

THE INTERACTION OF LEARNING AND SELF-ORGANIZATION IN PATTERN RECOGNITION

M. I. Shlezinger

Kibernetika, Vol. 4, No. 2, pp. 81-88, 1968.

UDC 51:681.14:155

INTRODUCTION

The solution of the pattern recognition problem in its statistical formulation [1-3] requires that over a certain space  $V$  of patterns there should be given a distribution  $p(v/k)$  of patterns subject to the condition that they belong to the  $k$ -th image; in addition one needs to know the a priori probabilities of the images  $p_k$  ( $k = 1, 2, \dots, s$ , where  $s$  is the number of images). On the basis of these data it is possible to construct a decision function  $h(v)$  indicating that the pattern  $v$  is to be assigned to the image  $h$ . It is possible to find a decision function such that the probability of erroneous classification is minimized.

Learning and self-organization in recognition systems takes place under the condition that the distribution  $p(v/k)$  is not known completely but only correct to an unknown parameter  $a_k$ . In such a case the problem of learning or self-organization consists in determining these parameters. The unknown parameters are determined on the basis of a so-called learning sequence, i. e., a certain set of patterns presented to the recognition system. During learning the presentation of each pattern is accompanied by an indication of which image it belongs to. A pattern is not assigned to an image during self-organization. This is the difference between learning and self-organization in pattern recognition.

FORMULATION OF THE PROBLEM

Let us introduce the following notation. The family of unknown parameters will be denoted by  $A$ :

$$A = \{a_1, a_2, \dots, a_s; p_1, p_2, \dots, p_s\}. \quad (1)$$

Among the unknown parameters we also include the a priori probabilities of the images.

The distribution of patterns belonging to the  $k$ -th image will be denoted by  $p(v/a_k)$ . The distribution of patterns belonging to the whole family of images will be denoted by  $P(v)$ . The following equation is obvious:

$$P(v) = \sum_{k=1}^s p_k \cdot p(v/a_k). \quad (2)$$

It can be seen from (2) that the distribution of patterns belonging to the whole family of images is generally speaking dependent on the family of unknown parameters  $A$ . For this reason this distribution will be denoted by  $P(v/A)$ :

$$P(v/A) = \sum_{k=1}^s p_k \cdot p(v/a_k). \quad (3)$$

We shall denote by  $V_k$  the sample of patterns belonging to the  $k$ -th image, and by  $V$  the sample of patterns belonging to the whole family of images. Then it is clear that the learning sequence presented to the recognition system in a learning mode is the family of samples  $V_k$  ( $k = 1, 2, \dots, s$ ), while in a self-organizing mode the recognition system is presented with the sample  $V$ . The unknown parameters must be determined on the basis of this initial material. Since the initial data is stochastic in character it is not possible to determine the unknown parameters exactly, but only to find estimates for them.

The maximum likelihood estimates are in a certain sense optimal. As is shown in [4], maximum likelihood estimates are effective if for the given parameter there is an effective estimate. In addition, with an increasing sample size the maximum likelihood estimate always approaches an effective one. In view of this we can formulate the problem of learning and self-organization as follows.

**The learning problem.** The distribution of patterns belonging to the  $k$ -th image is known correct to an unknown parameter  $a_k$ , i. e., a function of two variables  $p(v/a_k)$  is given. A sample  $V_k$  of patterns  $v_1, v_2, \dots, v_m$  belonging to the  $k$ -th image is given. A maximum likelihood estimate is to be obtained for the unknown parameter  $a_k$ . It is known [4] that this estimate is the value of  $a_k$  which maximizes the expression

$$l(a_k) = \sum_{i=1}^m \log p(v_i/a_k). \quad (4)$$

Estimates for the a priori probabilities of images can be obtained only when the patterns of various images have been sampled at random according to the a priori probabilities of images. In this case the estimates for the a priori probabilities are determined by quantities which are proportional to the sample size  $V_k$ .

