Doklady Mathematics, Vol. 69, No. 1, 2004, pp. 145–148. Translated from Doklady Akademii Nauk, Vol. 394, No. 2, 2004, pp. 175–178.
Original Russian Text Copyright © 2004 by Vorontsov.
English Translation Copyright © 2004 by MAIK "Nauka/Interperiodica" (Russia).

COMPUTER SCIENCE

Combinatorial Bounds for Learning Performance

K. V. Vorontsov

Presented by Academician Yu.I. Zhuravlev July 2, 2003

Received July 3, 2003

Cross-validation functionals and their upper bounds are considered that characterize the generalization performance of learning algorithms. The initial data are not assumed to be independent, identically distributed (i.i.d.) or even to be random. The effect of localization of an algorithm family is described, and the concept of a local growth function is introduced. New performance bounds for monotone classifiers are obtained, which are nontrivial for small data sets and do not depend on the family complexity.

The learning problem can be described as follows. We are given an object space X, an output space Y, and a set $\mathfrak{A}$ of mappings from X to Y, called algorithms. There exists a target function $y^*: X \to Y$ not necessarily in $\mathfrak{A}$ whose values $y_i = y^*(x_i)$ are known only on the objects of a finite training set $X^l = \{x_1, x_2, \ldots, x_l\}$. It is necessary to construct an algorithm $a^* \in \mathfrak{A}$ satisfying the local constraints $a^*(x_i) = y_i$ $(i = 1, 2, \ldots, l)$ and the universal constraints $a^* \in \mathfrak{A}_u$, where the set of algorithms $\mathfrak{A}_u \subseteq \mathfrak{A}$ is determined by the specific features of a particular problem [2]. The desired algorithm a^* must approximate the target function y^* not only on the objects of the training set but also on the entire set X. This requirement can be formalized by using various quality functionals, some of which will be considered below.

The frequency of errors made by an algorithm $a \in \mathfrak{A}$ on a set $X^p = \{x_1, x_2, \ldots, x_p\} \subset X$ is

$$\nu(a, X^p) = \frac{1}{p}\sum_{i=1}^{p} I(x_i, a(x_i)),$$

where $I(x, y)$ is an error indicator that takes a value of 1 if the output y is erroneous for object x and takes a value of 0 otherwise. The error indicator is usually defined as a function of the deviation of the output y from the correct output $y^*(x)$, for example, $I(x, y) = [|y - y^*(x)| \geq \delta]$ for a given $\delta > 0$. Here and below, square brackets are used to denote a mapping of a logical result to a number: [False] = 0 and [True] = 1.

Definition 1. A learning method is a mapping μ that takes an arbitrary finite training set X^l with given outputs $Y^l = \{y_1, y_2, \ldots, y_l\}$ to an algorithm $a = \mu(X^l, Y^l)$. The method μ is also said to generate an algorithm a from the training set X^l.

It is assumed that a learning method μ generates algorithms by choosing them from a family of algorithms $A \subseteq \mathfrak{A}_u$. Assuming that y^* is fixed, we will use the shortened notation $\mu(X^l)$.

An algorithm a is called correct on a data set X^l if $\nu(a, X^l) = 0$. A method μ is called correct on X^l if the algorithm $\mu(X^l)$ is correct on X^l. In the general case, the correctness of a method on a training set does not guarantee that the algorithm generated will perform well on other data sets.

Consider several functionals that characterize the generalization performance of a learning method out of the training set.

1. The hold-out functional $\nu(\mu(X^l), X^k)$ is the frequency of errors on a given testing set X^k. A shortcoming of this functional is that it fixes a generally arbitrary partition of $X^l \cup X^k$ into a training and a testing set. If the value of $\nu(\mu(X^l), X^k)$ is sufficiently small, there is no guarantee that $\nu(\mu(X_1^l), X_1^k)$ will again be small for another partition of the same set (X_1^l, X_1^k). Thus, various partitions of the set should be taken into account while the quality functional is constructed. In what follows, we assume that l and k are arbitrary fixed numbers and $L = l + k$.

2. The complete cross-validation functional is defined as

$$Q_c^{l,k}(\mu, X^L) = \frac{1}{N}\sum_{n=1}^{N} \nu(\mu(X_n^l), X_n^k),$$

where (X_n^l, X_n^k), $n = 1, 2, \ldots, N$, are all possible partitions of X^L into a training and a testing subset of length l and k, respectively, and $N = C_L^l$.

