# Lessons Learned in the Challenge: Making Predictions and Scoring Them

Jukka Kohonen and Jukka Suomela

Helsinki Institute for Information Technology, Basic Research Unit,
Department of Computer Science, P.O. Box 68,
FI-00014 University of Helsinki, Finland
jukka.kohonen@cs.helsinki.fi,
jukka.suomela@cs.helsinki.fi

**Abstract.** In this paper we present lessons learned in the Evaluating Predictive Uncertainty Challenge. We describe the methods we used in regression challenges, including our winning method for the Outaouais data set. We then turn our attention to the more general problem of scoring in probabilistic machine learning challenges. It is widely accepted that scoring rules should be proper in the sense that the true generative distribution has the best expected score; we note that while this is useful, it does not guarantee finding the best methods for practical machine learning tasks. We point out some problems in local scoring rules such as the negative logarithm of predictive density (NLPD), and illustrate with examples that many of these problems can be avoided by a distance-sensitive rule such as the continuous ranked probability score (CRPS).

## 1 Introduction

In this paper we present lessons learned in the *Evaluating Predictive Uncertainty Challenge* (EPUC). The challenge was organised by Joaquin Quiñonero Candela, Carl Edward Rasmussen, and Yoshua Bengio, and the deadline for submission was in December 2004. The challenge consisted of five tasks: two classification tasks (where the targets are discrete, in this case binary), and three regression tasks (where the targets are continuous). We describe the methods we used in the regression tasks, and some lessons to learn from the methods and from the results. We have included our winning method for the 'Outaouais' data set, our abuse of the scoring method in the 'Gaze' data set, as well as our miserable failure with the 'Stereopsis' data set.

Inspired by observations made in the regression tasks, we will turn our attention to the more general problem of *scoring* in probabilistic machine learning challenges. Probabilistic predictions take the form of discrete distributions for classification tasks, and of continuous distributions for regression tasks. Scoring in classification is better understood, especially in the case of binary classification. In this paper, we focus on regression.

Ideally, the scoring function would guide competitors' work: by selfishly maximising their own score they would also work towards a common good in machine learning research and practice. It is widely accepted that scoring rules should be

J. Quiñonero-Candela et al. (Eds.): MLCW 2005, LNAI 3944, pp. 95–116, 2006.

*proper* in the sense that the true generative distribution has the best expected score. We note that while this is useful, it does not guarantee finding the best methods for practical machine learning tasks.

We will discuss what else is required of a scoring rule in addition to properness. We will point out some problems in using *local* scores, which depend only on the predictive density exactly at the true target value. We illustrate with examples that many of these problems can be avoided by using *distance sensitive* scores, which also depend on how much predictive probability mass is placed *near* the true target. As an example of a local rule we will consider the *logarithmic score* and the corresponding loss function, *negative logarithm of predictive density* (NLPD). As an example of a distance sensitive rule we will consider the *continuous ranked probability score* (CRPS).

Finally, we will briefly discuss how one can *represent* continuous predictions. Both in challenges and in practical applications there is obviously a need for a finite representation. We observe that a *sample* can be used as a very simple representation of an arbitrary distribution, provided that one is using a non-local scoring rule such as CRPS.

This paper is organised as follows. In Section 2, we describe the methods we used in the challenge. The general problem of scoring in probabilistic challenges is discussed in Section 3. Section 4 is devoted to discussing what other properties of scoring functions would be useful in addition to properness. Finally, we discuss in Section 5 how one can represent continuous predictions by finite samples. We conclude by proposing experimenting with distance sensitive scoring rules in future probabilistic challenges.

## 2 Selected Regression Methods

In the EPUC challenge, probabilistic predictions were required. Instead of a single *point estimate* of the target value, we were required to predict a probability distribution for each target, expressing how likely we thought each possible value was, given the training data and the known input values for the target.

The predictions were evaluated by using the so called NLPD loss. This loss function, and scoring in general, will be discussed in detail in Sections 3 and 4. For now, it is enough to note that the score was a function of the predicted probability *density* at the location of the true target.

No other information of data sets was given in addition to raw data and the name of the data set. The competitors did not know what kind of phenomenon they were dealing with.

Each data set was divided into three parts: training, validation, and test data. For simplicity, we will usually regard both training and validation data as training data, as this was the setting during the final phase of the challenge.

### 2.1 Outaouais

In the so called ‘Outaouais’ data set, the amount of data was relatively large. There were 37 input variables, and as many as 29 000 training samples. Some

of the input variables contained only discrete values, while some other input dimensions were continuous.

No obvious easy solutions or quick wins were found by visual observation. To gain more information on the data, we tried $k$-nearest-neighbour methods with different values of $k$, different distance metrics, etc. We noticed that very small values of $k$ produced relatively good predictions, while the results with larger neighbourhoods were much worse.

We next focused on 1-nearest-neighbour and studied the data more closely, checking which input variables were typically close to each other for nearest neighbours. We noticed that there seemed to be a surprisingly large number of discrete input variables whose values were often *equal* for a pair of nearest neighbours.

The discrete dimensions were clearly somewhat dependent. Starting with an initial set of possibly dependent discrete dimensions, we formed a collection of input dimensions which could be used to group all data points into classes. As a greedy heuristic, we kept adding dimensions which left much more than one training point in most classes.

The results were surprising. We found a set of 23 dimensions which classified all training input into approximately 3 500 classes, each typically containing 1 to 14 training points. Next we checked if any of these classes occurred in the test data, too. It turned out that both training and test input could be classified into approximately 3 500 classes, each typically containing 13 or 14 points. Almost all classes contained both training and test points. Thus, given a test point we almost always had some training points in the same class.

Next we focused on those classes which contained large numbers of training samples. For each class, the data points looked as if they were time series data. There was one dimension which we identified as "time". Naturally we do not know if the dimension actually represents time. However, having this kind of metaphors to support human thinking proved to be useful.

Fig. 1 shows training points from five different classes. This figure illustrates well typical behaviour in all classes: usually the target values changed slowly with time within each class, but there were also some classes with more variation.

If we had had to just predict the values, we could have fitted a smooth curve within each class. However, in this challenge we needed probabilistic predictions. We had 29 000 training points. Thus we could calculate *empirical* error distributions for *pairs* of samples within one class, conditioned on the discretised distance in the "time" dimension.

In other words, we first answered the following question: If we know that two points, $x_1$ and $x_2$, are in the same class, and the "time" elapsed between measuring $x_1$ and $x_2$ is roughly $T$, what is the expected distribution of the difference of target values $y_1$ and $y_2$?

