Does snooping help?

V. V. V'yugin [1]

Institute for Information Transmission Problems,
Russian Academy of Sciences,
Bol'shoi Karetnyi per. 19, Moscow GSP-4, 101447, Russia

and

Computer Learning Research Centre
Royal Holloway, University of London
Egham, Surrey TW20 0EX, England

Technical Report

CLRC-TR-99-06

December 1999

Royal Holloway University of London

Department of Computer Science
Egham, Surrey TW20 0EX, England

[1]Address for correspondence: Troitskaya street 10-275, Moscow 129090, Russia

Abstract

One of the most efficient way of improving the performance of learning algorithms is "snooping", i.e. using some information about the data to be predicted for choosing the parameters of the learning algorithm or the learning algorithm itself. Allowing different degrees of snooping α makes it possible to attain a better performance $\mathrm{Loss}_P(x)$ of a prediction strategy P on the given data set x. We study the "snooping curves" $L_x(\alpha) = \inf_{K(P)\leq\alpha} \mathrm{Loss}_P(x)$, where $K(P)$ is the Kolmogorov complexity of the prediction strategy P. Our main result states that "anything is possible": we prove that any non-increasing function is a limit point of the family functions $\{L_x\}$. Our framework is that of on-line prediction; for simplicity we assume that sequences x are binary.

1 Prediction strategies

Suppose we are given a data set $x_1, x_2, \ldots, x_i \ldots$; in this paper we consider only the simplest case, where $x_i \in \{0, 1\}$. Our goal is to predict the elements of the data set on-line: we predict x_1, then predict x_2 given x_1, ..., then predict x_{i+1} given $x_1, x_2, \ldots x_i$, etc. At every step i the loss is measured by the some loss function $\lambda(x_i, p_i)$, where the forecast is a real number p_i and the actual outcome is x_i. We suppose that this function is computable. For example, we consider the squared difference $\lambda(x_i, p_i) = (x_i - p_i)^2$ and the log-loss function $\lambda(x_i, p_i) = -\log p_i$ if $x_i = 1$ and $\lambda(x_i, p_i) = -\log(1 - p_i)$ if $x_i = 0$ (in the following log means the logarithm to the base 2). Other loss functions are considered in Vovk [6], Haussler, Kivinen, Warmuth [2] (log-loss, Hellinger, etc); Vovk and Watkins [8] (financial theory); Yamanishi [11] (logistic), etc]. An interesting theoretical possibility (and sometimes an unsightly phenomenon of the real life) is "snooping", or choosing the learning algorithm depending on the sequence x. Of course, we can always built x into the learning algorithm and attain ideal performance, so we have to limit snooping; to do so, we will use the notion of Kolmogorov complexity of a finite object (see, e.g., Li and Vitanyi [3]). Since the computable prediction strategy S is defined by some algorithm, by complexity $K(S)$ of this prediction strategy S we mean the Kolmogorov complexity of this algorithm. By definition of Kolmogorov complexity $K(S)$ is equal to the length of the shortest program computing values of S under some optimal way of programming.

It is natural to suppose that all predictions are given according to a *prediction strategy* (or *learning algorithm*) $p_i = S(x_1 x_2 \ldots x_{i-1})$. The total loss incurred by Predictor who follows the strategy S over the first n trials $x_1, x_2, \ldots, x_n$ is defined

$$\mathrm{Loss}_S(x_1 x_2 \ldots x_n) = \sum_{i=1}^{n} \lambda(x_i, S(x_1 x_2 \ldots x_{i-1})).$$

The main task is to minimize the total loss suffered on a sequence $x = x_1 x_2 \ldots x_n$ of outcomes. The corresponding game-theoretic interpretation is given in Vovk [7] or in Vovk and Watkins [8].

"Honest" learning algorithms S are those with small Kolmogorov complexity. Knowing x allows us to choose more complicated S (tuned for this particular x), but we will impose the restriction $K(S) \le \alpha$, where α is a positive constant reflecting the degree of snooping allowed.