**The self-organization problem.** A sample  $V$  of patterns  $v_1, v_2, \dots, v_m$  from the totality of images is given. A maximum likelihood estimate is to be obtained for the unknown parameters  $a_1, a_2, a_3, \dots, a_s$  and for the a priori probabilities  $p_1, p_2, p_3, \dots, p_s$ , i. e., we seek the maximum of the expression

$$L(A) = \sum_{i=1}^m \log P(v_i/A).$$

Substituting into this expression for  $P(v/A)$  the right-hand side of (3), we obtain

$$L(A) = \sum_{i=1}^m \log \sum_{k=1}^s p_k \cdot p(v_i/a_k). \quad (5)$$

## AN ALGORITHM FOR SOLVING THE SELF-ORGANIZATION PROBLEM

It is easy to understand that the problem of finding the maximum of the expression (4) is not essentially different from the problem of maximizing the expression

$$\sum_{i=1}^m \alpha_{ik} \log p(v_i/a_k). \quad (6)$$

The learning problem is reduced to finding the maximum of (6) when each presentation of a pattern in the learning sequence is accompanied not by an accurate indication of the image to which the pattern belongs, but by a probability  $\alpha_{ik}$  that the  $i$ -th pattern belongs to the  $k$ -th image. The estimates for the a priori probabilities are in this case determined by quantities proportional to  $\sum_{i=1}^m \alpha_{ik}$ .

We assume that an algorithm is known for the learning process, i.e., that there is an algorithm for finding the value of the parameter  $a_k$  at which expression (6) is maximized. In that case we can also formulate an algorithm for self-organization, i.e., an algorithm for finding the values of all unknown parameters and a priori probabilities maximizing expression (5).

The initial data for the algorithm are the presented patterns  $v_1, v_2, v_3, \dots, v_m$ , belonging to the whole family of images.

The algorithm is an iterative procedure which, from the parameters

$$A^{(t)} = \{a_1^{(t)}, a_2^{(t)}, \dots, a_s^{(t)}, p_1^{(t)}, p_2^{(t)}, \dots, p_s^{(t)}\}$$

determines the new values of the parameters

$$A^{(t+1)} = \{a_1^{(t+1)}, a_2^{(t+1)}, \dots, a_s^{(t+1)}, p_1^{(t+1)}, p_2^{(t+1)}, \dots, p_s^{(t+1)}\},$$

which have a greater likelihood than  $A^{(t)}$ .

A single iteration of the algorithm, i.e., the derivation of the parameters  $A^{(t+1)}$  from the earlier parameters  $A^{(t)}$ , consists of two stages.

**Stage 1.** Calculation of the quantities  $\alpha_{ik}(A^{(t)})$  ( $i = 1, 2, \dots, m; k = 1, 2, \dots, s$ ) according to the formula

$$\alpha_{ik}(A^{(t)}) = \frac{p_k^{(t)} \cdot p(v_i/a_k^{(t)})}{\sum_{k=1}^s p_k^{(t)} \cdot p(v_i/a_k^{(t)})}. \quad (7)$$

As can be seen from formula (7), the quantity  $\alpha_{ik}(A^{(t)})$  is the probability that the pattern  $v_i$  belongs to the  $k$ -th image, given that the family of images is actually characterized by the parameters  $A^{(t)}$ . The calculation of these parameters is the most time-consuming part of the recognition system. Therefore it can be stated that the first stage is a recognition algorithm whose results are the a posteriori probabilities  $\alpha_{ik}$ , which are the initial data for the second stage.

**Stage 2.** This consists in finding the quantities  $a_k^{(t+1)}$  ( $k = 1, 2, 3, \dots, s$ ), maximizing the expression

$$l(a_k) = \sum_{i=1}^m \alpha_{ik} \log p(v_i/a_k),$$

and the values  $p_k^{(t+1)}$  ( $k = 1, 2, \dots, s$ ) which are proportional to the quantities  $\sum_{i=1}^m \alpha_{ik}$ .

It is easy to see that the second stage solves the learning problem in our formulation. The results of this stage are the quantities  $a_k^{(t+1)}, p_k^{(t+1)}$  ( $k = 1, 2, \dots, s$ ), which will be used in the first stage of the next iteration. The limiting values  $A^{(t)}$  are maximum likelihood estimates for the unknown parameters  $A$ .

Thus, it is clear that the self-organizing algorithm is a multiple repetition of two procedures, recognition algorithm and learning algorithm.

