# Evaluating Predictive Uncertainty Challenge

Joaquin Quiñonero-Candela[1,2,3], Carl Edward Rasmussen[1], Fabian Sinz[1],
Olivier Bousquet[1,4], and Bernhard Schölkopf[1]

[1] Max Planck Institute for Biological Cybernetics,
Spemannstr. 38, D-72076 Tübingen, Germany
{carl, fabee, bernhard.schoelkopf}@tuebingen.mpg.de
[2] Fraunhofer FIRST.IDA, Kekuléstr. 7, D-12489 Berlin, Germany
joaquin@first.fraunhofer.de
[3] TU Berlin, SWT, Franklinstr. 28/29, D-10587 Berlin, Germany
[4] Pertinence, 32, rue des Jeûneurs, F-75002 Paris, France
olivier.bousquet@pertinence.com

**Abstract.** This Chapter presents the PASCAL[1] Evaluating Predictive Uncertainty Challenge, introduces the contributed Chapters by the participants who obtained outstanding results, and provides a discussion with some lessons to be learnt. The Challenge was set up to evaluate the ability of Machine Learning algorithms to provide good "probabilistic predictions", rather than just the usual "point predictions" with no measure of uncertainty, in regression and classification problems. Participants had to compete on a number of regression and classification tasks, and were evaluated by both traditional losses that only take into account point predictions and losses we proposed that evaluate the quality of the probabilistic predictions.

## 1 Motivation

Information about the uncertainty of predictions, or *predictive uncertainty*, is essential in decision making. Aware of the traumatic cost of an operation, a surgeon will only decide to operate if there is enough evidence of cancer in the diagnostic. A prediction of the kind "there is 99% probability of cancer" is fundamentally different from "there is 55% probability of cancer", although both could be summarized by the much less informative statement: "there is cancer". An investment bank trying to decide whether to invest or not in a given fund might react differently at the prediction that the fund value will increase by "10%± 1%" than at the prediction that it will increase by "10%± 20%", but it will in any case find any of the two previous predictions way more useful than the point prediction "the expected value increase is 10%". Predictive uncertainties are also used in active learning to select the next training example which will bring most information. Given the enormous cost of experiments

[1] Pattern Analysis, Statistical Modelling and Computational Learning (PASCAL) Network of Excellence, part of the IST Programme of the European Community, IST-2002-506778.

J. Quiñonero-Candela et al. (Eds.): MLCW 2005, LNAI 3944, pp. 1–27, 2006.

with protein binding chips, a drug making company will not bother making experiments whose outcome can be predicted with very low uncertainty.

Decisions are of course most often based on a loss function that is to be minimized in expectation. One common approach in Machine Learning is to assume knowledge of the loss function, and then train an algorithm that outputs decisions that directly minimize the expected loss. In a realistic setting however, the loss function might be unknown, or depend on additional factors only determined at a later stage. A system that predicts the presence of calcification from a mammography should also provide information about its uncertainty. Whether to operate or not will depend on the particular patient, as well as on the context in general. If the loss function is unknown, expressing uncertainties becomes crucial. Failing to do so implies throwing information away.

One particular approach to expressing uncertainty is to treat the unknown quantity of interest ("will it rain?") as a random variable, and make to predictions in the form of probability distributions, also known as *predictive distributions*. We will center our discussion around this specific representation of the uncertainty. But, how to produce reasonable predictive uncertainties? What is a reasonable predictive uncertainty in the first place?

Under the Bayesian paradigm, posterior distributions are obtained on the model parameters, that incorporate both the uncertainty caused by the noise, and by not knowing what the true model is. Integrating over this posterior allows to obtain the posterior distribution on the variables of interest; the predictive distribution arises naturally. Whether the resulting predictive distribution is meaningful depends of course on the necessary prior distribution, and one should be aware of the fact that inappropriate priors can give rise to arbitrarily bad predictive distributions. From a frequentist point of view, this will be the case if the prior is "wrong". From a Bayesian point of view, priors are neither wrong nor right, they express degrees of belief. Inappropriate priors that are too restrictive, in that they discard plausible hypotheses about the origin of the data, are sometimes still used for reasons of convenience, leading to unreasonable predictive uncertainties (Rasmussen and Quiñonero-Candela, 2005). If you believe your prior is reasonable, then the same should hold true for the predictive distribution. However, this distribution is only an updated belief — the extent to which it is in agreement with reality will depend on the extent to which the prior encompasses reality.

It is common in Machine Learning to not consider the full posterior distribution, but to rather concentrate on its mode, also called the Maximum a Posteriori (MAP) approach. The MAP approach being equivalent to maximum penalized likelihood, one could consider that any method based on minimizing a regularized risk functional falls under the MAP umbrella. The MAP approach produces predictions with no measure of the uncertainty associated to them, like "it will rain"; other methods for obtaining predictive uncertainties are then needed, such as Bagging for example (Breiman, 1996). More simplistic approaches would consist in always outputting the same predictive uncertainties, independently of the input, based on an estimate of the overall generalization error. This generalization error can in turn be estimated empirically by cross-validation, or theoretically by means Statistical Learning bounds on the generalization error. This simplistic approach should of course be regarded as a baseline, since any reasonable method that individually estimates predictive uncertainties depending on the input could in principle be superior.

It appears that there might not be an obvious way of producing good estimates of predictive uncertainty in the Machine Learning (or Statistical Learning) community. There is also an apparent lack of consensus on the ways of evaluating predictive uncertainties in the first place. Driven by the urgent feeling that it might be easier to validate the goodness of the different philosophies on the empirical battleground than on the theoretical, we decided to organize the Evaluating Predictive Uncertainty Challenge, with support from the European PASCAL Network of Excellence. The Challenge allowed different Machine Learning approaches to predictive uncertainty in regression and classification to be directly compared on identical datasets.

### 1.1 Organization of This Chapter

We begin by providing an overview and some facts about the Challenge in Sect. 2. We then move on to describing in detail the three main components of the Challenge: 1) in Sect. 3 we define what is meant by probabilistic predictions in regression and in classification, and explain the *format of the predictions* that was required for the Challenge, 2) in Sect. 4 we present the *loss functions* that we proposed for the Challenge, and 3) Section 5 details the five *datasets*, two for classification and three for regression, that we used for the Challenge. In Sect. 6 we present the results obtained by the participants, and in Sect. 7 we focus in more detail on the methods proposed by the six (groups of) participants who contributed a Chapter to this book. The methods presented in these six contributed chapters all achieved outstanding results, and all the dataset winners are represented. Finally, Sect. 8 offers a discussion of results, and some reflection on the many lessons learned from the Challenge.

## 2 An Overview of the Challenge

The Evaluating Predictive Uncertainty Challenge was organized around the following website: http://predict.kyb.tuebingen.mpg.de. The website remains open for reference, and submissions are still possible to allow researchers to evaluate their methods on some benchmark datasets.

The results of the Challenge were first presented at the NIPS 2004 Workshop on Calibration and Probabilistic Prediction in Machine Learning, organized by Greg Grudic and Rich Caruana, and held in Whistler, Canada, on Friday December 17, 2004. The Challenge was then presented in more depth, with contributed talks from some of the participants with best results at the PASCAL Challenges Workshop held in Southampton, UK, on April 11, 2005.

Using the website, participants could download the datasets (described in Sect. 5), and submit their predictions. Immediately after submission, the results obtained where displayed in a table, and sorted according to the loss (given in Sect. 4). Inspired by the NIPS 2003 Feature Selection Challenge (Guyon et al, 2005), we divided the Challenge chronologically into two parts. In the first part the competing algorithms were evaluated on a "validation set", with no limitation on the number of submissions. In the second part, shorter, of duration one week, the validation targets were made available and participants had to make a limited number of final submissions on the "test set". The final ranking of the Challenge was built according to the test performance.

The reason for having a validation set evaluation in the first part is to allow for temporary assessment and comparison of the performance of the different submissions. Simply put, to make the challenge more "fun" and encourage participation by immediately allowing to see how the participants were doing in comparison to others. To discourage participants from trying to guess the validation targets by making very many submissions, the targets associated to the validation set were be made public at the start of the second part of the Challenge, one week before the submission deadline. The participants could then use them to train their algorithms before submitting the test predictions.

Unlike in the NIPS 2003 Feature Selection Challenge (Guyon et al, 2005), participants did not need to submit on every of the five datasets to enter the final ranking. Individual rankings were made for each of the datasets. Indeed, as discussed in Sect. 5, the nature of the datasets was so diverse that one could hardly expect the same algorithm to excel in all of them. Our intention was to evaluate algorithms and methods rather than participants.

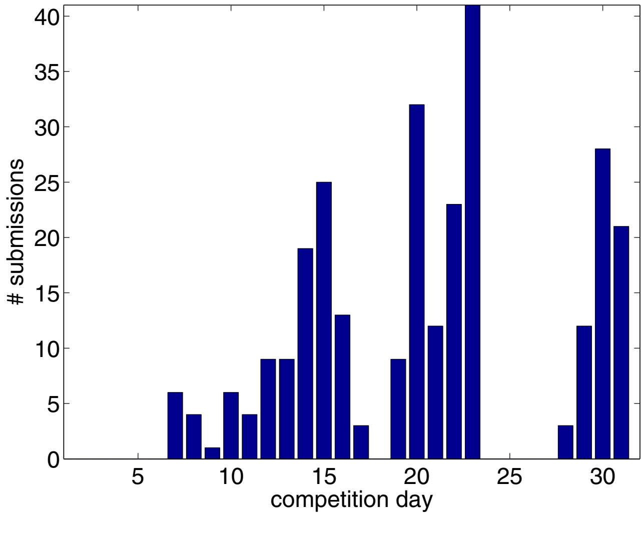


**Fig. 1.** Number of valid submissions on each day of the Challenge. Notice the break between the first and the second phase of the Challenge: the 68 valid test submissions were made on days 28 to 31.

