Self-calibrating Probability Forecasting

Vladimir Vovk, Glenn Shafer and Ilia Nouretdinov

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

On-line Compression Modelling Project

Working Paper #9

7 June, 2003

Project web site:
http://vovk.net/kp

Abstract

The problem of probability forecasting is an extension of the standard classification problem; the latter's goal of finding the "best" label for the test object is replaced by the goal of finding conditional probabilities, given the test object, for possible values of the test object's label. We introduce a new class of algorithms for on-line probability forecasting which we call "Venn probability machines" and study under the assumption of randomness (the object/label pairs are independent and identically distributed). The most important advantage of these algorithms is that they are automatically well-calibrated in a strong non-asymptotic sense. Our experimental results demonstrate that a 1-Nearest Neighbour Venn probability machine performs reasonably well on a standard benchmark data set, and one of our theoretical results asserts that a simple Venn probability machine asymptotically approaches the true conditional probabilities regardless, and without knowledge, of the true probability distribution generating the examples.

Contents

1 Introduction

We are interested in the on-line version of the problem of probability forecasting: we observe pairs of objects and labels sequentially, and after observing the nth object x_n the goal is to give a probability distribution p_n for its label; as soon as p_n is output, the label y_n of x_n is disclosed and can be used for computing future probability forecasts. A good review of early work in this area is Dawid [2]. In this introductory section we will assume that $y_n \in \{0, 1\}$; we can then take p_n to be a real number from the interval $[0, 1]$ (the probability that $y_n = 1$ given x_n); our exposition here will be very informal.

The standard view ([2], pp. 213–216) is that the quality of probability forecasting systems has two components: "reliability" and "resolution". At the crudest level, reliability requires that the forecasting system should not lie, and resolution requires that it should say something useful. To be slightly more precise, consider the first n forecasts p_i and the actual labels y_i.

The most basic test is to compare the overall average forecast probability $\overline{p}_n := n^{-1} \sum_{i=1}^{n} p_i$ with the overall relative frequency $\overline{y}_n := n^{-1} \sum_{i=1}^{n} y_i$ of 1s among y_i. If $\overline{p}_n \approx \overline{y}_n$, the forecasts are "unbiased in the large".

A more refined test would look at the subset of i for which p_i is close to a given value p^*, and compare the relative frequency of $y_i = 1$ in this subset, say $\overline{y}_n(p^*)$, with p^*. If

$$\overline{y}_n(p^*) \approx p^* \text{ for all } p^*, \tag{1}$$

the forecasts are "unbiased in the small", "reliable", "valid", or "well-calibrated"; in later sections, we will use "well-calibrated", or just "calibrated", as a technical term. Forecasting systems that pass this test at least get the frequencies right; in this sense they do not lie.

It is easy to see that there are reliable forecasting systems that are virtually useless. For example, the definition of reliability does not require that the forecasting system pay any attention to the objects x_i. In another popular example, the labels follow the pattern

$$y_i = \begin{cases} 1 \text{ if } i \text{ is odd} \\ 0 \text{ otherwise.} \end{cases}$$

The forecasts $p_i = 0.5$ are reliable, at least asymptotically (0.5 is the right relative frequency) but not as useful as $p_1 = 1, p_2 = 0, \ldots$; the "resolution" (which we do not define here) of the latter forecasts is better.

In this paper we construct forecasting systems that are automatically reliable. To achieve this, we allow our prediction algorithms to output sets

of probability distributions P_n instead of single distributions p_n; typically the sets P_n will be small (see §5).

This paper develops the approach of [9], [10] and [12], which show that it is possible to produce valid, asymptotically optimal, and practically useful p-values; the p-values can be then used for region prediction. Disadvantages of p-values, however, are that their interpretation is less direct than that of probabilities and that they are easy to confuse with probabilities; some authors have even objected to any use of p-values (see, e.g., [1]). In this paper we use the methodology developed in the previous papers to produce valid probabilities rather than p-values.