We impose the following restrictions on the loss function $\lambda(\omega, p)$: there exists a positive computable real number b such that

- ⋄ (1) $\lambda(0, p) \ge b$ or $\lambda(1, p) \ge b$ for each p;
- ⋄ (2) There exists a computable real number $\hat{p}$ such that $\lambda(0, \hat{p}) = \lambda(1, \hat{p}) = b$;
- ⋄ (3) $\lambda(0, 0) = \lambda(1, 1) = 0$;

- ⋄ (4) The loss function is perfectly mixable (see below).

The log-loss function and squared difference satisfy these conditions with $b=1$ and $b=\frac{1}{4}$, accordingly.

These conditions looks unnatural but they follow from the more natural conditions communicated to author by Yura Kalnishkan:

- ⋄ (5) $\lambda(0,p)=\lambda(1,1-p)$ for all p (symmetry condition);
- ⋄ (6) $\lambda(0,\frac{1}{2})\neq 0$ (or $\lambda(1,\frac{1}{2})\neq 0$);
- ⋄ (7) $\lambda(0,0)=0$ (or $\lambda(1,1)=0$);
- ⋄ (8) $\lambda(0,p)\leq\lambda(0,p')$ and $\lambda(1,p)\geq\lambda(1,p')$ if $p\leq p'$ (monotonicity condition).

Without loss of generality we can normalize an arbitrary loss function $\lambda(\omega,p)$ satisfying conditions (1)-(4) $\lambda'(\omega,p)=b^{-1}\lambda(\omega,p)$ for some b (satisfying (1)-(4)). So, in the following we suppose that our loss function satisfies conditions (1)-(4) with $b=1$.

By "perfectly mixable" loss functions we mean the following. Let us fix $\eta>0$ (the *learning rate*). Put $\beta=e^{-\eta}\in(0,1)$. The loss function (game) is *perfectly η-mixable* if for every sequence $(p_1,p_2,\ldots)$ of predictions and a sequence $(r_1,r_2,\ldots)$ of nonnegative weights summing to 1 there exists a prediction p such that, for all y

$$\lambda(y,p)\leq\log_\beta\sum_{i=1}^{\infty}\beta^{\lambda(y,p_i)}r_i.$$

The loss function (game) is *perfectly mixable* if it is perfectly η-mixable for some $\eta>0$. It is known that many popular loss functions such as the log-loss function, square-loss function, Cover's loss function, long-short loss function, Kullback–Leibler loss function, χ^2 loss function, Hellinger loss function etc. (see, e.g., [5], [2], [6] [8]) are perfectly mixable.

The important construction in this field is the Vovk's aggregating algorithm AA [5], [6], [8], Section 2. In the case of perfectly mixable loss functions this algorithm given a finite sequence of predictive strategies $S_1,S_2,\ldots S_k$ and weights $r_1,r_2\ldots r_k$, summing to 1, allows us to define theirs "mixture" - a prediction strategy S such that

$$\mathrm{Loss}_S(x)\leq\log_\beta\sum_{i=1}^{k}r_i\beta^{\mathrm{Loss}_{S_i}(x)} \tag{1}$$

for all x, where $\beta=e^{-\eta}$ and the corresponding loss function is η-mixable. A variant of Vovk's construction is given in Section A below.

2 Snooping curves

We will consider loss functions satisfying conditions (1)-(4) with $b = 1$. For any data sequence x we consider the "snooping curve" (or "snooping function")

$$L_x(\alpha) = \inf_{K(S|n)\leq\alpha} \mathrm{Loss}_S(x),$$

where n is the length of the finite binary sequence x. In the following the length of x will be denoted by $l(x)$.

An analogous function (for the logarithmic loss function) as a measure of stochasticity was introduced by Kolmogorov in the seventies [1], [4], [3] and studied by Levin and V'yugin [9], [10].