We created 27 empirical distributions, one for each discretised time difference. Actually only 14 distributions are needed, others are mirror images. Fig. 2 shows histograms of three of these empirical distributions. For comparison, we also show Gaussian distributions with the same mean and the same variance. The empirical

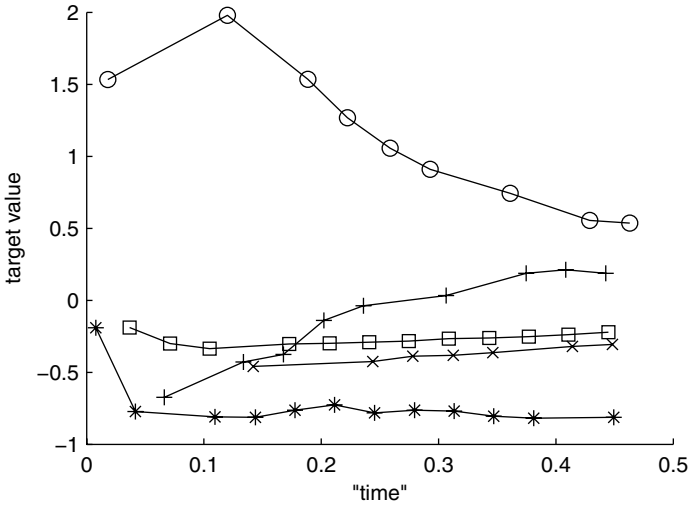


**Fig. 1.** Examples of some classes in the 'Outaouais' data set. Each set of connected dots corresponds to training points in a certain class. From the figure it is obvious that within one class, target values change only slightly in a short time interval.

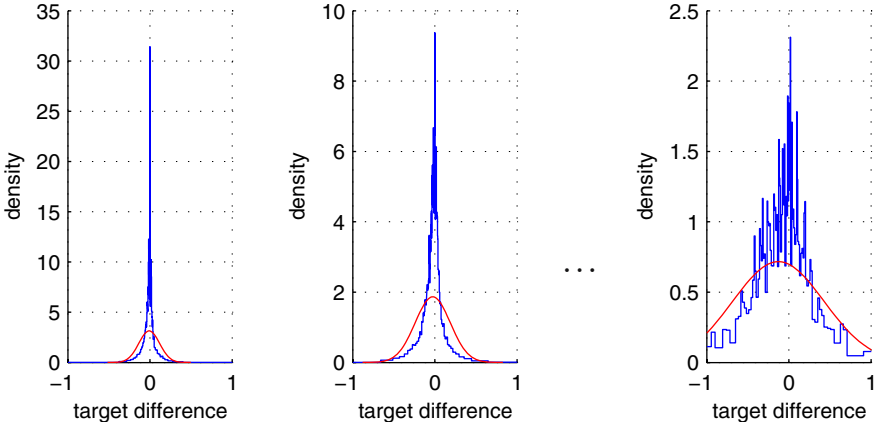


**Fig. 2.** Three precomputed error distributions for the 'Outaouais' data set, contrasted to Gaussians with the same mean and the same variance. The first figure corresponds to the shortest "time" interval (approx. 0.034 units) while the last figure corresponds to the longest "time" interval (approx. $13 \cdot 0.034 = 0.442$ units). Compare with Fig. 1.

distributions are clearly non-Gaussian, and their Pearson kurtoses range from 6 to 22, while a Gaussian would have a kurtosis of 3.

Now we were ready for prediction: For a given test input, we classified it, and picked the nearest neighbour value within the same class, measuring the distance in the "time" dimension. We discretised the distance and took the corresponding empirical error distribution. Then we predicted the target value of the neighbour

plus the error distribution. Thus, each prediction had a shape similar to one of the graphs in Fig. 2, shifted horizontally.

Our mean square error (0.056) was worse than what some other competitors had achieved (0.038). However, the NLPD loss was the lowest: -0.88 for us, -0.65 for the second place. Thus, our predictive distributions were more accurate.

If our method is viewed as a dimensionality reduction, one can see that 23 input dimensions (the ones used for classification) were converted into one class identifier; 1 "time" dimension was used as is; and 13 dimensions were discarded. Thus we reduced the dimensionality from 37 to 2 with very simple methods.

There is at least one thing to learn here: surprisingly naive methods may work *if* you can use large amounts of real data to estimate probability distributions. One may also consider whether the construction of training and test data was really compatible with the intended practical application. In practice one might have to make predictions before there are any known samples in the same class. Did we learn the phenomenon or just abuse the construction of the data?

### 2.2 Gaze

In the 'Gaze' data set, input was 12-dimensional, and there were only 450 training and validation samples. Sparsity of the data called for some kind of dimensionality reduction.

We visually inspected the $(x_j, y)$ scatter plots of each input variable $x_i$ versus the target $y$, for $j = 1, \ldots, 12$. Two of the input dimensions, $j = 1$ and $j = 3$, revealed a definite, albeit noisy regression structure (Fig. 3). The other 10 input dimensions appeared less useful for predicting $y$, and were discarded.

In the validation data, a few $x_3$ values were conspicuously low (Fig. 3); likewise in the test data. To avoid huge losses from erroneous regression estimates, we applied a manually chosen outlier detection rule ($x_3 < 0$). The outlying $x_3$ values were simply replaced with the sample mean. The values after this preprocessing step will be denoted $\tilde{x}_3$.

For further reduction of dimensionality, we linearly combined $x_1$ and $\tilde{x}_3$ into one quantity, $z = w_1 x_1 + w_3 \tilde{x}_3$, with $w$ chosen by cross-validation so as to maximise prediction accuracy within the training data.

We now had a one-dimensional regression problem of predicting $y$ from $z$. For this task, we chose a standard regression method, namely local linear regression [1] where the local weights were assigned using a Gaussian kernel.

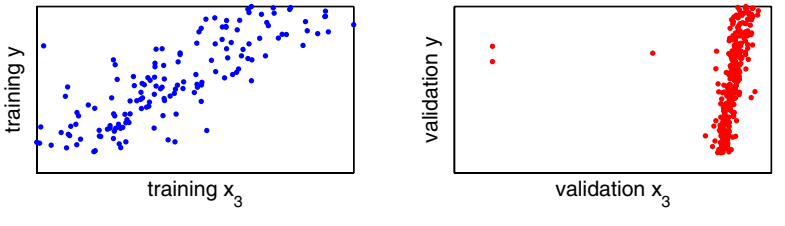


**Fig. 3.** 'Gaze' scatter plots of input variable $x_3$ versus regression target $y$

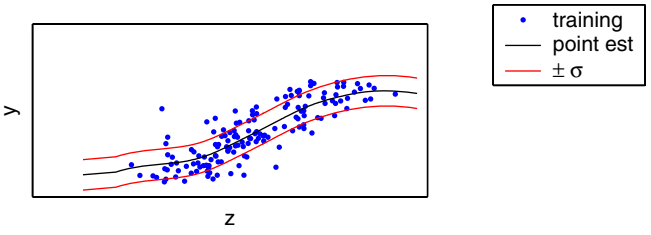


**Fig. 4.** LLR point estimates and standard error bounds for 'Gaze' data

Local linear regression (LLR) provides, for each unknown target, a point estimate $\hat{y}$. For probabilistic prediction, we also need the distribution of the error $\varepsilon = \hat{y} - y$. A standard practice is to estimate its variance with a local or global average of the squared errors for the training data [2]; or more generally, with an arbitrary smoother of the squared errors. After experimenting with different smoothers, we decided that a homoscedastic (i.e. constant-variance) error model with variance $\hat{\sigma}^2 = \sum_{i=1}^{n} (\hat{y} - y)^2 / n$ was accurate enough. Furthermore, the error distribution appeared more or less normal in the training and validation data sets.