2 Probability forecasting and calibration

From this section we start rigorous exposition. Let $\mathcal{P}(\mathbf{Y})$ be the set of all probability distributions in a measurable space $\mathbf{Y}$. We use the following protocol in this paper:

MULTIPROBABILITY FORECASTING
Players: Reality, Forecaster
Protocol:
FOR $n = 1, 2, \ldots$:
Reality announces $x_n \in \mathbf{X}$.
Forecaster announces $P_n \subseteq \mathcal{P}(\mathbf{Y})$.
Reality announces $y_n \in \mathbf{Y}$.

In this protocol, Reality generates *examples* $z_n = (x_n, y_n)$ consisting of two parts, *objects* x_n and *labels* y_n. After seeing the object x_n Forecaster is required to output a prediction for the label y_n. The usual probability forecasting protocol requires that Forecaster output a probability distribution; we relax this requirement by allowing him to output a family of probability distributions (and we are interested in the case where the families P_n become smaller and smaller as n grows).

In this paper we make the simplifying assumption that the label space $\mathbf{Y}$ is finite; in many informal explanations it will be assumed binary, $\mathbf{Y} = \{0, 1\}$. To avoid unnecessary technicalities, we will also assume that the families P_n chosen by Forecaster are finite and have no more than K elements; they will be represented by a list of length K (elements in the list can repeat). A *probability machine* is a measurable strategy for Forecaster in our protocol,

where at each step he is required to output a sequence of K probability distributions.

The problem of calibration is usually treated in an asymptotic framework. Typical asymptotic results, however, do not say anything about finite data sequences; therefore, in this paper we will only be interested in the non-asymptotic notion of calibration.

The *n-space* Π_n is the set of all sequences $p_1 y_1 \dots p_n y_n$ such that $p_i \in \mathcal{P}(\mathbf{Y})$ and $y_i \in \mathbf{Y}$, $i = 1, \dots, n$. Intuitively, such sequences $p_1 y_1 \dots p_n y_n$ arise when a sequence $y_1 \dots y_n$ is predicted using a probability distribution: p_i is the conditional distribution for y_i given all the previous examples $z_1 \dots z_{i-1}$ and the new object x_i. An *n-event* is a measurable subset of Π_n. The *capacity* $C(E)$ of an n-event E is defined as the supremum of $C_R(E)$ over all probability distributions R in $\mathbf{Z}^n$ and all choices of regular conditional distributions under R, where $C_R(E)$ is the R-probability that $p_1 y_1 \dots p_n y_n$ will belong to E, p_i being the chosen regular conditional distribution of y_i under R given $z_1, \dots, z_{i-1}$ and x_i. Intuitively, the smallness of $C(E)$ means that we do not expect E to happen if p_i are conditional probabilities.

A *calibration n-event* is an n-event E that is invariant w.r. to permutations: if

$$(p_1 y_1 \dots p_n y_n) \in E,$$

then

$$(p_{\pi(1)} y_{\pi(1)} \dots p_{\pi(n)} y_{\pi(n)}) \in E$$

for any permutation $\pi : \{1, \dots, n\} \to \{1, \dots, n\}$. (This definition is motivated by the fact that the requirement (1) is invariant w.r. to permutations. Effectively, a calibration n-event is a set of n-bags[1] of elements of $\mathcal{P}(\mathbf{Y}) \times \mathbf{Y}$.) Intuitively, miscalibration is evidenced by the happening of a calibration n-event with small $C(E)$ (for an example of such an event, see (2) below).

An *n-multievent* is a measurable set of sequences $P_1 y_1 \dots P_n y_n$, where each P_i is a list of K probability distributions in $\mathbf{Y}$ and $y_i \in \mathbf{Y}$, $i = 1, \dots, n$, which is invariant under permutations of each P_i (we are not interested in the order in which the elements of P_i are listed). The *capacity* $C(E)$ of an

[1]By "n-bag" we mean a collection of n elements, not necessarily distinct. "Bag" and "multiset" are synonymous, but we prefer the former term in order not to overload the prefix "multi".

n-multievent E is

$$C\left(\bigcup_{\omega\in\{1,\dots,K\}^n} E_\omega\right),$$

where: E_ω is the set of all sequences $P_1(\omega_1)y_1\dots P_n(\omega_n)y_n$; $P_1y_1\dots P_ny_n$ ranges over the sequences in E; and $P_i(\omega_i)$ stands for the ω_ith component of the sequence P_i. An n-multievent is a *calibration n-multievent* if it is invariant w.r. to permutations, in the same sense as for calibration n-events.

