Switching between Predictors with an Application in Density Estimation

Tim van Erven* Steven de Rooij Peter Grünwald
Tim.van.Erven@cwi.nl

Centrum voor Wiskunde en Informatica (CWI)
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands

April 5, 2007

Abstract

Universal coding is the standard technique for combining multiple predictors. This technique is explicitly used in minimum description length modeling, and implicitly in Bayesian modeling. Using universal coding, one can predict nearly as well as the best single predictor. When the predictors are themselves universal codes for models (sets of predictors) with varying number of parameters, however, we may often achieve smaller loss by switching between predictors in a different manner, which takes the local relative behaviour of the predictors into account. In this paper we present the switch-code, which implements this idea. It can be applied to coding, model selection, prediction and density estimation problems. As a proof of concept we give a particular application to histogram density estimation. We show that the switch-code achieves smaller redundancy, $O(n^{1/3} \log \log n)$, than standard universal coding, which achieves $O(n^{1/3}(\log n)^{2/3})$.

1 Introduction

In prediction and data compression tasks the goal is often to combine or choose between several prediction strategies, where more than one of these strategies is potentially successful. For example, in order to predict tomorrow's stock prices one might consult a number of experts as well as a couple of statistical models of varying sophistication, each of which might make different predictions. An important question is then how these predictions should be combined, in this case to maximise profit.

Here we consider sequential prediction strategies (predictors): functions from finite sequences over a sample space $\mathcal{X}$ to probability distributions on the next outcome. Such predictors are sometimes also called prequential forecasting systems [3]. We write $P(x_{n+1}|x_1, \ldots, x_n)$ for the probability of $x_{n+1} \in \mathcal{X}$ given previous observations $x^n = (x_1, \ldots, x_n) \in \mathcal{X}^n$, and we abbreviate $P(x^n|x^m) := \prod_{i=m+1}^{n} P(x_i|x^{i-1})$ and write $P(x^n)$ when $m = 0$. The performance of a predictor P on a sequence x^n is measured

*Corresponding author

by the accumulated log loss $-\log_2 P(x^n) = \sum_{i=1}^{n} -\log_2 P(x_i|x^{i-1})$. This loss measure is appropriate in, for example, data compression settings, since we can think of $-\log_2 P(x^n)$ as the number of bits needed to encode the data using the code based on P.

Universal Prediction In this setting, *universal prediction* (in data compression applications known as *universal coding*) is a widely used method of combining predictors. It has the advantage that its (accumulated) loss on x^n is never much higher than the loss $L_{\text{best}}(x^n)$ of the best original predictor. For example, consider two predictors P_1 and P_2. We construct a universal predictor using their Bayesian mixture as follows:

$$P_{\text{mix}}(x^n) := P_1(x^n)w(1) + P_2(x^n)w(2). \tag{1}$$

Here, w is a prior distribution on the original predictors. Clearly, for all $x^n \in \mathcal{X}^n$, we have $-\log_2 P_{\text{mix}}(x^n) - L_{\text{best}}(x^n) \geq 0$; but if, for instance, we use the uniform prior $w(1) = w(2) = 1/2$, we are also guaranteed that $-\log_2 P_{\text{mix}}(x^n) - L_{\text{best}}(x^n) \leq 1$.

The Central Idea Thus it is sensible to use universal prediction if we are satisfied with the loss of the best predictor under consideration, which is standard in minimum description length (MDL) and Bayesian approaches to prediction. However, we argue that it is often possible to combine predictors in such a way that the result achieves a lower loss than the best original predictor! This is possible if the identity of the best predictor *changes with the sample size in a predictable way*. This may occur, for example, if the predictors P_1 and P_2 are themselves universal predictors for nested models (sets of predictors) $\mathcal{M}_1$ and $\mathcal{M}_2$, $\mathcal{M}_1 \subset \mathcal{M}_2$. In this case P_1 typically predicts better at small sample sizes while P_2 predicts better eventually, because $\mathcal{M}_2$ has more parameters that need to be learned than $\mathcal{M}_1$. We provide an example in Figure 1, which shows the difference in accumulated loss for two predictors P_1 and P_2 on "The War of the Worlds" by H.G. Wells. P_1 is the Krichevsky-Trofimov (KT) [5] predictor for first-order Markov chains, P_2 is the KT predictor for second-order Markov chains. Clearly P_1 is the best predictor for about the first 50 000 outcomes, after which it is overtaken by P_2. Ideally, we should therefore predict the initial 50 000 outcomes using P_1 and the rest using P_2. However, P_{mix} only starts to behave like P_2 when its accumulated loss becomes lower than the accumulated loss of P_1. Thus, in the shaded area P_{mix} behaves like P_1 while P_2 accumulates less loss on those outcomes!