Assuming normally distributed errors, we could thus predict $N(\hat{y}_i, \hat{\sigma}^2)$ for target $i$, where $\hat{y}$ is the point estimate from local linear regression, and $\hat{\sigma}^2$ is the global estimate of error variance. Such predictions are illustrated in Fig. 4.

A closer look at the training and validation data revealed that all target values were integers ranging from 122 to 1000. Since arbitrary predictive distributions were allowed, it seemed pointless to assign any significant probability mass to non-integral target values. Doing so could, in fact, be seen as a failure to report a pronounced feature of the target distribution.

Accordingly, we concentrated the predicted probability on the integers. Because the number of quantiles seemed to be limited by allowed file size, we did this only for a part of the distribution. We discretised the predicted Gaussian into 7 equiprobable brackets, delimited by the $i/7$ quantiles for $i = 1, \ldots, 6$. The probability in the central bracket was then mostly reallocated around the integers within the bracket (Fig. 5). Limited only by the floating point precision, the spikes on the integers were $2 \cdot 10^{-13}$ units wide. As a result, each spike could be assigned a very high probability density, about $1.5 \cdot 10^{10}$. If a target indeed coincides with such a spike, we would thus gain a negative logarithmic score of $-\log_2 1.5 \cdot 10^{10} \approx -23$ for that target.

What did we learn from this regression task? The methods we applied were quite standard. The distinctive feature of our second-ranking solution was exploiting the integrality of the targets. It would be fair to say that this was an abuse of the scoring method. On the other hand, *assuming* that the scoring method faithfully represents what kind of prediction is being sought for, one could argue that NLPD essentially mandates that the competitors submit every bit of information they possibly can about the test targets, including the fact that they are precisely on the integers. We will return to this topic in Section 4.

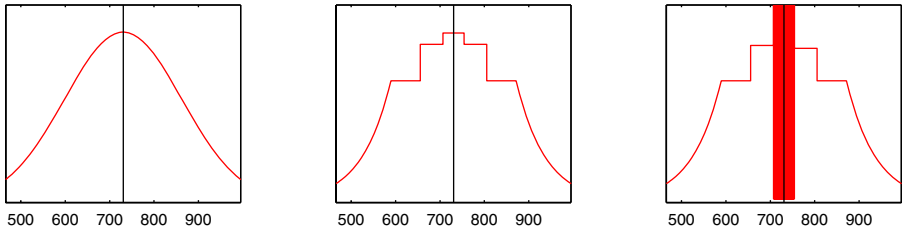


**Fig. 5.** Successive prediction stages for 'Gaze'. Left: The original Gaussian. Centre: Discretised Gaussian. Right: Centre bracket replaced with narrow spikes on integers.

### 2.3 Stereopsis

The 'Stereopsis' data set had only 4 input dimensions. Visual inspection of the data showed clear, regular structure. The name of the set was an additional hint: the word "stereopsis" means stereoscopic vision. Based on studies, we formed a hypothesis of the physical phenomenon that had created the data.

The assumed model was as follows: The input data consists of two coordinate pairs, $(x_1, y_1)$ and $(x_2, y_2)$. Each pair corresponds to the location of the image of a calibration target, as seen by a video camera. The target data corresponds to the distance $z$ between the calibration target and a fixed surface. Both cameras are fixed. The calibration target is moved in a $10 \times 10 \times 10$ grid. The grid is almost parallel to the surface from which distances are measured.

Naturally we had no idea if this model is right. However, having some visual model of the data helps in choosing the methods. Having a model in mind, we proceeded in two phases.

1. We first *classified* the data into 10 distance classes. Each class was supposed to correspond to one $10 \times 10$ surface in the grid. Distances (training target values) within each class are close to each other.
2. Within each class, we fitted a *low-order surface* to the training points.

The first part, classification, seemed trivial. We used a linear transformation to reduce dimensionality to 1, and used 9 threshold values to separate the classes.

In the second part, the physical model guided the selection of the parameterisation of each surface. It turned out that simply mapping the coordinates of one camera, $(x_1, y_1)$, to the polynomial $(1, x_1, y_1, x_1 y_1)$ made it possible to find a highly accurate linear fitting. Higher order terms seemed to cause only over-fitting, while leaving the $x_1 y_1$ term out produced worse results in validation.

This particular polynomial makes sense if we assume that the grid is formed by defining four corner points of each surface and interpolating linearly. Errors due to lens distortions may be large in $x$ and $y$ dimensions, but their effect on almost parallel surfaces in the $z$ dimension are minor.

Given this prediction method, there are two error sources in these predictions. Firstly, there is the possibility of a classification error. However, we assumed that classifications are correct. When the classifier was formed by using only training

points, all validation samples were classified correctly and with large margins. This assumption turned out to be a fatal mistake.

Secondly, there is the distance between the surface and the true target. We assumed that this error is primarily contributed by Gaussian noise in measurements. Variance was estimated for each surface by using the training samples. Thus, we submitted simple Gaussian predictions.

The results were a bit disappointing. There was a huge NLPD loss. It turned out that 499 of the 500 test samples were predicted well. 1 of the 500 samples was completely incorrect. This was a classification mistake and one huge loss was enough to ruin the score of the entire prediction. We, obviously, should not have trusted the simple classification method too much.

In addition to that single classification mistake, is there anything one can learn from this effort? If our guess of the model is correct, and the real objective was training a computer vision system to estimate distances, this method is completely useless. However, it did predict almost all test points well. This is due to learning the structure of the calibration process, not due to learning how to calculate the distance from stereo images. The lesson learned: One needs to be careful when choosing the training and validation data. For example, random points in continuous space instead of a grid could have helped here to avoid this problem.

