An interval approach to recognition of numerical matrices*

A.V. Prolubnikov
Omsk State University
a.v.prolubnikov@mail.ru

Abstract

We present an interval approach to recognition of numerical matrices. For solution of the problem, we construct systems of interval linear equations that associated with given numerical pattern matrices. Considering a system of interval linear equations as a family of real linear equations systems, we use a measure of variation of these systems solutions as a measure of closeness between matrices. As an application of the approach, we use it for recognition of raster images that are distorted in the course of noising. The results of computational experiments are presented.

Keywords: pattern recognition, interval analysis
AMS subject classifications: 68T10

1 Introduction

Registration of data by technical means is often complicated by noise that interfere the registration process. Measurement errors are examples of such noise. A common problem is to recognize the patterns under specified constraints on the noise. Considering the data presented in the matrix form, we are given a set of pattern matrices and the matrix that is obtained from some pattern matrix in the course of noising. We need to identify this pattern matrix. An obvious example of the problem under consideration is recognition of raster images.

Existing methods of recognition for such problems may be divided in two classes: those that are use preliminary learning stage and those that does not. Learning implies that a processing unit is capable of changing its input/output behavior as a result of changes in the input. Algorithms on the basis of the theory of morphological analysis, Kora-type algorithms, algorithms based on neural networks are some examples of the first class methods. The learning stage is absent or degenerated in the second class methods. The examples of these methods are the nearest neighbor method, the k nearest neighbor method, the potential function method, the Parzen window method.

In the presented approach, we propose an original measure of closeness between matrices and the algorithm which may be attributed to the second class. For given

*Submitted: 30 december 2012; Revised: 10 october 2013; Accepted: ?? ???????? ????.

input numerical matrices, we construct systems of interval linear equations. Considering an interval system of equations as a set of real systems of equations, we perform recognition by minimizing a measure of variation of the real systems solutions. For estimation of the variation, we use the Lebesgue measure of the united solution sets of the interval linear systems.

2 Preliminaries

We use the following notations and definitions throughout the paper.

We use boldface font for intervals, interval vectors and interval matrices. We use a special font to denote input matrices: we denote the k-th pattern matrix as $\mathsf{A}^{(k)}$ and the matrix to be recognized as A, $\mathsf{a}_{ij}^{(k)}$ and a_{ij} denote elements of these matrices.

For $A \in \mathbb{R}^{m\times n}$, $A = (a_{ij})$, we use p-norm $\|A\|_p$:

$$\|A\|_p = \left(\sum_{i=1}^{m}\sum_{j=1}^{n} |a_{ij}|^p\right)^{1/p},$$

$p \in \mathbb{R}$, $p \geq 1$. Also we use the vector norm $||x||_\infty$ for $x \in \mathbb{R}^n$:

$$||x||_\infty = \max_{1\leq i\leq n} |x_i|.$$

For $A \in \mathbb{R}^{m\times n}$, the matrix norm induced by $\|\cdot\|_\infty$ is denoted as $||A||_\infty$. The norm may be computed as

$$||A||_\infty = \max_{1\leq i\leq m}\left(\sum_{j=1}^{n} |a_{ij}|\right).$$

Definition 1. Let $\boldsymbol{A}x = \boldsymbol{b}$ be an interval linear system of equations, $\boldsymbol{A} \in \mathbb{IR}^{n\times n}$, $\mathrm{b} \in \mathbb{IR}^n$. Then the *united solution set* of this system is the set

$$\Xi(\boldsymbol{A}, \boldsymbol{b}) = \left\{x \in \mathbb{R}^n \;\middle|\; \exists A \in \boldsymbol{A}, \exists b \in \boldsymbol{b} : Ax = b\right\}.$$

Definition 2. A *comparison matrix* of a matrix $A \in \mathbb{R}^{n\times n}$ is a matrix $\langle A\rangle \in \mathbb{R}^{n\times n}$ with elements

$$\langle A\rangle_{ij} = \begin{cases} |a_{ij}|, & \text{if } i = j,\\ -|a_{ij}|, & \text{if } i \neq j.\end{cases}$$

Definition 3. An *M-matrix* is a matrix $A \in \mathbb{R}^{n\times n}$ such that it may be presented as $A = sI - P$, where P is a non-negative matrix and $s > \rho(P)$, $\rho(P)$ is a spectral radius of P. A matrix $\boldsymbol{A} \in \mathbb{IR}^{n\times n}$ is an *interval M-matrix* if every real matrix $A \in \boldsymbol{A}$ is an M-matrix.

Definition 4. An *H-matrix* is a matrix $A \in \mathbb{R}^{n\times n}$ such that its comparison matrix is an M-matrix. An *interval H-matrix* is a matrix $\boldsymbol{A} \in \mathbb{IR}^{n\times n}$ such that every real matrix $A \in \boldsymbol{A}$ is an H-matrix.