Computing Center, Russian Academy of Sciences,
ul. Vavilova 40, Moscow, 119991 Russia
e-mail: voron@ccas.ru

3. A complete cross-validation functional insensitive to a minor fraction of errors ε made on the testing set, $0 \le \varepsilon < 1$:

$$Q_\varepsilon^{l,k}(\mu, X^L) = \frac{1}{N}\sum_{n=1}^{N}[\nu(\mu(X_n^l), X_n^k) > \varepsilon].$$

Theorem 1. *The functionals* $Q_c^{l,k}$ *and* $Q_\varepsilon^{l,k}$ *are related by the two-sided bounds*

$$\varepsilon Q_\varepsilon^{l,k} \le Q_c^{l,k} \le \varepsilon + (1-\varepsilon)Q_\varepsilon^{l,k}.$$

4. A complete cross-validation functional insensitive to minor deviations of the frequency of errors on the testing set from the frequency of errors on the learning set:

$$Q_{\nu,\varepsilon}^{l,k}(\mu, X^L)$$
$$= \frac{1}{N}\sum_{n=1}^{N}[\nu(\mu(X_n^l), X_n^k) - \nu(\mu(X_n^l), X_n^k) > \varepsilon].$$

If μ is a correct method on all subsets of length l, then $Q_\varepsilon^{l,k}$ coincides with $Q_{\nu,\varepsilon}^{l,k}$. In the general case, they are related by the inequality $Q_{\nu,\varepsilon}^{l,k} \le Q_\varepsilon^{l,k}$.

5. Let X be a probability space, X^l and X^k be i.i.d. random sets, and A be a given algorithm family. Vapnik and Chervonenkis have proposed a probability functional of uniform convergence of the error frequency on two sets, for which in the case $l = k$ they have obtained the upper bound [1]

$$P_{\nu,\varepsilon}^{l,k}(A) = \mathrm{P}\{\sup_{a \in A}(\nu(a, X^k) - \nu(a, X^l)) > \varepsilon\}$$
$$\le 1.5\Delta^A(L)e^{-\varepsilon^2 l},$$

where $\Delta^A(L)$ is the growth function of an algorithm family A. It is defined as the number of different binary vectors $(\beta_1, \beta_2, \ldots, \beta_L)$, $\beta_i = I(x_i, a(x_i))$ generated by all possible algorithms $a \in A$ on all sets X^L. If A has a finite VC-dimension h, then $\Delta^A(L) \le 1.5\frac{L^h}{h!}$.

The complete cross-validation functionals $Q_c^{l,k}$, $Q_\varepsilon^{l,k}$, and $Q_{\nu,\varepsilon}^{l,k}$ will be called combinatorial. In contrast to the probability functional $P_{\nu,\varepsilon}^{l,k}$, they depend on the learning method and a particular set, which does not need to be random. Under suitable probabilistic assumptions, we can proceed from combinatorial to probability functionals by taking the expectation:

$$\mathrm{E}Q_c^{l,k}(\mu, X^L) = \mathrm{P}\{I(\mu(X^l), x) = 1\},$$
$$\mathrm{E}Q_\varepsilon^{l,k}(\mu, X^L) = \mathrm{P}\{\nu(\mu(X^l), X^k) > \varepsilon\}, \tag{1}$$
$$\mathrm{E}Q_{\nu,\varepsilon}^{l,k}(\mu, X^L)$$
$$= \mathrm{P}\{\nu(\mu(X^l), X^k) - \nu(\mu(X^l), X^l) > \varepsilon\} \le P_{\nu,\varepsilon}^{l,k}(A).$$

It follows that any upper bounds of combinatorial functionals can easily be extended to the corresponding probability functionals. Moreover, inequality (1) implies that the Vapnik–Chervonenkis bound also holds for $\mathrm{E}Q_{\nu,\varepsilon}^{l,k}$.

It turns out that this bound, and even a stronger one, is valid for $Q_{\nu,\varepsilon}^{l,k}(\mu, X^L)$ with arbitrary μ and X^L. A strengthening of the bound is associated with the effect of localization of the growth function, which lies in the fact that for a fixed set, only a finite part of A can be obtained by learning, while the remaining algorithms are not used.

Definition 2. The local algorithm family generated by method μ on a set X^L is the set of algorithms

$$A_L^l(\mu, X^L) = \{\mu(X_n^l) \mid n = 1, 2, \ldots, N\}, \quad N = C_L^l.$$

Definition 3. The local growth function $\Delta_L^l(\mu, X^L)$ of method μ on a set X^L is the number of distinct binary vectors $(\beta_1, \beta_2, \ldots, \beta_L)$, $\beta_i = I(x_i, a(x_i))$ generated by all algorithms $a \in A_L^l(\mu, X^L)$.

The local growth function does not exceed $\Delta^A(L)$ and is bounded above by C_L^l, while $\Delta^A(L) \le 2^L$.