## 3 About Challenges and Scoring

Probabilistic machine learning challenges, such as the EPUC challenge, can give us new empirical information on applying machine learning in practical problems. Ideally, one would gain information on which methods work well in practice. One could also learn more on how to choose the right tool for a given problem, and how to choose parameters. However, as we will see, one needs to be careful when choosing the scoring rules used in the competition.

The quality of a machine learning method can be defined in various ways. We narrow the scope by ignoring issues such as computational complexity. We will focus on how useful the predictions would be in a practical application.

### 3.1 Notation and Terminology

For scoring rules, we use the notation used by Gneiting and Raftery [3]. Let $P$ be the predictive distribution and let $x$ be the true target. A *scoring rule* is any function $S(P, x)$. Given a distribution $Q(x)$, we use $S(P, Q)$ for the expected value of $S(P, x)$ under $Q$, i.e. $S(P, Q) = \int S(P, x)\, Q(x)\, dx$.

If $P$ is a probability distribution, we use $P(x)$ to denote its density function, and $P(X \leq x)$ to denote cumulative density.

### 3.2 Modelling a Challenge

In our model of a competition, training inputs $X = (\mathbf{x}_j)$, training targets $Y = (y_j)$, and test inputs $T = (\mathbf{t}_i)$ are given to competitors while true test targets $U = (u_i)$ are not yet published.

The competition consists of three phases:

1. Each competitor $k \in K$ chooses a machine learning method $f_k$, forms a hypothesis $h_k = f_k(X, Y)$, and use the hypothesis to form a personal probability distribution $Q_{k,i} = h_k(\mathbf{t}_i)$.
2. The competitor chooses a prediction $P_{k,i} = g_k(Q_{k,i})$ by using any function $g_k$. The prediction $P_k = (P_{k,i})$ and a description of the method $(f_k, g_k)$ are reported to the organiser.
3. Each competitor is assigned a score $s_k = \frac{1}{|U|} \sum S(P_{k,i}, u_i)$.

In our model, $f_k$ encodes essentially everything one needs in order to re-use the same method in a new, similar problem. In addition to a machine learning algorithm, it describes all rules the expert used for, say, choosing the right parameters. Evaluating $f_k(X, Y)$ may require not only computer resources but also work by a human expert.

Competitors typically have no a priori knowledge on the phenomenon. Thus a competitor's personal probability distribution, $Q_{k,i}$, is formed solely by using the method $f_k$, as described above.

However, a competitor does not necessarily want to report her honest personal probability distribution. Perhaps her expected personal utility would be maximised by reporting an overconfident prediction. This could be caused either by the peculiar nature of her utility function, or by the general characteristics of the scoring method being applied in the competition. Instead of denying the possibility of such human behaviour, we have added in our model the mapping $g_k$ which the competitor uses for forming her prediction. This model, where the competitors seek to maximise their personal utilities, is in line with Bernardo and Smith's argumentation that "the problem of reporting inferences is essentially a special case of a decision problem" [4, p. 68].

We are explicitly requiring that the methods are revealed. Otherwise there is little one can learn from the results of the competition. However, the score does not depend on the structure of $f_k$. The whole point of these competitions is evaluating methods by their practical results, not by their theoretical merits.

The score does not depend directly on $Q_{k,i}$, either. While the competition organisers could, in principle, use $f_k$ to re-calculate $Q_{k,i}$, the amount of human work involved could be huge. Thus we are left with scoring the reported predictions, $P_{k,i}$.

The EPUC challenge conforms to this model. The requirement of reporting $(f_k, g_k)$ is implemented by asking the winners to present their methods.

### 3.3 Results of a Challenge

Let us assume that the competitor $\tilde{k}$ achieved one of the highest scores. In spite of all limitations mentioned above, we would like to be able to learn something on $f_{\tilde{k}}$. Ideally, $f_{\tilde{k}}$ should now be among the strong candidates for use in similar practical machine learning problems. Obviously, this is not always true, a trivial counterexample is a competition where all participants were novices and all predictions were useless.

Competitors have many degrees of freedom for choosing what to do in the competition. On the other hand, there is relatively little what the organiser of the competition can do in order to affect the quality of the results. Assuming the data sets are fixed, the organiser can only choose the set of competitors $K$ and the scoring rule $S$.

In this paper, we concentrate on the task of choosing the scoring rule. We will not discuss the task of choosing the competitors. However, we briefly note that in order to find good methods $f_k$, the set of competitors should preferably contain a number of leading machine learning experts. On the other hand, participation is voluntary. If the experts notice that the scoring rules of the competition are poorly designed, they may be reluctant to participate. Thus choosing the scoring rules plays a role even in the task of choosing the competitors.

### 3.4 Scoring and Linearity of Utilities

A score as such has little meaning. In this paper, we assume that each competitor has a utility function which depends linearly on her score, $s_k$. This is a strong assumption. It may be hard to implement in a competition. If reputation, fame and publicity are the only prize, typically the winner takes it all.

However, a non-linear dependency makes it hard to analyse competitions. If, for example, only the winning score has a high utility, competitors are encouraged to take a risk with overconfident predictions. A small chance for a winning score would have a better expected utility than a safe medium-level score. In such a setting, the winners could be those who were lucky, not those who used the best methods. Competitors can also be risk-averse; in that case they might choose to play it safe and report underconfident predictions. Both risk-seeking and risk-averse patterns of behaviour have been observed in probability forecasting competitions; see, for example, Sanders [5].

Implementing a linear utility has been studied in the literature. A typical construction involves a single-prize lottery where the winning probabilities are proportional to the scores. See, for example, Smith's construction [6].

### 3.5 Scoring in a Challenge

There is a rich literature on scoring probabilistic predictions, both for discrete probability distributions (classification) and for continuous probability distributions (regression) [7, 8, 9]. Much of the work is related to atmospheric sciences for obvious reasons [5, 10, 11, 12]. In this subsection, we will look at the scoring from the point of view of probabilistic machine learning challenges.

Matheson and Winkler [13] summarise three common uses for scoring rules. First, scoring rules can be used to encourage assessors *make careful assessments*. Second, scoring rules can be used to *keep assessors honest*. Finally, scoring rules can be used to *measure the goodness* of predictions. We will soon see that this list applies well to a machine learning challenge.

The direct requirements for a scoring rule are two-fold:

1. The scoring rule should encourage experts to work seriously in order to find a good method, $f_k$.
2. The final score should reflect how good the method $f_k$ is.

These two requirements correspond directly to two applications on Matheson and Winkler's list: encouraging good assessments and measuring the goodness of predictions.