The Switch-Code For such cases, we have developed an alternative method to combine predictors P_1 and P_2 into a single predictor P_{sw}, which we call the *switch-code* [10]. Given a switch-point s at which to switch from P_1 to P_2, it predicts according to

$$P_{\text{sw}}(x_{n+1}|x^n, s) := \begin{cases} P_1(x_{n+1}|x^n) & \text{if } n < s \\ P_2(x_{n+1}|x^n) & \text{otherwise.} \end{cases} \tag{2}$$

The optimal switch-point, however, will typically depend on the data, which leads us to define the unconditional switch-code as

$$P_{\text{sw}}(x^n) := \sum_{s=0}^{\infty} P_{\text{sw}}(x^n|s)w(s) = \sum_{s=0}^{\infty} \prod_{j=1}^{n} P_{\text{sw}}(x_j|x^{j-1}, s)w(s), \tag{3}$$

where w is a prior on the sample size at which we should switch from P_1 to P_2. As Figure 1 shows, P_{sw} behaves like P_1 initially, but in contrast to P_{mix} it starts to mimic P_2 *almost immediately* after P_2 starts making better predictions (here we instantiated w to the universal prior on the integers, see the end of Section 4). In Section 2 we bound the individual sequence redundancy of the switch-code and in Section 3 we show how $P_{\text{sw}}(x^n)$ can be computed in $O(n)$ time.