## BASIS OF THE ALGORITHM

**Lemma 1.** Let  $\alpha_i$  ( $i = 1, 2, \dots, s$ ) be positive constants, and let  $x_i$  be variables such that  $\sum_{i=1}^s x_i = c$ . The maximum value of  $f = \sum_{i=1}^s \alpha_i \log x_i$  is reached when the values of the  $x_i$  are proportional to the values of the  $\alpha_i$ .

The lemma can easily be proved for  $s = 2$ , and then generalized for any  $s$  by mathematical induction.

**Theorem 1.** Let  $A^{(t)}, A^{(t+1)}$  be the values of the unknown parameters obtained respectively after the  $t$ -th and the  $(t+1)$ -th iteration of the self-organization algorithm. Then if  $A^{(t)} \neq A^{(t+1)}$ , we have  $L(A^{(t)}) < L(A^{(t+1)})$ .

**Proof.** Since  $\sum_{k=1}^s \alpha_{ik} = 1$  for all  $i$  (see formula (7)), we can express  $L(A^{(t)})$  as follows:

$$\begin{aligned} L(A^{(t)}) &= \sum_{i=1}^m \log \sum_{k=1}^s p_k^{(t)} \cdot p(v_i/a_k^{(t)}) = \\ &= \sum_{k=1}^s \sum_{i=1}^m \alpha_{ik}(A^{(t)}) \log p_k^{(t)} + \\ &+ \sum_{k=1}^s \sum_{i=1}^m \alpha_{ik}(A^{(t)}) \log p(v_i/a_k^{(t)}) - \\ &- \sum_{i=1}^m \sum_{k=1}^s \alpha_{ik}(A^{(t)}) \log \frac{p_k^{(t)} \cdot p(v_i/a_k^{(t)})}{\sum_{k=1}^s p_k^{(t)} \cdot p(v_i/a_k^{(t)})}. \end{aligned} \quad (8)$$

We can express  $L(A^{(t+1)})$  similarly:

$$\begin{aligned} L(A^{(t+1)}) &= \sum_{i=1}^m \log \sum_{k=1}^s p_k^{(t+1)} \cdot p(v_i/a_k^{(t+1)}) = \\ &= \sum_{k=1}^s \sum_{i=1}^m \alpha_{ik}(A^{(t)}) \log p_k^{(t+1)} + \\ &+ \sum_{k=1}^s \sum_{i=1}^m \alpha_{ik}(A^{(t)}) \log p(v_i/a_k^{(t+1)}) - \\ &- \sum_{i=1}^m \sum_{k=1}^s \alpha_{ik}(A^{(t)}) \log \frac{p_k^{(t+1)} \cdot p(v_i/a_k^{(t+1)})}{\sum_{k=1}^s p_k^{(t+1)} \cdot p(v_i/a_k^{(t+1)})}. \end{aligned}$$

We now prove three inequalities.

$$\sum_{k=1}^s \sum_{i=1}^m \alpha_{ik}(A^{(t)}) \log p_k^{(t)} < \sum_{k=1}^s \sum_{i=1}^m \alpha_{ik}(A^{(t)}) \log p_k^{(t+1)}; \quad (9)$$

$$\begin{aligned} & \sum_{k=1}^s \sum_{i=1}^m \alpha_{ik}(A^{(t)}) \log p(v_i/a_k^{(t)}) < \\ & < \sum_{k=1}^s \sum_{i=1}^m \alpha_{ik}(A^{(t)}) \log p(v_i/a_k^{(t+1)}); \end{aligned} \quad (10)$$

$$\begin{aligned} & \sum_{i=1}^m \sum_{k=1}^s \alpha_{ik}(A^{(t)}) \log \frac{p_k^{(t)} \cdot p(v_i/a_k^{(t)})}{\sum_{k=1}^s p_k^{(t)} \cdot p(v_i/a_k^{(t)})} \geq \\ & \geq \sum_{i=1}^m \sum_{k=1}^s \alpha_{ik}(A^{(t)}) \log \frac{p_k^{(t+1)} \cdot p(v_i/a_k^{(t+1)})}{\sum_{k=1}^s p_k^{(t+1)} \cdot p(v_i/a_k^{(t+1)})}, \end{aligned} \quad (11)$$

where at least one of the first two inequalities is strictly satisfied.

Let us prove inequality (9).

