confidence with the difference between the largest and second largest p-values. In combination with credibility this carries the same information as our suggestion. The pair (confidence, credibility) still appears to us simpler and more intuitive, but we believe that this is a matter of taste.

What is randomness?

To motivate the definition of conformal predictors we start the article from the notion of randomness. Alan Hutchinson's comments give us an opportunity to discuss further terminological and philosophical issues surrounding this notion.

The word 'random' is loaded with a plethora of different meanings. Several years ago we even tried to avoid it altogether in our lectures and articles, using 'typical' instead. But the noun 'typicalness' was so awkward and both 'random' and 'randomness' so well established that we reverted to the old usage. Kolmogorov, who started the modern stage of the theory of randomness, was only interested in randomness with respect to the uniform distribution on a finite set, and in this case the word 'random' (as well as its Russian counterpart 'случайный') matches the common usage perfectly. Later on his followers started generalizing Kolmogorov's concept to arbitrary probability measures and statistical models; although the mismatch between the technical and ordinary senses of the word 'random' became apparent, the term was not changed.

We think that Part 1 of Mr Hutchinson's contribution is very well illustrated by Dr Long's aphoristic statement that 'in many applications, one expects randomness between examples and structure within them'. A 'seemingly random training set' is a bad starting point if there is too much randomness within examples, but randomness between examples helps: it enables us to make provably valid stochastic statements about the future. Another point we would like to emphasize is that we do not have to learn the true probability distribution P to make good predictions (as repeatedly pointed out by Vladimir Vapnik in [42] and [43]); in fact, conformal predictors, despite producing reasonable predictions, do not provide us with any information about P.

As Mr Hutchinson says, our initial reaction to his idea of a computable universal randomness test was that such a test is unlikely to exist except in very simple and uninteresting cases. This impression was based on our experience so far (for a given computable test it is usually easy to find another computable test that is much more powerful on some data). However, our experience only covers a small part of machine learning, and it is by no means our intention to discourage research in this direction.

Philosophy

Prof. Vapnik asks our opinion about philosophical aspects of transductive inference. To a large degree, we are his pupils on this subject (the reader can consult his books [42, 43] and the afterword to the second English edition of his classic [41]). It appears that the role of transduction is constantly increasing. The muddle-headed transduction, to borrow David Bell's delightful metaphor, is obviously the right way of reasoning in the complex social world surrounding us. But even in physics, the traditional abode of the most general and precise rules (physical theories), pure induction encounters serious difficulties: we have two very general sets of rules, quantum mechanics and general relativity, but they contradict each other. Induction appears to be becoming subordinate to

transduction; for example, as in this article, induction might make transduction more computationally efficient.

At this point it is useful to remind the reader that this article always makes the assumption of randomness. The general ideas such as induction and transduction become incomparably more manageable. This is a very simple-minded world: the usual philosophical picture of constant creation of and struggle between scientific theories (e.g., [22], [32]) becomes irrelevant. But we have to start somewhere.

As Prof. Bell can see, despite our interest in transduction, our article is still very much simple-minded. In its current embryonic state all rigorous machine learning has to be such, and it is likely to stay this way for some time. The only thing we can hope to do now is to nick a few interesting topics here and there from more muddle-headed areas such as experimental AI or philosophy, and try to prove something about them.

Predecessors of conformal prediction

This topic was raised by Glenn Shafer. Of course, the vast majority of our comments are not new to him, and they are mostly addressed to people who are not experts in this field. Indeed, our work is closely connected to that of Kei Takeuchi and his predecessors mentioned by Prof. Shafer: Sam Wilks, who introduced in 1941 the notion of tolerance regions, Abraham Wald, who in 1943 extended Wilks's idea to the multidimensional case, and John Tukey, Donald Fraser, John Kemperman (and many other researchers), who in the 1940s and 1950s contributed to generalizing Wald's idea.

From the very beginning of the theory there were two versions of tolerance regions, which we might call inductive (involving two parameters, denoted ϵ and δ in our article) and transductive (involving only one parameter). We will be discussing only the latter version.