Definition 4. The incorrectness degree of method μ on a set X^L is the maximum frequency of errors made on all training subsets of length l:

$$\sigma_L^l(\mu, X^L) = \max_{n = 1, 2, \ldots, N}\nu(\mu(X_n^l), X_n^l).$$

If μ is correct on all subsets of length l, then $\sigma_L^l = 0$. In what follows, we use the shortened notation Δ_L^l, A_L^l, and σ_L^l, with the arguments (μ, X^L) dropped.

Theorem 2. *For any* μ *and* X^L,

$$Q_{\nu,\varepsilon}^{l,k}(\mu, X^L) < \Delta_L^l(\mu, X^L) \cdot \Gamma_L^l(\varepsilon, \sigma_L^l), \tag{2}$$

where $\Gamma_L^l(\varepsilon, \sigma) = \max\limits_m\sum\limits_s\frac{C_m^s C_{L-m}^{l-s}}{C_L^l}$, *with* m *ranging from* $\lceil\varepsilon k\rceil$ *to* $k + \sigma l$, *and with* s *ranging from* $\max\{0, m-k\}$ *to* $\min\left\{\left\lfloor\frac{l}{L}(m-\varepsilon k)\right\rfloor, \sigma l\right\}$.

$\Gamma_L^l(\varepsilon, \sigma)$ will be referred to as a combinatorial factor.

Corollary 1. *The bound given by* (2) *is not decreasing with respect to* σ, *since* $\Gamma_L^l(\varepsilon, \sigma)$ *is not decreasing with respect to* σ. *The least value is reached at* $\sigma = 0$, *when the method is correct*:

$$\Gamma_L^l(\varepsilon, 0) = \frac{C_{L-\lceil \varepsilon k \rceil}^l}{C_L^l} \le \left(\frac{k}{L}\right)^{\varepsilon k}.$$

Corollary 2. *For* $l = k$ *and any* (μ, X^L), *the quality functional* $Q_{\nu,\varepsilon}^{l,k}$ *satisfies the Vapnik–Chervonenkis bound up to the replacement of the growth function of the entire family by a local growth function*:

$$Q_{\nu,\varepsilon}^{l,k}(\mu, X^L) < 1.5\Delta_L^l(\mu, X^L)e^{-\varepsilon^2 l} \le 1.5\Delta^A(L)e^{-\varepsilon^2 l}. \quad (3)$$

Note that there is no reason to take the same value for l and k, except for the convenience of estimating Γ_L^l. That is why we consider the general case of arbitrary l and k.

This result means that learning performance can be described not only in terms of probability theory, but also in terms of set-depending combinatorial functionals based the idea of complete cross-validation. Bound (3) is valid for an arbitrary set, which is not necessarily random and independent.

In probability theory, independence means the invariance of the probability measure under all permutations of the elements in a set. In combinatorial setting, instead of the independence of a set, it is sufficient to assume the invariance of the quality functional under all permutations of a set (the symmetry of the functional). Note that all of the combinatorial functionals introduced above are symmetric. This constraint is much weaker, because it is imposed on the quality functional used rather than on the initial data. Thus, the nature of bound (3) is purely combinatorial and is implied by the discrete nature of an error indicator $I(x, y)$ and by the symmetry of the quality functional.

For combinatorial functionals, one can derive tighter bounds depending on the properties of a specific set. In particular, the effect of localization of the growth function can be taken into account in such bounds. By virtue of (1), the probability functional of uniform convergence of frequencies can be considered an upper bound for the complete cross-validation functional. The accuracy in this bound is lost because of the redundant requirement of uniform convergence.

The ratio of the right- to left-hand sides of (3) can be represented as

$$\frac{\Delta(A) \cdot 1.5e^{-\varepsilon^2 l}}{Q_{\nu,\varepsilon}^{l,k}} = \frac{\Delta(A)}{\Delta_L^l} \cdot \frac{1.5e^{-\varepsilon^2 l}}{\Gamma_L^l} \cdot \frac{\Delta_L^l \Gamma_L^l}{Q_{\nu,\varepsilon}^{l,k}}.$$

In each of the fractions, the numerator is an upper bound on the denominator. Three factors on the right-hand side of the equality describe the respective three basic causes of overestimated probability bounds for learning performance. The first cause is that the effect of localization is neglected. The complexity of the finite algorithm subfamily A_L^l resulting from learning can be considerably lower than the complexity of the entire family A. The second cause is associated with the relative error in the exponential bound for the combinatorial factor, which noticeably increases with l, in contrast to the absolute error. The third cause is the error in the decomposition of the complete cross-validation functional into the product of the local growth function Δ_L^l and the combinatorial factor Γ_L^l.

A promising approach to improving the accuracy of bounds is to give up the complexity characteristics of an algorithm family. Bounds of this kind are known for stable algorithms [5] and convex hulls of classifiers [4]. We consider one more case, when the target function is a priori known to be monotone or nearly monotone. Practical significance of monotone classifiers is discussed in [6]. Methods for designing monotone algorithms on finite sets are considered in [3] for classification and regression problems.