Let us abbreviate the term interval linear system of equations" as *ILSE*.

3 Recognition of numerical matrices using the Lebesgue measure of united solution set of ILSE

3.1 General idea

The investigated problem is formulated as follows. We are given a set of N $n \times n$-matrices $\mathsf{A}^{(1)}, \ldots, \mathsf{A}^{(N)} \in \mathbb{R}^{n\times n}$. The matrix $\mathsf{A} = (\mathsf{a}_{ij})$ is obtained from some matrix $\mathsf{A}^{(p)} = (\mathsf{a}_{ij}^{(p)})$ in the course of noising, $p \in \{1, \ldots, N\}$. It is known that values of elements of $\mathsf{A}^{(p)}$ may vary within the intervals $[\mathsf{a}_{ij}^{(p)} - \Delta, \mathsf{a}_{ij}^{(p)} + \Delta]$, $\Delta \geq 0$ $(i, j = \overline{1, n})$. We need to identify p.

Without loss of generality, we may assume that the input matrices $\mathsf{A}^{(1)}, \ldots, \mathsf{A}^{(N)}, \mathsf{A}$ are square. In other case, if the matrices are $m \times n$-matrices and $m < n$, then we append $n - m$ zero rows to every input matrix, and if $n < m$, then we append $m - n$ zero columns.

For solution of the problem, we use a special measure of closeness between matrices. Suppose we have two matrices $A, B \in \mathbb{R}^{n\times n}$, $A = (a_{ij})$, $B = (b_{ij})$. Let us construct an interval matrix $\boldsymbol{C}$ using matrices A and B. Elements $(\boldsymbol{C})_{ij}$ of the matrix $\boldsymbol{C}$ are the intervals

$$(\boldsymbol{C})_{ij} = [\min\{a_{ij}, b_{ij}\}, \max\{a_{ij}, b_{ij}\}]. \tag{1}$$

These intervals specify the variations of the matrix B elements that are needed to obtain the matrix A by modifying elements of B.

Assuming that A is obtained from B as a result of noising, we measure aggregate variation of the matrix B elements by using the Lebesgue measure of the united solution set $\Xi(\boldsymbol{C}, b)$ for some rigth-hand side vector $b \in \mathbb{R}^n$. The Lebesgue measure of this set depends on a mutual disposition of the matrices elements and it depends continuously on their changes. We use the Lebesgue measure $\mu(\Xi(\boldsymbol{C}, b))$ as a measure of closeness $\delta_\Xi(A, B)$ between the matrices A and B:

$$\delta_\Xi(A, B) = \mu(\Xi(\boldsymbol{C}, b)).$$

The less the value of $\delta_\Xi(A, B)$, the closer A to B.

If A is the matrix that was obtained from the pattern matrix $\mathsf{A}^{(p)}$ during the course of noising and if elements of the matrix $\mathsf{A}^{(p)}$ have been varied in some restricted intervals during the process, then the value of the measure will be small enough to recognize $\mathsf{A}^{(p)}$ from the other candidates. Assuming this, we use the heuristic

$$p = \arg\min_k \delta_\Xi(\mathsf{A}, \mathsf{A}^{(k)}) \tag{2}$$

for recognition.

The set $\Xi = \Xi(\boldsymbol{C}, b)$ is a union of not more than 2^n polyhedrons [**?**]. These polyhedrons are intersections of Ξ with orthants of $\mathbb{R}^n$. The problem of description of this set has an exponential computational complexity by itself, so the computation of the Lebesgue measure of Ξ also has an exponential complexity. Thereby we just estimate the Lebesgue measure of Ξ. We do this by computing the Lebesgue measure of some approximation of the interval hull $\square\Xi$ of this set. An interval hull $\square\Xi$ is a box such that $\Xi \subset \square\Xi$ and $\square\Xi \subseteq \boldsymbol{W}$ for every box $\boldsymbol{W}$ such that $\Xi \subset \boldsymbol{W}$.

Let $\boldsymbol{X} = ([\underline{x}_1, \overline{x}_1], \ldots, [\underline{x}_n, \overline{x}_n])^\top$ be an approximation of $\square\Xi$ that was obtained using some interval algorithm *Encl*. $\boldsymbol{X}$ is a box such that $\square\Xi \subseteq \boldsymbol{X}$. The Lebesgue measure μ of $\boldsymbol{X}$ is computed as

$$\mu(\boldsymbol{X}) = (\overline{x}_1 - \underline{x}_1) \cdot \ldots \cdot (\overline{x}_n - \underline{x}_n).$$

For every $k \in \{1, \ldots, N\}$, let us construct matrices $\boldsymbol{C}^{(k)}$ according to (1) and taking A as A and $\mathsf{A}^{(k)}$ as B. The following natural suggestion is a base for the further presented recognition algorithm: for some right-hand side vector $b \in \mathbb{R}^n$, the less the variation of solutions of the real systems that give $\boldsymbol{C}^{(k)}x = b$, the more likely that A was obtained from the pattern matrix $\mathsf{A}^{(k)}$.