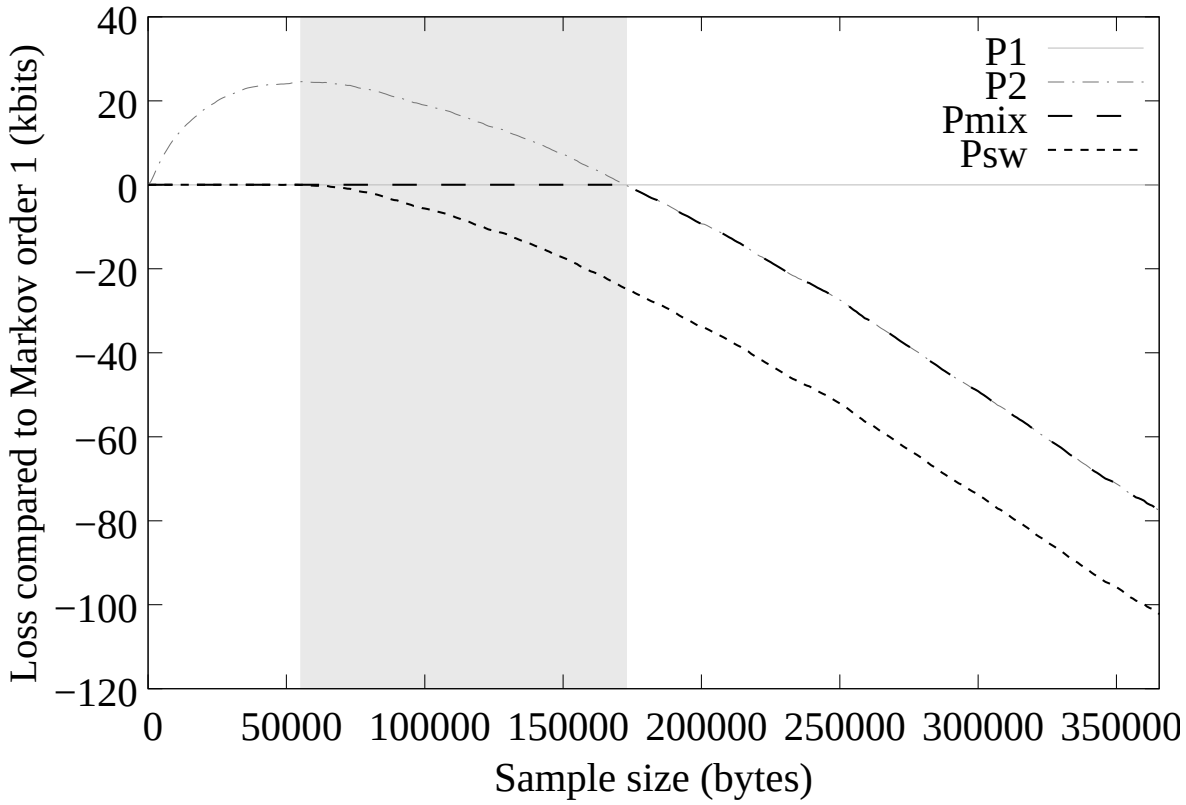

Figure 1: Accumulated loss difference on prefixes of "The War of the Worlds".

Applications In the discussion of our results in Section 5 we argue that the switch-code potentially outperforms standard universal predictors (like e.g. P_{mix}) whenever P_1 and P_2 are themselves universal predictors relative to some underlying models $\mathcal{M}_1$ and $\mathcal{M}_2$. This setting is sometimes called *twice-universal prediction* [9]. It is encountered in (a) Bayesian and MDL approaches to statistical model selection, prediction and density estimation problems [4]; and (b) in some state-of-the-art data compressors such as the context-tree weighting method [12]. In this preliminary paper, we merely highlight a single application where the use of the switch-code improves over other, existing methods: nonparametric density estimation based on histograms with bins of equal width.

Density estimation with histograms is considered by Rissanen, Speed and Yu (RSY) [8, 13], who show that in estimating a differentiable density that is bounded away from zero and infinity, it is asymptotically optimal in expectation to let the number of histogram bins increase as $\lceil n^{1/3} \rceil$ with the sample size n. Although they show that this strategy achieves the minimax optimal expected redundancy, $O(n^{1/3})$, using a fixed function of the sample size to determine the number of bins leads to concerns about the efficiency of the procedure when the estimated density does *not* satisfy the assumed conditions. In Section 4 we generalise the switch-code to switch between a sequence of (more than two) predictors that use increasingly many bins. There we also present a theorem, which will be proved in a subsequent full paper, showing that in the setting of RSY it achieves close to optimal redundancy, $O(n^{1/3} \log \log n)$, while still determining the number of histogram bins automatically from the data. In the same setting the standard Bayesian mixture/MDL approach only achieves redundancy $O(n^{1/3} (\log n)^{2/3})$ [13]. This means that, based on the switch-code, we obtain a flexible nonparametric density estimator that has not specifically been designed for the restricted RSY setting; nevertheless, if RSY's assumptions do apply, then in contrast

to the standard Bayesian/MDL estimator the switch-code estimator converges almost at the minimax-optimal rate that can be achieved within this restricted setting.

2 Redundancy of the Switch-Code

We measure the efficiency of the switch-code in terms of its individual sequence redundancy. For any two predictors P and P', the individual sequence redundancy of P' relative to P on data sequence x^n is defined as

$$\mathcal{R}(P, P', x^n) := \log P(x^n) - \log P'(x^n). \tag{4}$$

The following proposition provides an upper bound on the individual sequence redundancy of the switch-code relative to $P_{\mathrm{sw}}(\cdot|s)$ for any switch-point s and to the original predictors P_1 and P_2.

Proposition 1. *Let P_1 and P_2 be arbitrary predictors and let $P_{sw}(\cdot|s)$ and P_{sw} be the corresponding switch-code with known switch-point s and prior w, respectively, as in (2) and (3). Then, for any data sequence x^n,*

$$\mathcal{R}(P_1, P_{sw}, x^n) \leq -\log \sum_{s=n}^{\infty} w(s), \tag{5}$$

$$\mathcal{R}(P_{sw}(\cdot|s), P_{sw}, x^n) \leq -\log w(s). \tag{6}$$

Proof. The proof follows from the observations that $P_{\mathrm{sw}}(x^n|s) = P_1(x^n)$ for $s \geq n$, such that $P_{\mathrm{sw}}(x^n) \geq \sum_{s=n}^{\infty} w(s) \cdot P_1(x^n)$, and $P_{\mathrm{sw}}(x^n) \geq w(s) \cdot P_{\mathrm{sw}}(x^n|s)$ for all $s \in \mathbb{N}$. □