We say that a probability machine is *finitarily calibrated* if, for any n, any probability distribution q in $\mathbf{Z}$, and any calibration n-multievent E, the q^n-probability that the sequence $P_1y_1\dots P_ny_n$ of Forecaster and Reality's moves belongs to E never exceeds $C(E)$. Intuitively, a probability machine is finitarily calibrated if we can treat the multiprobabilities it outputs as containing the genuine conditional probabilities, as far as calibration is concerned.

Remark To make our finitary notion of calibration clearer, we will give a simple example of a calibration n-event and its modification providing a simple example of a calibration n-multievent, with upper bounds on their capacity. For any $\epsilon > 0$ and n, the capacity of the calibration n-event

$$\frac{1}{n}\sum_{i=1}^{n}(x_i - p_i) \geq \epsilon \tag{2}$$

does not exceed $e^{-2\epsilon^2 n}$ (by the Hoeffding-Azuma inequality [3]); this implies that the capacity of the calibration n-multievent

$$\frac{1}{n}\sum_{i=1}^{n}\inf_{p\in P_i}(x_i - p) \geq \epsilon$$

does not exceed $e^{-2\epsilon^2 n}$.

3 Self-calibrating probability forecasting

Now we will describe a general algorithm for multiprobability forecasting. Let $\mathbb{N}$ be the sets of all positive integer numbers. A sequence of measurable functions $A_n : \mathbf{Z}^n \to \mathbb{N}^n$, $n = 1, 2, \dots$, is called a *taxonomy* if, for any

$n \in \mathbb{N}$, any permutation π of $\{1, \ldots, n\}$, any $(z_1, \ldots, z_n) \in \mathbf{Z}^n$, and any $(\alpha_1, \ldots, \alpha_n) \in \mathbb{N}^n$,

$$(\alpha_1, \ldots, \alpha_n) = A_n(z_1, \ldots, z_n) \Longrightarrow (\alpha_{\pi(1)}, \ldots, \alpha_{\pi(n)}) = A_n(z_{\pi(1)}, \ldots, z_{\pi(n)}).$$

In other words,

$$A_n : (z_1, \ldots, z_n) \mapsto (\alpha_1, \ldots, \alpha_n) \tag{3}$$

is a taxonomy if every α_i is determined by the bag $\wr z_1, \ldots, z_n \int$ and z_i. The *Venn probability machine associated with* (A_n) is the probability machine which outputs the following $K = |\mathbf{Y}|$ probability distributions p_y, $y \in \mathbf{Y}$, at the nth step: complement the new object x_n by the postulated label y; consider the division of $\wr z_1, \ldots, z_n \int$, where z_n is understood (only for the purpose of this definition) to be (x_n, y), into groups (or *types*) according to the values of A_n (i.e., z_i and z_j are assigned to the same group if and only if $\alpha_i = \alpha_j$, where the αs are defined by (3)); find the empirical distribution p_y of the labels in the group containing the nth example $z_n = (x_n, y)$. A *Venn probability machine* (VPM) is the Venn probability machine associated with some taxonomy.

Theorem 1 *Any Venn probability machine is finitarily calibrated.*

It is clear that VPM depends on the taxonomy only through the way it splits the examples $z_1, \ldots, z_n$ into groups; therefore, we may specify only the latter when constructing specific VPMs.

Remark The notion of VPM is a version of Transductive Confidence Machine (TCM) introduced in [11] and [7], and Theorem 1 is a version of Theorem 1 in [9]; as we already mentioned, the main difference between VPM and TCM is that the former replaces the p-values used by TCM with probabilities. Paper [9] shows that the conclusion of its Theorem 1 is also true for a computationally efficient modification of TCM, Inductive Confidence Machine. An inductive version of VPM can also be defined and Theorem 1 can be extended to cover this version.