The presented algorithm is a nearest neighbor algorithm that uses measure of closeness δ_Ξ as a distance between matrices. Denoting an approximation of $\square\Xi^{(k)}$ as $\boldsymbol{X}^{(k)}$, we formulate the recognition algorithm:

The algorithm for recognition of numerical matrices

Input: the pattern matrices $\mathsf{A}^{(1)}, \ldots, \mathsf{A}^{(N)}$ and the matrix A that was obtained from some pattern matrix in the course of noising.
Output: index p of one of the pattern matrices as a result of recognition.

1. For $\mathsf{A}^{(k)}$ and A, obtain $\boldsymbol{C}^{(k)}$ according to (1), $k = \overline{1, N}$.

2. Using an algorithm *Encl*, compute $\boldsymbol{X}^{(k)}$, $k = \overline{1, N}$.

3. Find p such that $p = \arg\min_k \mu(\boldsymbol{X}^{(k)})$.

3.2 Modifications of the matrices $\boldsymbol{C}^{(k)}$

To perform recognition efficiently, we do the following modifications of input matrices. First, we increase every element of every input matrix by the same value υ ($\upsilon > 0$):

$$\mathsf{a}_{ij} := \mathsf{a}_{ij} + \upsilon, \ \mathsf{a}_{ij}^{(k)} := \mathsf{a}_{ij}^{(k)} + \upsilon, \tag{3}$$

$k = \overline{1, N}$. The transformations (3) preserve distance between elements in different positions of a single matrix and preserve distance between elements in the same positions of two different matrices too. Thus, these transformations preserve all of the input information that we may use for recognition purposes.

As a result of this modification, we decrease the ratio $\Delta/|\mathsf{a}_{ij}^{(k)}|$ for elements of the matrices. The ratio $\Delta/|\mathsf{a}_{ij}^{(k)} + \upsilon|$ decreases with growth of υ. Such a decreasing is necessary for the next reason. If the value Δ is greater or equal to absolute values of the pattern matrices elements, then, for $C \in \boldsymbol{C}^{(k)}$, the vectors $x = C^{-1}e$ may differ so much that we cannot take any decision analysing distances between them. But if the radius Δ is small enough with respect to moduli of pattern matrices elements, then we often may take such a decision.

For example, in matrices of black-and-white images, white pixels are encoded as 1's and black pixels are encoded as 0's. If some pixels of the pattern image are inverted in the course of noising, then Δ is greater or equal than every element of the input matrices. We cannot perform recognition using our heuristic at this case. But if we modify the input matrices according to (3) using $\upsilon = 10$ and having $\Delta / |\mathsf{a}_{ij}^{(k)} + \upsilon| \le 0.1$ as a result, then we obtain efficient recognition.

Transformations (3) of the input matrices are equivalent to the transformations of the interval matrices $\boldsymbol{C}^{(k)}$:

$$\boldsymbol{C}^{(k)} := \boldsymbol{C}^{(k)} + \upsilon \boldsymbol{E}, \tag{4}$$

where $\boldsymbol{E}$ is a matrix with elements $(\boldsymbol{E})_{ij} = [1, 1]$, $i, j = \overline{1, n}$.

As a result of the second modification of the matrices $\boldsymbol{C}^{(k)}$, we have diagonally dominant matrices $\boldsymbol{C}^{(k)}$. The diagonal dominance is needed for obtaining the enclosures $\boldsymbol{X}^{(k)}$ using the interval Gauss-Seidel method.

Let $\boldsymbol{D} \in \mathbb{IR}^{n\times n}$ be a diagonal matrix with elements $(\boldsymbol{D})_{ii} = [d, d]$, $i = \overline{1, n}$, where

$$d = 2\max\left\{ \max_{i=\overline{1,n}} \left(\sum_{j\neq i} |\mathsf{a}_{ij}| \right), \max_{i=\overline{1,n}} \left(\sum_{j\neq i} |\mathsf{a}_{ij}^{(1)}| \right), \ldots, \max_{i=\overline{1,n}} \left(\sum_{j\neq i} |\mathsf{a}_{ij}^{(N)}| \right) \right\}.$$

Taking such d, we modify the matrices $\boldsymbol{C}^{(k)}$:

$$\boldsymbol{C}^{(k)} := \boldsymbol{C}^{(k)} + \boldsymbol{D}. \tag{5}$$

3.3 Choice of a right-hand side vector of ILSE

For the interval system of equations $\boldsymbol{C}x = \boldsymbol{b}$, the right-hand side $\boldsymbol{b}$ is a vector that defines the set $\Xi(\boldsymbol{C}, \boldsymbol{b})$ for a given matrix $\boldsymbol{C}$. The choice of $\boldsymbol{b}$ is important to efficiency of the presented algorithm. We choose a real vector as a right-hand side vector. Such selection of $\boldsymbol{b}$ allows us to obtain more precise enclosures of the united solution sets, because it decreases the distance between $\Xi(\boldsymbol{C}, \boldsymbol{b})$ and $\square\Xi(\boldsymbol{C}, \boldsymbol{b})$ [**?**].