Consider a classification problem in which X is a partially ordered set, $Y = \{0, 1\}$, the error indicator is given by $I(x, y) = |y^*(x) - y|$, and a learning method μ chooses algorithms from the set A of all monotone mappings of X to Y.

Definition 5. The nonmonotonicity degree of a set X^L is the lowest frequency of errors made by monotone algorithms on X^L:

$$\delta(X^L) = \min_{a \in A} \nu(a, X^L).$$

A set X^L is called monotone if $x_i \le x_j$ implies $y_i \le y_j$ for all $i, j = 1, 2, \ldots, L$. A set is monotone if and only if $\delta(X^L) = 0$. If a method μ generates algorithms with the minimum frequency of errors on the training set in the class of all monotone functions A, then that method is correct on any monotone set [3].

Definition 6. The upper and lower wedges of an object $x_i \in X^L$ are the respective sets

$$W_0(x_i) = \{x_k \in X^L |\ x_i < x_k \text{ and } y_k = 0\},$$
$$W_1(x_i) = \{x_k \in X^L |\ x_k < x_i \text{ and } y_k = 1\}.$$

The wedge cardinality $w_i = |W_{y_i}(x_i)|$ characterizes the depth to which x_i is embedded in the class it belongs to. The smaller the cardinality w_i, the nearer the object to the boundary of the class. For boundary objects,

$w_i = 0$. If a monotone algorithm makes an error on x_i, it also makes an error on all objects of $W_{y_i}(x_i)$. The proof of the following theorem relies heavily on this fact.

Theorem 3. *If* μ *generates an algorithm with the minimum frequency of errors made on the training set in the class of all monotone functions and if the nonmonotonicity degree* of X^L *is equal to* δ, *then*

$$Q_c^{l,k}(\mu, X^L) \le \frac{1}{L} \sum_{\substack{i=1\\ w_i < \delta L + k}}^{L} \sum_{s=0}^{\min\{\delta L, l, w_i\}} \frac{C_{w_i}^{s} C_{L-1-w_i}^{l-s}}{C_{L-1}^{l}}. \qquad (4)$$

Corollary 3. *The bound does not monotonically decrease in* δ *and is minimal at* $\delta = 0$ *when the set is monotone and* μ *is correct*:

$$Q_c^{l,k}(\mu, X^L) \le \frac{1}{L} \sum_{\substack{i=1\\ w_i < k}}^{L} \frac{C_{L-1-w_i}^{l}}{C_{L-1}^{l}} \le \frac{1}{L} \sum_{\substack{i=1\\ w_i < k}}^{L} \left(\frac{k}{L}\right)^{w_i}.$$

The bound obtained, in contrast to complexity bounds, never exceeds unity. The largest value of 1 is reached when $w_i = 0$ for all $i = 1, 2, \ldots, L$. This is the case where both classes consist of pairwise incomparable objects and the set splits into two antichains. The smallest value is reached when the set is monotone and linearly ordered. The number of wedges of cardinality w then does not exceed 2 for all $w = 1, 2, \ldots, k$, whence $Q_c^{l,k} \le \frac{2}{l}$.

The VC-dimension of the class of monotone classifiers is infinite, since there are exactly 2^L dichotomies for a set of length L consisting of pairwise incomparable elements. Thus, the Vapnik–Chervonenkis classical theory fails to give performance bounds in this case. It is well known [6] that the effective VC-dimension of the class of monotone functions does not exceed the length of the maximal antichain in X^L. Bound (4) is much sharper, especially for small size sets.

ACKNOWLEDGMENTS

The author is grateful to Corresponding Member of the RAS K.V. Rudakov for his encouragement and valuable remarks.

This work was supported in part by the Russian Foundation for Basic Research (project no. 02-01-00325) and the Foundation for Supporting National Science.

REFERENCES

1. V. N. Vapnik, *Estimation of Dependences Based on Empirical Data* (Nauka, Moscow, 1979; Springer-Verlag, New York, 1982).
2. Yu. I. Zhuravlev and K. V. Rudakov, in *Problems of Applied Mathematics and Computer Science,* Ed. by O. M. Belotserkovskiĭ (Moscow, 1987), pp. 187–198.
3. K. V. Rudakov and K. V. Vorontsov, Dokl. Akad. Nauk **367**, 314 (1999) [Dokl. Math. **60**, 139 (1999)].
4. P. Bartlett, IEEE Trans. Inform. Theory **44**, 525 (1998).
5. O. Bousquet and A. Elisseeff, J. Machine Learning Res., No. 2, 499 (2002).
6. J. Sill, Discrete Appl. Math. (Special Issue on VC Dimension) **86**, 95 (1998).