Computational Mathematics and Mathematical Physics, Vol. 44, No. 11, 2004, pp. 1997–2009.
Translated from Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki, Vol. 44, No. 11, 2004, pp. 2099–2112.

Combinatorial Substantiation of Learning Algorithms

K. V. Vorontsov

Dorodnitsyn Computing Center, Russian Academy of Sciences, ul. Vavilova 40, Moscow, 119991 Russia

e-mail: voron@ccas.ru

Received January 30, 2004

Abstract—Combinatorial cross-validation functionals that characterize the generalization performance of learning algorithms are considered. Upper bounds are derived that are tighter than those in the Vapnik–Chervonenkis statistical theory. The initial data set is not assumed to be independent, identically distributed, or even random. The effect of localization of an algorithm family is described, and the concept of a local growth function is introduced. The basic principles of statistical theory are revised by using the combinatorial approach. The basic causes of complexity bound overestimation are analyzed.

Keywords: computational learning theory, learning method, VC-dimension, local growth function, local effective VC-dimension.

In learning theory, the generalization performance of a learning algorithm is characterized by the probability of an error. Unfortunately, this hypothetical quantity cannot be calculated or sometimes even satisfactorily evaluated, for example, in the case of small data sets. At the same time, in practice, any learning system deals only with finite data sets, both training and testing. Therefore, it is reasonable to characterize the generalization performance of algorithms with respect to finite data sets. Learning performance is empirically quantified by using independent testing sets, bootstrap, or cross-validation [1]. It is shown in this paper that upper bounds for cross-validation performance functionals can be derived without resorting to probability theory. Moreover, those bounds are tighter than traditional probability bounds.

The basic concepts related to the learning problem are introduced in Section 1. The basic principles of the Vapnik–Chervonenkis statistical theory [2] required for further consideration are briefly described in Section 2. Combinatorial functionals that characterize the performance of a given learning method on a given finite data set are defined in Section 3.

In Section 4, upper bounds for combinatorial functionals are derived, which are similar to bounds on the probability of a uniform convergence of the error frequencies on two subsets. The similarity of these bounds suggests that cross-validation functionals are, at least, no worse than probability functionals when used to characterize the generalization performance. However, there are conceptual differences between them. First, combinatorial bounds hold for any (not necessarily random, independent, and identically distributed (i.i.d.)) data set. Second, combinatorial bounds depend not on the complexity of the entire algorithm family but rather on the complexity of its local subset consisting of algorithms really obtained in training. Thus, the uniform convergence principle becomes redundant, and the algorithm family turns out to be a secondary construction with respect to the learning method.

The relationship between combinatorial and probability performance functionals is discussed in Section 5. The upper bounds for combinatorial functionals can easily be extended to the corresponding probability functionals. Thus, we can say that the "correspondence principle" holds for the transition from the statistical theory to the more accurate combinatorial theory.

In Section 6, the combinatorial approach is invoked to revise the basic principles of the statistical theory, including the correctness property of learning algorithms, the functional of uniform relative convergence of frequencies, the method of structural risk minimization, and the concept of effective VC-dimension. The basic causes of overestimated complexity bounds for learning performance are analyzed in Section 7.

This paper presents the proofs of the theorems stated in [3].

1. THE LEARNING PROBLEM

We are given an object space X, an output space Y, and a set $\mathfrak{A}$ of mappings from X to Y, which are called algorithms, meaning that they are effectively computable functions. It is assumed that there exists a target

function $y^*: X \longrightarrow Y$ not necessarily in $\mathfrak{A}$, whose values $y_i = y^*(x_i)$ are known only on the objects of a finite training set $X^l = \{x_1, \ldots, x_l\}$.

The learning problem is to construct an algorithm $a^* \in \mathfrak{A}$ satisfying the following three requirements.

First, it must return given outputs on training objects: $a^*(x_i) = y_i$, $i = 1, 2, \ldots, l$. Here, equality is regarded as exact or approximate, depending on the particular problem under consideration. In [4], these requirements are called *local constraints* to emphasize that they concern a finite number of training objects and admit effective verification in a finite number of steps.

Second, additional general constraints can be imposed on a^* as on a mapping from X to Y. For example, these can be symmetry, continuity, smoothness, monotonicity, etc., constraints or their combinations. Requirements of this kind are called in [4] *universal constraints* to emphasize that they are independent of a particular training set and are related to the mapping as a whole. As a rule, they do not admit effective verification and are taken into account on the stage of the algorithm design. Generally, the universal constraints are expressed by the condition $a^* \in \mathfrak{A}_u$, where $\mathfrak{A}_u$ is a given set of algorithms specific of the problem.

Third, the desired algorithm a^* must display the ability to generalize, i.e., to 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 of objects $X^p = \{x_1, \ldots, x_p\}$ is

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

where $I(x, y)$ is an *indicator function* that takes 1 if the output y is erroneous for object x and takes 0 otherwise. The choice of an indicator is problem-specific, and it depends primarily on the nature of Y. In classification problems when $Y = \{0, 1\}$, it is usually defined as

$$I(x, y) = |y - y^*(x)|$$

for a given function $\delta(x)$. Here and below, square brackets are used to denote the natural mapping of a logical quantity to a number: [False] = 0 and [True] = 1.

The use of a binary error indicator allows one to use a uniform approach to a wide class of problems, including both classification and regression.

2. STATISTICAL THEORY OF LEARNING

The Vapnik–Chervonenkis statistical theory [2, 5] assumes that X is a probability space and all the sets considered are i.i.d. The learning process constructs an algorithm a^* from a given family of algorithms $A \subset \mathfrak{A}$ by *minimizing the empirical risk*:

$$a^* = \operatorname*{arg\,min}_{a \in A} \nu(a, X^l).$$

The family can contain many algorithms that minimize the empirical risk. However, details of the learning method are disregarded, and it is assumed that any of them can be selected.