The Challenge ran for 31 days, and attracted 20 groups of participants. A total of 280 submissions were made, of which 68 were "final" submissions on the test set. Figure 1 shows the number of submissions that were made each day of the Challenge.

The website opened for submissions on November 10 2004, and closed on December 10 2004. The second phase of the Challenge, with validation targets available and predictions to be made on the test inputs, started on December 3. The test results were made public on December 11. The website remains open for submission. After the closing deadline, some interesting submissions were made, which we include in the results section. Some of the contributed chapters were also written by participants who made very good post-Challenge submissions.

### 2.1 Design of the Website

When we designed the webpage for the *Evaluating Predictive Uncertainty* Challenge we had two objectives in mind. First, to build it in as flexible a way as possible way in order to be able to do minor changes very easily, like for example including additional losses, even during the competition. The second objective was a high degree of automation, to be able to for example give instant feedback whenever a submission was made. This way the participants were able to compare their preliminary scores with those the other participants.

The webpage consists of two separate parts, *appearance* and *functionality*, that are kept disjoint possible. An overview is given in Fig. 2. The website's appearance, was programmed with the use of PHP and CSS. PHP (*PHP Hypertext Preprocessor*) is a widely used open source script language, specially suited for easy website development, that can be embedded into HTML code. We used it to define the website's global structure on a higher level, that is to dynamically create HTML code. CSS (*cascading style-sheets*) is a simple standard for defining the style of a website. While the website's structure was created by PHP via HTML, CSS was used to define its final look. PHP was also used to implement a part of the website's functionality like managing the `ftp` upload and the interaction with external applications. The remaining functional part was implemented using Python and MySQL. Python is an interpreted, interactive, object-oriented programming language that combines a very clear syntax with a remarkable power. Although it is not open source, it is absolutely free. We used it in the project for mathematical computations, to compute the scores of the submissions, and to verify that the submissions were correctly formatted. MySQL is a key part of LAMP (*Linux, Apache, MySQL, PHP/Perl/Python*) and the world's most popular open source database. We used it to maintain a database of all information relevant to the submissions, as well ad the error scores under the different losses we used.

The appearance of the Challenge website is shown in 3. The structural framework of the website was implemented by the exclusive use of PHP. The structure of the navigation bar is defined in an separate file, used by formatting functions to

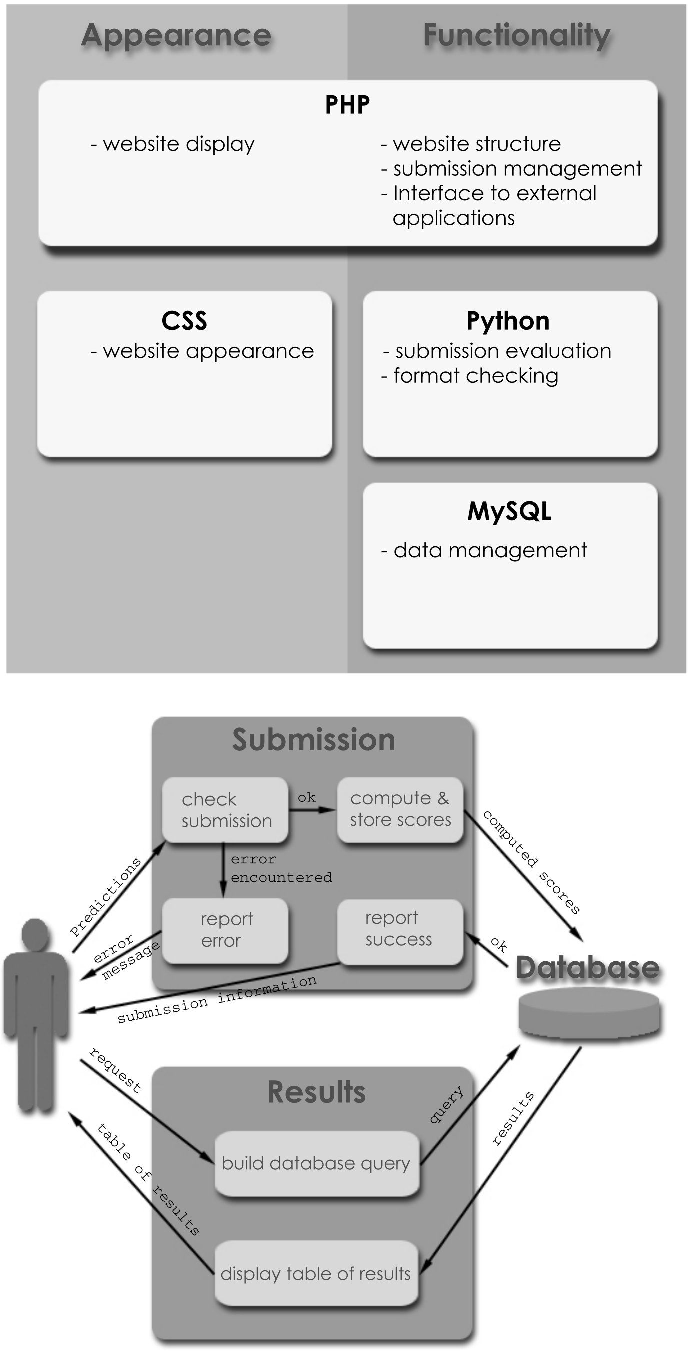
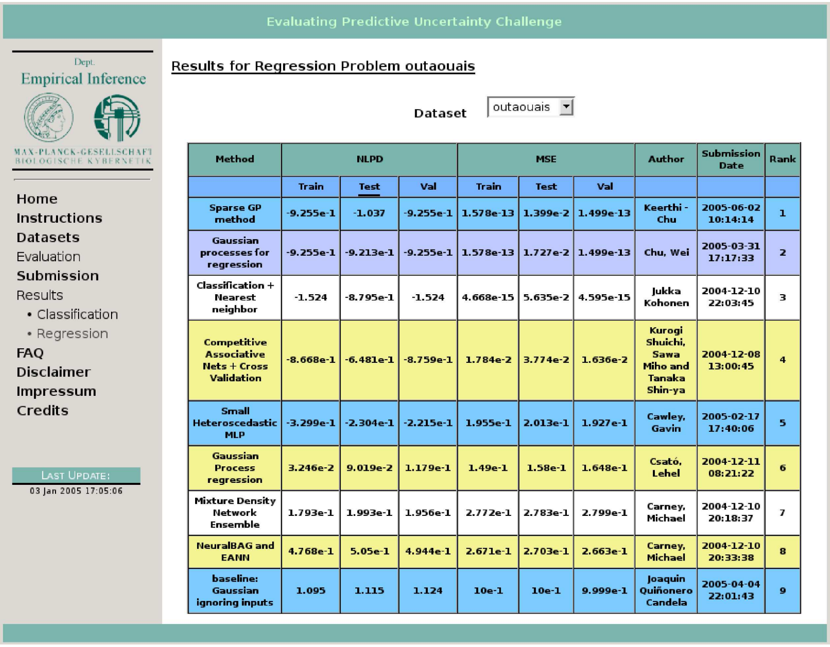


**Fig. 2.** Top: The website's functional units and the programming languages used to implement them. Bottom: Interaction control between user, website and database.



Results for Regression Problem outaouais

Dataset: outaouais

| Method | NLPD | | | MSE | | | Author | Submission Date | Rank |
|---|---|---|---|---|---|---|---|---|---|
| | Train | Test | Val | Train | Test | Val | | | |
| Sparse GP method | -9.255e-1 | -1.037 | -9.255e-1 | 1.578e-13 | 1.399e-2 | 1.499e-13 | Keerthi - Chu | 2005-06-02 10:14:14 | 1 |
| Gaussian processes for regression | -9.255e-1 | -9.213e-1 | -9.255e-1 | 1.578e-13 | 1.727e-2 | 1.499e-13 | Chu, Wei | 2005-03-31 17:17:33 | 2 |
| Classification + Nearest neighbor | -1.524 | -8.795e-1 | -1.524 | 4.668e-15 | 5.635e-2 | 4.595e-15 | Jukka Kohonen | 2004-12-10 22:03:45 | 3 |
| Competitive Associative Nets + Cross Validation | -8.668e-1 | -6.481e-1 | -8.759e-1 | 1.784e-2 | 3.774e-2 | 1.636e-2 | Kurogi Shuichi, Sawa Miho and Tanaka Shin-ya | 2004-12-08 13:00:45 | 4 |
| Small Heteroscedastic MLP | -3.299e-1 | -2.304e-1 | -2.215e-1 | 1.955e-1 | 2.013e-1 | 1.927e-1 | Cawley, Gavin | 2005-02-17 17:40:06 | 5 |
| Gaussian Process regression | 3.246e-2 | 9.019e-2 | 1.179e-1 | 1.49e-1 | 1.58e-1 | 1.648e-1 | Csató, Lehel | 2004-12-11 08:21:22 | 6 |
| Mixture Density Network Ensemble | 1.793e-1 | 1.993e-1 | 1.956e-1 | 2.772e-1 | 2.783e-1 | 2.799e-1 | Carney, Michael | 2004-12-10 20:18:37 | 7 |
| NeuralBAG and EANN | 4.768e-1 | 5.05e-1 | 4.944e-1 | 2.671e-1 | 2.703e-1 | 2.663e-1 | Carney, Michael | 2004-12-10 20:33:38 | 8 |
| baseline: Gaussian ignoring inputs | 1.095 | 1.115 | 1.124 | 10e-1 | 10e-1 | 9.999e-1 | Joaquin Quiñonero Candela | 2005-04-04 22:01:43 | 9 |

**Fig. 3.** Screenshot of the website's result page

determine the actual HTML code. That way new pages can easily be integrated in or removed from the existing website structure. Formatting functions are also used to put together the navigation bar itself, the contents of the different pages and to produce the final HTML code. All this is transparent to the users, all that is sent to them is pure HTML.