Let $\epsilon > 0$. A function S^ϵ mapping each training set to a subset of the example space $\mathbf{Z}$ is called a *conservative ϵ-tolerance predictor* if the probability of the event

$$z_{l+1} \in S^\epsilon(z_1, \dots, z_l)$$

is at least $1 - \epsilon$ (for all sizes l and for independent and identically distributed examples $z_1, \dots, z_{l+1}$). In practice one usually considers systems of conservative ϵ-tolerance predictors S^ϵ, $\epsilon \in (0, 1)$, which are nested: $S^{\epsilon_1} \subseteq S^{\epsilon_2}$ when $\epsilon_1 \geq \epsilon_2$. For brevity, we will refer to such systems of conservative ϵ-tolerance predictors as *tolerance predictors*.

The parallel between tolerance predictors and valid confidence predictors is obvious. For example, given a tolerance predictor S we can define a valid confidence predictor Γ by the formula

$$\Gamma^\epsilon(x_1, y_1, \dots, x_l, y_l, x_{l+1}) := \{Y \in \mathbf{Y} : (x_{l+1}, Y) \in S^\epsilon(x_1, y_1, \dots, x_l, y_l)\}.$$

So what do the conformal predictors contribute to the theory of tolerance regions?

The most important contribution of conformal prediction is perhaps the general definition of nonconformity measures. In our book ([51], p. 257) we describe a version of an important procedure due to Tukey for computing nonconformity scores (using our terminology). However, it appears to us that Tukey's procedure (and its predecessors due to Wilks, Wald, and several other researchers) can be used efficiently only in the case of traditional low-dimensional statistical data sets, and to process data sets that are common in machine learning one needs the general definition, as given in this article. An important advance towards the general definition of nonconformity measures was made by Takeuchi in the recently found manuscript [39], a hand-out for his lecture at Stanford University in 1979. According to the information we have been able to gather after Prof. Shafer's talk at the discussion of our article, the chronology of events seems to be slightly different from his description. The Stanford lectures (or lecture) happened in the late rather than early 1970s (namely, in July 1979), after the publication of [38] in 1975. To our knowledge, Takeuchi's idea of nonconformity measures for multi-dimensional tolerance regions has never been published, even in Japanese. We are lucky to have the three-page handwritten manuscript [39]. Takeuchi's definition of nonconformity is rather narrow (based on parameter estimation), and he does not state it formally; he gives only one example of its use in a multi-dimensional situation. However, there is little doubt that if Takeuchi had continued work in this direction, he would have arrived at the general definition.

For a much fuller historical account, including our predecessors in machine learning (but not including [39], which was found only in July 2006), see [51], especially Section 10.2.

Applications in medicine and biology

Zhiyuan Luo and Tony Bellotti describe in detail the use of conformal predictors in medical applications; we have little to add to their very clear description. Medicine appears to be an especially suitable field for this technique. Consider, for example, the problem of automated screening for a serious disease. We would like to declare a person clean of the disease only if we are confident that he or she really is; if we are not, the test results should be passed on to a human doctor. The guaranteed validity of automated screening systems based on conformal prediction is obviously of great value; even if such a system is badly designed, this will be reflected in its efficiency (extra work for human doctors), but the patients can be assured that validity will never suffer. This guarantee depends, of course, on the assumption of randomness being satisfied, but in this particular application it appears reasonable.

In biological applications, the most natural use of conformal prediction is to filter out, e.g., uninteresting genes. Prof. Liu discusses the difficult problem of setting thresholds for deciding when a gene should be passed on to a biologist for a further analysis. There might not be universally applicable principles for making such decisions. The whole process of analysis might involve several iterations, with the thresholds lowered or raised depending on the results

obtained.

Assumptions

Prof. Shafer eloquently points out the narrowness of the assumption of randomness (called the i.i.d. assumption by several discussants). We agree that it is rather narrow (and one of us has been concerned since the late 1980s with prediction free of any stochastic assumptions: see, e.g., [46], [47]), but we will start from its defence.