The performance of a^* is characterized by either the probability of an error or the frequency $\nu(a^*, X^k)$ of errors on some unknown testing set X^k. This quantity cannot be calculated explicitly, but it is proved to be quite close to the empirical risk $\nu(a^*, X^l)$ if the uniform deviation of the error frequencies on two sets is small:

$$P_\varepsilon^{lk}(A) = P\{\sup_{a \in A}(\nu(a, X^k) - \nu(a, X^l)) > \varepsilon\}. \tag{2.1}$$

In general, the algorithm returned by the learning method is unknown in advance. For this reason, one estimates a maximum deviation on the worst algorithm. If $P_\varepsilon^{lk}(A) \longrightarrow 0$ as $l \longrightarrow \infty$, then the frequencies of errors on two sets are said to converge uniformly. This is a sufficient condition for the *learnability* of an algorithm family.

For $l = k$ and any probability distribution on the object space, we have the estimate (see [5])

$$P_\varepsilon^{lk}(A) \le \Delta^A(2l) \cdot 1.5e^{-\varepsilon^2 l}, \tag{2.2}$$

where $\Delta^A(2l)$ is a complexity measure called the growth function of the algorithm family A.

Definition 1. *The growth function* $\Delta^A(L)$ of an algorithm family A is the maximum number of distinct binary vectors $[I(x_i, a(x_i))]_{i=1}^L$ generated by all possible algorithms $a \in A$ on an arbitrary set $\{x_1, \ldots, x_L\}$.

Obviously, $\Delta^A(l)$ does not exceed 2^L. The smallest h for which $\Delta^A(h) < 2^h$ is called the VC-dimension of $\Delta^A(l)$. If such an h does not exist, then the family is said to have an infinite VC-dimension. If A has a finite VC-dimension h, then the growth function is a polynomial in L:

$$\Delta^A(L) \le C_L^0 + \ldots + C_L^h \le 1.5\frac{L^h}{h!}. \tag{2.3}$$

In this case, the uniform convergence of frequencies takes place. Thus, the learning performance in statistical theory can be estimated if one knows the sample size and the VC-dimension of the algorithm family.

The calculation or estimation of the VC-dimension is a complicated problem for most particular families. It is well known that the VC-dimension of linear decision rules is equal to the number of free parameters or to the dimension of the linear space in which the separating hyperplane is constructed. Bounds on the VC-dimension were obtained for neural networks [6], decision trees and decision forest [7], correct algebraic closures of the estimate-calculating algorithm submodel [8], committees of linear inequalities [9], and many others.

Statistical bounds substantiate the method of *structural risk minimization*, which tries to select a submodel of algorithms having an optimal complexity. In this method, a structure of nested subfamilies $A_1 \subset \ldots \subset A_h = A$ of increasing VC-dimension is fixed, and the learning problem is solved for each subfamily. The algorithm providing the least upper bound for $\nu(a, X^k)$ is selected from the resulting algorithms.

Unfortunately, bound (2.2) is highly overestimated. The values it gives for the sufficient size l of training sets are considerably greater than numbers of objects encountered in practice. Table 1 shows l as a function of the VC-dimension h, accuracy ε, and the quality P_ε^{lk}. The right part of the table corresponds to $P_\varepsilon^{lk} = 1$ and demonstrates the range of applicability of bound (2.2). For smaller l, the bound exceed unity, i.e., becomes trivial. The overestimated bound in structural risk minimization may lead to excessive simplification of algorithms [10].

The overestimation of statistical bounds follows from their excessive generality. They correspond to the worst case and do not take into account three important characteristics of the problem and of the learning process. First, these are the characteristics of the objects distribution: they can belong to a subspace of lower dimension. Moreover, this exclusive case is typical of many machine learning tasks, because its dependent or nearly dependent features. Second, these are the characteristics of the target function itself: it can be smooth, symmetric, monotonic, or can have other specific properties, which sharply narrows the space of admissible solutions. Third, these are the characteristics of the learning method: it may have the ability to fit a given task bounding effectively the working subfamily of algorithms really obtained by learning.

3. COMBINATORIAL FUNCTIONALS OF LEARNING PERFORMANCE

The principle of empirical risk minimization within a fixed family can be criticized as an insufficiently accurate formalization of the learning process. First, it is unclear where the boundary of the family lies. It may happen that a very large family is formally written, while in practice the learning process generates algorithms only from its small part. Second, many algorithms are capable of minimizing empirical risk, but we always select a single solution. The explicit specification of the method that gives this solution would take into account the specific features of the learning process. Third, some learning methods exhibiting good performance in practice do not minimize the empirical risk. They include cross-validation methods and techniques based on external criteria, in particular, group method of data handling (GMDH) [11], explicit optimization of margins [12], boosting [13], bagging [14], etc.

Definition 2. A *learning method* μ is a mapping μ that transforms an arbitrary finite training set X^l to an algorithm $a = \mu(X^l)$. The method μ is also said to construct an algorithm a from the training set X^l.

A learning method μ is assumed to construct algorithms by selecting them from a family $A \subseteq \mathfrak{A}_u$. It is also assumed that μ is symmetric; i.e., the result $\mu(X^l)$ does not change under an arbitrary permutation of the elements in the training set.

Definition 3. 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.

Table 1

h	$P_\varepsilon^{lk} = 0.01$				$P_\varepsilon^{lk} = 1$			
	$\varepsilon = 0.01$	$\varepsilon = 0.05$	$\varepsilon = 0.1$	$\varepsilon = 0.2$	$\varepsilon = 0.01$	$\varepsilon = 0.05$	$\varepsilon = 0.1$	$\varepsilon = 0.2$
0	60106	2404	601	150	14054	562	140	35
2	295074	9012	1946	408	245330	6963	1423	273
5	673222	19884	4192	848	623320	17823	3664	711
10	1307418	38160	7974	1589	1257471	36095	7444	1452
20	2579359	74855	15572	3082	2529396	72789	15043	2944
50	6401335	185193	38433	7575	6351365	183127	37903	7437
100	12775769	369275	76581	15075	12725798	367208	76051	14937

In general, the small frequency $\nu(\mu(X^l), X^l)$ of errors on a given training set X^l does not guarantee that the algorithm will perform well on other data sets.