**Example Process Flow.** Let us describe the interaction between the different single components given above during the submission of predictions. This is also shown in the right diagram of figure 2. After checking the validity of informations entered by the user into the form of the submission page, the submission is uncompressed into a temporary directory and the format of the prediction files is checked. If errors are found at this stage, they are collected and jointly reported to the user. If no errors are found, the information related to the submission, like the description of the method, the submission time-stamp, the name of the participant, etc, are stored in a MySQL table and the evaluation scores are computed and inserted into the database. After moving the submitted file to a backup directory, a "successful submission" message is given to the user. At this point, the results of this submission are already available from the results table. If the user enters the results page, the evaluation scores for this challenge type and the default dataset are requested from database, sorted according to a default score, formatted by PHP and displayed. The user can change sorting the results according to a different loss, request the descriptions of other submissions, or access the results for a different dataset. Every time she does so, a completely new result table is requested from the database.

## 3 Probabilistic Predictions in Regression and Classification

The two modelling tasks addressed in the Challenge were binary classification and scalar output regression. For classification let the two classes be labelled by "+1" and "-1". Probabilistic predictions were required: for a each test input $\boldsymbol{x}_*$, the participant was required to provide the predictive (or posterior) probability of the label of that case being of class "+1":

$$p(y_* = +1|\boldsymbol{x}_*) \in [0, 1] \ , \qquad p(y_* = -1|\boldsymbol{x}_*) = 1 - p(y_* = +1|\boldsymbol{x}_*) \ . \tag{1}$$

For regression, participants were required to specify the probability density function of the output $y_*$ associated to the test input $\boldsymbol{x}_*$. Two possibilities are offered. The first, simpler one, is to describe the predictive density in a parametric form by means of a Gaussian density function. The predictive mean $m_*$ and variance $v_*$ need to be specified:

$$p(y_*|\boldsymbol{x}_*) \sim \frac{1}{\sqrt{2\pi v_*}} \exp\left(-\frac{\|y_* - m_*\|^2}{2v_*}\right) \ . \tag{2}$$

In some situations more complex predictive densities are appropriate (for example multi-modal). To allow participants to approximately specify any predictive density function we allowed them to describe it by means of any given number $N$ of quantiles $[q_{\alpha_1}, \dots, q_{\alpha_N}]$ such that:

$$p(y_* < q_{\alpha_j} \,|\, x_*) = \alpha_j \ , \qquad 0 < \alpha_j < 1 \ . \tag{3}$$

Imposing $0 < \alpha_j < 1$ avoids that some regions of the output space be given zero probability, which is unreasonable under the loss we use (see Sect. 4). The remaining probability mass, equal to $\alpha_1 + (1 - \alpha_N)$, is accounted for by two exponential tails of the form $\hat{p}(y|x) \propto \exp(-|y|/b)$.

Figure 4 gives an example of a predictive density being specified by quantiles. The participants need to specify the quantiles and their values. To recover the estimated predictive density $\hat{p}(y_*|x_*)$ from the quantiles, we need to distinguish between three cases:

1. if $q_{\alpha_1} \geq y_* > q_{\alpha_N}$ and $\alpha_i$ and $\alpha_{i+1}$ are such that $q_{\alpha_i} \geq y_* > q_{\alpha_{i+1}}$ then

$$\hat{p}(y_*|\boldsymbol{x}_*) = \frac{\alpha_{i+1} - \alpha_i}{q_{\alpha_{i+1}} - q_{\alpha_i}} \ , \tag{4}$$

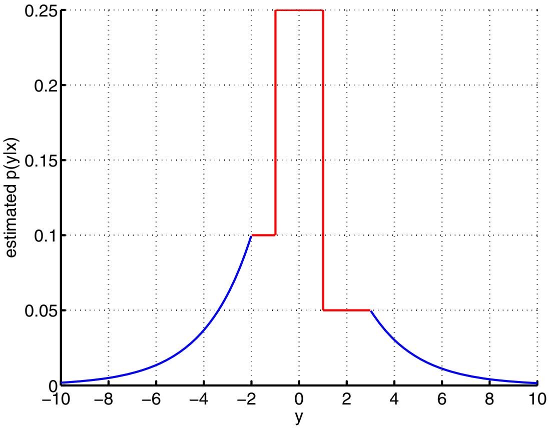


**Fig. 4.** Specifying the predictive density with quantiles. Example where the quantiles $q_{0.2} = -2$, $q_{0.3} = -1$, $q_{0.8} = 1$ and $q_{0.9} = 3$ are specified. The exponential tails guarantee that distribution integrates to 1.

2. if $y < q_{\alpha_1}$ then from the lower exponential tail:

$$\begin{aligned}\hat{p}(y_*|\boldsymbol{x}_*) &= z_1 \exp\left(-\frac{|y_* - q_{\alpha_1}|}{b_1}\right) \quad , \\ z_1 = \hat{p}(q_{\alpha_1}|\boldsymbol{x}_*) &= \frac{\alpha_2 - \alpha_1}{q_{\alpha_2} - q_{\alpha_1}} \quad , \\ \int_{-\infty}^{q_{\alpha_1}} \hat{p}(q_{\alpha_1}|\boldsymbol{x}_*) = \alpha_1 &\iff b_1 = \frac{\alpha_1}{z_1} \quad .\end{aligned} \tag{5}$$

3. if $q_{\alpha_N} \geq y$ then from an upper exponential tail:

$$\begin{aligned}\hat{p}(y_*|\boldsymbol{x}_*) &= z_N \exp\left(-\frac{|y_* - q_{\alpha_N}|}{b_N}\right) \quad , \\ z_N = \hat{p}(q_{\alpha_N}|\boldsymbol{x}_*) &= \frac{\alpha_N - \alpha_{N-1}}{q_{\alpha_N} - q_{\alpha_{N-1}}} \quad , \\ \int_{q_{\alpha_N}}^{\infty} \hat{p}(q_{\alpha_1}|\boldsymbol{x}_*) = (1 - \alpha_N) &\iff b_N = \frac{(1 - \alpha_N)}{z_N} \quad .\end{aligned} \tag{6}$$

In addition to the loss that takes into account the probabilistic nature of the predictions we will also compute the standard mean squared error loss (see Sect. 4). Since we only obtain predictive densities from the participants, we need to compute their mean, which is the optimal point estimator under the squared loss. For the case where quantiles are specified, computing the predictive mean is easily done by computing the following three contributions:

- The contribution of the quantiles to the mean is:

$$m_q = \sum_{i=1}^{N-1} \left[\frac{q_{\alpha_j} + q_{\alpha_{j+1}}}{2}\right] (\alpha_{j+1} - \alpha_j) \tag{7}$$

- The contribution of the lower exponential tail is:

$$m_{lt} = z_1 \int_0^\infty (q_{\alpha_1} - y_*) \exp\left(-\frac{y_*}{b_1}\right) = z_1 (q_{\alpha_1} b_1 - {b_1}^2) = \alpha_1 (q_{\alpha_1} - \frac{\alpha_1^2}{z_1}) \quad (8)$$

- Similarly, the contribution of the upper exponential tail is:

$$\begin{aligned} m_{ut} &= z_N \int_0^\infty (q_{\alpha_N} + y_*) \exp\left(-\frac{y_*}{b_N}\right) = z_N (q_{\alpha_N} b_N + {b_N}^2) \\ &= (1 - \alpha_N) \left[ q_{\alpha_N} + \frac{(1 - \alpha_N)^2}{z_N} \right] \end{aligned} \quad (9)$$

The estimate of the mean is obtained by adding up the terms:

$$m = m_q + m_{lt} + m_{ut} . \quad (10)$$

## 4 Loss Functions Proposed

Algorithms that perform well under classical losses, for hard decisions in classification and, scalar predictions in regression, do not necessarily perform well under losses that take into account predictive uncertainties. For this reason, we did evaluate the performance with losses of both natures.

In Sect. 4.1 we describe the losses used for classification, and in Sect. 4.2 those used for regression. We will denote the actual target associated to input $\boldsymbol{x}_i$ by $t_i$. In classification $t_i$ will take the value “+1” or “-1”, and in regression a value in $\mathbb{R}$. In Sect. 4.3 we justify the use of losses based on the logarithm for the evaluation of probabilistic predictions.