However, the scores reflect the quality of the method only indirectly. There are two layers of indirection: First, we are scoring $f_k$ by using a sample. The second issue is that we are not scoring the quality of the competitor's personal probability $h_k(\mathbf{t}_i)$ but the quality of her reported probability $g_k(h_k(\mathbf{t}_i))$ for an arbitrary $g_k$.

For the first issue there is little we can do besides using a relatively large and representative test set. By using a sample mean $s_k = \frac{1}{|U|} \sum S(P_{k,i}, u_i)$ we are estimating the expected value of the score and determining the quality of the estimator is standard statistics.

The second issue is more subtle. Ideally we would like to have $g_k(x) = x$ for all $k$. This is needed not only for making it easier to analyse the competition but also for encouraging careful assessments. If the connection between competitors' work and the final score becomes more indirect and complicated, the competitors cannot see where they should focus their attention.

One can naturally ask: why not simply *require* that competitors use the identity function as $g_k$. We were assuming, after all, that competitors report both $f_k$ and $g_k$, and the reported $f_k$ and $g_k$ could be manually checked by a competition organiser, if needed. Unfortunately, competitors could incorporate the mapping $g_k$ in their reported $f_k$ and let $g_k$ be the identity function. Nothing would be gained but the goal of the competition would be missed. The method $f_k$ would no longer be a good way of estimating probabilities, it would only be a way of maximising the score. Its use in practical problems would be limited.

Instead of *requiring*, the competitors should be *encouraged*, by the design of the challenge, to use an identity function as $g_k$. This is what Matheson and Winkler refer to as keeping assessors honest. If the scoring rule is such that for a competitor, the identity function is the best choice of $g_k$ in terms of her expected score, we gain a lot. The competition organiser can announce this fact and the competitors can check it. Now the competitors can concentrate on developing a method to estimate true probabilities of the phenomenon.

A scoring method with this useful property is called *proper*. Focusing on proper scoring rules is similar to focusing on truthful or strategyproof mechanisms in the field of mechanism design.

### 3.6 Proper Scoring Rules

The problem of keeping assessors honest is easily expressed and well understood. See, for example, Gneiting and Raftery [3] for a modern treatment on the subject.

A scoring rule is called *proper* if $S(Q, Q) \geq S(P, Q)$ for all $P$ and $Q$. A scoring rule is *strictly proper* if it is proper and if $S(Q, Q) = S(P, Q)$ only if $Q = P$. If a predictor's personal probability is $Q$, under proper scoring she gains nothing, on average, by predicting anything else than $Q$. Under strictly proper scoring she always loses, on average, by predicting anything else than $Q$.

Proper scoring rules have gained a lot of attention in the literature. There are large families of proper and strictly proper scoring rules [3, 7, 13]. However, properness by itself has only limited use in encouraging careful assessments and measuring the goodness of predictions. Strict properness essentially guarantees that *if* one works hard enough to make a perfect prediction by deducing the true generative distribution, one is expected to gain more than other competitors. However, a useless prediction can achieve *almost* as high a score as the perfect prediction. We will later see examples of this. As a competitor's utility depends both on the score and on the work required, an easy and highly scored solution is inviting. This problem will be reflected also in measuring the goodness of predictions. If competitors were encouraged to focus on highly scored but useless predictions, that is also what the highest scoring methods are expected to be.

In summary, properness is useful, it allows a large variety of alternative scoring rules, and it is not enough by itself. Properness is a good starting point. The next section illustrates what else should be expected from a good scoring rule.

## 4 Beyond Properness

While keeping assessors honest can be formulated and solved in a uniform way for all problems, encouraging careful assessments and measuring the goodness depends on the application. This is intimately tied to the question of how we value the information provided by different kinds of probabilistic predictions.

Scoring rules can be divided into *local* and *non-local* rules. In a local rule [14], the score of a predictive distribution $P$ depends on the predictive density at the true target value only, that is, on $P(x)$. A non-local scoring rule may take into account also other characteristics of the predictive distribution. An interesting class of non-local scoring rules are *distance sensitive* rules [15, 16], which favour predictions that place probability mass *near* the target value, even if not exactly at the target.

It is an intriguing question whether a scoring rule should be local or not. Statistical inference problems may be vaguely divided into "pure inference problems" where the goal is simply to gain information about the targets, and "practical problems" where we seek information in order to solve a particular decision problem. Bernardo [4, 14] states that in a pure inference setting, a local scoring rule should be used. However, in many practical settings there seems to be a need for non-local rules.

In order to help make this discussion more concrete, we introduce two proper scoring rules which have been proposed for scoring continuous predictions: the negative log predictive density (NLPD), which is a local rule, and the continuous ranked probability score (CRPS), which is non-local.

### 4.1 NLPD

NLPD is the scoring function which was used, for example, in the EPUC challenge. NLPD stands for *negative log estimated predictive density*. It is a loss function: large values imply poor performance. To make it compatible with our framework, we derive a scoring function by changing the sign. The result is simply the *logarithmic score* (see, for example, Matheson and Winkler [13]):

$$S_{\mathrm{NLPD}}(P, x) = \log P(x). \tag{1}$$

In this text we will use the terms NLPD score and the logarithmic score interchangeably. Both refer to the scoring function defined in Equation (1).

The NLPD score is obviously local. In fact, under suitable smoothness conditions, it is essentially the *only* proper scoring rule which is also local [14]. If locality is indeed desirable, this is a strong argument in favour of the NLPD.

### 4.2 CRPS

CRPS stands for *continuous ranked probability score*. CRPS is a generalisation of the idea of the *ranked probability score* (RPS), introduced by Epstein [11] for scoring in probabilistic classification.

Epstein observed that existing proper scoring rules did not use the concept of distance. However, the classes may represent, for example, ranges of measurements. In this case the classes are not independent.

Epstein uses weather forecasting as an example: It is assumed that the classes represent consecutive temperature ranges, $A$ predicts (0.1, 0.3, 0.5, 0.1), $B$ predicts (0.5, 0.3, 0.1, 0.1), and the fourth class corresponds to the observed temperature. If the ordering of the classes is ignored, both predictions would obtain the same score.

However, these predictions are not equivalent for a typical *user* of the prediction. Given the prediction $B$, the user would be prepared for much colder weather than given the prediction $A$. Epstein developed this line of thought further to estimate the expected utility of the user of the prediction. For example, given the prediction $B$ above, the user's utility is low as she had to prepare for cold weather while preparation was not needed. With a number of simplifying assumptions, and after normalising the scores, Epstein derived his recommendation for a scoring rule, the ranked probability score. Murphy [17] showed that this score is proper.