The frequency $\nu(\mu(X^l), X^k)$ of errors on a given test set X^k that, in general, does not intersect with X^l is also an incomplete characteristic of learning performance. A shortcoming of this functional is that it fixes an arbitrary partition $X^l \cup X^k$ of the data set into a training and a testing subset. 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 $X_1^l \cup X_1^k$ of the same set. This leads to the natural requirement that the functional characterizing learning performance on a finite set must be invariant under arbitrary permutations of the set.

Let l and k be arbitrary fixed numbers, $L = l + k$, and $X^L = \{x_1, \ldots, x_L\}$ be a given set. Denote by (X_n^l, X_n^k), $n = 1, 2, \ldots, N$ all possible partitions of X^L into a training and a testing subset of size l and k, respectively. The number of partitions is equal to C_L^l.

The following functionals characterize the generalization performance of a learning method μ on a finite set X^L and have the required invariance.

1. The complete cross-validation functional [1] is defined as

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

2. The functional of mean deviation of the frequency of errors on the testing set from the frequency of errors on the learning set is

$$Q_d^{lk}(\mu, X^L) = \frac{1}{N}\sum_{n=1}^{N} (\nu(\mu(X_n^l), X_n^k) - \nu(\mu(X_n^l), X_n^l))_+,$$

where $(z)_+ = z\ [z > 0]$ for any real z.

3. The cross-validation functional insensitive to a minor fraction of errors ε made on the testing set is defined as

$$Q_\varepsilon^{lk}(\mu, X^L) = \frac{1}{N}\sum_{n=1}^{N} [\nu(\mu(X_n^l), X_n^k) > \varepsilon], \quad 0 \le \varepsilon \le 1.$$

4. The 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 is defined as

$$Q_{\nu,\varepsilon}^{lk}(\mu, X^L) = \frac{1}{N}\sum_{n=1}^{N} [\nu(\mu(X_n^l), X_n^k) - \nu(\mu(X_n^l), X_n^l) > \varepsilon], \quad 0 \le \varepsilon \le 1.$$

The mean frequency of errors on the training set is defined as

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

In what follows, we drop the arguments (μ, X^L) of functionals and the superscripts *lk* indicating that the functionals depend on both training and testing sets sizes.

The definitions above imply that $Q_c \le Q_d + \bar{\nu}_L^l$ and $Q_{\nu,\varepsilon} \le Q_\varepsilon$. If μ is a correct learning method on all subsets of size l, then $\bar{\nu}_L^l = 0$, $Q_c = Q_d$, and $Q_{\nu,\varepsilon} = Q_\varepsilon$. The following two-sided bounds are less obvious.

Lemma 1. *For arbitrary* μ, X^L, *and* $\varepsilon \in [0, 1]$,

$$\varepsilon Q_\varepsilon < Q_c \le \varepsilon + (1-\varepsilon)Q_\varepsilon, \quad \varepsilon Q_{\nu,\varepsilon} < Q_d \le \varepsilon + (1-\varepsilon)Q_{\nu,\varepsilon},$$

$$\varepsilon Q_{\nu,\varepsilon} < Q_c \le \varepsilon + (1-\varepsilon)Q_{\nu,\varepsilon} + \bar{\nu}_L^l.$$

Proof. The first two bounds follow directly from the definitions and

$$\varepsilon[x > \varepsilon] < x \le \varepsilon + (1-\varepsilon)[x > \varepsilon],$$

which holds for any x and ε in $[0, 1]$.

The third bound follows from the first and the second one:

$$\varepsilon Q_{\nu,\varepsilon} \le \varepsilon Q_\varepsilon < Q_c \le Q_d + \bar{\nu}_L^l \le \varepsilon + (1-\varepsilon)Q_{\nu,\varepsilon} + \bar{\nu}_L^l.$$

The lemma is proved.

These bounds suggest that the functionals are interchangeable. The choice of a particular functional is not very important and can be inspired by a priori preferences or the convenience of bounds derivation.

4. LOCAL COMPLEXITY AND BOUNDS FOR LEARNING PERFORMANCE

In practice, the target function and the learning method are always fixed and the training set is finite. For this reason, only a finite part of the family can be obtained by learning, while the other algorithms remain unused or very rarely used. We shall call this effect the localization of the algorithm family. The most interesting are situations when the complexity of a local subfamily is considerably lower than that of the entire family A.

The localization effect removes the artificial ban against using complex algorithms. It is more important to develop a learning method capable of fitting a given task and of localizing a suitable effectively working domain of the family rather than of limiting a VC dimension of the entire family. A perfect learning method must generate algorithms close to the target function. It is of no matter how many algorithms that are not close to the target function are contained in the family. This property will be referred to as the *localizing ability* of a learning method.

Definition 4. The local 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 5. 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 $[I(x_i, a(x_i))]_{i=1}^L$ generated by all algorithms a in $A_L^l(\mu, X^L)$.

The local growth function differs substantially from the growth function of the entire family $\Delta^A(L)$. The former depends on a particular set, the learning method, and the ratio between l and k. The local growth function is bounded above by C_L^l, while $\Delta^A(L) \le 2^L$. It does not exceed $\Delta^A(L)$.