### 4.1 Losses for Classification

We used three losses for classification. The classic average classification error (relative number of errors, or 0/1 loss), the negative log probability (log loss, or negative cross entropy), and the “lift loss”. The final ranking was established according to the log loss, the two other losses being used only for comparison.

**The Average Classification Error**

$$L = \frac{1}{n} \left[ \sum_{\{i|t_i=+1\}} \mathbf{1}\{p(y_i = +1|\boldsymbol{x}_i) < 0.5\} + \sum_{\{i|t_i=-1\}} \mathbf{1}\{p(y_i = +1|\boldsymbol{x}_i) \geq 0.5\} \right] \quad (11)$$

where $\mathbf{1}\{z\}$ is an indicator function, equal to 1 if z=true, and to 0 if z=false. This is the classic 0/1 loss, obtained by thresholding the predictive probabilities at 0.5. Its minimum value is 0, obtained when no test (or validation) examples are missclassified; it is otherwise equal to the fraction of missclassified examples relative to the total number of examples.

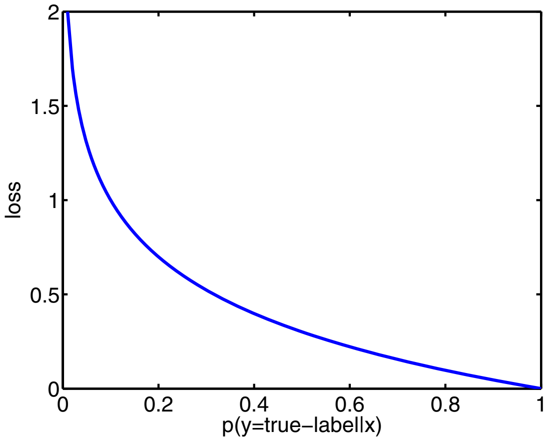


**Fig. 5.** NLP loss when predicting the class of a single test point that actually belongs to class "+1". Observe how the loss goes to infinity as the model becomes increasingly certain that the point belongs to the wrong class.

### The Negative Log Probability (NLP) Loss

$$L = -\frac{1}{n}\left[\sum_{\{i|t_i=+1\}} \log p(y_i = +1|\boldsymbol{x}_i) + \sum_{\{i|t_i=-1\}} \log\left[1 - p(y_i = +1|\boldsymbol{x}_i)\right]\right] \quad (12)$$

Notice that this loss penalizes both over and under-confident predictions. Overconfident predictions can be infinitely penalized, which should discourage predictive probabilities equal to zero or one. Zero is the minimum value of this loss, that could be achieved if one predicted correctly with 100% confidence. If one predicts otherwise, the worse one predicts, the larger the loss. This loss is also referred to as "negative cross-entropy loss". Figure 5 shows NLP loss incurred when predicting the class of a single point $\boldsymbol{x}_i$ that belongs to class "+1". The figure illustrates how the penalty becomes infinite as the predictor becomes increasingly certain that the test point belongs to the wrong class.

An interesting way of using this loss, is to give it relative to that of the random uninformative predictor, that always predicts 0.5. If one takes the difference between the log loss of a given algorithm and that of the random predictor one obtains the average gain in information (in bits if one takes base 2 logarithms).

**The LIFT Loss.** Although we decided not to rely on this loss to rank the submissions, which we ranked according to the log loss instead, we have decided to still explain it here, since it might be useful to some readers for other purposes. The "LIFT loss" is based on the area under the lift loss curve, and is minimum when that area is maximum. We define it in such a way that it is equal to 1 for an average random predictor. As we will explain, the LIFT loss is the area lost

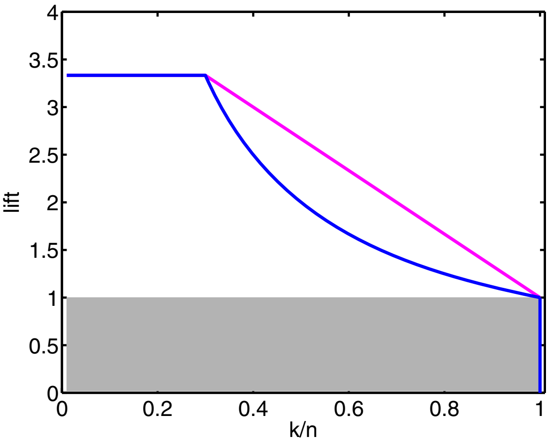


**Fig. 6.** Explaining the LIFT loss. The curve is the lift loss of the ideal predictor, and the line above it is a simple upper bound on it. The shaded region is the area under the average loss curve of a random predictor. The LIFT loss is defined as the ratio between two areas. The numerator is given by the area encompassed by the upper bound lift curve and the lift curve of the predictor being evaluated. The denominator is given by the area encompassed between the upper bound lift curve and that of the average random predictor. In this way the LIFT loss is the area lost relative to the ideal predictor, normalized by the loss of the average random predictor.

to the ideal predictor by the evaluated predictor, normalized by the area lost to the ideal predictor by the average random predictor. The reason why we build a loss based on the area under the lift loss, rather than looking at a particular value of the lift loss is similar to the reason why the area under the Area Under the ROC Curve (AUC) (Hanley and McNeil, 1982) has become a popular loss. In the absence of a specific point at which to evaluate the lift loss, we go for a measure that integrates over all its values.

The lift loss is obtained by first sorting the predictive probabilities with $p_i = p(y_i = +1|x_i)$ for the $n$ test points in decreasing order: $p_{s_1} \geq p_{s_2} \geq \ldots \geq p_{s_n}$. The obtained reordering contained in the $s_i$'s is applied to the test targets, and for $k = 1, \ldots, n$ the lift loss is defined as:

$$l(k/n) = \frac{1}{\bar{n}_+} \frac{1}{k} \sum_{i=1}^{k} \mathbf{1}\{t_{s_i} = +1\} \ , \qquad \bar{n}_+ = \frac{n_+}{n} \ , \tag{13}$$

where $n_+$ is the number of test examples that belong to class "+1". Notice that the lift loss is always, positive, that $l(1) = 1$ and that $l(k/n) \leq 1/\bar{n}_+$.

Figure 6 shows in blue the lift curve for an ideal predictor that would get a perfect ordering. In the figure we have set $\bar{n}_+ = 0.3$. For $0 \leq k/n \leq n_+$, all $y_{s_k}$="+1", and therefore the lift loss is equal to $1/n_+$ (from Eq. 13). For $k/n > n_+$, all $y_{s_k}$="-1" and therefore the lift loss is $l(k/n) = n/k$. The average lift loss of a random predictor is $l(k/n) = 1$ for all $k$. The shaded gray region in the figure represents the area under the average lift loss of such a random predictor, whose surface is equal to 1. In magenta we show a simple linear upper bound to the ideal lift curve, where the $n/k$ decaying part of the ideal loss is replaced by a linear upper bound.

The area under the upper bound curve to the lift loss of the ideal predictor is given by:

$$\mathrm{A_I} = 1 + \frac{1}{2}\left(\frac{1}{\bar{n}_+} - 1\right)(\bar{n}_+ + 1) \ , \tag{14}$$

while the area under the lift loss curve for the predictor we want to evaluate is given by

$$\mathrm{A} = \frac{1}{n}\sum_{k=1}^{n} l(k/n) \ . \tag{15}$$

In order to obtain a loss that is equal to 1 for the average random predictor, we define the LIFT loss as the ratio between the area lost by the predictor being evaluated and the area lost by the average random predictor:

$$L = \frac{\mathrm{A_I} - \mathrm{A}}{\mathrm{A_I} - 1} \tag{16}$$

Notice that $L \gtrapprox 0$ is the minimum loss, $L \approx 1$ is the average loss of a random predictor, and $L > 1$ is worse than random.

### 4.2 Losses for Regression

We used two losses to evaluate performance in the regression tasks. The first is the classic average normalized mean squared error (nMSE), which only takes into account the means of the predictive distributions (these are the optimal point estimates under the nMSE loss). The second loss is the average negative log predictive density (NLPD) of the true targets. We used the NLPD to rank the results of the participants.

**The nMSE Loss**

$$L = \frac{1}{n}\sum_{i=1}^{n} \frac{(t_i - m_i)^2}{\mathrm{var}(t)} \tag{17}$$

where $m_i$ is the mean of the predictive distribution $p(y_i|\boldsymbol{x}_i)$. Observe that we normalize the MSE wrt. to the variance of the true targets: predicting the empirical mean of the training targets, independently of the test input, leads thus to a normalized MSE of close to 1. In practice of course, we don't know the variance of the true test targets, and we simply estimate $\mathrm{var}(t)$ empirically by computing the sample variance of the test targets.