If the right-hand side $\boldsymbol{b}$ is a real vector b, then

$$\Xi(\boldsymbol{C}, b) = \left\{ x \in \mathbb{R}^n \;\middle|\; \exists C \in \boldsymbol{C} : Cx = b \right\}.$$

As a result of the transformation (5), the matrices $\boldsymbol{C}^{(k)}$ become invertible, i.e., every $C \in \boldsymbol{C}^{(k)}$ becomes an invertible matrix. For an invertible matrix $\boldsymbol{C}$, we have

$$\Xi(\boldsymbol{C}, b) = \left\{ C^{-1} b \;\middle|\; C \in \boldsymbol{C} \right\}.$$

Let us consider the set $\Xi(\boldsymbol{C}, b)$ as the image of the set of real matrices $\boldsymbol{C}$. $\Xi(\boldsymbol{C}, b)$ is the image of the mapping L_b:

$$L_b : \boldsymbol{C} \to \Xi(\boldsymbol{C}, b),$$

$$L_b(C) = C^{-1} b$$

for $C \in \boldsymbol{C}$. If we denote elements of C^{-1} as σ_{ij}, then, for the i-th component of some $x \in \Xi(\boldsymbol{C}, b)$, we have

$$x_i = \sum_{j=1}^{n} \sigma_{ij} b_j = \sum_{j=1}^{n} \left(\frac{C_{ji}}{\det C} \right) b_j,$$

where $C \in \boldsymbol{C}$ such that $x = C^{-1} b$, C_{ji} are cofactors of elements of C. Thus, the vector b components are weights of C^{-1} columns that we use to obtain components of x.

These considerations give us another condition on a right-hand side vector of the interval linear systems of equations that we construct: all of the components of the right-hand side vector must be equal to each other. So, if we use the vector $e =$

$(1,\ldots,1)^\top$ as a right-hand side vector, then components of x are equally dependent on all of cofactors of C, i.e., the components are equally dependent on every element of the matrix.

It is not so if we choose a vector with non-equal components as a right-hand side vector. For example, if the unit vector e_i of a standard basis in $\mathbb{R}^n$ is chosen for this purpose, then

$$x = \frac{1}{\det C}\Bigg(C_{1i},\ldots,C_{ni}\Bigg)$$

for $x \in \Xi(\boldsymbol{C}, e_i)$, $x = L_{e_i}(C)$. At this case, elements of the i-th column of C have no impact on the values $C_{1i},\ldots,\ C_{ni}$. As a result, components of x are not equally dependent on every element of the matrix C.

The elements of C are specified by elements of input matrices. We proceed from the presupposition that all of input matrices elements should be equally accounted in the process of recognition. So we consider the systems of interval linear equations of the form $\boldsymbol{C}^{(k)}x = e$.

3.4 Estimation of ILSE solution sets

For the matrix $\boldsymbol{C}^{(k)}$ that is obtained using (1) and subjected to transformations (4) and (5), the comparison matrix $\langle\boldsymbol{C}^{(k)}\rangle$ is an M-matrix, i.e., $\boldsymbol{C}^{(k)}$ is an H-matrix. If we use the interval Gauss-Seidel method to compute $\boldsymbol{X}^{(k)}$, then every sufficiently large box will be improved in the process of its iteration [**?**]. Thus, taking sufficiently large box as an initial approximation, we may compute the enclosures $\boldsymbol{X}^{(k)}$ of the sets $\Xi^{(k)}$. For this purpose, we take the box $\boldsymbol{W} = ([-w,w],\ldots,[-w,w])^\top \in \mathbb{IR}^n$ which contains all of the solutions sets $\Xi^{(k)}$ for some $w > 0$.

Without any restriction we may assume that all elements of every matrix $C \in \boldsymbol{C}^{(k)}$ are positive. In other case, we may perform the preliminary transformations (4) with $\upsilon_0 > 0$ such that υ_0 is greater than module of every negative element of A and $\mathsf{A}^{(k)}$ for $k = \overline{1,n}$.