Definition 6. The *incorrectness degree* of method μ on a set X^L is the maximum frequency of errors made on all training subsets of size l:

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

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. *Let the incorrectness degree of method* μ *on a set* X^L *be* $\sigma = \sigma_L^l(\mu, X^L)$. *Then, for any* $\varepsilon \in [0, 1)$,

$$Q_{\nu,\varepsilon}^{lk}(\mu, X^L) < \Delta_L^l(\mu, X^L)\Gamma_L^l(\varepsilon, \sigma), \tag{4.1}$$

where $\Gamma_L^l(\varepsilon, \sigma)$ *is defined as*

$$\Gamma_L^l(\varepsilon, \sigma) = \max_{m \in M(\varepsilon,\sigma)} \sum_{s \in S(\varepsilon,\sigma)} \frac{C_m^s C_{L-m}^{l-s}}{C_L^l},$$

$$M(\varepsilon, \sigma) = \{m \mid \varepsilon k < m \le k + \sigma l\},$$

$$S(\varepsilon, \sigma) = \{s \mid \max(0, m-k) \le s \le \sigma l,\ s < (m - \varepsilon k) l/L\}.$$

Proof. On A_L^l, we introduce an equivalence relation: for arbitrary a and a' in A_L^l,

$$a \sim a' \Leftrightarrow (\forall x \in X^L) I(x, a) = I(x, a'),$$

i.e., algorithms are equivalent if they make errors on the same objects of X^L. This relation splits A_L^l into classes, denoted by A_{mi}, where $m = 0, 1, \ldots, L$ is the number of errors made on X^L by algorithms of a given class; $i = 1, 2, \ldots, \Delta_m$ is the index of a class among all the classes whose algorithms make n errors; and Δ_m is the number of ways of obtaining m errors on X^L by all possible algorithms of A_L^l. The number of all equivalence classes is equal to the local growth function of method μ on X^L:

$$\Delta_L^l = \Delta_0 + \ldots + \Delta_L. \tag{4.2}$$

Equivalence on algorithms generates equivalence on partitions if, for arbitrary n and u in $\{1, 2, \ldots, N\}$, we set $n \sim u \Leftrightarrow \mu(X_n^l) \sim \mu(X_u^l)$. This yields the equivalence classes $N_{mi} = \{n \mid \mu(X_n^l) \in A_{mi}\}$ on the partitions set, which are in one-to-one correspondence with the classes A_{mi}.

Summing the partitions over each equivalence class separately, we write the quality functional

$$Q_{\nu,\varepsilon}^{lk} = \frac{1}{N} \sum_{m=0}^{L} \sum_{i=1}^{\Delta_m} \sum_{n \in N_{mi}} [\nu(\mu(X_n^l), X_n^k) > \nu(\mu(X_n^l), X_n^l) + \varepsilon].$$

The functional value does not change if the algorithm $\mu(X_n^l)$, $n \in N_{mi}$ is replaced by an arbitrary element a_{mi} of A_{mi}. Since the sum vanishes for $m \le \varepsilon k$ and $m > k + \sigma l$, it suffices to sum only over $m \in M(\varepsilon, \sigma)$:

$$Q_{\nu,\varepsilon}^{lk} = \sum_{m \in M(\varepsilon,\sigma)} \sum_{i=1}^{\Delta_m} \underbrace{\frac{1}{N} \sum_{n \in N_{mi}} [\nu(a_{mi}, X_n^k) > \nu(a_{mi}, X_n^l) + \varepsilon]}_{\gamma_{mi}}. \tag{4.3}$$

The inner sum γ_{mi} is estimated from above by replacing the equivalence class N_{mi} with the set of all partitions. Denoting by s the number of errors on the training set ($0 \le s \le \sigma l$), we sum the partitions for each s separately:

$$\gamma_{mi} \le \frac{1}{N} \sum_{n=1}^{N} [\nu(a_{mi}, X_n^k) > \nu(a_{mi}, X_n^l) + \varepsilon] = \sum_{s=0}^{\sigma l} \left[\frac{m-s}{k} > \frac{s}{l} + \varepsilon\right] \frac{1}{N} \sum_{n=1}^{N} \left[\nu(a_{mi}, X_n^l) = \frac{s}{l}\right].$$

The inner sum is equal to $C_m^s C_{L-m}^{l-s}$, which is the number of partitions of the set of size L into two subsets such that exactly s errors out of m ones are contained in the subset of size l. Thus,

$$\gamma_{mi} \le \sum_{s \in S(\varepsilon,\sigma)} \frac{C_m^s C_{L-m}^{l-s}}{C_L^l}.$$

Since this quantity is independent of i, it can be taken out of the sum over i. By using (4.2), we arrive at

$$Q^{lk}_{\nu,\varepsilon} \le \sum_{m \in M(\varepsilon,\sigma)} \Delta_m \gamma_{mi} \le \Delta^l_L \max_{m \in M(\varepsilon,\sigma)} \gamma_{mi}.$$

Substituting the bound for γ_{mi} into this inequality gives the desired result. Theorem 2 is proved.

Corollary 1. For $l = k$ and arbitrary μ and X^L, the functional $Q^{lk}_{\nu,\varepsilon}$ satisfies the Vapnik–Chervonenkis bound up to the replacement of the growth function of the entire family by the local growth function:

$$Q^{lk}_{\nu,\varepsilon}(\mu, X^L) \le \Delta^l_L(\mu, X^L) 1.5 e^{-\varepsilon^2 l} \le \Delta^A(L) \cdot 1.5 e^{-\varepsilon^2 l}. \tag{4.4}$$

The proof is derived from the following facts. The local growth function $\Delta^l_L(\mu, X^L)$ does not exceed the growth function $\Delta^A(L)$ of the entire family, the combinatorial factor $\Gamma^l_L(\varepsilon, \sigma)$ is a nondecreasing function of σ, and $\Gamma^l_L(\varepsilon, 1) \le 1.5 e^{-\varepsilon^2 l}$ for $l = k$ (see [2]).

The strengthening of the bound is achieved primarily due to the modification of the quality functional caused by discarding the redundant requirement of uniform convergence. This result was first mentioned in [15].

Note that, in general, there is no reason to take the same value for l and k, except for the convenience of estimating the combinatorial factor.

By Lemma 1, similar bounds hold for other combinatorial functionals. A more accurate technique produces somewhat stronger upper bounds for Q^{lk}_d and Q^{lk}_c.

Theorem 3. *Let the incorrectness degree of method μ on a set X^L be $\sigma = \sigma^l_L(\mu, X^L)$. Then, for any $\varepsilon \in [0, 1)$,*

$$Q^{lk}_d(\mu, X^L) < \varepsilon + \Delta^l_L(\mu, X^L) \tilde{\Gamma}^l_L(\varepsilon, \sigma), \tag{4.5}$$

where

$$\tilde{\Gamma}^l_L(\varepsilon, \sigma) = \max_{m \in M(\varepsilon,\sigma)} \sum_{s \in S(\varepsilon,\sigma)} \left(\frac{ml - sL}{lk} - \varepsilon \right) \frac{C^s_m C^{l-s}_{L-m}}{C^l_L},$$

and $M(\varepsilon, \sigma)$ and $S(\varepsilon, \sigma)$ are the same as in Theorem 2.