4 Discussion of the Venn probability machine

In this somewhat informal section we will discuss the intuitions behind VPM, considering only the binary case $\mathbf{Y} = \{0, 1\}$. We start with the almost trivial

Bernoulli case, where the objects x_i are absent,[2] and our goal is to predict, at each step $n = 1, 2, \ldots$, the new label y_n given the previous labels $y_1, \ldots, y_{n-1}$. The most naive probability forecast is $p_n = k/(n-1)$, where k is the number of 1s among the first $n-1$ labels. (Often "regularized" forms of $k/(n-1)$, such as Laplace's rule of succession $(k+1)/(n+1)$, are used.)

In the Bernoulli case there is only one natural VPM: the multiprobability forecast for y_n is $\{k/n, (k+1)/n\}$. Indeed, since there are no objects x_n, it is natural to take the one-element taxonomy A_n at each step, and this produces the VPM $p_n = \{k/n, (k+1)/n\}$. It is clear that the diameter $1/n$ of P_n for this VPM is the smallest achievable.

Now let us consider the case where x_n are present. The probability forecast $k/(n-1)$ for y_n will usually be too crude, since the known population $z_1, \ldots, z_{n-1}$ may be very heterogeneous. A reasonable statistical forecast would take into account only objects x_i that are similar, in a suitable sense, to x_n. A simple modification of the Bernoulli forecast $k/(n-1)$ is as follows:

1. Split the available objects $x_1, \ldots, x_n$ into a number of types.
2. Output k'/n' as the predicted probability that $y_n = 1$, where n' is the number of objects among $x_1, \ldots, x_{n-1}$ of the same type as x_n and k' is the number of objects among those n' that are labelled as 1.

At the first stage, a delicate balance has to be struck between two contradictory goals: the types should be as large as possible (to have a reasonable sample size for estimating probabilities); the types should be as homogeneous as possible. This problem is sometimes referred to as the "reference class problem"; according to Kılınç [4], John Venn was the first to formulate and analyse this problem with due philosophical depth.

The procedure offered in this paper is a simple modification of the standard procedure described in the previous paragraph:

0. Consider the two possible completions of the known data

$$(z_1, \ldots, z_{n-1}, x_n) = ((x_1, y_1), \ldots, (x_{n-1}, y_{n-1}), x_n) :$$

in one (called the 0-*completion*) x_n is assigned label 0, and in the other (called the 1-*completion*) x_n is assigned label 1.

[2]Formally, this correspond in our protocol to the situation where $|\mathbf{X}| = 1$, and so x_n, although nominally present, do not carry any information.

1. In each completion, split all examples $z_1, \ldots, z_{n-1}, (x_n, y)$ into a number of types, so that the split does not depend on the order of examples ($y = 0$ for the 0-partition and $y = 1$ for the 1-partition).

2. In each completion, output k'/n' as the predicted probability that $y_n = 1$, where n' is the number of examples among $z_1, \ldots, z_{n-1}, (x_n, y)$ of the same type as (x_n, y) and k' is the number of examples among those n' that are labelled as 1.

In this way, we will have not one but two predicted probabilities that $y_n = 1$; but in practically interesting cases we can hope that these probabilities will be close to each other (see the next section).

Venn's reference class problem reappears in our procedure as the problem of avoiding over- and underfitting. A taxonomy with too many types means overfitting; it is punished by the large diameter of the multiprobability forecast (importantly, this is *visible*, unlike the standard approaches). Too few types means underfitting (and poor resolution).

Important advantages of our procedure over the naive procedure are: our procedure is self-calibrating; there exists an asymptotically optimal VPM (see §6); we can use labels in splitting examples into types (this will be used in the next section).

5 Experiments

In this section, we will report the results for a natural taxonomy applied to the well-known USPS data set of hand-written digits; this taxonomy is inspired by the 1-Nearest Neighbour algorithm. First we describe the taxonomy, and then the way in which we report the results for the VPM associated with that taxonomy.