The proposition shows that the redundancy of the switch-code does not grow with the sample size, but only depends on the index of the optimal switch point s, i.e. the index where P_2 becomes a better predictor than P_1. Since $P_{\mathrm{sw}}(\cdot|s = 0) = P_2$, it also follows from (6) that the switch-code incurs at most constant individual sequence redundancy with respect to P_2. It is important to observe that $\mathcal{R}(P_2, P_{\mathrm{sw}}, x^n)$ can in fact be much smaller than zero in cases where $s = 0$ is not the optimal switch-point on the data.

The worst-case redundancy with respect to P_1 generally *does* grow with the sample size, at a rate which is determined by the prior w. One can ensure that the redundancy with respect to P_1 is bounded by a constant, by defining w such that it assigns prior mass to indices beyond any sample size that is ever obtained. To be perfectly safe, we could take this idea to an extreme by including infinity in the domain of switch-points; we could then specify $w(\infty) > 0$, which (since now $P_{\mathrm{sw}}(\cdot|s = \infty) = P_1$) guarantees constant worst-case redundancy of at most $-\log w(\infty)$ bits.

3 Computing the Switch-Code

As the definition, (3), of the switch-code involves a sum of infinitely many terms, it may not be immediately clear that it can in fact be computed efficiently. Here we show how the probability $P_{\mathrm{sw}}(x^n)$ of a data sequence x^n can be computed sequentially as a function of the predictions of P_1 and P_2, requiring only $O(n)$ computation time.

The probability can be decomposed as follows:

$$P_{\mathrm{sw}}(x^{n+1}) = \sum_{s=0}^{\infty} P_{\mathrm{sw}}(x^{n+1}|s)w(s)$$
$$= P_1(x_{n+1}|x^n) \sum_{s=n+1}^{\infty} P_{\mathrm{sw}}(x^n|s)w(s) + P_2(x_{n+1}|x^n) \sum_{s=0}^{n} P_{\mathrm{sw}}(x^n|s)w(s). \quad (7)$$

Dividing by $P_{\mathrm{sw}}(x^n)$, we find that the predictive distribution $P_{\mathrm{sw}}(x_{n+1} \mid x^n) = P_{\mathrm{sw}}(x^{n+1})/P_{\mathrm{sw}}(x^n)$ is a mixture of the predictions $P_1(x_{n+1}|x^n)$ and $P_2(x_{n+1}|x^n)$ with the following respective mixture weights $W_1(x^n)$ and $W_2(x^n)$:

$$W_1(x^n) = \frac{\sum_{s=n+1}^{\infty} P_{\mathrm{sw}}(x^n|s)w(s)}{\sum_{s=0}^{\infty} P_{\mathrm{sw}}(x^n|s)w(s)}, \qquad W_2(x^n) = 1 - W_1(x^n). \quad (8)$$

Our goal is to efficiently update these mixture weights each time a new outcome has to be predicted. To this end, observe that $P_{\mathrm{sw}}(x^n|s) = P_1(x^n)$ for $s \geq n$, so that the expression for $W_1(x^n)$ reduces to:

$$W_1(x^n) = \frac{P_1(x^n) \sum_{s=n+1}^{\infty} w(s)}{\sum_{s=0}^{n} P_{\mathrm{sw}}(x^n|s)w(s) + P_1(x^n) \sum_{s=n+1}^{\infty} w(s)}$$
$$= \frac{P_1(x^n) \sum_{s=n+1}^{\infty} w(s)}{\sum_{s=0}^{n} P_1(x^s)P_2(x^n|x^s)w(s) + P_1(x^n) \sum_{s=n+1}^{\infty} w(s)}.$$