Evidently, the snooping function is non-negative and non-decreasing. For any sequence x we have by (2) $L_x(\alpha + c) \leq l(x)$ for each non-negative α, c is a constant. A more refined estimate is

$$L_x(\alpha + c) \leq n - \alpha$$

for all positive $\alpha \leq n$, where $n = l(x)$ is the length of x and c is a non-negative constant. To prove this inequality we define the following prediction strategy

$$S(x_1x_2\dots x_{i-1}) = \begin{cases} x_i, \text{ if } i \leq \alpha, \\ \hat{p}, \text{ otherwise }, \end{cases}$$

where the number $\hat{p}$ satisfies the condition (2). By definition $K(S|n) \leq \alpha + c$ for some positive constant c and $\mathrm{Loss}_S(x) \leq n - \alpha$.

We present the general description of all possible forms of snooping functions for different x. We prove that for any finite simple function $f(\alpha)$ (i.e. function, whose domain is a union of finite number of intervals, and the function is constant on each of them) there is a finite binary sequence x such that $L_x(\alpha)$ is close to $f(\alpha)$.

Theorem 1 *For any $\delta > 0$ the constants c_1 and c_2 exist such that for any positive n and k and for any sequence*

$$\alpha_1, \dots, \alpha_k, m_1, \dots, m_k$$

of positive integer numbers satisfying

$$c_1 \leq \alpha_1 < \dots < \alpha_k$$

and

$$n > m_1 > \dots > m_k,$$

there exists a binary sequence x of the length n such that for $1 \leq i \leq k+1$ and all α satisfying the inequality

$$\alpha_{i-1} + \log_2 i + c_2 \leq \alpha \leq \alpha_i$$

the following estimate holds

$$m_i - (k-i)(c_\eta \alpha_k + 1) \le L_x(\alpha) \le m_i,$$

where $c_\eta = (1+\delta)(\eta \log e)^{-1}$ *and we put* $\alpha_0 = 0$, $\alpha_{k+1} = \infty$, $m_{k+1} = 0$.

Proof. It is naturally to include the information on the sequence $I = (\alpha_1, \ldots, \alpha_k, m_1, \ldots, m_k)$ in the condition, i.e. we consider the conditional Kolmogorov complexity $K(S|n, I)$ of a prediction strategy given n and I (see Li and Vitanyi [3]). For simplicity of the exposition we omit these conditions and write $K(S)$.

For any prediction strategy P we consider the loss incurred over a fragment $y_{m+1} \ldots y_k$ of a sequence $y_1 \ldots y_n$, where $m < k \le n$.

$$\mathrm{Loss}_P(y_{m+1} \ldots y_k | y_1 \ldots y_m) = \sum_{i=m+1}^{k} \lambda(y_i, P(y_1 \ldots y_{i-1})).$$

Define $m'_i = m_i - m_{i+1}$, where $i = 1, 2 \ldots, k$. For any α_i let $P_1, P_2, \ldots, P_{k(i)}$ be all computable prediction strategies P such that $K(P) \le \alpha_i$. We have $k(i) < 2^{\alpha_i + 1}$. By means of AA (Section A) given the sequence $m_1, \ldots, m_{k+1}$ and $1 \le i \le k$ we can define an averaging prediction strategy P_{α_i} such that for any $y = y_1 \ldots y_n$

$$\beta^{\mathrm{Loss}_{P_{\alpha_i}}(y_{m+1} \ldots y_n | y_1 \ldots y_m)} \ge \sum_{j=1}^{k(i)} (k(i))^{-1} \beta^{\mathrm{Loss}_{P_j}(y_{m+1} \ldots y_n | y_1 \ldots y_m)}, \qquad (2)$$

where $m = m_1 - m_i$. By construction of AA we can compute all values of $P_{\alpha_i}(z)$, where $l(z) \le n$, with an arbitrary degree of accuracy by means the program p_i with the following properties. The length of p_i is $\le \alpha_i$ and this program has the maximal terminating time among all programs computing values $P_j(z)$ for all $j \le k(i)$ and all z of the length $\le n$ with given corresponding degree of accuracy. The important feature of this program is that using p_i we can compute also all values of $P_{\alpha_j}(z)$ for all $j \le i$.