By definition (stage 2 of the algorithm),  $p_k^{(t+1)}$  is proportional to  $\sum_{i=1}^m \alpha_{ik}(A^{(t)})$ . Also, obviously  $\sum_{k=1}^s p_k^{(t+1)} = \sum_{k=1}^s p_k^{(t)}$ , since both of these sums are equal to 1. Consequently, the conditions of the lemma are satisfied, and the expression

$$\sum_{k=1}^s \sum_{i=1}^m \alpha_{ik}(A^{(t)}) \log p_k \text{ when } p_k = p_k^{(t+1)}$$

reaches an absolute maximum. Hence the inequality (9) is valid. Here equality will be obtained only when  $p_k^{(t)} = p_k^{(t+1)}$  for any  $k$ , since in view of the lemma the expression in question has a unique maximum.

Let us now prove inequality (10).

By definition (stage 2 of the algorithm),  $a_k^{(t+1)}$  ensures the maximum of the expression

$$\sum_{i=1}^m \alpha_{ik}(A^{(t)}) \log p(v_i/a_k),$$

and therefore the inequality

$$\begin{aligned} & \sum_{i=1}^m \alpha_{ik}(A^{(t)}) \log p(v_i/a_k^{(t)}) < \\ & < \sum_{i=1}^m \alpha_{ik}(A^{(t)}) \log p(v_i/a_k^{(t+1)}). \end{aligned} \quad (12)$$

is valid.

Summing expression (12) with respect to the subscript  $k$ , we obtain the inequality (10). Here equality will be obtained only when  $a_k^{(t)} = a_k^{(t+1)}$  for any  $k$ .

According to the conditions of the theorem,  $A^{(t)} \neq A^{(t+1)}$ . This means that at least one unknown parameter or one a priori probability has been changed as a result of the  $(t+1)$ -th iteration, and also that at least one of the inequalities (9) and (10) is strictly satisfied.

Let us prove inequality (11).

We consider the expression

$$\sum_{k=1}^s \alpha_{ik}(A^{(t)}) \log \frac{p_k \cdot p(v_i/a_k)}{\sum_{k=1}^s p_k \cdot p(v_i/a_k)}. \quad (13)$$

By definition (see (7)), when  $p_k = p_k^{(t)}$ ,  $a_k = a_k^{(t)}$  the logarithmic expression is equal to the coefficient  $\alpha_{ik}(A^{(t)})$ . Consequently, according to Lemma 1, when  $p_k = p_k^{(t)}$ ,  $a_k = a_k^{(t)}$  expression (13) reaches its maximum, whence we immediately have

$$\begin{aligned} & \sum_{k=1}^s \alpha_{ik}(A^{(t)}) \log \frac{p_k^{(t)} \cdot p(v_i/a_k^{(t)})}{\sum_{k=1}^s p_k^{(t)} \cdot p(v_i/a_k^{(t)})} \geq \\ & \geq \sum_{k=1}^s \alpha_{ik}(A^{(t)}) \log \frac{p_k^{(t+1)} \cdot p(v_i/a_k^{(t+1)})}{\sum_{k=1}^s p_k^{(t+1)} \cdot p(v_i/a_k^{(t+1)})}. \end{aligned}$$

Summing this over the subscript  $i$ , we obtain inequality (11).

From inequalities (9)–(11) it follows that  $L(A^{(t)}) < L(A^{(t+1)})$ , and the theorem is proved.

**Corollary.** The quantity  $L(A^{(t+1)}) - L(A^{(t)})$  tends to zero with increasing  $t$ .

**Proof.** From Theorem 1,  $L(A^{(0)})$ ,  $L(A^{(1)})$ , ...,  $L(A^{(t)})$  is a monotonically increasing sequence. It is bounded from above by the value of the actual maximum of likelihood. As is known, such a sequence has a limit. The equation

$$\lim_{t \rightarrow \infty} [L(A^{(t+1)}) - L(A^{(t)})] = 0 \quad (14)$$

is a necessary condition for the existence of a limit for  $L(A^{(t)})$ . Therefore Eq. (14) is valid, which is what we wanted to prove.

Theorem 1 and its corollary reflect an important property of the algorithm. Nevertheless from these we cannot immediately conclude that the algorithm converges, i.e., that a limit exists for the values directly computable by the algorithm.