Proof. The inequality $(x)_+ \le \varepsilon + (x - \varepsilon)_+$ holds for any $x \in \mathbb{R}$ and $\varepsilon \ge 0$ and implies that

$$Q^{lk}_d \le \varepsilon + \frac{1}{N} \sum_{n=1}^{N} (\nu(\mu(X^l_n), X^k_n) - \nu(\mu(X^l_n), X^l_n) - \varepsilon)_+.$$

By analogy with the proof of the preceding theorem, we introduce classes A_{mi} of equivalent algorithms and classes N_{mi} of equivalent partitions and then choose a single element a_{mi} from each A_{mi}, where m is the number of errors made by the algorithm a_{mi} on the entire set X^L, $m = 0, 1, \ldots, L$, $i = 1, 2, \ldots, \Delta_m$. Summing the partitions over the equivalence classes separately gives

$$Q^{lk}_d \le \varepsilon + \frac{1}{N} \sum_{m=0}^{L} \sum_{i=1}^{\Delta_m} \sum_{n \in N_{mi}} (\nu(a_{mi}, X^k_n) - \nu(a_{mi}, X^l_n) - \varepsilon)_+.$$

Since the sum vanishes for $m \le \varepsilon k$ and $m > k + \sigma l$, it suffices to sum only over $m \in M(\varepsilon, \sigma)$:

$$Q^{lk}_d \le \varepsilon + \sum_{m \in M(\varepsilon,\sigma)} \sum_{i=1}^{\Delta_m} \underbrace{\frac{1}{N} \sum_{n \in N_{mi}} (\nu(a_{mi}, X^k_n) - \nu(a_{mi}, X^l_n) - \varepsilon)_+}_{\tilde{\gamma}_{mi}}. \tag{4.6}$$

The inner sum $\tilde{\gamma}_{mi}$ is estimated from above by replacing the equivalence class N_{mi} with the set of all partitions. Denoting by s the number of errors on the training set ($0 \le s \le \sigma l$), we sum the partitions for each s separately:

$$\tilde{\gamma}_{mi} \le \frac{1}{N}\sum_{n=1}^{N}(\nu(a_{mi}, X_n^k) - \nu(a_{mi}, X_n^l) - \varepsilon)_+$$

$$= \sum_{s=0}^{\sigma l}\left[\frac{m-s}{k} > \frac{s}{l} + \varepsilon\right]\left(\frac{m-s}{k} - \frac{s}{l} - \varepsilon\right)\frac{1}{N}\sum_{n=1}^{N}\left[\nu(a_{mi}, X_n^l) = \frac{s}{l}\right].$$

The inner sum is equal to $C_m^s C_{L-m}^{l-s}$, which is the number of partitions of the set of size L into two subsets such that exactly s errors out of m ones are contained in the subset of size l. Thus,

$$\tilde{\gamma}_{mi} \le \sum_{s \in S(\varepsilon, \sigma)}\left(\frac{m-s}{k} - \frac{s}{l} - \varepsilon\right)\frac{C_m^s C_{L-m}^{l-s}}{C_L^l}.$$

Since this quantity is independent of i, it can be taken out of the sum over i in (4.6). By using (4.2), we arrive at

$$Q_d^{lk} \le \varepsilon + \sum_{m \in M(\varepsilon, \sigma)} \Delta_m \tilde{\gamma}_{mi} \le \varepsilon + \Delta_L^l \max_{m \in M(\varepsilon, \sigma)} \tilde{\gamma}_{mi}.$$

Substituting the bound for $\tilde{\gamma}_{mi}$ into this inequality gives the desired result. The theorem is proved.

Theorem 3 and $Q_c^{lk} \le Q_d^{lk} + \bar{\nu}_L^l$ imply an upper bound for the cross-validation functional.

Corollary 2. Let the incorrectness degree of method μ on a set X^L be $\sigma = \sigma_L^l(\mu, X^L)$. Then, for any $\varepsilon \in [0, 1)$,

$$Q_c^{lk}(\mu, X^L) < \bar{\nu}_L^l + \varepsilon + \Delta_L^l(\mu, X^L)\tilde{\Gamma}_L^l(\varepsilon, \sigma). \tag{4.7}$$

Bounds (4.5) and (4.7) involve the artificially introduced parameter ε. To eliminate it, we have to solve an additional minimization problem over ε.

5. PROBABILITY FUNCTIONALS AND THE CORRESPONDENCE PRINCIPLE

The results above allow us to propose a nonprobabilistic theory of learning performance.

Unlike P_ε^{lk}, the combinatorial functionals depend on the learning method and a particular data set, which is not necessarily i.i.d. Again assuming that X is a probability space and X^L is a random i.i.d. set, the expectation of combinatorial functionals takes the form of probabilistic quality functionals:

$$EQ_c^{lk}(\mu, X^L) = P\{I(\mu(X^l), x) = 1\},$$

$$EQ_\varepsilon^{lk}(\mu, X^L) = P\{\nu(\mu(X^l), X^k) > \varepsilon\},$$

$$EQ_{\nu,\varepsilon}^{lk}(\mu, X^L) = P\{\nu(\mu(X^l), X^k) - \nu(\mu(X^l), X^l) > \varepsilon\}.$$

The first line expresses the well-known fact that the complete cross-validation Q_c gives an unbiased estimator of the probability of an error [2]. The other functionals are also unbiased estimators of the corresponding probability functionals and have a clear interpretation.

Any upper bound for a combinatorial functional can easily be transformed into an upper bound for the corresponding probability functional by taking the expectation of the both sides of the inequality.