As a result of (5), every $C \in \boldsymbol{C}^{(k)}$ is diagonally dominant. For $C \in\ \boldsymbol{C}^{(k)}$, let

$$R_i(C) = c_{ii} - \sum_{j\neq i} c_{ij},\ \ R_*(C) = \min_{1\leq i\leq n} R_i(C).$$

We have [**?**]

$$||C^{-1}||_\infty \leq \frac{1}{R_*(C)}.$$

By (4) and (5), we have $R_*(C) \geq (n-1)\upsilon$, so

$$||C^{-1}||_\infty \leq \frac{1}{(n-1)\upsilon}.$$

If $x = C^{-1}e \in \Xi^{(k)}$, then

$$||x||_\infty \leq ||C^{-1}||_\infty ||e||_\infty,$$

i.e.,

$$||x||_\infty \leq \frac{1}{(n-1)\upsilon}. \tag{6}$$

The value at the right-hand side of (6) allows us to do the choice of the initial approximation $\boldsymbol{W}$ for the interval Gauss-Seidel method. We take $w = 1/((n-1)\upsilon)$ for the box $\boldsymbol{W}$.

As it follows from (6), a large growth of υ may lead to an inappropriate decreasing of values $||x||_\infty$ for $x \in \Xi^{(k)}$. As a result of such a decreasing, we may have a situation when the Lebesgue measures of the enclosures $\boldsymbol{X}^{(k)}$ do not reflect any specificity of pattern matrices. As diagonal dominance becomes excessively large, the values $\mu(\Xi^{(k)})$ may become too small. At this case, comparison of the values $\boldsymbol{X}^{(k)}$ will not give us good recognition. Such decreasing is not admissible due to computational errors of the enclosure methods we may use. For efficient recognition, we need a rather large deviation of the value $\mu(\boldsymbol{X}^{(p)})$ from the other values $\mu(\boldsymbol{X}^{(k)})$. When we perform our computational experiments, we take $\upsilon = 10\,a$, where $a = \max |\mathsf{a}_{ij}^{(k)}|$, $i, j = \overline{1, n}$, $k = \overline{1, N}$.

3.5 Computational complexity

The complexity of the presented algorithm depends on the enclosure method we use and it is equal to $O(N \cdot \mathit{cmpl}(\mathit{Encl}, n))$, where $\mathit{cmpl}(\mathit{Encl}, n)$ is a computational complexity of the algorithm Encl which we use to obtain the enclosures $\boldsymbol{X}^{(k)}$. If $\mathit{cmpl}(\mathit{Encl}, n) = O(n^2)$, then we get an algorithm with the lowest order of complexity among the algorithms that may be designed for solution of the considered problem: since every procedure that processes elements of $n \times n$-matrix has overall complexity that is not less than $O(n^2)$, the computational complexity of any algorithm for solution of the problem is not less than $O(n^2)$.

If we use the interval Gauss-Seidel method, denote it as GS, then computational complexity of the algorithm is equal to $O(N \cdot \mathit{cmpl}(GS, n)) = O(N \cdot N_{GS} \cdot n^2)$, where N_{GS} is a number of the interval Gauss-Seidel iterations we perform. We take $N_{GS} = 20$ for computational experiments.

4 Computational experiments

At the two first computational experiments that we perform, we investigate the heuristic efficiency in application to recognition of digits images. The images we use are presented in black-and-white and grayscale modes. They have resolution 20×20, 35×35, 50×50 and 100×100 pixels. We have done our experiments for Times New Roman, Arial and Courier New fonts and for the font that is presented in the Figure 1. As the experiments show, the last font is the most difficult for recognition. We use it in the first and in the second experiment we perform.

The element $\mathsf{a}_{ij}^{(k)}$ of a pattern image matrix $\mathsf{A}^{(k)}$ may take one of the two values:

$$\mathsf{a}_{ij}^{(k)} = \begin{cases} c_1, & \text{if pixel at } ij \text{ position is white,} \\ c_2, & \text{if pixel at } ij \text{ position is black.} \end{cases}$$

If the images are black-and-white, then $c_1 = 1$ and $c_2 = 0$. If they are grayscale, then c_1 and c_2 may take any two values in the range of 0 to 255.

Figure 1: Pattern images of digits.

Suppose we have black-and-white images. Let us take some $Q \in [0, 100]$ as a level of noise. The value of Q specifies a percent of pixels subjected to noise. We noise the pattern image doing the following modification of its pixels. For every pixel, we generate a random integer value $q \in [0, 100]$ using uniform distribution. If $q \in [0, Q]$, then we invert pixel, else the pixel stays the same. If we noise image in such a manner for $Q = 0$, then we have the initial image as a result of noising. If $Q = 50$, then, on average, 50% of the image pixels are inverted. If $Q = 100\%$, then every pixel is inverted in the course of noising. For every ordered pair of pattern matrices $(\mathsf{A}^{(i)}, \mathsf{A}^{(j)})$, we get A noising $\mathsf{A}^{(i)}$ and then try to recognize it. We perform 100 trials for every such a pair of the matrices during the experiment. The *recognition efficiency* P is a percent of right choices that we make:

$$P = \frac{\text{the number of right choices}}{\text{the number of trials}} \times 100.$$

If the images are presented in a grayscale mode, then we change the value of pixel in predefined interval instead of inverting it.