In order to prove the convergence of the algorithm the following theorem is useful.

**Theorem 2.** The quantity  $\alpha_{ik}(A^{(t+1)}) - \alpha_{ik}(A^{(t)})$  tends to zero with increasing  $t$ .

**Proof.** We introduce the notation

$$\Delta^{(t)} = L(A^{(t+1)}) - L(A^{(t)}).$$

On the basis of expressions (7) and (8) we can write

$$\begin{aligned} \Delta^{(t)} &= \sum_{k=1}^s \sum_{i=1}^m \alpha_{ik}(A^{(t)}) \log \frac{p_k^{(t+1)} \cdot p(v_i/a_k^{(t+1)})}{p_k^{(t)} \cdot p(v_i/a_k^{(t)})} - \\ & - \sum_{i=1}^m \sum_{k=1}^s \alpha_{ik}(A^{(t)}) \log \frac{\alpha_{ik}(A^{(t+1)})}{\alpha_{ik}(A^{(t)})}. \end{aligned}$$

In the course of proving Theorem 1 we have shown that

$$\sum_{k=1}^s \sum_{i=1}^m \alpha_{ik}(A^{(t)}) \log \frac{p_k^{(t+1)} \cdot p(v_i/a_k^{(t+1)})}{p_k^{(t)} \cdot p(v_i/a_k^{(t)})} > 0.$$

Therefore

$$\sum_{i=1}^m \sum_{k=1}^s \alpha_{ik}(A^{(t)}) \log \frac{\alpha_{ik}(A^{(t)})}{\alpha_{ik}(A^{(t+1)})} < \Delta^{(t)}.$$

Since in the last expression, in view of the lemma, all terms following the summation sign with respect to  $i$  are nonnegative, for all  $i$  we have

$$0 < \sum_{k=1}^s \alpha_{ik}(A^{(t)}) \log \frac{\alpha_{ik}(A^{(t)})}{\alpha_{ik}(A^{(t+1)})} < \Delta^{(t)}.$$

From this, and also in view of the fact that  $\Delta^{(t)}$  approaches zero, we can write

$$\lim_{t \rightarrow \infty} \sum_{k=1}^s \alpha_{ik}(A^{(t)}) \log \frac{\alpha_{ik}(A^{(t)})}{\alpha_{ik}(A^{(t+1)})} = 0 \quad (i = 1, 2, \dots, m). \quad (15)$$

Expression (15) is valid only when

$$\lim_{t \rightarrow \infty} (\alpha_{ik}(A^{(t+1)}) - \alpha_{ik}(A^{(t)})) = 0 \quad (i = 1, 2, \dots, m; k = 1, 2, \dots, s),$$

since  $\sum_{k=1}^s \alpha_{ik}(A^{(t)}) \log \alpha_{ik}$  has a unique maximum when  $\sum_{k=1}^s \alpha_{ik} = 1$  and it is reached when

$$\alpha_{ik} = \alpha_{ik}(A^{(t)}).$$

This proves the theorem.

We introduce some new concepts and notation which will be used later.

The set  $m \cdot s$  of numbers  $\alpha_{ik}$  ( $i = 1, 2, \dots, m; k = 1, 2, \dots, s$ ) will be denoted by  $\vec{\alpha}$ . The set of all possible values of  $\vec{\alpha}$  will be denoted by  $\Omega$ . Some subsets of  $\Omega$  will be denoted by  $\{\vec{\alpha}\}$ . It is obvious that one iteration of the algorithm can be regarded as a realization of some function  $F(\vec{\alpha})$  mapping the set  $\Omega$  into itself.

We shall say that the vector  $\vec{\alpha}^{(t)}$  tends to the set  $\{\vec{\alpha}\}$  if

$$\lim_{t \rightarrow \infty} \min_{\vec{\alpha} \in \{\vec{\alpha}\}} (\vec{\alpha}^{(t)} - \vec{\alpha}) = 0.$$

It is clear that if the set  $\{\vec{\alpha}\}$  contains a unique point, then the vector  $\vec{\alpha}^{(t)}$  has a limit in the ordinary sense.