The assumption of randomness is non-parametric. No assumptions whatsoever are made about the probability distribution generating each example. In many situations this assumption is close to being satisfied; think, e.g., of a sequence of zip codes passing through a given post office (over a period of time that is not too long). It is an interesting and widely applicable assumption.

Besides, it is clear that some stochastic assumption is needed in order to obtain valid stochastic measures of confidence. Taking into account the strength of guarantees that can be derived, we find the assumption surprisingly weak. In Chapter 8 of [51] we further generalize the method of conformal prediction to cover a wide range of 'on-line compression models', and in Section 8.6 we derive conformal predictors for the Markov model (cf. numbers 2 and 3 on Prof. Shafer's list).

It can be counted as a disadvantage of conformal prediction that it depends *heavily* on the assumption of randomness. Our discussion will be general, but we will couch it, for concreteness, in terms of support vector machines. The support vector method can also be said to depend on the assumption of randomness: the theorems about support vector machines obtained in [42]–[43] always make this assumption. What is important in typical applications, however, is not the theorems but the predictions themselves, which are more precise for support vector machines than for many other methods. Support vector machines can always be applied and the results will be useful unless the assumption is violated dramatically. Of course, conformal predictors can also be always applied, but the measures of confidence are an integral part of their predictions, and the validity of these measures is much more sensitive to violations of the assumption of randomness (or assumptions expressed by other on-line compression models).

Drago Indjic raises the question of applying confidence and credibility to active experimental design. In the limited framework of this article, the objects x_i, being components of the i.i.d. examples, are themselves i.i.d. Active experimentation destroys this property. If this article's approach were followed, one would need relatively long sequences of i.i.d. examples between active interventions, and this appears wasteful. Combining active experimental design with confidence and credibility without waste would require developing a suitable on-line compression model, perhaps a version of the Gauss linear model ([51], Section 8.5).

The topic of experimental design is continued by Glenn Hawe. The analogy between two-stage/one-stage varieties of cost-effective optimization and induction/transduction is striking, but implementing his idea will again require a

different on-line compression model. The assumption of randomness, so central in our article, is quite different from the assumption of 'low bumpiness'. Finding a suitable on-line compression model might not be easy, but it is definitely worth pursuing.

Dr Long's idea of using conformal prediction in reinforcement learning also requires another on-line compression model. A good deal of further work is still needed.

This brings us back to the limitations of the assumption of randomness. It makes many applications (such as active experimental design and reinforcement learning) problematic. The assumption can be weakened or modified (see [51] for numerous examples), but it is always good to have at our disposal methods of prediction that do not depend on any stochastic assumptions. As Prof. Shafer says, such probability-free methods are being actively explored in prediction with expert advice (also known as 'universal prediction of individual sequences' and 'competitive on-line prediction'), with some recent breakthroughs. In many applications (such as typical medical applications) the assumption of randomness is convincing and the measures of confidence provided by conformal predictors are really needed. In other areas, particularly those in which no human intervention is envisaged, conformal prediction is less useful, and if, additionally, the assumption of randomness is violated, the case for prediction with expert advice becomes very strong.

Acknowledgements

We are grateful to Akimichi Takemura for sharing [39] with us.

References

[1] Peter Auer. Using confidence bounds for exploitation-exploration tradeoffs. *Journal of Machine Learning Research*, 3:397–422, 2002.

[2] John S. Bell. *Speakable and Unspeakable in Quantum Mechanics*. Cambridge University Press, Cambridge, 1987. See p. 27.

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

[4] Ulisses M. Braga-Neto and Edward R. Dougherty. Is cross-validation valid for small-sample microarray classification? *Bioinformatics*, 20:374–380, 2004.

[5] Bernard Bru. The Bernoulli code. *Electronic Journal for History of Probability and Statistics*, 2(1), June 2006. Available on-line at `http://www.jehps.net`.

[6] Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, Cambridge, England, 2006.

[7] Joshua W. Comley and David L. Dowe. General Bayesian networks and asymmetric languages. In *Proceedings of the Hawaii International Conference on Statistics and Related Fields*, June 2003.