Since the data set is relatively small (9298 examples in total), we have to use a crude taxonomy: two examples are of the same type if their nearest neighbours have the same label; therefore, the taxonomy consists of 10 types. The distance between two examples is defined as the Euclidean distance between their objects (which are 16×16 matrices of pixels and represented as points in $\mathbb{R}^{256}$).

The algorithm processes the nth object x_n as follows. First it creates the 10×10 matrix A whose entry $A_{i,j}$, $i, j = 0, \ldots, 9$, is computed by assigning i to x_n as label and finding the fraction of examples labelled j among the

examples in the bag $\wr z_1, \ldots, z_{n-1}, (x_n, i)\wr$ of the same type as (x_n, i). The *quality* of a column of this matrix is its minimum entry. Choose a column (called the *best* column) with the highest quality; let the best column be j_{best}. Output j_{best} as the prediction and output

$$\left[\min_{i=0,\ldots,9} A_{i,j_{\text{best}}}, \max_{i=0,\ldots,9} A_{i,j_{\text{best}}}\right]$$

as the interval for the probability that this prediction is correct. If the latter interval is $[a, b]$, the complementary interval $[1-b, 1-a]$ is called the *error probability interval*. In Figure 1 we show the following three curves: the cumulative error curve

$$E_n := \sum_{i=1}^{n} \text{err}_i,$$

where $\text{err}_i = 1$ if an error is made at step i and $\text{err}_i = 0$ otherwise; the *cumulative lower error probability curve*

$$L_n := \sum_{i=1}^{n} l_i,$$

and the *cumulative upper error probability curve*

$$U_n := \sum_{i=1}^{n} u_i,$$

where $[l_i, u_i]$ is the error probability interval output by the algorithm for the label y_i; the values E_n, L_n and U_n are plotted against n. The plot confirms that the error probability intervals are calibrated.

6 Universal Venn probability machine

The following result asserts the existence of a universal VPM (it can be easily constructed using the histogram approach to probability estimation [3]).

Theorem 2 *There exists a Venn probability machine such that, if the examples are generated from p^∞ and A is an open set in $\mathcal{P}(\mathbf{Z})$ containing p, from some n on we will have $P_n \subseteq A$, where P_n are the multiprobabilities produced by the Venn probability machine.*

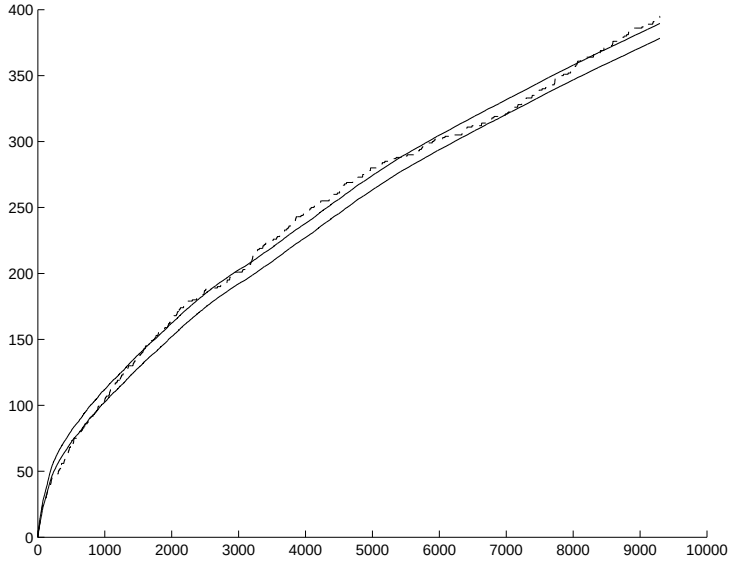

Figure 1: On-line performance of the 1-Nearest Neighbour VPM on the USPS data set (9298 hand-written digits, randomly permuted). The dashed line shows the cumulative number of errors E_n and the solid ones the cumulative upper and lower error probability curves U_n and L_n. The mean error E_N/N is 0.0425 and the mean probability interval $(1/N)[L_N, U_N]$ is $[0.0407, 0.0419]$, where $N = 9298$ is the size of the data set. This figure is not significantly affected by statistical variation (due to the random choice of the permutation of the data set).