Let us define a sequence y of the length m_1 by mathematical induction on i. Let $1 \le i \le k$ and $y_1 \ldots y_{m_1 - m_i}$ is defined. We suppose that this sequence is empty when $i = 1$.

For $m_1 - m_i < s \le m_1 - m_{i+1}$ define

$$y_s = \begin{cases} 1, & \text{if } \lambda(1, P_{\alpha_i}(y_1 \ldots y_{s-1})) > 1 - 2^{-(s+1)}, \\ 0, & \text{if } \lambda(0, P_{\alpha_i}(y_1 \ldots y_{s-1})) > 1 - 2^{-(s+1)}. \end{cases}$$

Since these conditions are overlapped, we choose that the first was computed.

So, for any $1 \le i \le k$ we defined a sequence $y_1 \ldots y_{m_1 - m_{i+1}}$.

The total loss incurred over $y_{m_1 - m_j + 1} \ldots y_{m_1 - m_{j+1}}$ by the prediction strategy P_{α_j} is estimated from below

$$\mathrm{Loss}_{P_{\alpha_j}}(y_{m_1 - m_j + 1} \ldots y_{m_1 - m_{j+1}} | y_1 \ldots y_{m_1 - m_j}) \ge m'_j - 1.$$

By (2) for any $i \le j \le k$ it holds

$$\mathrm{Loss}_P(y_{m_1-m_j+1}\dots y_{m_1-m_{j+1}}|y_1\dots y_{m_1-m_j}) \ge m'_j - c_\eta\alpha_j - 1 \qquad (3)$$

for each prediction strategy P such that $K(P) \le \alpha_j$.

Put $y = y_1\dots y_{m_1}$ and $x = y0^{n-m_1}$ (0 is concatenated $n - m_1$ times).

We have $K(P) \le \alpha_i \le \alpha_{i+1} \le \dots \le \alpha_k$. Summing the inequalities (3) for $j = i, i+1\dots k$ we obtain

$$\begin{aligned}
\mathrm{Loss}_P(y_1\dots y_n) = \mathrm{Loss}_P(y_1\dots y_{m_1-m_i}) &+ \\
\mathrm{Loss}_P(y_{m_1-m_i+1}\dots y_{m_1-m_{i+1}}|y_1\dots y_{m_1-m_i}) &+ \\
\mathrm{Loss}_P(y_{m_1-m_{i+1}+1}\dots y_{m_1-m_{i+2}}|y_1\dots y_{m_1-m_{i+1}}) &+ \dots + \\
\mathrm{Loss}_P(y_{m_1-m_k+1}\dots y_{m_1-m_{k+1}}|y_1\dots y_{m_1-m_k}) &\ge \\
m'_i + m'_{i+1} + \dots + m'_k - (k-i)(c_\eta\alpha_k + 1) &\ge \\
m_i + (k-i)(c_\eta\alpha_k + 1)&.
\end{aligned}$$

for each strategy P such that $K(P) \le \alpha_i$.

In order to obtain the upper bounds for any $i = 1, 2\dots k$ consider the prediction strategy U_i, depending on $y_1\dots y_{m_1-m_{i+1}}$, such that for any sequence $z_1\dots z_{j-1}$

$$U_i(z_1z_2\dots z_{j-1}) = \begin{cases} y_j, \text{ if } 1 \le j \le m_1 - m_{i+1}, \\ \hat{p}, \text{ if } m_1 - m_{i+1} < j \le m_1, \\ 0, \text{ otherwise }, \end{cases}$$

where y_j is the j-th bit of the sequence y, and the number $\hat{p}$ satisfies the condition (2). By definition

$$\mathrm{Loss}_{U_i}(x) = m_{i+1}.$$

We have also $K(U_i) \le \alpha_i + 2\log_2 i + c$ for some constant c, since the prediction strategy U_i can be defined by the number i and by the program p_i. $\square$

Let $\nu(n)$ and $\mu(n)$ be two non-negative non-decreasing unbounded functions such that $\mu(n) < n$ for all n and $\nu(n) = o(\mu(n))$ as n tends to infinity. We will study the asymptotic behaviour of snooping curve in the rectangle $0 \le \alpha \le \nu(n)$ and $0 \le \beta \le \mu(n)$. To do this, consider the family of "normalized" functions

$$f_x(t) = \frac{L_x(\nu(n)t)}{\mu(n)},$$

where $n = l(x)$ and $0 \le t \le 1$.