[8] Joshua W. Comley and David L. Dowe. Minimum message length and generalized Bayesian nets with asymmetric languages. In Peter Grünwald, Mark A. Pitt, and In Jae Myung, editors, *Advances in Minimum Description Length: Theory and Applications*, pages 265–294. MIT Press, 2005.

[9] David R. Cox and David V. Hinkley. *Theoretical Statistics*. Chapman and Hall, London, 1974.

[10] A. Philip Dawid. Probability forecasting. In Samuel Kotz, Norman L. Johnson, and Campbell B. Read, editors, *Encyclopedia of Statistical Sciences*, volume 7, pages 210–218. Wiley, New York, 1986.

[11] A. Philip Dawid. Discussion of the papers by Rissanen and by Wallace and Dowe. *Computer Journal*, 42(4):323–326, 2000.

[12] Arthur P. Dempster. An overview of multivariate data analysis. *Journal of Multivariate Analysis*, 1:316–346, 1971.

[13] David L. Dowe and Chris S. Wallace. Kolmogorov complexity, minimum message length and inverse learning. In *Proceedings of the Fourteenth Australian Statistical Conference*, page 144, 1998.

[14] Peter Gács. Uniform test of algorithmic randomness over a general space. *Theoretical Computer Science*, 341:91–137, 2005.

[15] Alexander Gammerman and A. R. Thatcher. Bayesian diagnostic probabilities without assuming independence of symptoms. *Methods of Information in Medicine*, 30:15–22, 1991.

[16] Murray Gell-Mann. *The Quark and the Jaguar*. W. H. Freeman, 1994. See p. 34.

[17] Hans-Martin Gutmann. A radial basis function method for global optimization. *Journal of Global Optimization*, 19:201–227, 2001.

[18] David J. Hand. Classifier technology and the illusion of progress (with discussion). *Statistical Science*, 21:1–14, 2006.

[19] Donald R. Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 21:345–383, 2001.

[20] Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13:455–492, 1998.

[21] Kevin B. Korb. Calibration and the evaluation of predictive learners. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 73–77, 1999.

[22] Thomas S. Kuhn. *The Structure of Scientific Revolutions*. University of Chicago Press, Chicago, 1962. Third edition: 1996.

[23] Yann LeCun, Bernhard E. Boser, John S. Denker, Donnie Henderson, R. E. Howard, Wayne E. Hubbard, and Lawrence D. Jackel. Handwritten digit recognition with a back-propagation network. In David S. Touretzky, editor, *Advances in Neural Information Processing Systems 2*, pages 396–404. Morgan Kaufmann, San Mateo, CA, 1990.

[24] Ming Li and Paul Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer, New York, second edition, 1997.

[25] Per Martin-Löf. The definition of random sequences. *Information and Control*, 9:602–619, 1966.

[26] Thomas Melluish, Craig Saunders, Ilia Nouretdinov, and Vladimir Vovk. Comparing the Bayes and typicalness frameworks. In Luc De Raedt and Peter A. Flach, editors, *Proceedings of the Twelfth European Conference on Machine Learning*, volume 2167 of *Lecture Notes in Computer Science*, pages 360–371, Heidelberg, 2001. Springer.

[27] John S. Mill. *A System of Logic*. 1843. See p. 130.

[28] Ilia Nouretdinov, Tom Melluish, and Vladimir Vovk. Ridge Regression Confidence Machine. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 385–392, San Francisco, CA, 2001. Morgan Kaufmann.

[29] Ilia Nouretdinov and Vladimir Vovk. Criterion of calibration for transductive confidence machine with limited feedback. *Theoretical Computer Science*, 364:3–9, 2006. Special issue devoted to the ALT'2003 conference.

[30] Harris Papadopoulos, Konstantinos Proedrou, Vladimir Vovk, and Alexander Gammerman. Inductive Confidence Machines for regression. In Tapio Elomaa, Heikki Mannila, and Hannu Toivonen, editors, *Proceedings of the Thirteenth European Conference on Machine Learning*, volume 2430 of *Lecture Notes in Computer Science*, pages 345–356, Berlin, 2002. Springer.