In order to simplify notation, let us define a probability distribution function $R_x$ based on the observed value $x$, such that $R_x(X \leq i) = 0$ for all $i < x$, and $R_x(X \leq i) = 1$ for all $i \geq x$. Now we can present RPS as follows:

$$S_{\mathrm{RPS}}(P, x) = 1 - \frac{1}{n-1} \sum_{i=1}^{n-1} (P(X \leq i) - R_x(X \leq i))^2. \tag{2}$$

Here $n$ is the number of classes. We see that the ranked probability score is a linear transform of the *square error* between the predicted and observed *cumulative* distribution functions.

This can be directly generalised to a continuous distribution. Thus we obtain the continuous ranked probability score which was introduced, and proved to be proper, by Matheson and Winkler [13]:

$$S_{\mathrm{CRPS}}(P, x) = -\int (P(X \leq u) - R_x(X \leq u))^2 \, w(u) \, du. \tag{3}$$

Here $w(u)$ is arbitrary weight. NLPD and CRPS scores are illustrated in Fig. 6.

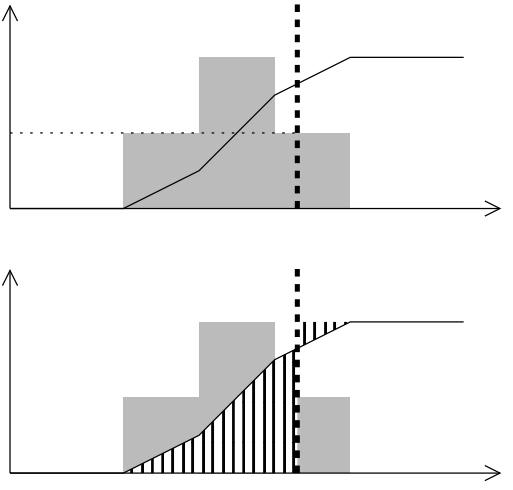

**Fig. 6.** An illustration of the NLPD and CRPS scores. In these figures, grey areas illustrate predicted probability density functions. Solid lines are used to show the corresponding cumulative distribution. Dashed vertical bars show the true target. The first part, (a), shows the NLPD score: it is logarithm of the predicted density near the true target values, as indicated by the horizontal dotted line. The second part, (b), shows the CRPS score: it is the square error between the predicted and observed cumulative distribution functions, the error is illustrated by a vertical striped pattern.

The CRPS is a non-local rule, and in general, it favours predictions that put a lot of probability mass near the target. We will refer to this property as *distance-sensitivity*. Its exact definition varies in the literature; CRPS is sensitive to distance according to Staël von Holstein's "tail sums" definition [15], but not according to Murphy's "symmetric sums" definition [16].

### 4.3 Locality Versus Distance-Sensitivity

Let us now continue the discussion of the relative merits of local and distance sensitive scoring rules. We will here accept the view of Bernardo [14] that statistical inference about an unknown quantity is essentially a decision problem, where one tries to maximise the expected *utility* of information attained by doing inference. A central question is then how to define the utility of information.

Recall the distinction of "pure inference problems" and "practical problems". In a practical problem, the application at hand may dictate a particular scoring function, related to the end utility of making decisions based on partial information. Such a scoring function is quite often non-local.

On the other hand, in a pure inference problem, there is no immediate practical application. One simply wants to gain knowledge about the targets. Bernardo states that in such a setting, one should *maximise the expected gain of information* [4, p. 72]. This leads to requiring a local scoring function, and if also properness and smoothness are desired, essentially choosing the NLPD.

While Bernardo's argument is otherwise compelling, it rests on an important hidden assumption: that if one desires to gain knowledge (about the value of a target quantity), one should then indeed maximise the *amount of information* gained. This implicitly means that all information about the target value is treated as equal in value; for example, that learning the tenth decimal of an unknown quantity is just as valuable as learning its first decimal.

For regression tasks, where the target values are continuous, we find such equivalence rather unnatural even in pure inference; more so if the inference task is in any way related to a practical setting, such as inferring the value of a physical magnitude. This is particularly evident when we consider how probabilistic predictions can be used.

When using a probabilistic prediction, the predicted distribution tells us how likely various undesirable or difficult situations are. Then, the costs of preparing for those situations can be compared to the probabilities and the potential damages in case of no protection. We list here some examples from various fields:

1. Weather prediction: Difficult situations can be, for example, very high or low temperatures, heavy rain, storms, etc. The costs of preparing for those situations can range from carrying an umbrella to cancelling flights.
2. Financial sector: Undesirable situations can be financial risks in investments. Preparing for those situations may involve, for example, bidding a lower price and the possibility of losing a deal.
3. Measuring distances: An undesirable situation can be, for example, a robot colliding with an obstacle. Preparing for that situation might require slowing down and thus spending more time.

In all of these examples, if the undesirable situation did *not* realise, the preparations were done in vain. In that case, all other factors being equal, a prediction which estimated a high risk of an undesirable situation is more costly than a prediction which did not do so.

Furthermore, usually some undesirable situations are more severe than others. The more extreme situations are possible, the more costly preparations we may need to do. Thus, the practical use of a prediction depends on the distance between the true target and the predicted probability mass.

In all examples described above, the concept of distance plays a role: In weather prediction, the practical significance of small and large differences in predicted temperatures was already illustrated in Epstein's [11] example in Section 4.2 above. In financial sector, if an investment was actually highly profitable,

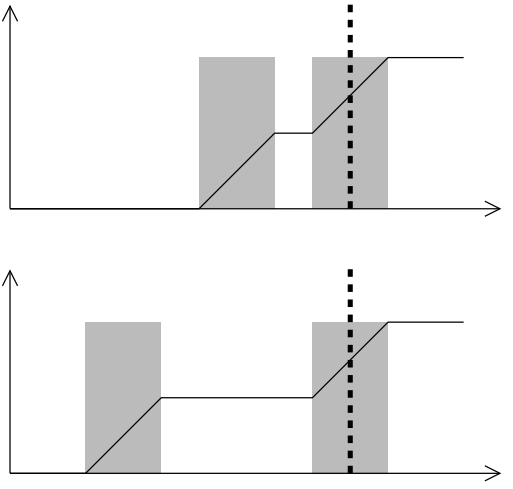

**Fig. 7.** An illustration of the concept of distance. Both predictions, (a) and (b), will gain the same NLPD score, while the CRPS score is much lower in case (b). The CRPS score corresponds well to the practical applicability of the predictions: given the prediction (b), the user of the prediction needs to be prepared for a wide range of different values, making her utility lower.

having a prediction of possible low profits may be relatively harmless while having a prediction of possible large losses may lead into the wrong decision. When measuring distances, if a wall is actually at the distance of 200 cm, predicting a possible obstacle at the distance of 196 cm is typically harmless, while predicting a possible obstacle at 5 cm may make robot navigation much harder.