**The NLPD Loss**

$$L = -\frac{1}{n}\sum_{i=1}^{n} \log p(y_i = t_i|\boldsymbol{x}_i) \tag{18}$$

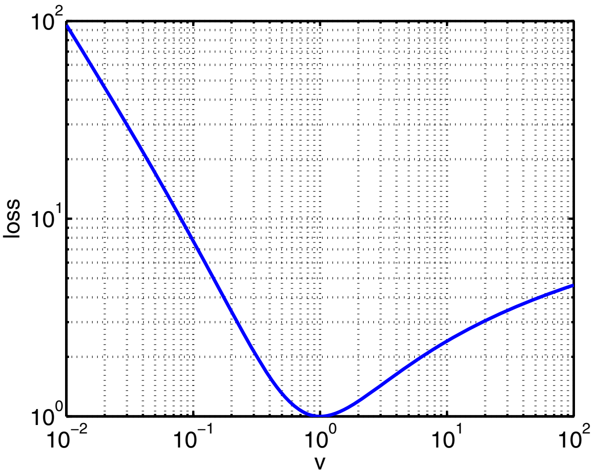


**Fig. 7.** NLPD loss (up to a constant) incurred when predicting at a single point with a Gaussian predictive distribution. In the figure we have fixed $\|t_i - m_i\|^2 = 1$ and show how the loss evolves as we vary the predictive variance $v_i$. The optimal value of the predictive variance is equal to the actual squared error given the predictive mean.

This loss penalizes both over and under-confident predictions. To illustrate this, let us take a closer look at the case of Gaussian predictive distributions. For a predictive distribution with mean $m_i$ and variance $v_i$ the NLPD loss incurred for predicting at input $\boldsymbol{x}_i$ with true associated target $t_i$ is given by:

$$L_i = \frac{1}{2}\left[\log v_i + \frac{(t_i - m_i)^2}{v_i}\right] + c \ , \tag{19}$$

where $c$ is a constant, independent of $m_i$ and $v_i$. Given $m_i$, the optimal value of $v_i$ is $(t_i - m_i)^2$. Figure 7 illustrates the variation of $L_i$ as a function of $v_i$ when $(t_i - m_i)^2 = 1$.

The NLPD loss favours conservative models, that is models that tend to be under-confident rather than over-confident. This is illustrated in Fig. 7, and can be deduced from the fact that logarithms are being used. An interesting way of using the NLPD is to give it relative to the NLPD of a predictor that ignores the inputs and always predicts the same Gaussian predictive distribution, with mean and variance the empirical mean and variance of the training data. This relative NLPD translates into a gain of information with respect to the simple Gaussian predictor described.

### 4.3 Discussion About Losses

Both log losses, NLP and NLPD, have the property of infinitely penalizing wrong predictions made with zero uncertainty. It might be argued that this is too strong a penalty. However, on the one hand if one is to take probabilistic predictions seriously, it might be desirable for consistency to discourage statements made with 100% confidence, that turn out to be wrong. On the other hand, think about the binary classification problem. If $n$ data points are observed, it might seem ambitious to have predictive uncertainties smaller than $1/n$: one has just not observed enough data to be more confident than that! So one obvious technique to avoid infinite penalties in classification would be to replace those predictive probabilities smaller than $1/n$ by $1/n$, and those larger than $1 - 1/n$ by $1 - 1/n$.

In regression, using the NLPD can be dangerous for certain specific types of outputs. Take for example the case where in a regression problem the outputs take values from a (potentially large) finite discrete set. One obvious strategy to minimize the NLPD in that case would be to distribute the available probability mass equally on tiny intervals one around each discrete output value. Since the NLPD only cares about density, the NLPD can be made arbitrarily small by decreasing the width of the intervals. Of course, there are machine precision limitations in practice. In this Challenge we had two datasets, Stereopsis (with outputs very close to discrete) and Gaze (with discrete outputs), where the NLPD could be exploited in this way (see Sect. 5). One way out of this issue would be to limit the minimum interval size when specifying predictive distributions by means of histograms, detailed in Sect. 4.2. The contributed Chapter by Kohonen and Suomela addresses this potential problem with the NLPD, and proposes an alternative loss for probabilistic predictions in regression.

For classification, the mutual information between the true class labels and the predicted class labels is sometimes used as a measure of performance. The mutual information however is an aggregate measure, that only depends on the conditional probabilities of predicting one class given that another class is true. It is totally insensitive to individual predictive probabilities, and therefore useless for our purposes. The Area Under the ROC Curve (AUC) is another common measure of performance, for classifiers that are able to output some number whose magnitude relates to the degree of belief that a point belongs to one class rather than to the other. The AUC score is fully determined by the *ordering* of these scalar predictions, and does not capture anything at all about calibration. In fact, the AUC score ignores the fact that the outputted numbers are probabilities. These are the reasons why we did not used the AUC score in this Challenge.

## 5 Datasets Proposed

We proposed two datasets for classification, and 3 for regression tasks for the Challenge, summarized in Table 1. All datasets are "real world data" in the sense that they were not synthesized nor fabricated, but rather measured or extracted from a real phenomenon.

The Gatineau and Outaouais datasets come from industry, and we are unfortunately not allowed to reveal any details about them. They were kindly donated by Yoshua Bengio, to whom we are very grateful.

**Table 1.** Datasets proposed for the Challenge. dim: input dimension. # Tr, # Val and # Test are respectively the number of training, validation and test cases. SV and ST are respectively the number of submissions during the validation and during the test phase of the Challenge.

| Classification | | | | | | |
|---|---|---|---|---|---|---|
| Name | dim | # Tr | # Val | # Test | SV | ST |
| Catalysis | 617 | 873 | 300 | 700 | 44 | 11 |
| Gatineau | 1092 | 3000 | 2176 | 3000 | 52 | 27 |
| Regression | | | | | | |
| Name | dim | # Tr | # Val | # Test | SV | ST |
| Stereopsis | 4 | 192 | 300 | 500 | 18 | 8 |
| Gaze | 12 | 150 | 300 | 427 | 50 | 16 |
| Outaouais | 37 | 20000 | 9000 | 20000 | 22 | 5 |

**Catalysis.** This dataset comes from the Yeast Functional Catalog[2], and was kindly prepared by Alexander Zien at the Max Planck Institute for Biological Cybernetics. The binary targets are obtained from assigning the functional categories of all yeast proteins to one of two classes. These two classes roughly correspond to presence (or absence) of catalytic activity. The inputs are gene expression levels of the genes encoding those proteins. The dataset is quite balanced, there are approximately as many positive as negative examples.

**Gatineau.** (Secret data) This is a very unbalanced binary classification dataset, with less than 10% positive examples. The data is also very hard to model, which makes the average classification (0/1 Loss) useless in practice. Models have to compete in terms of their probabilistic predictions.

**Stereopsis.** This dataset was collected at the Max Planck Institute for Biological Cybernetics, for a detailed account see (Sinz et al, 2004). The dataset was obtained by measuring the 3 dimensional location of a pointer attached to a robot arm by means of two high resolution cameras. The resulting 4 dimensional inputs correspond to the two pairs of coordinates on both cameras focal planes. Figure 8 illustrates one particularity of this dataset, that turns out to be of central importance when analyzing the results: when collecting the data, measurements were taken at a set of parallel planes, giving the impression that the variable to be estimated (the depth) was in fact naturally clustered around the discrete set of distances of the planes to the cameras.

**Gaze.** This dataset was also collected at the Max Planck Institute for Biological Cybernetics, with the help of Kienzle to whom we are very grateful. The targets are the pixel value of the horizontal position of a target displayed on a computer monitor. The corresponding 12-dimensional inputs are a set of measurements from head mounted cameras, that focus on markers on the monitor

[2] `http://mips.gsf.de`

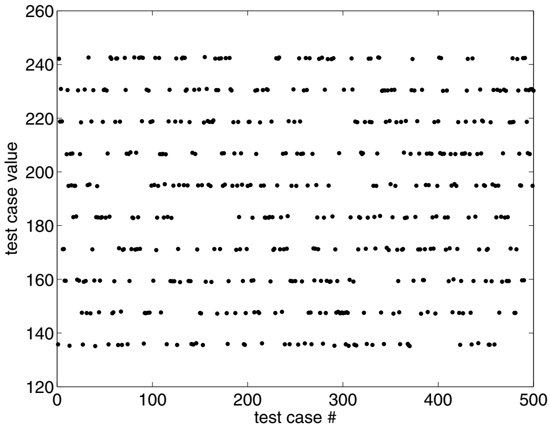


**Fig. 8.** Test targets of the Stereopsis datasets plotted against their index. The targets are clearly clustered around what appears to be 10 discrete values. In fact, there is structure within each "cluster". This discretization is solely an artifact of the way the data was collected, and has nothing to do with its nature.

and estimate the positions of the eyes of the subject looking at the monitor. This experimental setup is prone to severe outliers, since the cameras occasionally loose their calibration. It was indeed the case that there were severe outliers in the data, which the participants had to deal with, as reported in their contributed Chapters in the following. Another strong peculiarity of this dataset was that, being pixel values, the targets were discrete! This was exploited for instance by Kurogi et. al (see their contributed Chapter in this Volume) to "abuse" the NLPD loss. See Sect. 4.3 for a discussion on abusing the NLPD loss. This is just an example of the fact that losses and datasets should not be independent, but rather the opposite, see Sect. 8.

**Outaouais.** (Secret data) This is a regression dataset with very structured inputs, strongly clustered. This was noticed and exploited by Kohonen and Suomela, see Sect. 6.

## 6 Results of the Challenge

We now give the results of the Challenge for each of the datasets, following the order in which we presented them in table 1. We only provide a short list of the best performing entries. The complete tables can be found online, in the Challenge webpage: `http://predict.kyb.tuebingen.mpg.de`. The names of the participants who have contributed a Chapter to this Volume are shown in **bold** characters in the results tables. All dataset winners have contributed a chapter to this volume, in addition to some other participants with best results. The contributed Chapters are presented in Sect. 7.