We consider the space $L_\infty = L_\infty([0,1])$ with the norm $||f|| = \sup\limits_{x\in[0,1]} |f(x)|$.

Then from Theorem 1 it follows

Corollary 1 *Each non-increasing function whose graph is in the unit square is a limit point in metrics L_∞ of the family functions $\{f_x\}$, where $x \in \Xi$.*

The most interesting case is $\nu(n) = O(\log n)$.

Notice, that Theorem 1 cannot explain the situation in the case, where $\nu(n) = o(\mu(n))$ does not hold. It is an open problem to describe snooping curves in the case when the values of α and $L_x(\alpha)$ are of the same order.

Acknowledgements

I am grateful to Volodya Vovk for an idea of "snooping" interpretation and to Yura Kalniskan for useful discussions.

A Appendix: Aggregating algorithm

For any prediction strategy P we consider the loss incurred over a fragment $y_{m+1} \dots y_n$ of a sequence $y_1 \dots y_n$, where $m < n$.

$$\mathrm{Loss}_P(y_{m+1} \dots y_n | y_1 \dots y_m) = \sum_{i=m+1}^{n} \lambda(y_i, P(y_1 \dots y_{i-1})).$$

We present the Vovk's aggregating algorithm in a form convenient to us.

Proposition 1 *Let a loss function $\lambda(\omega, p)$ be η-mixable and $\beta = e^{-\eta}$. For any $1 \le m \le n$, any finite sequence $P_1, P_2, \dots, P_k$ of computable prediction strategies and a sequence $r(1), \dots, r(k)$ of nonnegative real numbers, summing to 1, a computable prediction strategy P exists such that for any binary sequence $y_1 \dots y_n$*

$$\mathrm{Loss}_P(y_{m+1} \dots y_n | y_1 \dots y_m) \le \log_\beta \sum_{i=1}^{k} r(i) \beta^{\mathrm{Loss}_{P_i}(y_{m+1} \dots y_n | y_1 \dots y_m)} \tag{4}$$

Proof. Remember that for any prediction strategy P the loss incurred on a fragment $y_{m+1} \dots y_n$ of a sequence $y_1 \dots y_n$ is defined

$$\mathrm{Loss}_P(y_{m+1} \dots y_n | y_1 \dots y_m) = \sum_{i=m+1}^{n} \lambda(y_i, P(y_1 \dots y_{i-1})).$$

Let $r_0(i) = r(i), i = 1, \dots, k$. Trials $y_1, \dots, y_m$ are used as parameters. After each trial $y_{m+j}, j = 1, \dots, n-m$ the weights are updated as follows

$$r_j(i) = \beta^{\lambda(y_{m+j}, P_i(y_1 \dots y_{m+j-1}))} r_{j-1}(i).$$

As follows from the definition

$$r_j(i) = \beta^{\mathrm{Loss}_{P_i}(y_{m+1} \dots y_{m+j} | y_1 \dots y_m)} r_0(i).$$

After each trial y_{m+j-1} a *superprediction* g_j is defined

$$g_j(\omega) = \log_\beta \sum_{i=1}^{k} \beta^{\lambda(\omega, P_i(y_1 \dots y_{m+j-1}))} r^*_{j-1}(i),$$

where

$$r^*_{j-1}(i) = \frac{r_{j-1}(i)}{\sum_{s=1}^k r_{j-1}(s)}$$

are normalized weights.

Since the loss function is η-mixable for any $j = 1, \ldots, n-m$ a real number p_j exists (can be effectively computed with arbitrary degree of accuracy) such that

$$\lambda(\omega, p_j) \leq g_j(\omega)$$

for all ω $(= 0, 1$ in our case). Define $P(y_1 \ldots y_{m+j-1}) = p_j$.