The concept of distance is commonly used in many non-probabilistic loss functions designed for regression tasks: consider, for example, the square loss. We argue that it should also be taken into account in probabilistic loss functions. Fig. 7 illustrates how the CRPS score takes the distance into account, while the NLPD score ignores it. It should be noted that in this example, both predictions contain exactly the same amount of information, while the practical value of this information differs. In practical applications, not every bit of information is worth the same. We will elaborate this issue further in the next section.

### 4.4 Information Which Is of Little Use

Typically, knowing the finest details of a probability distribution function is of very little use. When predicting a real value in the range $[0, 1]$, accurately predicting the first decimal digit is of much higher practical use than accurately predicting the second (or tenth) decimal digit. This issue is closely related to the concept of distance: knowing the first decimal digit corresponds to a prediction where all probability mass is concentrated in a small range.

Let us assume that $A$ predicts correctly the first decimal digit of the true target (all values with the right 1st digit having the same probability) and $B$ predicts

correctly the second decimal digit. Both predictors gain the same amount of information. Furthermore, the NLPD score will be the same for both predictors. However, we can easily see that the CRPS score is much higher for the prediction $A$. Actually, with respect to the CRPS score, the prediction $B$ is not much better than a prediction where nothing is known. This is well in balance with the practice: knowing, say, the second decimal digit of rainfall is typically of no practical use if the more significant digits are uncertain. The user of the prediction would have to be prepared for all kinds of weathers.

In the EPUC challenge, the problem of useless information can be illustrated by the 'Stereopsis' data set. As mentioned above in Section 2.3, one possible approach consists of classifying points to 10 distance classes, and predicting within each class. Our failure could have been avoided by assigning a positive probability for each of these classes, and by giving predictions with 10 narrow Gaussians instead of only 1 narrow Gaussian distribution. This may gain a good NLPD score, yet be of no practical use in a computer vision application.

### 4.5 Point Masses

In practical regression settings, one often encounters the problem of point masses. By this we mean that the predictor has some reason to believe that in an otherwise continuous target domain, there are special values which have a nonzero mass. There are several different sources of this problem.

Fig. 8 illustrates how a predictor may use this information. Part (a) presents the original prediction before information on point masses is used. Part (b) shows how a competitor may slightly modify her prediction to reflect her belief that there are point masses.

The spikes can be made arbitrarily high, and at the same time their probability mass can be made very small simply by limiting their width. Thus adding spikes can leave the predicted density outside the spikes virtually unchanged.

If the true target does not match any of those spikes, neither NLPD nor CRPS score is considerably changed when comparing predictions (a) and (b). Thus, adding spikes is relatively harmless from a competitor's point of view.

However, if the true target indeed happens to match one of the predicted spikes, the NLPD score for prediction (b) is arbitrarily high while the CRPS score for prediction (b) is still essentially the same as for prediction (a). Thus, the NLPD score strongly encourages working towards finding some discrete point masses, while the CRPS score does not reward for it unless the point masses are large enough to considerably change the density function.

If there is at least one match with *any* predicted spike in *any* of the test targets, the NLPD score is dominated by that spike. If the NLPD score is used and any such point masses are found, there is little reason for predictors to make any efforts to model any other aspects of the phenomenon.

Thus, this problem, too, is primarily related to encouraging careful assessments. It seriously affects measuring the goodness of predictions: the existence of such point masses may be trivial and uninteresting, while NLPD score may be dominated by them.

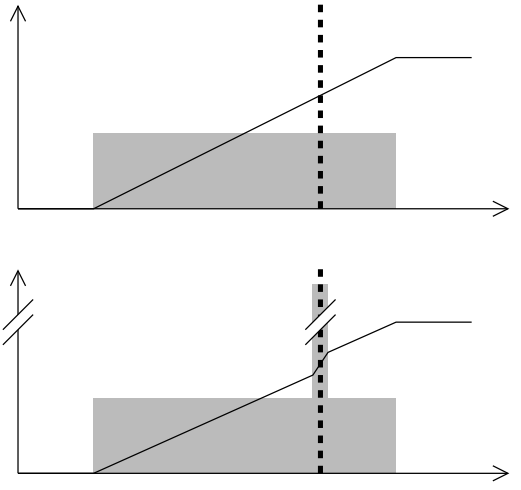

**Fig. 8.** An illustration of the problem of point masses. Part (a) shows the original prediction. Part (b) shows how a competitor may slightly modify her prediction if she has a reason to believe there is a nonzero point mass. The competitor has added an extremely narrow spike in the predicted density function. The probability mass of the spike can be made low, and density everywhere else can be left virtually unchanged.

In practice the format in which the predictions are submitted may not allow arbitrarily high and arbitrarily narrow spikes. For example, the range and precision of IEEE floating point numbers effectively limited this problem in the EPUC challenge. However, such limits are quite unsatisfactory and arbitrary and still allow manipulating scores by using some narrow spikes.

Thus, using *any* local scoring rule which is based on the value of the probability density function at the location of the true target should be avoided in this kind of challenges unless data sets are selected very carefully in order to avoid the problem of point masses. Non-local scoring rules which are based on comparing cumulative distribution functions are better in this respect.

Now we will have a look at some practical examples of point masses. We have three categories of point mass problems. These categories are based on features of the data sets of the EPUC challenge, and thus clearly relevant in the context of probabilistic challenges.

**Known Targets.** The first category is the case of known targets. For example, the first version of the 'Stereopsis' data set in the EPUC challenge accidentally contained some overlap between training and test data. This should not affect the results of the challenge: each competitor has the same knowledge, and the scores of these overlapping known points would simply be an additive constant in final scores. However, for the NLPD score, this additive constant would be infinity, ruining the final scores. For the CRPS score, the constant would be zero.

**Special Values in the Target Domain.** The second category deals with special values in the target domain. For example, the 'Outaouais' training data

contained 250 points where the target was exactly zero. Thus, it is likely that also the test data set of comparable size contains some targets that are zero.

Such special values may occur if *missing data* are represented as zeroes. There may also be a natural reason why a continuous physical variable really has a nonzero point mass somewhere, precipitation being a good example [3].

**Discrete Target Domain.** The third category is the case of discrete target domain. The 'Gaze' data set had integral target values. Modifying predictions to reflect this trivial fact improved NLPD scores considerably.

Discrete target values are actually relatively typical in practical applications:

- The target domain may actually be integral; the prediction task might deal with counting some occurrences.
- Financial quantities such as money and shares are almost always expressed with a fixed precision.
- Devices which measure physical quantities usually work with finite precision.

One may argue that at least discrete target domains could be dealt with by interpreting them as classification tasks instead of regression tasks, and by asking for discrete predictions instead of continuous predictions. However, the set of possible values may be large or infinite, making this approach impractical.