The entries made before the validation targets were released are marked with a less than sign '<', meaning "before" the final submission period. The entries made after the deadline of December 10th 2004 (post-Challenge entries) are marked with a greater than sign '>', meaning "after". The remaining entries (with no mark) were made after the validation targets were available, and before the submission deadline of December 10th, 2004. The entries made before the validation targets were released only benefited from the training targets, while the final entries benefited both from the training and validation targets. The test targets have never been released, therefore the post-Challenge entries had only the training and validation targets available. Some of the participants who made post-Challenge entries have also contributed invited chapters to this volume.

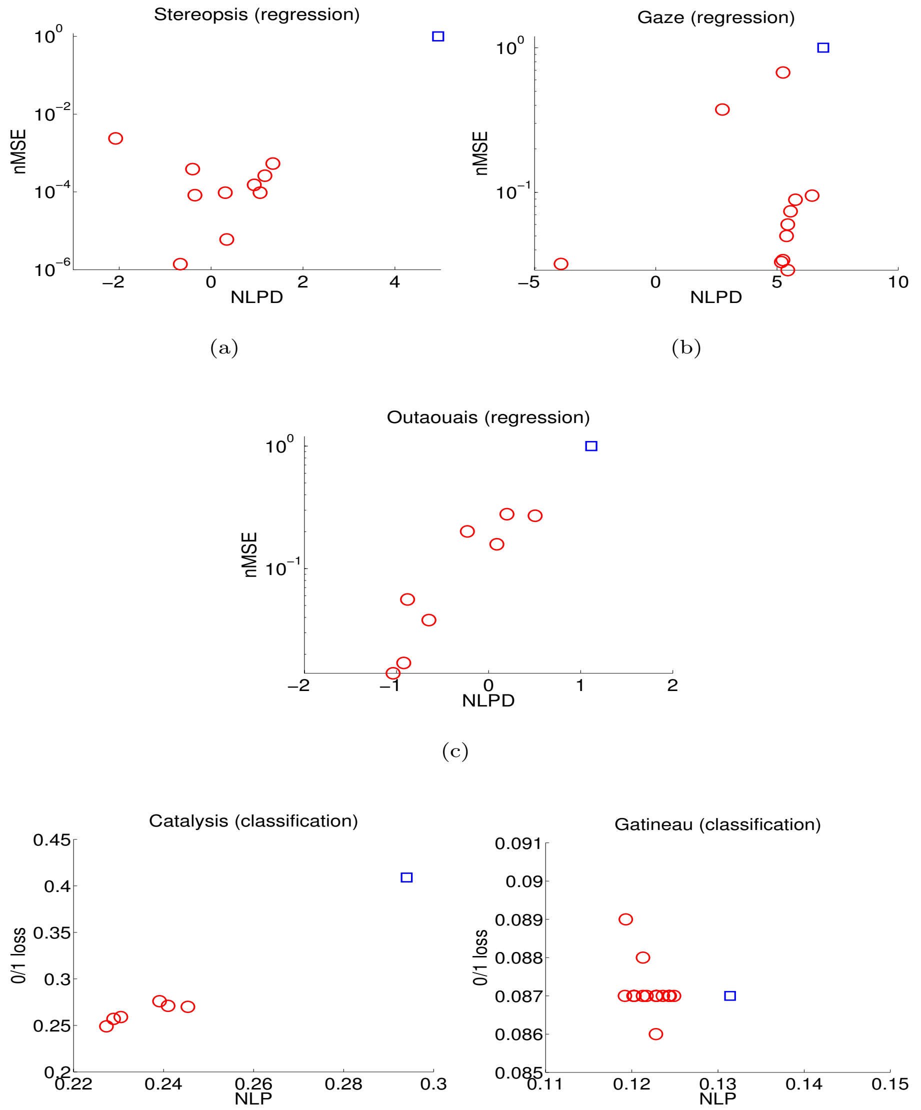
The results are compared to a baseline method. In classification, the baseline outputs the empirical training class frequencies independently of the inputs. In regression, the baseline is a Gaussian predictive distribution independent of the inputs, with mean and variance equal to the empirical mean and variance of the training targets. In Fig. 9 we present a scatter plot of the entries in the tables, one loss versus the other, for each dataset.

**Catalysis (Classification)**

| Method | NLP | 01L | Author |
|---|---|---|---|
| Bayesian NN | 0.2273 | 0.249 | **Neal, R** |
| < Bayesian NN | 0.2289 | 0.257 | **Neal, R** |
| SVM + Platt | 0.2305 | 0.259 | **Chapelle, O** |
| > Bagged R-MLP | 0.2391 | 0.276 | **Cawley, G** |
| > Bayesian Logistic Regression | 0.2401 | 0.274 | **Neal, R** |
| Feat Sel + Rnd Subsp + Dec Trees | 0.2410 | 0.271 | **Chawla, N** |
| Probing SVM | 0.2454 | 0.270 | Zadrozny, B & Langford, J |
| **baseline: class frequencies** | **0.2940** | **0.409** | |

(NLP: average negative log probability, 01L: average zero-one loss)

The winner was Radford Neal with Bayesian Neural Networks. Radford Neal also produced the second best entry, with the same model but learning only from the training targets during the "validation" part of the Challenge, therefore not benefitting from the validation targets. The third best submission is a support vector machine by Olivier Chapelle, that used Platt scaling (Platt, 1999) to produce calibrated probabilistic predictions. There is another support vector machine submission by Zadrozny and Langford, with lower ranking, that used Probing (Langford and Zadrozny, 2005) to obtain probabilistic predictions. Cawley's post-Challenge submission based on neural networks uses Bagging (Breiman, 1996) instead of Bayesian averaging. Bayesian logistic regression, a post-Challenge submission by Radford Neal, outperforms Nitesh Chawla's



(a)

(b)

(c)

(d)

(e)

**Fig. 9.** Visualization of results, non-probabilistic loss vs. probabilistic loss. The circles represent the participant's entries, the square the baseline method that ignores the inputs. Top (a-c): Regression, NLPD vs. nMSE. Outaouais is the dataset for which both losses are most highly correlated. For Stereopsis, the entry with lowest NLPD has the highest nMSE, and for Gaze there are a number of submissions with very low nMSE that have a very high NLPD: this might be due to the outliers present in this dataset. Bottom (d-e): Classification, NLP vs. 0/1 Loss. While for Catalysis both losses seem correlated, for Gatineau the 0/1 Loss is vacuous, and the only informative loss is really the NLP.

decision trees, which won the Gatineau dataset. This might be an indication that the performance of these methods is quite dataset dependent.

**Gatineau (Classification)**

| Method | NLP | 01L | Author |
|---|---|---|---|
| Feat Sel + Rnd subsp + Dec Trees | 0.1192 | 0.087 | **Chawla, N** |
| Feat Sel + Bagging + Dec Trees | 0.1193 | 0.089 | **Chawla, N** |
| Bayesian NN | 0.1202 | 0.087 | **Neal, R** |
| < Bayesian NN | 0.1203 | 0.087 | **Neal, R** |
| Simple ANN Ensemble | 0.1213 | 0.088 | Ohlsson, M |
| EDWIN | 0.1213 | 0.087 | Eisele, A |
| > Bayesian Logistic Regression | 0.1216 | 0.088 | **Neal, R** |
| > ANN with L1 penalty | 0.1217 | 0.087 | Delalleau, O |
| > CCR-MLP | 0.1228 | 0.086 | **Cawley, G** |
| Rnd Subsp + Dec Trees | 0.1228 | 0.087 | **Chawla, N** |
| Bagging + Dec Trees | 0.1229 | 0.087 | **Chawla, N** |
| > R-MLP | 0.1236 | 0.087 | **Cawley, G** |
| Probing J48 | 0.1243 | 0.087 | Zadrozny, B & Langford, J |
| > Bagged R-MLP (small) | 0.1244 | 0.087 | **Cawley, G** |
| SVM + Platt | 0.1249 | 0.087 | **Chapelle, O** |
| **baseline: class frequencies** | **0.1314** | **0.087** | |

(NLP: average negative log probability, 01L: average zero-one loss)

The 0/1 loss is not informative for the Gatineau dataset: under this loss, none of the methods beats a baseline classifier that always predicts class '-1'. The dataset is very unbalanced, with about only 9% examples from the less frequent class '+1', which lead most methods to also classify all test examples as members of class '-1'. In this situation probabilistic predictions become of great importance. The contestants managed to perform significantly better than the baseline classifier, which outputs a probability of belonging to class '+1' of 0.087, independently of the input. This probability is equal to the empirical class frequency. The two winning entries, by Nitesh Chawla, correspond to decision trees with feature selection and averaging. For the winner entry averaging consists in randomly sub-sampling the feature space, and for the second best entry in Bagging. Interestingly both ensemble methods give very similar performance. Feature selection appears to be decisive for improving the performance of the decision trees used, as can be seen from the decision tree entries without feature selection. Radford Neal's Bayesian Neural Network achieved the 3rd and 4th best results, when trained on training and validation, and training targets only respectively. Other Neural Networks are represented, in Delalleau and Cawley's post-Challenge entries. Interestingly, SVMs with Platt scaling perform much worse on this dataset than on Catalysis.