By mathematical induction on $t = 1, \ldots n-m$ we shall prove that

$$\sum_{j=1}^t g_j(y_{m+j}) = \log_\beta \sum_{i=1}^k \beta^{\mathrm{Loss}_{P_i}(y_{m+1} \ldots y_{m+t} | y_1 \ldots y_m)} r(i). \tag{5}$$

The inequality (4) will be follow from (5) when $t = n-m$.

When $t = 1$ we have

$$g_1(y_{m+1}) = \log_\beta \sum_{i=1}^k \beta^{\lambda(y_{m+1}, P_i(y_1 \ldots y_m))} r(i).$$

When $t > 1$ we have

$$\log_\beta \sum_{i=1}^k \beta^{\mathrm{Loss}_{P_i}(y_{m+1} \ldots y_{m+t} | y_1 \ldots y_m)} r(i) -$$

$$\log_\beta \sum_{i=1}^k \beta^{\mathrm{Loss}_{P_i}(y_{m+1} \ldots y_{m+t-1} | y_1 \ldots y_m)} r(i) =$$

$$\log_\beta \frac{\sum_{i=1}^k \beta^{\mathrm{Loss}_{P_i}(y_{m+1} \cdots y_{m+t-1} | y_1 \cdots y_m) + \lambda(y_{m+t}, P_i(y_1 \cdots y_{m+t-1}))} r(i)}{\sum_{i=1}^k \beta^{\mathrm{Loss}_{P_i}(y_{m+1} \cdots y_{m+t-1} | y_1 \cdots y_m)} r(i)} =$$

$$\log_\beta \sum_{i=1}^k \beta^{\lambda(y_{m+t}, P_i(y_1 \ldots y_{m+t-1}))} r^*_{t-1}(i) = g_t(y_{t+m}).$$

□

References

[1] Cover, T.M., Gács, P. and Gray, R.M. (1989) Kolmogorov's contributions to information theory and algorithmic complexity. *Ann. Probab.*, **17**, 840–865.

[2] Haussler, D., Kivinen, J. and Warmuth, M.K. (1994) Tight worst-case loss bounds for predicting with expert advice. Technical Report UCSC-CRL-94-36, University of California at Santa Cruz, revised December 1994. Short version in P. Vitányi, editor, *Computational Learning Theory*, Lecture Notes in Computer Science, volume 904, pages 69–83, Springer, Berlin, 1995.

[3] Li, M. and Vitányi, P. (1997) *An Introduction to Kolmogorov Complexity and Its Applications*. Springer, New York, 2nd edition.

[4] Shen, A. (1983) The concept of (α, β)-stochasticity in the Kolmogorov sense and its properties. *Sov. Math. Dokl.*, **28**, 295–299.

[5] Vovk, V. (1990) Aggregating strategies. In M. Fulk and J. Case, editors, *Proceedings of the 3rd Annual Workshop on Computational Learning Theory*, pages 371–383, San Mateo, CA, 1990. Morgan Kaufmann.

[6] Vovk, V. (1998) A game of prediction with expert advice. *J. Comput. Syst. Sci.*, **56**, 153–173.

[7] Vovk, V., Gammerman, A. (1999) Complexity estimation principle, *The Computer Journal* **42:4**, 318–322.

[8] Vovk, V, Watkins, C.J.H.C. (1998) Universal portfolio selection, *Proceedings of the 11th Annual Conference on Computational Learning Theory*, 12–23.

[9] V'yugin, V.V. (1987) On the defect of randomness of a finite object with respect to measures with given complexity bounds, *SIAM Theory Probab. Appl.* **32**, 508–512.

[10] V'yugin, V.V. (1999) Algorithmic complexity and stochastic properties of finite binary sequences, *The Computer Journal*, **42:4**, 294–317.

[11] Yamanishi, K. (1995) Randomized approximate aggregating strategies and their applications to prediction and discrimination, in *Proceedings, 8th Annual ACM Conference on Computational Learning Theory*, pp 83–90, Assoc. Comput. Mach, New York.