The probability functionals introduced above provide a more accurate characterization of learning performance than the Vapnik–Chervonenkis functional P_ε^{lk}, because they are free from the redundant requirement of uniform convergence. As follows from definitions, P_ε^{lk} is an overestimated upper bound for $EQ_{\nu,\varepsilon}^{lk}$,

which in fact describes the learning performance:

$$EQ^{lk}_{\nu,\varepsilon}(\mu, X^L) \le P^{lk}_{\varepsilon}(A). \tag{5.1}$$

Thus, the correspondence principle holds for the transition from the Vapnik–Chervonenkis statistical theory to the more accurate combinatorial theory of learning performance.

Bounds in the statistical theory are derived under assumption that objects are drawn independently from an unknown probability distribution. Now, it turns out that the same bound holds for an arbitrary data set, which is not necessarily i.i.d. Moreover, the use of probabilistic quality functionals may lead to excessive intermediate steps in the derivation of bounds and to their degraded accuracy (a typical example is the basic lemma in the statistical theory [2, p. 219].

The rejection of the independence assumption seems to be surprising at first glance. In probability theory, the independence of a set means the invariance of a probability measure under all possible permutations of the set. In the proof of Theorem 2, the same role is played by the invariance of the quality functional under all possible permutations of the set (the symmetry of the functional). This requirement can be viewed as a weakening of the independence hypothesis under which the constraint is moved from the initial data to the quality functional. Note that all the combinatorial functionals defined above are symmetric.

Thus, the nature of bounds (4.1) and (4.5) is purely combinatorial rather than probabilistic and follows from the discrete nature of the error indicator $I(x, y)$ and from the symmetry of the quality functional.

6. SOME PRINCIPLES OF STATISTICAL THEORY REVISITED

1. The combinatorial bounds derived above depend on the incorrectness degree σ. In the Vapnik–Chervonenkis theory, the extreme cases were only considered: $\sigma = 0$ (deterministic case) and $\sigma = 1$. It is of interest to analyze intermediate situations with $0 < \sigma < 1$.

The combinatorial factor $\Gamma^l_L(\varepsilon, \sigma)$ is a monotonically nondecreasing function of σ. It has its minimum at $\sigma = 0$ when the learning method is correct. The maximum value is reached at $\sigma = 1$, when there is no prior knowledge of the number of errors made in the learning.

In the case of a correct learning method, the combinatorial factor can be simplified:

$$\Gamma^l_L(\varepsilon, 0) = \frac{C^l_{L-\lceil \varepsilon k\rceil}}{C^l_L} \le \left(\frac{k}{L}\right)^{\varepsilon k}.$$

If h is the VC-dimension of a local subfamily, then (2.3) and Theorem 2 give the following bound:

$$Q^{lk}_{\varepsilon} = Q^{lk}_{\nu,\varepsilon} < (C^0_L + \ldots + C^h_L)\frac{C^l_{L-\lceil \varepsilon k\rceil}}{C^l_L}.$$

Table 2 presents the required size l of a training set computed from this equation. The results are much better than those presented in Table 1.

As σ increases, $\Gamma^l_L(\varepsilon, \sigma)$ grows very quickly and its value becomes comparable (in order) with $\Gamma^l_L(\varepsilon, 1)$ for $\sigma \approx \varepsilon$. Thus, acceptable numerical bounds for l (at least, not exceeding 10^3) can be obtained only for a correct learning method and families of an extremely low local VC-dimension.

The combinatorial approach provides a new view of construction of correct algorithms. Obviously, the structure of the algorithms has to be more complex to ensure their correctness. According to the statistical theory, this leads to a considerable increase in the growth function, against which a decrease in the combinatorial factor is unnoticeable. Thus, the statistical theory suggests that the correctness on a training set is not expedient. In the combinatorial approach, the more complex structure of an algorithm does not necessarily lead to a considerable increase in the local growth function. In this case, the correctness of the learning algorithm becomes extremely desired, because this sharply reduces the combinatorial factor. Note that the construction of correct algorithmic structures is a central idea in the algebraic approach to pattern recognition [16].

2. The statistical theory acknowledges that uniform convergence is a redundant requirement. To estimate the frequency $\nu(a, X^k)$ of errors on a testing set from the frequency $\nu(a, X^l)$ of errors on a training set, it suffices to require uniform convergence in the domain of minimal frequencies rather than on the entire family. The statistical theory says that this domain is difficult to describe explicitly and proposes a partial solution

Table 2

h	$Q_\varepsilon^{lk} = 0.01$				$Q_\varepsilon^{lk} = 1$			
	$\varepsilon = 0.01$	$\varepsilon = 0.05$	$\varepsilon = 0.1$	$\varepsilon = 0.2$	$\varepsilon = 0.01$	$\varepsilon = 0.05$	$\varepsilon = 0.1$	$\varepsilon = 0.2$
0	800	160	80	40	200	40	10	5
2	2900	460	200	85	2100	300	130	50
5	6300	980	420	170	5500	820	340	130
10	12000	1840	780	315	11200	1680	700	275
20	23500	3560	1510	600	22800	3420	1430	560
50	58200	8780	3710	1470	57400	8620	3630	1430
100	107000	17500	7380	2920	107000	17340	7300	2880

to the problem. Specifically, the functional of uniform relative deviation of frequencies on two subsets is introduced with the bound obtained in [2]

$$P\left\{\sup_{a\in A}\frac{\nu(a, X^k)-\nu(a, X^l)}{\sqrt{\nu(a, X^L)}}>\varepsilon\right\}<\Delta^A(L)\max_{m_0\le m\le m_1}\sum_{s=s_0(m)}^{s_1(m)}\frac{C_m^s C_{L-m}^{l-s}}{C_L^l},$$

where

$$m_0 = \lceil(\varepsilon k)^2/L\rceil,\quad s_0(m) = \max(0, m-k),$$

$$m_1 = L,\quad s_1(m) = \lfloor(m-\varepsilon k\sqrt{m/L})l/L\rfloor.$$

It is easy to show that this bound can be derived from combinatorial formula (1.4) by substituting $\varepsilon\sqrt{m/L}\longrightarrow\varepsilon$. It only estimates the combinatorial factor in a different manner but does not describe the effect of the family localization. No relative bounds are required when we use (4.1).