**Stereopsis (Regression)**

| Method | NLPD | nMSE | Author |
|---|---|---|---|
| Mixture of Bayesian Neural Nets | -2.077 | 2.38e-3 | **Snelson** & **Murray** |
| Compet Assoc Nets + Cross Val | -0.669 | 1.39e-6 | **Kurogi, S et al** |
| > Mixt of LOOHKRR Machines | -0.402 | 3.86e-4 | **Cawley, G** |
| > Gaussian Process Regression | -0.351 | 8.25e-5 | **Chapelle, O** |
| > Inflated Var MLP Committee | 0.309 | 9.59e-5 | **Cawley, G** |
| KRR + Regression on the variance | 0.342 | 9.60e-5 | **Chapelle, O** |
| < Hybrid: Neural Net | 0.940 | 1.52e-4 | Lewandowski, A |
| Mixture Density Network Ensemble | 1.171 | 2.62e-4 | Carney, M |
| **baseline: empirical Gaussian** | **4.94** | **1.002** | |
| Modelling the experimental setting | 209.4 | 2.49e-4 | **Kohonen & Suomela** |

(NPLD: negative log predictive density, nMSE: normalized mean squared error)

The winning entry, by Snelson and Murray, had the worst nMSE loss. However, this entry achieved the lowest NLPD by providing multi-modal predictive distributions, which is a natural choice given the clustered nature of the outputs, see Fig. 8. The entry by Kohonen and Suomela scored extremely low under the NLPD loss with unimodal Gaussian predictive distributions, with too small variances. As detailed in their chapter, this might not be a problem as long as the prediction falls within the right cluster. However, a single prediction that fell in the wrong cluster blew the NLPD loss. Excluding that case, Kohonen and Suomela's entry would have ranked first in Stereopsis. In their chapter, Kohonen and Suomela discuss the appropriateness of the NLPD loss. The second best entry, competitive associative networks, achieved a nMSE loss an order of magnitude smaller than the second best. It did not win because it provided under-confident unimodal, Gaussian predictive distributions. Mixtures of leave-one-out heteroscedastic kernel ridge regressors (LOOHKRR) (post-Challenge) was third, with unimodal Gaussian predictive distributions as well.

**Gaze (Regression)**

| Method | NLPD | nMSE | Author |
|---|---|---|---|
| Compet Assoc Nets + Cross Val | -3.907 | 0.032 | **Kurogi, S et al** |
| LLR Regr + Resid Regr + Int Spikes | 2.750 | 0.374 | **Kohonen** & **Suomela** |
| > LOOHKRR | 5.180 | 0.033 | **Cawley, G** |
| > Heteroscedastic MLP Committee | 5.248 | 0.034 | **Cawley, G** |
| Gaussian Process regression | 5.250 | 0.675 | Csató, L |
| KRR + Regression on the variance | 5.395 | 0.050 | **Chapelle, O** |
| < Neural Net | 5.444 | 0.029 | Lewandowski, A |
| Rand Forest with OB enhancement | 5.445 | 0.060 | Van Matre, B |
| NeuralBAG and EANN | 5.558 | 0.074 | Carney, M |
| Mixture Density Network Ensemble | 5.761 | 0.089 | Carney, M |
| **baseline: empirical Gaussian** | **6.91** | **1.002** | |

The winners, Kurogi et al. with competitive associative networks, achieved a NLPD loss spectacularly lower than that of the second best entry. The authors took advantage of a flaw of the NLPD loss for this dataset. Indeed, the outputs of the Gaze dataset take discrete values. Kurogi et al. provided predictive distributions by means of quantiles, to specify predictive histograms with one bin around each discrete output level. By making the bins small enough, any arbitrarily low value of the NLPD can be achieved. This inappropriateness of the NLPD loss for discrete-valued regression problems was also exploited by the second best entry, although to a lesser extent. More details are given in the chapter contributed by Kohonen and Suomela. The remaining entries did not abuse the NLPD loss. The lowest nMSE loss was achieved by Lewandowski with a neural network to estimate the predictive mean, and another network to estimate the predictive variance. This entry did not achieve excellent predictive uncertainties. It must be noted though, that it did only used the training targets, and not validation targets, for training. The best entry during made before the deadline, that did not abuse the NLPD loss was a Gaussian process by Lehel Csató. Leave-one-out heteroscedastic kernel ridge regression (LOHKRR), a post-Challenge submission, ranked third. This submission provided Gaussian predictive distributions, with one regressor to model the mean, and another to model the variance. A committee of multi-layer perceptrons, also post-Challenge, ranked fourth.

**Outaouais (Regression)**

| Method | NLPD | nMSE | Author |
|---|---|---|---|
| > Sparse GP method | -1.037 | 0.014 | Keerthi & Chu |
| > Gaussian Process regression | -0.921 | 0.017 | Chu, Wei |
| Classification + Nearest Neighbour | -0.880 | 0.056 | **Kohonen, J** |
| Compet Assoc Nets + Cross Val | -0.648 | 0.038 | **Kurogi S et al** |
| > Small Heteroscedastic MLP | -0.230 | 0.201 | **Cawley, G** |
| Gaussian Process regression | 0.090 | 0.158 | Csató, L |
| Mixture Density Network Ensemble | 0.199 | 0.278 | Carney, M |
| NeuralBAG and EANN | 0.505 | 0.270 | Carney, M |
| **baseline: empirical Gaussian** | **1.115** | **1.000** | |

The winning entry before the deadline, by Kohonen and Suomela, was not achieved by any conventional Machine Learning "black box" method, but rather by a "data-mining" approach. Nearest neighbours were used to make predictions. The input space was divided into clusters, and a cluster dependent distribution of the outputs was empirically estimated, for each cluster. Test predictive distributions were subsequently obtained by attributing the test input to one of the clusters. Kohonen and Suomela won in spite of not having the best nMSE score. Competitive associative networks ranked second, achieving the lowest nMSE loss before the submission deadline. It is interesting to see that two post-Challenge submissions outperform all the rest both in terms of nMSE and NLPD loss. These two submissions are based on Gaussian Processes: the winning entry managed to use the entire training set thanks to a sparse approximation, while the second used a full GP trained only on a subset of the training data.

## 7 Presentation of the Invited Chapters

This volume includes six additional contributed chapters, written by participants who achieved outstanding results in the Evaluating Predictive Uncertainty Challenge. All dataset winners and seconds are represented, as well as the authors of some of the post-Challenge submissions. There is high variety in the methods used. In classification, neural networks are used with Bayesian averaging by Radford Neal, and with Bagging by Gavin Cawley. Decision trees are used with Bagging and with random sub-samples of the inputs by Nitesh Chawla. Support vector machines and Gaussian Processes are used by Olivier Chapelle. In regression neural networks are used with Bayesian averaging by Ed Snelson and Iain Murray, and as committees by Gavin Cawley. Competitive associative networks with cross-validation, which can be seen as a gating network of local experts, are used by Shuichi Kurogi, Miho Sawa and Shinya Tanaka. Kernel methods are represented as Gaussian processes, in Olivier Chapelle's submission, and as heteroscedastic leave-one-out kernel ridge regression on the mean and on the variance by Gavin Cawley. Datamining is used in Jukka Kohonen's submission to the Outaouais dataset, where he used nearest neighbours together with a gating classifier. Jukka Kohonen and Jukka Suomela do also provide the single submission that was not made using a "black box" model: for Stereopsis, they deduce from the name of the dataset the physical underlying model of two cameras looking at one object. In their chapter, Jukka Suomela and Jukka Kohonen additionally provide with a discussion on the kind of losses that seem appropriate for evaluating probabilistic predictions.

The contributed Chapters are, in order of appearance in this volume:

**Bayesian Neural Networks**
*Radford M. Neal*
The author describes his use of Bayesian neural networks for the Catalysis and Gatineau datasets. Use was made of the author's publicly available[3] Flexible Bayesian Modelling (FBM) software. Since no information was revealed about the datasets at the time of the competition, the author decided to use vague priors with a complex neural network architecture. The author describes how model complexity is automatically adjusted through Bayesian averaging. In addition, the author comments on his post-Challenge entry, based on Bayesian logistic regression, which achieved a fair performance.

**A Pragmatic Bayesian Approach to Predictive Uncertainty**
*Iain Murray and Ed Snelson*
The authors explain how they used a Bayesian approach tailored to the Stereopsis dataset. First, a probabilistic classifier based on Radford Neal's FBM software

[3] `http://www.cs.utoronto.ca/∼radford/fbm.software.html`

serves as a soft gating network, that allows the combination of a mixture of local regression experts, each trained on a cluster of the Stereopsis outputs, see Fig. 8.

**Decision Trees with Feature Selection and Random Subspaces**
*Nitesh V. Chawla*
The author first explains why decision trees are not suited for probabilistic classification when used directly, nor when used with over-simplistic smoothing schemes such as Laplace or m-estimates. He then argues that ensemble methods allow to obtain large improvements in the predictive probabilities from decision trees. He discusses the use of two ensemble methods: random subsets and Bagging. The author also points out the importance that feature selection had for his good results. Finally, a discussion is given on how to improve performance on highly unbalanced datasets, such as Gatineau.