**Lemma 2.** If the vector  $\vec{\alpha}^{(t)}$  approaches the set  $\{\vec{\alpha}\}$  consisting of a finite number of points, and the quantity  $|\vec{\alpha}^{(t+1)} - \vec{\alpha}^{(t)}|$  approaches zero, then the vector  $\vec{\alpha}^{(t)}$  approaches one of the points of the set  $\{\vec{\alpha}\}$ .

**Proof.** Let us assume that the vector  $\vec{\alpha}^{(t)}$  does not approach any element of the set  $\{\vec{\alpha}\}$ . In this case with very large  $T$  and very small  $\varepsilon$  there are two vectors  $\vec{\alpha}^{(t)}$  and  $\vec{\alpha}^{(t+1)}$  ( $t > T$ ) such that the distance between them will be greater than  $\delta - 2\varepsilon$ , where  $\delta$  is the distance between two neighboring elements of the set  $\{\vec{\alpha}\}$ . However, this contradicts the condition that the distance between  $\vec{\alpha}^{(t)}$  and  $\vec{\alpha}^{(t+1)}$  approaches zero.

The lemma is proved.

We shall say that the set  $\{\vec{\alpha}\}^{(t)}$  approaches the set  $\{\vec{\alpha}\}$  if any vector  $\vec{\alpha}^{(t)} \in \{\vec{\alpha}\}^{(t)}$  approaches  $\{\vec{\alpha}\}$ .

The roots of the equation

$$|\vec{\alpha} - F(\vec{\alpha})| = 0, \quad (16)$$

where  $F(\vec{\alpha})$  is a function realizable by one iteration of the algorithm, will be called the stationary point of the algorithm. As will be shown later, the roots of

Eq. (16) are roots of a system of likelihood equations representing the classical necessary conditions for the maximum of the likelihood function.

Let us consider an equation differing in its right-hand side from Eq. (16):

$$|\vec{\alpha} - F(\vec{\alpha})| = \varepsilon^{(t)}. \quad (17)$$

To Eq. (17) there corresponds a set  $\{\vec{\alpha}\}^{(t)}$  of roots. We assume that Eq. (16) has a solution which is stable and not very different from the right-hand side, i.e., that the set  $\{\vec{\alpha}\}^{(t)}$  of roots of Eq. (17) approaches the set  $\{\vec{\alpha}\}$  of roots of Eq. (16) if  $\varepsilon^{(t)}$  approaches zero.

It is obvious that each of the vectors  $\vec{\alpha}^{(1)}, \vec{\alpha}^{(2)}, \dots, \vec{\alpha}^{(t)}, \dots$  is a solution of the corresponding equations  $|\vec{\alpha} - F(\vec{\alpha})| = \varepsilon^{(1)}, \varepsilon^{(2)}, \dots, \varepsilon^{(t)}, \dots$ , where  $\varepsilon^{(t)}$  approaches zero. In view of this it can be stated that the vector  $\vec{\alpha}^{(t)}$  approaches the set of roots of Eq. (16).

If we assume that Eq. (16) has a finite number of roots, then in view of Theorem 2 and Lemma 2 we can say that the sequence of vectors  $\vec{\alpha}^{(1)}, \vec{\alpha}^{(2)}, \dots, \vec{\alpha}^{(t)}$ , approaches one of the roots of Eq. (16).

The following theorem defines the properties of the roots of Eq. (16).

**Theorem 3.** If a certain vector  $\vec{\alpha}$  is a stationary point of the algorithm, then the quantities  $p_k^*$  and  $a_k^*$ , determined at the second stage of a single iteration of the algorithm, are roots of the system of equations

$$\frac{\partial L(A)}{\partial a_k} = 0 \quad (k = 1, 2, \dots, s), \quad (18)$$

$$\frac{\partial L(A)}{\partial p_k} - \lambda = 0 \quad (k = 1, 2, \dots, s), \quad (19)$$

$$\sum_{k=1}^s p_k = 1. \quad (20)$$

**Proof.** We introduce the notation

$$L^{(1)} = \sum_{k=1}^s \sum_{i=1}^m \alpha_{ik} \log p_k;$$

$$L_k^{(2)} = \sum_{i=1}^m \alpha_{ik} \log p(v_i/a_k);$$

$$L_i^{(3)} = \sum_{k=1}^s \alpha_{ik} \log \frac{p_k \cdot p(v_i/a_k)}{\sum_{k=1}^s p_k \cdot p(v_i/a_k)}.$$