3. Combinatorial quality functionals have an undoubted advantage over their probability analogs: the former can be measured from a given data set. This makes it possible to reject the overestimated upper bounds in structural risk minimization. Moreover, nested subfamilies of growing VC-dimensions do not need to be constructed in this case. A more general approach is, given a fixed set of learning methods $\mu_1, \ldots, \mu_T$, to select the best one by using the cross-validation criterion on a given finite training set. Note that it is this approach that was recommended for practical applications in [2], although without revealing a clear connection with the basic theoretical results. Empirical studies [10] also show that this model selection technique is preferable to the structural risk minimization and to the minimum description length principle [17], which both represent various formalisms for the notion of complexity.

4. In [18], the effective VC-dimension was introduced and it was shown that statistical bounds remain valid if the VC-dimension is replaced by the effective VC-dimension. A method for measuring the effective VC-dimension from a given data set was also proposed in [18] for two-class classification problems. Empirical measurements of the VC-dimension are expedient for two reasons. First, one fails to obtain analytical bounds on the VC-dimension for many families of algorithms. Second, the effective VC-dimension in a particular problem can be considerably lower than the complete VC-dimension.

The idea of the method is to measure, for different sample sizes l, the value of the functional on the left-hand side of the inequality

$$Q_{\sup}^{lk}(A) = \frac{1}{N}\sum_{n=1}^{N}[\sup_{a\in A}(\nu(a, X_n^k)-\nu(a, X_n^l))>\varepsilon]<C\frac{L^h}{h!}e^{-\varepsilon^2 l},$$

where C is a constant and $k = l$.

Next, it is assumed that, for some values of C and h, the dependence of the left-hand side on the sample size has the same algebraic expression as the right-hand side. The corresponding value of h is called the *effective VC-dimension.* This assumption is well confirmed in the case of linear decision rules [18].

An elegant method that avoids the calculation of the supremum was proposed for measuring $Q^{lk}_{\sup}(A)$. In classification problems with two classes, the maximization of $\nu(a, X^k_n) - \nu(a, X^l_n)$ is equivalent to the minimization of $\nu(a, \tilde{X}^k_n) + \nu(a, X^l_n)$, where $\tilde{X}^k_n$ is obtained from X^k_n by replacing the original classifications with erroneous ones. If μ minimizes the empirical risk, then the difference of the frequencies is maximized by the algorithm $a_n = \mu(\tilde{X}^k_n \cup X^l_n)$. Then

$$Q^{lk}_{\sup}(A) = \frac{1}{N}\sum_{n=1}^{N} [\nu(a_n, X^k_n) - \nu(a_n, X^l_n) > \varepsilon]. \tag{6.1}$$

The measurement itself is to estimate this sum from a smaller number of partitions chosen at random. The measurement accuracy can easily be estimated by the law of large numbers.

The effective VC-dimension takes into account a particular distribution of objects but ignores specifics of the target function and of the learning method, because the algorithm is intentionally trained to make errors. In the case of linear decision rules, the effective VC-dimension is equal, with high accuracy, to the dimension of the subspace containing the set [18].

In the combinatorial approach, the uniform convergence functional $Q^{lk}_{\sup}(A)$ is replaced by the cross-validation functional $Q^{lk}_{\nu,\varepsilon}$. The measuring procedure (6.1) remains the same, with the only difference being that $a_n = \mu(X^l_n)$; i.e., it is not required to introduce artificial errors in the training set.

In this case, a new concept—the *local effective VC-dimension*—arises, which is the value of h for which the dependence of $Q^{lk}_{\nu,\varepsilon}$ on L is best approximated by

$$Q^{lk}_{\nu,\varepsilon}(\mu, X^L) \approx C\frac{L^h}{h!}\Gamma^l_L(\varepsilon, \sigma).$$

In contrast to the effective VC-dimension introduced in [18], the local effective VC-dimension takes into account everything: the features of the distribution of objects, the features of the target function, and the features of the learning method.

A comparison of the VC-dimension with the measured effective VC-dimension can reveal how well the method captures the effective VC-dimension of the object space [18].

A comparative measurement of the effective VC-dimension and the local effective VC-dimension can reveal how significant the localization effect is, i.e., how well the given learning method is adjusted to a particular target function on a particular training set.

7. CAUSES OF OVERESTIMATED COMPLEXITY BOUNDS

To analyze the causes of overestimated complexity bounds for learning performance, let us consider the ratio of the right- to left-hand sides of (4.4):

$$\frac{\Delta^A(L)\cdot 1.5e^{-\varepsilon^2 l}}{Q^{lk}_{\nu,\varepsilon}} = \frac{\Delta^A(L)}{\Delta^l_L}\,\frac{1.5e^{-\varepsilon^2 l}}{\Gamma^l_L}\,\frac{\Delta^l_L\Gamma^l_L}{Q^{lk}_{\nu,\varepsilon}}.$$

In each of the fractions, the numerator is an upper bound for the denominator. The three ratios on the right-hand side of the equality describe the respective three basic causes of overestimated complexity 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. To obtain bounds applicable in practice, one needs to calculate or tabulate rather complicated combinatorial expressions.

The third cause is the error in the decomposition of the cross-validation functional into the product of the local growth function Δ^l_L and the combinatorial factor Γ^l_L. This cause seems to be most important,

because it stems from the transition from performance analysis to complexity analysis and is associated with the nature of complexity bounds. This cause is intrinsic both to probabilistic and combinatorial bounds based on the growth function.

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 well known for stable algorithms [19] and convex hulls of classifiers [20]. In [19, 20], only the specific features of the learning method were taken into account, while the features of a particular training set and the target function were ignored. The resulting bounds are still highly overestimated, and the sufficient sample size is about 10^4 objects.