For simplicity we now assume that the original predictors P_1 and P_2 assign non-zero probability to the data. If this assumption is violated, $W_1(x^n)$ can still be computed equally efficiently. Letting $\delta(x^n) := P_1(x^n)/P_2(x^n)$,

$$W_1(x^n) = \frac{P_1(x^n) \sum_{s=n+1}^{\infty} w(s)}{P_2(x^n) \sum_{s=0}^{n} P_1(x^s)/P_2(x^s)w(s) + P_1(x^n) \sum_{s=n+1}^{\infty} w(s)}$$
$$= \frac{\vdots\, \delta(x^n) \,\vdots \left(1 - \vdots \sum_{s=0}^{n} w(s) \vdots\right)}{\vdots \sum_{s=0}^{n} \delta(x^s)w(s) \vdots + \delta(x^n)\left(1 - \sum_{s=0}^{n} w(s)\right)}. \quad (9)$$

Storage is required only for the values of the subexpressions in the dotted boxes. They can be updated to their new values in constant time when the next outcome x_{n+1} becomes available. As $\delta(x^n)$ can take on extremely large or small values, it may be necessary to store logarithms rather than the actual values.

4 Histogram Density Estimation

Rissanen, Speed and Yu [8] consider density estimation based on histogram models with equal-width bins relative to a restricted set $\mathcal{T}$ of 'true' densities on the unit interval $\mathcal{X} = [0, 1]$. The restriction on $\mathcal{T}$ is that there should exist constants $0 < c_0 < 1 < c_1$ such that for every $f \in \mathcal{T}$, for all $x \in \mathcal{X}$, $c_0 \leq f(x) \leq c_1$ and $|f'(x)| \leq c_f$, where f' denotes the first derivative of f and c_f may depend on f, but not on x. The densities are extended to sequences by taking products: $f(x^n) := \prod_{t=1}^{n} f(x_t)$ for $x^n \in \mathcal{X}^n$.

The histogram models are defined by their predictive densities: A model with m equal-width bins $[0, a_1]$, $(a_1, a_2]$, ..., $(a_{m-1}, 1]$, with $a_i = i/m$, predicts according to

$$f_m(x_{n+1} \mid x^n) := \frac{n_{x_{n+1}}(x^n) + 1}{n + m} \cdot m, \quad (10)$$

where $n_{x_{n+1}}(x^n)$ denotes the number of outcomes in x^n that fall into the same bin as x_{n+1}. For $\mathbf{m} = m_0, \ldots, m_{n-1}$, Rissanen, Speed and Yu prove an upper bound on the expected redundancy of the joint density $f_{\mathbf{m}}(x^n) := \prod_{t=1}^{n} f_{m_{t-1}}(x_t \mid x^{t-1})$:

Theorem 2 (Theorem 2 from [8]). *Suppose $m_0 = 1$ and $\lceil (t/\alpha)^{1/3} \rceil \leq m_t \leq \lceil t^{1/3} \rceil$ for some fixed $\alpha \geq 1$ for $t = 1, \ldots, n-1$. Then for every $f \in \mathcal{T}$*

$$\frac{1}{n} E_{X^n \sim f^n} \left[\log \frac{f(X^n)}{f_{\mathbf{m}}(X^n)} \right] \leq A_f n^{-2/3}, \tag{11}$$

where A_f is a constant dependent on f.

In [8] the theorem is proved only for $\alpha = 1$, but their proof remains valid for larger values of α, as long as α does not vary with t. We require this generalisation of Theorem 2 in our proof of Theorem 5, which is stated below.

Theorem 2 shows that the number of histogram bins should increase approximately as $n^{1/3}$ with the sample size n. We will consider *non-decreasing* sequences $m_0 \leq \ldots \leq m_{n-1}$ that satisfy the conditions of the theorem, because these sequences all correspond to the same sequence of predictors P_1, P_2, ... and we can generalise the switch-code to switch between the first k of these predictors: Suppose we have a vector $\mathbf{s} = s_1, \ldots, s_{k-1}$ of $k-1$ switch-points. For convenience, also define $s_0 = 0$, $s_k = \infty$. Given these switch-points, the switch-code P_{sw} for k predictors predicts according to

$$P_{\mathrm{sw}}(\cdot \mid x^n, \mathbf{s}) := P_i(\cdot \mid x^n), \quad \text{where } i = \arg\min_j s_{j-1} \leq n < s_j, \tag{12}$$

and we again define the unconditional switch-code based on prior w by

$$P_{\mathrm{sw}}(x^n) := \sum_{\mathbf{s} \in \mathbb{N}^{k-1}} P_{\mathrm{sw}}(x^n | \mathbf{s}) w(\mathbf{s}), \tag{13}$$

where w is now a prior on $k-1$ switch-points. Note that (12) and (13) reduce to (2) and (3) if $k = 2$.