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

[32] Karl R. Popper. *Logik der Forschung*. Springer, Vienna, 1934. An English translation, *The Logic of Scientific Discovery*, was published by Hutchinson, London, in 1959.

[33] Daniil Ryabko, Vladimir Vovk, and Alexander Gammerman. Online prediction with real teachers. Technical Report CS-TR-03-09, Department of Computer Science, Royal Holloway, University of London, 2003.

[34] Glenn Shafer. The unity and diversity of probability. *Statistical Science*, 5:435–444, 1990.

[35] Ilham A. Shahmuradov, Viktor V. Solovyev, and Alexander Gammerman. Plant promoter prediction with confidence estimation. *Nucleic Acids Research*, 33:1069–1076, 2005.

[36] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

[37] Stephen Swift, Allan Tucker, Veronica Vinciotti, Nigel Martin, Christine Orengo, Xiaohui Liu, and Paul Kellam. Consensus clustering and functional interpretation of gene expression data. *Genome Biology*, 5:R94, 2004.

[38] Kei Takeuchi. 統計的予測論 *(Statistical Prediction Theory)*. Baihūkan, Tokyo, 1975.

[39] Kei Takeuchi. Non-parametric prediction regions. Hand-out for a lecture at Stanford University (3 pages), 17 July 1979.

[40] Peter J. Tan and David L. Dowe. MML inference of oblique decision trees. In *Proceedings of the Seventeenth Australian Joint Conference on Artificial Intelligence*, volume 3339 of *Lecture Notes in Artificial Intelligence*, pages 1082–1088. Springer, 2004.

[41] Vladimir N. Vapnik. Оценивание зависимостей по эмпирическим данным. Nauka, Moscow, 1979. English translation: Springer, New York, 1982. Second English edition: *Estimation of Dependences Based on Empirical Data: Empirical Inference Science*. Information Science and Statistics. Springer, New York, 2006.

[42] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 1995. Second edition: 2000.

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

[44] 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.

[45] Veronica Vinciotti, Allan Tucker, Paul Kellam, and Xiaohui Liu. The robust selection of predictive genes via a simple classifier. *Applied Bioinformatics*, 5:1–12, 2006.

[46] Vladimir Vovk. Aggregating strategies. In Mark Fulk and John Case, editors, *Proceedings of the Third Annual Workshop on Computational Learning Theory*, pages 371–383, San Mateo, CA, 1990. Morgan Kaufmann.

[47] Vladimir Vovk. Competitive on-line statistics. *International Statistical Review*, 69:213–248, 2001.

[48] 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.

[49] Vladimir Vovk. Predictions as statements and decisions. In Gábor Lugosi and Hans Ulrich Simon, editors, *Proceedings of the Nineteenth Annual Conference on Learning Theory*, volume 4005 of *Lecture Notes in Artificial Intelligence.*, page page 4, Berlin, 2006. Springer. Full version: Technical Report `arXiv:cs.LG/0606093`, `arXiv.org` e-Print archive, June 2006.

[50] Vladimir Vovk, Alexander Gammerman, and Craig Saunders. Machine-learning applications of algorithmic randomness. In *Proceedings of the Sixteenth International Conference on Machine Learning*, pages 444–453, San Francisco, CA, 1999. Morgan Kaufmann.

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

[52] Chris S. Wallace. *Statistical and Inductive Inference by Minimum Message Length*. Information Science and Statistics. Springer, New York, 2005.

[53] Chris S. Wallace and David M. Boulton. An information measure for classification. *Computer Journal*, 11:185–195, 1968.

[54] Chris S. Wallace and David M. Boulton. An invariant Bayes method for point estimation. *Classification Society Bulletin*, 3(3):11–34, 1975.

[55] Chris S. Wallace and David L. Dowe. Minimum message length and Kolmogorov complexity. *Computer Journal*, 42:270–283, 1999.

[56] Juyang Weng. Muddy tasks and the necessity of autonomous mental development. In *Proceedings of the 2005 AAAI Spring Symposium Series, Developmental Robotics Symposium, Stanford University*, 2005.