There is reason to believe that acceptable numerical bounds can be obtained only by using a priori knowledge of the properties of the training set and the target function. Note that the correspondence between the training set (local information) and a priori constraints (universal information) are studied in detail in the theory of universal and local constraints [4, 21] in terms of the theory of categories and the algebraic approach to pattern recognition [16]. The algebraic theory makes it possible to verify the consistency of these two types of information and to constructively describe irredundant classes of algorithm models that ensure the existence of correct algorithms. However, generalization bounds are out of the scope of this theory. In general, the influence of a priori knowledge on the quality of the target function is the most complicated and least studied problem. The combinatorial approach substantially facilitates the development of this direction. For example, a nonprobabilistic bound for Q_c has been derived in the case when the target function is monotonic or nearly monotonic and the learning method generates only monotone mappings [3, 22]. That bound is always less than 1, does not depend on the complexity of the family (which is known to have an infinite VC-dimension), and is much tighter on small sets than the bounds obtained in [23, 24].

ACKNOWLEDGMENTS

I am deeply grateful to Yu.I. Zhuravlev for his encouragement and to my teacher K.V. Rudakov for his interest in this study and valuable remarks. This work was supported by the program "Algebraic and Combinatorial Methods in Mathematical Cybernetics" of the RAS Department of Mathematical Sciences, by the Russian Foundation for Basic Research (project nos. 02-01-00325 and 01-07-90242), and by the Russian Science Support Foundation.

REFERENCES

1. R. Kohavi, "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection," *Proceedings of 14th Int. Joint Conf. on Artificial Intelligence, Montreal, PQ, 1995, IJCAI-95* (Morgan Kaufmann, San Francisco, 1995), pp. 1137–1145.
2. V. N. Vapnik, *Estimation of Dependences Based on Empirical Data* (Nauka, Moscow, 1979; Springer-Verlag, New York, 1982).
3. K. V. Vorontsov, "Combinatorial Bounds for Learning Performance," Dokl. Akad. Nauk **394**, 175–178 (2004) [Dokl. Math. **69**, 145–147 (2004)].
4. Yu. I. Zhuravlev and K. V. Rudakov, "On the Algebraic Correction of Procedures for Data Processing (Transformation)," in *Problems in Applied Mathematics and Computer Science*, Ed. by O. M. Belotserkovskii (Nauka, Moscow, 1987), pp. 187–198 [in Russian].
5. V. Vapnik, *Statistical Learning Theory* (Wiley, New York, 1998).
6. M. Karpinski and A. Macintyre, "Polynomial Bounds for VC-Dimension of Sigmoidal Neural Networks," *Proceedings of the 27th Annual ACM Symp. on the Theory of Computing, Las Vegas, USA, 1995* (ACM, New York, 1995), pp. 200–208.
7. Yu. Yu. Dyulicheva, "Bound for the VSD of an *r*-Reduced Empirical Forest," Tavrich. Vest. Inform. Mat., No. 1, 31–42 (2003).
8. V. L. Matrosov, "The VC-Dimension of Algebraic Extensions of a Model for Estimate-Calculating Algorithms," Zh. Vychisl. Mat. Mat. Fiz. **24**, 1719–1730 (1984).
9. V. Mazurov, M. Khachai, and A. Rybin, "Committee Constructions for Solving Problems of Selection, Diagnostics, and Prediction," Proc. Inst. Math. **1**, 67–101 (2002).
10. M. J. Kearns, Y. Mansour, A. Y. Ng, and D. Ron, "An Experimental and Theoretical Comparison of Model Selection Methods," *8th Conf. Comput. Learning Theory, Santa Cruz*, 21–30 (1995).
11. A. G. Ivakhnenko and Yu. P. Yurachkovskii, *Modeling of Complex Systems from Experimental Data* (Radio i Svyaz', Moscow, 1987) [in Russian].
12. L. Mason, P. Bartlett, and J. Baxter, "Direct Optimization of Margins Improves Generalization in Combined Classifiers," *Tech. Report Dept. Systems Eng.* (Australian Natl. Univ., 1998).

13. R. E. Schapire, Y. Freund, W. S. Lee, and P. Bartlett, "Boosting the Margin: A New Explanation for the Effectiveness of Voting Methods," Ann. Stat. **26**, 1651–1686 (1998).
14. L. Breiman, "Bagging Predictors," Mach. Learning **24** (2), 123–140 (1996).
15. K. V. Vorontsov, "Quality of Dependences Estimated from Empirical Data," *Abstracts of the 7th All-Russia Conf. on Mathematical Methods in Pattern Recognition, Pushchino, 1995*, pp. 24–26 [in Russian].
16. Yu. I. Zhuravlev, "Correct Algebras over Sets of Incorrect (Heuristic) Algorithms, Parts I–III," Kibernetika, No. 4, 5–17 (1977); No. 6, 21–27 (1977); No. 2, 35–43 (1978).
17. J. Rissanen, "Modeling by Shortest Data Description," Automatica **14**, 465–471 (1978).
18. V. Vapnik, E. Levin, and Y. L. Cun, "Measuring the VC-Dimension of a Learning Machine," Neural Comput. **6**, 851–876 (1994).
19. O. Bousquet and A. Elisseeff, "Stability and Generalization," J. Mach. Learning Res., No. 2, 499–526 (2002).
20. P. Bartlett, "The Sample Complexity of Pattern Classification with Neural Networks: The Size of the Weights Is More Important than the Size of the Network," IEEE Trans. Inform. Theory **44**, 525–536 (1998).
21. K. V. Rudakov, "Universal and Local Constraints in the Correction of Heuristic Algorithms," Kibernetika, No. 2, 30–35 (1987).
22. K. V. Vorontsov, "Estimation of the Performance of a Monotone Decision Rule out of a Training Set," in *Proceedings of Int. Scientific Conf. on Intellectualization of Data Processing* (Krymsk. Nauch. Tsentr NAN Ukr., Tavrich. Nats. Univ., Simferopol, 2002).
23. A. N. Semochkin, *Bounds on the Performance Functional for Class of Algorithms with Universal Monotonicity Constraints*, Available from VINITI, No. 2965-B98 (1998).
24. J. Sill, "The Capacity of Monotonic Functions," Discrete Appl. Math. (Special Issue on VC-Dimension) **86**, 95–107 (1998).