## 5 Representing Predictions

One needs a finite representation for continuous probability distributions. A single Gaussian is not flexible enough in order to represent arbitrary predictions.

In the EPUC challenge, the other alternative was a *set of quantiles*, essentially a histogram with exponential tails (see Fig. 5 for an example). Quantiles are flexible but handling quantile predictions in, for example, mathematical software is a bit complicated. One typically needs to handle the histogram part and the tails separately, making program code more complicated and error-prone.

There is a need for simpler ways to represent continuous predictions, both in challenges and in practical applications. One possible idea would be using a *sample*. One could simply draw a finite set of sample values randomly from the predicted distribution and report those.

Naturally, one could then use density estimation to recover an approximation of the original distribution. However, this would not simplify matters at all, and one would also need to specify the parameters used in density estimation. By using suitable scoring methods, there is an easier solution.

One could interpret a finite sample literally as a probability distribution with a finite set of point masses. This is illustrated in Fig. 9. The density function would consist of infinitely high and narrow spikes, while the cumulative distribution function would consist of a finite number of steps.

In most cases, the true target value would not match exactly any point mass. The expected NLPD loss would be infinite. However, the shape of the *cumulative* distribution function would be close to the original distribution. Thus, if one

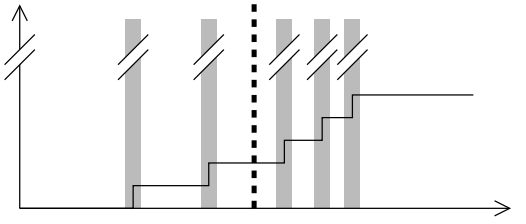

**Fig. 9.** An illustration of using samples to represent predictions. In this example, the prediction was given as a sample of five points. It corresponds to a probability distribution with five spikes.

used the CRPS score, a finite set of sample points would give approximately the same score as a quantile prediction, but with considerably less complexity. A key difference with quantile predictions is that there would be no need to handle tails in any special way. CRPS is well-defined even if the target is outside the range of the sample points.

This line of thought may be also used to guide the selection of the scoring method. Clearly, a finite sample cannot represent accurately all aspects of a continuous distribution, such as the shape of the density function in low-density areas. Such details are not much reflected in the CRPS score, either. If we have a practical problem where the CRPS score can be used, it means that we are not interested in such details, and thus we can use samples to represent predictions. Conversely, if we cannot use a sample, it may be because we *are* interested in such details, and then we probably should not be using CRPS.

While evaluating the CRPS score in equation (3) may be difficult for an arbitrary prediction [3], it is straightforward if a prediction is represented as a finite sample. One may actually interpret this process as a (possibly randomised) approach to approximate numerical integration. By letting the competitors perform sampling, they may use arbitrarily complicated predictions. Scoring will be also fair in the sense that the organiser of the challenge does not need to use any randomised or approximate method when evaluating submitted predictions.

Finite samples arise naturally in the context of *ensemble prediction*. For example, by running a weather model with several slightly perturbed initial conditions we can obtain an ensemble of different point predictions for, say, tomorrow's temperature. Typically the ensemble is then converted into a probability distribution of suitable form. But if predictions are represented as finite samples, no conversion is needed: the ensemble itself can serve as the representative sample.

## 6 Conclusion

In this paper, we reported our methods in the regression tasks of the Evaluating Predictive Uncertainty Challenge. The tasks also demonstrated some pitfalls in using the well-known NLPD score. We analysed the problem of organising a probabilistic machine learning challenge and proposed two possible improvements for future challenges:

1. One can avoid many pitfalls, if one uses a distance-sensitive scoring method such as CRPS.
2. Description and implementation of the scoring methods can be simplified, if predictions are represented as samples.

We accept that NLPD is the method of choice for the tasks it was designed for: truly continuous, pure inference tasks where every bit of information is worth the same. Unfortunately, one often encounters regression tasks that do not conform to this idealised model, even if they appear so on the surface.

CRPS is not the only possible solution. Whether there are other distance-sensitive scoring methods which reflect significantly better the practical value of predictions is still an open question.

We assumed that the competitors' utilities depend linearly on their scores. Further research is needed on this issue. Firstly, one can pay more attention on implementing linear utilities in challenges. Secondly, more research can be done on modelling challenges where the winner takes it all.

## Acknowledgements

We wish to thank the organisers and the PASCAL network for an interesting and thought-provoking challenge.

This work was supported in part by the Academy of Finland, Grant 202203, and by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors' views.

## References

1. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. Springer-Verlag (2001)
2. Härdle, W.: Applied Nonparametric Regression. Cambridge University Press (1990)
3. Gneiting, T., Raftery, A.E.: Strictly proper scoring rules, prediction, and estimation. Technical Report 463, Department of Statistics, University of Washington (2004)
4. Bernardo, J.M., Smith, A.F.M.: Bayesian Theory. John Wiley & Sons, Inc. (2000)
5. Sanders, F.: The verification of probability forecasts. Journal of Applied Meteorology **6** (1967) 756–761
6. Smith, C.A.B.: Consistency in statistical inference and decision. Journal of the Royal Statistical Society. Series B **23** (1961) 1–37
7. Savage, L.J.: Elicitation of personal probabilities and expectations. Journal of the American Statistical Association **66** (1971) 783–801
8. Winkler, R.L.: Probabilistic prediction: Some experimental results. Journal of the American Statistical Association **66** (1971) 678–685
9. Corradi, V., Swanson, N.R.: Predictive density evaluation. Technical Report 200419, Rutgers University, Department of Economics (2004)
10. Bremnes, J.B.: Probabilistic forecasts of precipitation in terms of quantiles using NWP model output. Monthly Weather Review **132** (2004) 338–347

11. Epstein, E.S.: A scoring system for probability forecasts of ranked categories. Journal of Applied Meteorology **8** (1969) 985–987
12. Hamill, T.M., Wilks, D.S.: A probabilistic forecast contest and the difficulty in assessing short-range forecast uncertainty. Weather and Forecasting **10** (1995) 620–631
13. Matheson, J.E., Winkler, R.L.: Scoring rules for continuous probability distributions. Management Science **22** (1976) 1087–1096
14. Bernardo, J.M.: Expected information as expected utility. The Annals of Statistics **7** (1979) 686–690
15. Staël von Holstein, C.A.S.: A family of strictly proper scoring rules which are sensitive to distance. Journal of Applied Meteorology **9** (1970) 360–364
16. Murphy, A.H.: The ranked probability score and the probability score: A comparison. Monthly Weather Review **98** (1970) 917–924
17. Murphy, A.H.: On the "ranked probability score". Journal of Applied Meteorology **8** (1969) 988–989