Any non-decreasing sequence $m_0 \leq \ldots \leq m_{n-1}$ may be fully specified by the indexes where it increases, i.e. by the switch-points between predictors. The reader may verify that the requirements of Theorem 2 are satisfied if the switch-points are chosen as in the following lemma.

Lemma 3 ($\mathbf{m}$ Specified by Switch-Points). *For arbitrary $\alpha \geq 1$ and k predictors, any non-decreasing sequence $\mathbf{m} = m_0, \ldots, m_{\alpha k^3}$ that is specified by $k-1$ switch-points $s_1 \leq \ldots \leq s_{k-1}$, satisfies the conditions of Theorem 2 if $i^3 \leq s_i \leq \alpha \cdot i^3$ for $i = 1, \ldots, k-1$.*

As with the two-predictor switch-code, we interpret the switch-points as the sample sizes at which we switch to the next predictor on the list. Hence the interpretation of $s_1, \ldots, s_{k-1}$ is clear if they are non-decreasing: $s_1 \leq \ldots \leq s_{k-1}$. The following lemma clarifies their interpretation if some parameters are decreasing, for instance if $s_3 < s_2$.

Lemma 4 (Decreasing Switch-Point Parameters). *Suppose $\mathbf{s} = s_1, \ldots, s_{k-1}$ are switch-point parameters such that $i^3 \leq s_i \leq \alpha \cdot i^3$ for $1 \leq i \leq k-1$. Let $\mathbf{s}' = s_1, \max\{s_1, s_2\}$, $\ldots, \max\{s_1, \ldots, s_{k-1}\}$. Then $P_{sw}(\cdot \mid \mathbf{s}) = P_{sw}(\cdot \mid \mathbf{s}')$ and $i^3 \leq s'_i \leq \alpha \cdot i^3$ for $1 \leq i \leq k-1$.*

Proof. $P_{\mathrm{sw}}(\cdot \mid \mathbf{s}) = P_{\mathrm{sw}}(\cdot \mid \mathbf{s}')$ can be verified directly from (12). For the second claim it is sufficient to verify that $\max\{s_i : i \leq j\} \leq \alpha \cdot j^3$ for $1 \leq j \leq k-1$, which can be done by induction on j. □

The switch-code for k predictors satisfies the following upper bound, which will be proved in the full paper.

Theorem 5 (Switch-Code Expected Redundancy)**.** *Let w be a mass function on $\mathbb{N}$ such that $\log 1/w(n) = \log(n+1) + O(\log\log n)$ and, for any number of models $k \geq \lceil n^{1/3} \rceil$, let P_{sw} denote the switch-code based on $w'(\mathbf{s}) = \prod_{i=1}^{k-1} w(s_i)$, where $\mathbf{s} = (s_1, \ldots, s_{k-1}) \in \mathbb{N}^{k-1}$. Then for every $f \in \mathcal{T}$*

$$\frac{1}{n} E_{X^n \sim f^n} \left[\log \frac{f(X^n)}{P_{sw}(X^n)} \right] \leq C_f n^{-2/3} \log\log n, \tag{14}$$

where C_f is a constant dependent on f.

Proof Outline. We use Theorem 2 to bound the expected redundancy for fixed sequences of switch-points. Our bound then arises from the additional cost of encoding the switch-points. This would take $O(n^{1/3} \log n)$ bits if we were to encode them exactly. However, taking $\alpha = 2$ we get by Lemma 3 that it is sufficient to specify the logarithm of the switch-points to integer precision, which reduces their codelength to $O(n^{1/3} \log\log n)$. □

An example of a mass function w that meets the requirements of the theorem is $w(n) = 2^{-L^*(n+1)}$, where $L^*(n)$ denotes the codelength of n for the universal code for the positive integers. $L^*(n) = c + \log n + \log\log n + \ldots$ for $n \in \mathbb{Z}^+$, where the sequence of nested logarithms includes all positive terms, $\mathbb{Z}^+$ denotes the positive integers and $c \approx 1.5$ is a positive constant [7].