This theorem shows that not only all VPMs are reliable but some of them also have asymptotically optimal resolution. The version of this result for p-values was proved in [10]; it is interesting that Theorem 2 is much easier to prove.

7 Comparisons

In this section we briefly and informally compare this paper's approach to standard approaches in machine learning.

Two most important approaches to analysis of machine-learning algorithms are Bayesian learning theory and PAC theory (the recent mixture,

the PAC-Bayesian theory, is part of PAC theory in its assumptions). This paper is in a way intermediate between Bayesian learning (no empirical justification for probabilities is required) and PAC learning (the goal is to find or bound the true probability of error, not just to output calibrated probabilities). An important difference of our approach from the PAC approach is that we are interested in the conditional distribution of the label given the new object, whereas PAC theory (even in its "data-dependent" version, as in [5, 8, 6]) tries to estimate the unconditional probability of error.

References

[1] James O. Berger and Mohan Delampady. Testing precise hypotheses. *Statistical Science*, 2:317–335, 1987. David R. Cox's comment: pp. 335–336.

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

[3] Luc Devroye, László Györfi, and Gábor Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer, New York, 1996.

[4] Berna E. Kılınç. The reception of John Venn's philosophy of probability. In Vincent F. Hendricks, Stig Andur Pedersen, and Klaus Frovin Jørgensen, editors, *Probability Theory: Philosophy, Recent History and Relations to Science*, pages 97–121. Kluwer, Dordrecht, 2001.

[5] Nick Littlestone and Manfred K. Warmuth. Relating data compression and learnability. Technical report, University of California, Santa Cruz, 1986.

[6] David A. McAllester. Some PAC-Bayesian theorems. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, pages 230–234, New York, 1998. Association for Computing Machinery.

[7] Craig Saunders, Alex Gammerman, and Vladimir Vovk. Transduction with confidence and credibility. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 722–726, 1999.

[8] John Shawe-Taylor, Peter L. Bartlett, R. C. Williamson, and M. Anthony. Structural risk minimization over data-dependent hierarchies. *IEEE Transactions on Information Theory*, 44:1926–1940, 1998.

[9] Vladimir Vovk. On-line Confidence Machines are well-calibrated, On-line Compression Modelling project, `http://vovk.net/kp`, Working Paper #1, April 2002.

[10] Vladimir Vovk. Universal well-calibrated algorithm for on-line classification, On-line Compression Modelling project, `http://vovk.net/kp`, Working Paper #3, November 2002.

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

[12] Vladimir Vovk, Ilia Nouretdinov, and Alex Gammerman. Testing exchangeability on-line, On-line Compression Modelling project, `http://vovk.net/kp`, Working Paper #5, February 2003.

On-line Compression Modelling Project
Working Papers

1. *On-line confidence machines are well-calibrated*, by Vladimir Vovk, April 2002. [FOCS'2002]

2. *Asymptotic optimality of Transductive Confidence Machine*, by Vladimir Vovk, May 2002. [ALT'2002]

3. *Universal well-calibrated algorithm for on-line classification*, by Vladimir Vovk, November 2002. [COLT'2003]

4. *Mondrian Confidence Machine*, by Vladimir Vovk, David Lindsay, Ilia Nouretdinov and Alex Gammerman, March 2003.

5. *Testing exchangeability on-line*, by Vladimir Vovk, Ilia Nouretdinov and Alex Gammerman, February 2003. [ICML'2003]

6. *Criterion of calibration for Transductive Confidence Machine with limited feedback*, by Ilia Nouretdinov and Vladimir Vovk, April 2003.

7. *Online region prediction with real teachers*, by Daniil Ryabko, Vladimir Vovk and Alex Gammerman, March 2003.

8. *Well-calibrated predictions from on-line compression models*, by Vladimir Vovk, April 2003.

9. *Self-calibrating probability forecasting*, by Vladimir Vovk, Glenn Shafer and Ilia Nouretdinov, June 2003.

Versions of some of these working papers have been or will be published in conference proceedings (given in brackets).