For the presented algorithm, results of the computational experiments are shown in the Tables 1 and 2 and are illustrated by the graphs in the Figure 2. The Table 1 shows the results of the computations for $Q \in [31, 50]$. The Table 2 shows the results of the computations for $Q = 45\%$ and $n = 50$. If $Q \leq 30\%$, then the recognition efficiency is not less than 99.9%. Graphs in the Figure 2 show that the recognition efficiency grows as the images resolution grows.

Table 1: The recognition efficiency P for the test images ($Q \in [31, 50]$).

$n\backslash Q$	31	32	33	34	35	36	37	38	39	40
20	99.6	99.6	99.4	99.2	99	98.6	98	97.3	96	95
35	99.9	100	100	99.8	99.8	99.7	99.5	99.4	99	98.3
50	100	100	100	100	100	100	99.9	99.9	99.8	99.7
100	100	100	100	100	100	100	100	100	100	100
$n\backslash Q$	41	42	43	44	45	46	47	48	49	50
20	93.3	90.7	88.1	84.1	79.9	75.3	69.2	63.1	57	49.7
35	97.5	96.6	94.5	91	87.2	82.2	76.2	67.9	60.1	49.4
50	99.5	99.3	98.4	97.3	95.2	90.6	84	74.6	62.7	49.9
100	100	100	100	99.9	99.7	99.1	97.2	99.1	75.1	50.7

The recognition efficiency of the presented algorithm is not any worse than the one for well-known algorithms. Thus, for Kora, R-method, TEMP and CORAL algorithms, $P = 80\%$ when $Q = 43\%$, the algorithm based on monochrome morphology has $P = 80\%$ when $Q = 45\%$ [**?**, **?**]. For neural networks algorithms, the recognition efficiency P is not greater than 90% when $Q \geq 46\%$ for the similar experiment with letter images [**?**] that we also have performed. If we use an appropriate resolution, then the presented heuristic gives equal or greater recognition efficiency.

Also, we have compared the recognition efficiency of the heuristic (2) with the one of the other two natural heuristics that may be used for recognition of numerical matrices. The heuristics are

$$p = \arg\min_k \rho_1(\mathsf{A}, \mathsf{A}^{(k)}) \tag{7}$$

Table 2: The results of the recognition tests for the level of noise $Q = 45\%$, $n = 50$. ij-th entry of the table is equal to the recognition efficiency for the experiment with the ordered pair $(\mathsf{A}^{(i)}, \mathsf{A}^{(j)})$, where $\mathsf{A}^{(i)}$ is the numerical matrix that is corresponds to digit i $(i, j = \overline{0, 9})$.

$i \setminus j$	0	1	2	3	4	5	6	7	8	9
0	—	97	92	100	91	86	96	94	85	97
1	98	—	98	98	98	98	94	92	98	99
2	94	94	—	94	99	97	100	96	98	89
3	99	96	96	—	97	96	97	91	100	95
4	100	93	99	96	—	92	99	99	93	95
5	94	97	95	97	93	—	94	99	90	96
6	96	93	100	95	97	94	—	95	99	98
7	99	99	99	91	98	96	96	—	99	99
8	76	92	93	96	89	84	92	97	—	94
9	96	97	93	89	94	95	98	97	96	–

Figure 2: The growth of recognition efficiency for growing image resolution.

and

$$p = \arg\min_k \rho_2(\mathsf{A}, \mathsf{A}^{(k)}), \tag{8}$$

where $\rho_1 = \|\mathsf{A} - \mathsf{A}^{(k)}\|_1$ and $\rho_2 = \|\mathsf{A} - \mathsf{A}^{(k)}\|_2$. First, consider the heuristic (8).

Let S be a percent of the recognition problems for which the heuristic (8) does not gives the right solution, while we can obtain it using the heuristic (2). It is may be seen that if the values c_1 and c_2 become closer to each other and the radius Δ grows, then the value of S grows.

In the experiments for the evaluation of S, we use the same set of the pattern images as we have used previously but the images are presented now in a grayscale mode with the values $c_1 = 110$ and $c_2 = 120$. The Table 3 shows the values of S, which we have obtain for different values of Δ. The Table 4 shows a growth of S for $c_1 = 119$ and $c_2 = 120$. The percent S has a tendency to growth as the radius Δ grows, if only the value of $|c_1 - c_2|$ is rather small comparing to Δ.

Note that the algorithm we have presented is not applicable for recognition of

Table 3: The percent S for the level of noise $Q = 44\%$, $c_1 = 110$, $c_2 = 120$.