5 Discussion

Related Work The idea of universal coding and prediction is to construct a "universal" predictor that performs nearly as well as the best single predictor in some given comparison class of individual predictors. The switch-code embodies a prediction strategy that, in some situations, performs considerably better. Namely, it aims to perform nearly as well as the best *sequence* of different predictors in the given class. The idea to construct prediction strategies that perform nearly as well as the best sequence of different predictors, rather than the best single predictor, is not at all new. It was considered earlier, by, for example, Bousquet and Warmuth [1]; in a data compression context, a similar idea was explored by Volf [11].

The main difference from these earlier works is that the switch-code has been specifically designed for settings where we would normally consider twice-universal coding. Then it commonly happens that the optimal predictor depends on the sample size, where, as the sample grows, the model on which to base the optimal predictor becomes more and more complex; and this phenomenon is exploited by the switch-code. In contrast, [1] describes a prediction strategy that is optimized for the situation where the optimal predictor changes over time in completely arbitrary ways, but not too often.

In contrast to standard universal codes, the switch-code selects a model not just by evaluating the overall past performance of a predictor, but also by considering its "local" behaviour compared to competing predictors, which makes it related to *leave-one-out cross-validation* [4]. Thus, it implicitly estimates not just the past behaviour, but also the future behaviour of each predictor. The fact that the two are related, but different, is investigated from a Bayesian perspective by Chickering and Heckerman [2]. In this context, we should also point out a note by MacKay [6] on the relation between Bayesian model comparison and leave-one-out cross-validation, where he predicts that "cross-validation would be the better method for predicting generalisation error".

Future Work Twice-universal prediction is commonly applied not just in statistical model selection and density estimation (which we considered here), but also in data compression. For example, the celebrated context-tree-weighting (CTW) [12] algorithm for data compression may be viewed as an instance of twice-universal prediction. Thus, it may be the case that some versions of the switch-code may improve state-of-the-art data compressors such as CTW. We are currently investigating this possibility.

Acknowledgements We would like to thank Yishay Mansour for directing our attention to the difference between past overall predictive performance and anticipated future predictive performance in a brief remark over lunch at the COLT 2005 conference, which initiated this line of research.

This work was supported in part 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] O. Bousquet and M. K. Warmuth. Tracking a small set of experts by mixing past posteriors. *Journal of Machine Learning Research*, 3:363–396, 2002.

[2] D. M. Chickering and D. Heckerman. A comparison of scientific and engineering criteria for Bayesian model selection. *Statistics and Computing*, 10:55–62, 2000.

[3] A. P. Dawid. Statistical theory: The prequential approach. *Journal of the Royal Statistical Society, Series A*, 147, Part 2:278–292, 1984.

[4] P. D. Grünwald. *The Minimum Description Length Principle*. The MIT Press, 2007.

[5] R. E. Krichevsky and V. K. Trofimov. The performance of universal encoding. *IEEE Transactions on Information Theory*, 27(2):199–207, March 1981.

[6] D. J. C. MacKay. Bayesian methods for neural networks – FAQ: Relation between Bayes and GCV. Retrieved April 3, 2007 from `http://www.inference.phy.cam.ac.uk/mackay/Bayes_FAQ.html`.

[7] J. Rissanen. *Stochastic Complexity in Statistical Inquiry*, volume 15 of *Series in Computer Science*. World Scientific, 1989.

[8] J. Rissanen, T. P. Speed, and B. Yu. Density estimation by stochastic complexity. *IEEE Transactions on Information Theory*, 38(2):315–323, March 1992.

[9] B. Ryabko. Twice-universal coding. *Problems of Information Transmission*, 20:173–177, 1984.

[10] T. A. L. van Erven. The momentum problem in MDL and Bayesian prediction. Master's thesis, University of Amsterdam, Amsterdam, The Netherlands, May 2006. Available from http://www.cwi.nl/∼erven/publications/.

[11] P. A. J. Volf. *Weighting Techniques in Data Compression: Theory and Algorithms*. PhD thesis, Technische Universiteit Eindhoven, December 2002.

[12] F. Willems, Y. Shtarkov, and T. Tjalkens. The context-tree weighting method: basic properties. 41:653–664, 1995.

[13] B. Yu and T. P. Speed. Data compression and histograms. *Probability Theory and Related Fields*, 92:195–229, 1992.