**Heteroscedastic Kernel Regression Methods**
*Gavin Cawley, Nicola Talbot and Olivier Chapelle*
The approach proposed in this work is to directly model the predictive distribution. For regression, a Gaussian predictive distribution is chosen. Its mean and variance are explicitly modelled separately by kernel ridge regression, and learning is achieved by assuming that the loss is the NLPD, and directly minimizing it. A leave-one-out scheme is used to avoid biased variance estimates.

**Competitive Associative Nets and Cross-Validation for Estimating Predictive Uncertainty on Regression Problems**
*Shuichi Kurogi, Miho Sawa and Shinya Tanaka*
Competitive associative nets (CANs) are presented. These are piece-wise linear approximations to non-linear functions. The input space is divided into a Voronoi tessellation, with a linear model associated to each region. For the Stereopsis and Outaouais datasets, Gaussian predictive distributions were provided, where the means were directly obtained from CANs trained to minimize the leave-one-out mean squared error. The variances were then estimated within the Voronoi regions by means of K-fold cross-validation. For the Gaze dataset, the authors took advantage of the discrete outputs to abuse the NLPD. The authors specified the predictive distribution by means of quantiles, and concentrated all the mass around tiny intervals centered around the integer output values.

**Lessons Learned in the Challenge: Making Predictions and Scoring Them**
*Jukka Suomela and Jukka Kohonen*
The authors present their winning entry for the Outaouais dataset: a pragmatic data-mining approach, based on a gating classifier followed by nearest neighbour regression. They also explain how they abused the NLPD loss on the discrete outputs Gaze dataset, in a similar but less extreme way than Kurogi et al. This motivates a very important discussion by the authors, on the more general problem of defining good losses for evaluating probabilistic predictions in regression. The authors propose to use of the continuous ranked probability score (CRPS), which does not suffer from the disadvantages of the NLPD loss.

## 8 Discussion

The wealth of methods successfully used by the participants to the Challenge indicates that there was not a single universally good way of producing good predictive uncertainties. However, averaging was common in many of the best submissions, see Fig. 10 for a qualitative impression in classification. Both classification winners used averaging: Radford Neal used Bayesian averaging of neural networks, and Nitesh Chawla decision trees averaged over random subsets of the inputs. Chawla's bagged decision trees achieved second position. In regression, averaging was used by the winning entry for the Stereopsis dataset with a Bayesian mixture of neural networks. Other successful entries for regression that used averaging include mixtures of kernel ridge regressors, bagged multi-layer perceptrons (MLPs) and committees of MLPs. Leave-one-out cross-validation was also found in many successful entries. It was used for example by Kurogi, Sawa and Tanaka with competitive associative networks (CANs), and by Cawley, Talbot and Chapelle with kernel ridge regression.

In terms of architectures, neural networks had a strong presence, and generally achieved very good results. Other architectures, like decision trees, Gaussian Processes and support vector machines also gave good results. Interestingly, an approach from datamining by Jukka Kohonen won the Outaouais regression dataset, later outperformed by two post-Challenge Gaussian Processes entries.

The Challenge revealed a difficulty inherent to measuring in general. While the goal was to evaluate "honest" predictive uncertainties, in practice the loss

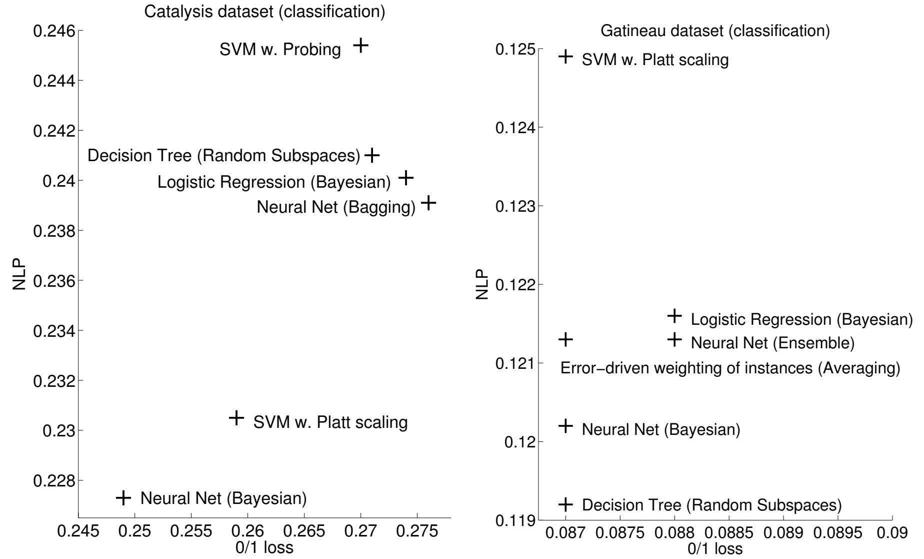


**Fig. 10.** Qualitative display of some classification results: 0/1 loss (average error rate) versus Negative Log Probability (NLP). Whenever averaging was used, the kind of averaging is indicated between brackets.

biased the predictive distributions of the participants. An example of this is the approach of Cawley, Talbot and Chapelle for regression, consisting in providing Gaussian predictive distributions tuned to minimize the NLPD loss. The authors would certainly have provided a different predictive distribution, if a different loss had been used.

The use of the NLPD loss turned out to be clearly inappropriate for the Gaze dataset. The outputs of this dataset take values from a finite discrete set. This encourages a simple strategy to achieve an arbitrarily small loss (the NLPD is unbounded from below). It is enough to specify a predictive histogram, with one bin encompassing each output discrete value. Making the bins narrow enough allows to arbitrarily increase the amount of probability density on the targets, and to therefore attain any arbitrarily small value of the NLPD, the being machine precision. This inadequacy of the NLPD for the Gaze dataset was exploited by two groups of participants, Kurogi, Sawa and Tanaka, and Snelson and Murray, who achieved respectively the best and second best results.

We have seen that the accuracy according to a point-prediction-based loss does not always give the same ranking as a loss which takes uncertainties into account, and that for some datasets like Gatineau, only the loss that evaluates probabilistic predictions is useful. However, it seems that defining good losses for probabilistic predictions is hard, since the losses might encourage strategies that are loss-dependent Maybe one way of encouraging unbiased and "honest" predictive distributions would be to apply several losses that encourage contradictory strategies. Another way could be not to reveal the loss under which predictions will be evaluated.

It would have been very interesting to empirically evaluate in this challenge a very recent paradigm for probabilistic predictions, based on "conformal predictions" (Vovk, Gammerman and Shafer, 2005).Conformal predictors are capable of producing accurate and reliable point predictions, while providing information about their own accuracy and reliability. This work was unfortunately published after the closing deadline of the Evaluating Predictive Uncertainty Challenge. Perhaps future competitions will allow to evaluate its practical utility.

## Acknowledgements

Many thanks to Olivier Chapelle for beta testing, to Yoshua Bengio for providing the Outaouais and Gatineau datasets, and to both for many discussions. Special thanks to Greg Grudic and Rich Caruana for accommodating the presentation of the challenge at the NIPS 2004 Workshop on Calibration and Probabilistic Prediction in Machine Learning.

We thank Alexander Zien for preparing for us the Catalysis dataset, and Wolf Kienzle for the Gaze dataset. We thank Sebastian Stark for his help setting up the mysql database for the Challenge. Many thanks to Jan Loderhose for allowing us to use his php framework as a basis to build ours.

We thank Florence d'Alché-Buc and Steve Gunn for putting together the Southampton PASCAL Challenges Workshop, where this Challenge was also presented and discussed. We are grateful for the support received by the PASCAL European Network of Excellence, IST Programme of the European Community, grant number IST-2002-506778. JQC and CER were partly supported by the German Research Council (DFG) through grant RA 1030/1. This work was partly done while JQC and OB where with the Max Planck Institute for Biological Cybernetics.

## References

Carl Edward Rasmussen and Joaquin Quiñonero-Candela. Healing the relevance vector machine by augmentation. In De Raet and Wrobel, editors, *Proceedings of the 22nd International Conference on Machine Learning*, pages 689–696, ACM Press, 2005.

Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

Isabelle Guyon, Steve Gunn, Asa Ben-Hur, and Gideon Dror. Result analysis of the nips 2003 feature selection challenge. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 545–552, Cambridge, Massachussetts, 2005. The MIT Press.

James A. Hanley and Barbara J. McNeil. The meaning and use ofthe Area under a Receiver Operating Characteristic ROC Curve. *Radiology*, 143(1):29–26, 1982.

F. Sinz, J. Quiñonero-Candela, G. H. Bakir, C. E. Rasmussen, and M.O. Franz. Learning depth from stereo. In Carl Edward Rasmussen, Henrich H. Bülthoff, Martin A. Giese, and Bernhard Schölkopf, editors, *Proc. 26 DAGM Pattern Recognition Symposium*, pages 245–252, Heidelberg, Germany, 2004. Springer.

John C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In Alexander J. Smola, Peter Bartlett, Bernhard Schölkopf, and Dale Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 61–74, Cambridge, MA, 1999. MIT Press.

John Langford and Bianca Zadrozny. Estimating class membership probabilities using classifier learners. In Robert G. Cowell and Zoubin Ghahramani, editors, *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, pages 198–205. Society for Artificial Intelligence and Statistics, 2005. (Available electronically at http://www.gatsby.ucl.ac.uk/aistats/).

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