Δ	10	25	50	75	100
$S, \%$	0	5.4	7.4	16.2	23.5
$P, \%$	100	99.93	99.79	99.72	99.81

Table 4: The percent S for the level of noise $Q = 44\%$, $c_1 = 119$, $c_2 = 120$.

Δ	10	25	50	75	100
$S, \%$	22.8	37.5	47.3	46.4	46.4
$P, \%$	99.71	99.6	99.8	99.72	99.82

raster images subjected to spatial displacements, i.e., rotations or shifts.

In the third computational experiment we compare the recognition efficiency of the heuristic (2) and the one of the heuristic (7) for the recognition problems that we construct in the following way.

Let A be some numerical matrix. Let us construct two matrices, both of which are obtained from A. We shall consider these matrices as pattern matrices denoting them as $\mathsf{A}^{(1)}$ and $\mathsf{A}^{(2)}$. As we getting $\mathsf{A}^{(1)}$ from A, we change a rather small number of the matrix A elements but the changes are large. As we getting $\mathsf{A}^{(2)}$, we change a majority of the matrix A elements but the changes are relatively small.

Denote by MP the percent of elements of A that we change obtaining $\mathsf{A}^{(2)}$. $100 - MP$ is the percent of elements of A that we change obtaining $\mathsf{A}^{(1)}$. Let $\mathsf{A}^{(1)} := \mathsf{A}$ and $\mathsf{A}^{(2)} := \mathsf{A}$. Construct the matrices $\mathsf{A}_M^{(1)}$ and $\mathsf{A}_M^{(2)}$ using randomness in such a manner that MP percent of their elements are 0's and $100 - MP$ percent of them are 1's. Fill these matrices independently. Then, let η_1, η_2 be random variables that take the values -1 and 1 with equal probabilities, and let ξ_1, ξ_2 be independent random variables which have uniform distribution on the interval $[0, 1]$. For matrices $\mathsf{A}^{(1)}$ and $\mathsf{A}^{(2)}$, let Δ_1 and Δ_2, respectively, be some predefined radiuses of the intervals in which elements of the matrices are changing. Let $\Delta_1 > \Delta_2$.

Using elements of $\mathsf{A}_M^{(1)}$, we change $100 - MP$ percent of elements of $\mathsf{A}^{(1)}$ in accordance with elements $(\mathsf{A}_M^{(1)})_{ij}$. If $(\mathsf{A}_M^{(1)})_{ij} = 0$, then we do not change the value of $\mathsf{a}_{ij}^{(1)}$, else

$$\mathsf{a}_{ij}^{(1)} := \mathsf{a}_{ij}^{(1)} + \lfloor \eta_1 \xi_1 \Delta_1 \rfloor ,$$

where $\lfloor \cdot \rfloor$ denotes the operation of taking an integer part of a real number. And let us change MP percent of $\mathsf{A}^{(2)}$ elements. Positions of these elements are specified by elements of $(\mathsf{A}_M^{(2)})_{ij}$. If $(\mathsf{A}_M^{(2)})_{ij} = 0$, then

$$\mathsf{a}_{ij}^{(2)} := \mathsf{a}_{ij}^{(2)} + \lfloor \eta_2 \xi_2 \Delta_2 \rfloor ,$$

else the value of $\mathsf{a}_{ij}^{(2)}$ stays the same.

Assuming now that the matrix A is the matrix to be recognized and that $\mathsf{A}^{(1)}$ and $\mathsf{A}^{(2)}$ are the pattern matrices, we try to recognize the matrix A. For large values of

MP, since the majority of the matrix A elements are equal to corresponding elements of $\mathsf{A}^{(1)}$, it is natural to suggest that the correct recognition is the recognition that gives $\mathsf{A}^{(1)}$ as a result. We can construct such matrices A, $\mathsf{A}^{(1)}$ and $\mathsf{A}^{(2)}$ that

$$\rho_1(\mathsf{A}, \mathsf{A}^{(1)}) > \rho_1(\mathsf{A}, \mathsf{A}^{(2)}), \tag{9}$$

when

$$\delta_\Xi(\mathsf{A}, \mathsf{A}^{(1)}) < \delta_\Xi(\mathsf{A}, \mathsf{A}^{(2)}). \tag{10}$$

The following matrices give us such an example:

$$\mathsf{A} = \begin{pmatrix} 0 & 0 & 4 & 1 \\ 1 & 3 & 1 & 0 \\ 1 & 2 & 0 & 2 \\ 0 & 4 & 0 & 1 \end{pmatrix}, \; \mathsf{A}^{(1)} = \begin{pmatrix} 0 & 0 & 4 & 57 \\ 1 & 3 & 1 & 0 \\ 1 & 2 & 0 & 2 \\ 0 & 4 & 0 & 47 \end{pmatrix}, \; \mathsf{A}^{(2)} = \begin{pmatrix} -3 & 0 & 14 & 1 \\ 5 & -3 & -3 & 8 \\ 8 & 0 & 6 & -1 \\ -2 & -6 & 0 & -7 \end{pmatrix}.$$