Expression (3) above can be written in the form

$$L(A) = L^{(1)} + \sum_{k=1}^s L_k^{(2)} - \sum_{i=1}^m L_i^{(3)}.$$

Since  $L^{(1)}$  and  $L_k^{(2)}$  are independent of  $a_k$  and  $p_k$  respectively, we can write

$$\frac{\partial L(A)}{\partial a_k} = \sum_{k=1}^s \frac{\partial L_k^{(2)}}{\partial a_k} - \sum_{i=1}^m \frac{\partial L_i^{(3)}}{\partial a_k}, \quad (21)$$

$$\frac{\partial L(A)}{\partial p_k} = \frac{\partial L^{(1)}}{\partial p_k} - \sum_{i=1}^m \frac{\partial L_i^{(3)}}{\partial p_k}. \quad (22)$$

The conditions of the theorem are equivalent to the following three conditions:

a) the value  $a_k^*$  maximizes the expression

$$\sum_{i=1}^m \alpha_{ik} \log p(v_i/a_k);$$

b) the quantity  $p_k^*$  is proportional to the values  $\sum_{i=1}^m \alpha_{ik}$ , where  $\sum_{k=1}^s p_k^* = 1$ ;

c) the quantity  $\alpha_{ik}$  is equal to

$$\alpha_{ik} = \frac{p_k^* \cdot p(v_i/a_k^*)}{\sum_{k=1}^s p_k^* \cdot p(v_i/a_k^*)}.$$

In view of condition (c) and Lemma 1 the quantity  $L_i^{(3)}$ , when  $a_k = a_k^*$ ,  $p_k = p_k^*$ , reaches an absolute maximum. Therefore the following equations are valid:

$$\frac{\partial L_i^{(3)}}{\partial a_k} = 0, \quad (23)$$

$$\frac{\partial L_i^{(3)}}{\partial p_k} = 0. \quad (24)$$

In view of condition (a) the quantity  $a_k^*$  ensures the maximum of the expression  $L_k^{(2)}$ . If  $L_k^{(2)}$  is differentiable the derivative  $\partial L_k^{(2)}/\partial a_k$  is equal to zero when  $a_k = a_k^*$ :

$$\frac{\partial L_k^{(2)}}{\partial a_k} = 0. \quad (25)$$

It is easy to see that the derivative  $\partial L^{(1)}/\partial p_k$  is independent of  $k$  when  $p_k = p_k^*$  (and  $p_k^*$  satisfies condition (1)), i.e., it is equal to some constant  $\lambda$ :

$$\frac{\partial L^{(1)}}{\partial p_k} = \lambda. \quad (26)$$

On the basis of Eqs. (23) and (25) we ascertain that expression (21) is equal to zero, and consequently Eq. (18) is valid.

In the same way Eq. (19) is a consequence of Eqs. (24) and (26).

Equation (20) is a direct consequence of condition (b).

The theorem is proved.

As a consequence of this theorem it can be stated that the algorithm described converges to that value of  $\vec{\alpha}^*$  which is a root of the system of equations (18)–(20). Generally speaking, not all roots of the system of equations correspond to the maximum of the function  $L(A)$ . Certain roots correspond to minima or to so-called false extrema of the function  $L(A)$ . However, in view of the fact that in the operation of the algorithm the estimates of the parameters can only be improved, the algorithm converges only to the point which corresponds to the maximum of the function  $L(A)$ . This is the main advantage of the proposed algorithm for finding the estimates compared with the method based on direct solution of the system of equations.

Finally it should be noted that the algorithm is also applicable when for each image several parameters have to be determined. All of the assertions of the paper, apart from Theorem 3, are valid regardless of what  $a_k$  stands for, whether a single unknown parameter or several parameters. As for Theorem 3, it can easily be formulated and proved for the case of many unknown parameters for a single image. For this purpose it is necessary to consider the partial derivative not with respect to the parameter  $a_k$ , but with respect to all parameters occurring in  $a_k$ . An experimental test of the algorithm has been carried out for just such a case, when the number of unknown parameters was very large. A description of the experiments will be contained in another paper. Here we merely mention that the results of the experiments were positive.