Here the values of elements of A are randomly and uniformly chosen from the interval $[0, 5]$, $MP = 90$, $\Delta_1 = 60$, $\Delta_2 = 10$, $\upsilon = 10$. For these matrices, we have

$$\rho_1(\mathsf{A}, \mathsf{A}^{(1)}) = 102 > 73 = \rho_1(\mathsf{A}, \mathsf{A}^{(2)}),$$

while

$$\delta_\Xi(\mathsf{A}, \mathsf{A}^{(1)}) \approx 0.312 < 2.359 \approx \delta_\Xi(\mathsf{A}, \mathsf{A}^{(2)}).$$

For an experiment, let A be a 10×10-matrix whith elements that are randomly and uniformly chosen integers from the interval $[110, 120]$. Taking $MP = 90\%$ and taking the same as above values of Δ_1 and Δ_2, we generate such matrices $\mathsf{A}^{(1)}$ and $\mathsf{A}^{(2)}$ that (9) holds. For more than 95% of the recognition trials, we have (10) and so we have $\mathsf{A}^{(1)}$ as a result of recognition. It is appropriate to accept these results as correct since 90% of corresponding elements of A and $\mathsf{A}^{(1)}$ are equal to each other, while 90% of corresponding elements of A and $\mathsf{A}^{(2)}$ are different.

As the percent MP decrease, the results of recognition become worse for the same values of Δ_1 and Δ_2. The results for some other values of MP and Δ_1 are shown in the Table 5. These results demonstrate that, using the presented heuristic, we take into account aggregate variation of matrix elements rather than large variations of a small number of the elements.

5 Conclusions

We present an interval approach to recognition of numerical matrices. The heuristic we use is a minimization of the proposed measure of closeness between two matrices. For computing the measure, we construct the interval linear system associated with the matrices. We take the Lebesgue measure of its united solution set as the measure of closeness. Using the heuristic, we construct a recognition algorithm. The recognition algorithm has quadratic computational complexity. The computational experiments show that the presented heuristic provides good results for recognition of raster images and that its recognition efficiency grows as a resolution of the images grows. Also experiments show that the heuristic takes into account some aggregate variation of matrix elements rather than large variations of a small number of the elements.

Acknowledgements

The author is grateful to S.P. Shary for his attention and comments.

Table 5: The recognition efficiency P for different values of MP and Δ_2 for the experiment with such A, $\mathsf{A}^{(1)}$ and $\mathsf{A}^{(2)}$ that $\rho_1(\mathsf{A}, \mathsf{A}^{(1)}) > \rho_1(\mathsf{A}, \mathsf{A}^{(2)})$ ($\Delta_1 = 10$ for all of the instances).

Δ_2	$MP, \%$	Δ_1	$P, \%$	$MP, \%$	Δ_1	$P, \%$
10	85	35	88	80	25	76
10	85	40	79.8	80	30	67
10	85	45	71.8	80	35	52.8
Δ_2	$MP, \%$	Δ_1	$P, \%$	$MP, \%$	Δ_1	$P, \%$
10	75	20	63	70	15	53
10	75	25	47.1	70	20	36
10	75	30	29.4	70	25	16

References

[1] Oettli, W. On the solution set of a linear system with inaccurate coefficients // SIAM Journal on Numerical Analysis. 1965. Vol. 2, 1. P. 115–118.

[2] Shary, S.P. Finite-dimensional interval analysis. [Electronical resource] URL: http://www.nsc.ru/interval/Library/InteBooks/SharyBook.pdf (the date of reference is 18.09.2013)

[3] Neumaier, A. Interval Methods for Systems of Equations. Cambridge: Cambridge University Press, 1990. P. 140.

[4] Ahlberg, J.H., Nilson, E.N. Convergence properties of the spline fit // J. SIAM. 1963. V. 11, 1. P. 95–104.

[5] Kirnos, E.A. A comparative analysis of morphological methods of image interpretations: An abstract of dissertation of the candidate of physics and mathematics (the speciality 05.13.18, mathematical modeling, numerical methods and complexes of programs). Moscow, 2004. RGB OD, 61:05-1/118.

[6] Kirnos, E.A., Pyt'ev, Yu.P., Djukova, E.V. Training the Kora-Type Algorithms // Pattern Recognition and Image Analysis. 2002. Vol. 12. 1. P. 19–24.

[7] Demuth, H., Beale, M. Neural Network Toolbox for Use with MATLAB. Users Guide. Version 4 // [Electronical resource] URL: http://cs.mipt.ru/docs/comp/eng/develop/software/matlab/nnet/main.pdf (the date of reference is 18.09.2013)