#### OTHER SOLUTIONS OF THE LEARNING PROBLEM

The first paper which is known to us about the self-organization problem is [6]. That paper, like [7], does not use the words "self-organization" and "recognition." Nevertheless, the problems solved in those papers are very close to ours. Expressing the contents of these papers in our terms, we can say that they consider the problem of finding the a priori probabilities of images according to a sample of patterns belonging to a whole family of images. The conditional distributions of patterns belonging to one or another class are assumed to be known completely. The problem of determining the a priori probabilities is solved for certain specific cases of one-dimensional distributions. In [6] it is stated that estimates for the a priori probabilities can be found by maximum likelihood, but no method is suggested for finding these estimates.

The paper [8] contains a rigorous formulation of the self-organization problem. This paper indicates that in a number of cases it is desirable to find estimates for the maximum likelihood. When finding such estimates requires a great deal of computation it is suggested that the minimum chi-square method should be used. According to this criterion the estimates are found by the method of steepest descent.

The papers [9] and [10] are very similar in content and solve the problem of self-organization for the case when the image patterns have a normal distribution.

The papers [11–13] contain various solutions of the self-organization problem, although their formulation is essentially different from the problem solved in this paper.

The concept of "self-organization" is often related to Rosenblatt-type perceptron systems. It was Rosenblatt who first used this term in the papers [14–15], although he did not present a rigorous formulation of the problem and, as has been shown in [16], did not solve the problem of self-organization.

The results of the present paper have been reported at the seminars on pattern recognition and automatic control theory of the Scientific Committee on Cybernetics AS USSR. The author is grateful to the participants of these seminars, whose friendly criticism facilitated a more rigorous formulation of the theorems.

## REFERENCES

1. T. Anderson, Introduction to Multivariable Statistical Analysis [Russian translation], GIFML, Moscow, pp. 175-183, 1963.
2. C. K. Chow, "An optimum character recognition system using decision function," IRE Trans. on El. Computers, EC-6, p. 247, 1957.
3. V. A. Kovalevskii, "The pattern recognition problem from the statistical point of view," collection: Reading Automata [in Russian], izd-vo Naukova dumka, Kiev, 1965.
4. I. G. Kramer, Mathematical Methods of Statistics [Russian translation], IL, 1949.
5. A. S. Barashko, et al., "ChARS—a correlational reading automaton with a shift register," collection: Reading Automata [in Russian], izd-vo Naukova dumka, Kiev, 1965.
6. G. Robbins, "Asymptotic subminimax solutions in complex statistical decision theory," collection: Matematika [Russian translation], 8, no. 2, 141-159, 1964.
7. G. Robbins, The empirical Bayesian approach in statistics, *ibid.*
8. A. V. Milen'kii, "Determining the statistical characteristics of recognized images in a self-organization regime," Kibernetika [Cybernetics], no. 3, 1967.
9. D. B. Cooper and P. W. Cooper, "Nonsuper-vized adaptive signal detection and pattern recognition," Inf. and Control, v. 7, no. 3, 1964.
10. O. G. Zhuravlev and I. Sh. Torgovitskii, "An optimal method of objective classification in pattern recognition problems," Avtomatika i telemekhanika, 26, 2062-2063, 1965.
11. M. I. Shlezinger, "Spontaneous discrimination of patterns," collection: Reading Automata [in Russian], izd-vo Naukova dumka, Kiev, 1965.
12. A. A. Dorofeyuk, "A learning algorithm for the machine recognition of patterns without a trainer, using the method of potential functions," Avtomatika i telemekhanika, no. 10, 78-87, 1966.
13. E. M. Braverman, "The method of potential functions in the learning of pattern recognition by a machine without a trainer," Avtomatika i telemekhanika, no. 10, 100-121, 1966.
14. F. Rosenblatt, "Two theorems of statistical separability in the perceptron," Symp. Mechaniz. Thought Proc., Nat. Phys. Lab., Teddington, England, 1958.
15. F. Rosenblatt, "Perceptron simulation experiments," Proc. IRE, v. 48, no. 3, 1960.
16. V. M. Glushkov, "The problem of self-organization in the perceptron," Zhurnal vychislitel'noi matematiki i matematicheskoi fiziki, 2, no. 6, 1962.

20 April 1967