Journal of Advances in Mathematics and Computer Science

37(1): 28-39, 2022; Article no.JAMCS.83290

ISSN: 2456-9968

(Past name: British Journal of Mathematics & Computer Science, Past ISSN: 2231-0851)

Stepwise Inversion of Large Matrices with the Gauss-Jordan Vector Transformation

Leon Bobrowski [a] **and Cezary Boldak** [a*]

[a]*Faculty of Computer Science, Bialystok University of Technology, Bialystok, Poland.*

Authors' contributions

This work was carried out in collaboration between both authors. Both authors read and approved the final manuscript.

Article Information

DOI: 10.9734/JAMCS/2022/v37i130429

Open Peer Review History:
This journal follows the Advanced Open Peer Review policy. Identity of the Reviewers, Editor(s) and additional Reviewers, peer review comments, different versions of the manuscript, comments of the editors, etc are available here: https://www.sdiarticle5.com/review-history/83290

Received: 01 January 2022
Accepted: 24 January 2022
Published: 25 January 2022

Original Research Article

Abstract

Data mining methods often include matrix inversion tasks. Big data mining requires fast and accurate inversion of high-dimensional arrays. The presented work describes the concept of stepwise matrix inversion using the basis exchange algorithm based on the Gauss-Jordan vector transformation. The article also includes a demonstrative numerical example.

Keywords: Large matrices; high-dimensional arrays; Gauss-Jordan transformation; linear programming.

2010 Mathematics Subject Classification: 65F05, 65K10, 90C05.

1 Introduction

The complexity of large matrices inversion is currently an important challenge in many practical problems in data exploration methods [1, 2, 3].

Various numerical techniques aimed at efficient and precise inverting of large matrices are currently being developed [4]. Basis exchange algorithms are also examined in this context [5, 6, 7]. They are based on the Gauss-Jordan transformation and as a result are similar to the *Simplex* algorithm used in linear programming [8].

*Corresponding author: E-mail: c.boldak@pb.edu.pl;

A new version of the basis exchange algorithm specified for matrices inversion with a Gauss-Jordan transformation is proposed and examined in the presented paper. A multistage process of reversing matrices with gradually increased computational complexity is described.

2 Sequence of Inverted Matrices

Consider a square, non-singular matrix $\mathbf{X}$ of the dimension $n \times n$:

$$\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^T. \tag{2.1}$$

The rows of the non-singular matrix $\mathbf{X}$ (2.1) are assumed to be composed of n linearly independent feature vectors $\mathbf{x}_j = [x_{j,1}, \ldots, x_{j,n}]^T$ belonging to a given n-dimensional feature space $\mathbf{F}[n]$ ($\mathbf{x}_j \in \mathbf{F}[n]$). In this case, there exists the inverse matrix $\mathbf{X}^{-1}$ ($\mathbf{X}\mathbf{X}^{-1} = \mathbf{I}$, where $\mathbf{I} = [\mathbf{e}_1, \ldots, \mathbf{e}_n]$ is the unit matrix):

$$\mathbf{X}^{-1} = [\mathbf{r}_1, \ldots, \mathbf{r}_n]. \tag{2.2}$$

The inverse matrix $\mathbf{X}^{-1}$ is composed of n columns $\mathbf{r}_i = [r_{i,1}, \ldots, r_{i,n}]^T$ which fulfill the below equations:

$$\begin{gathered}(\forall j \in \{1, \ldots, n\})\ \mathbf{x}_j^T \mathbf{r}_j = 1 \\ \text{and } (\forall i \in \{1, \ldots, n; i \neq j\})\ \mathbf{x}_j^T \mathbf{r}_i = 0.\end{gathered} \tag{2.3}$$

New methods for reversing the matrix are proposed and implemented to increase the efficiency of calculations and the size of the inverted matrices [4, 9]. In this context, we propose using the basis exchange algorithms for matrices inversion.

Let us consider the below family of the non-singular matrices (*bases*) $\mathbf{B}_k$:

$$\begin{aligned}&(\forall k \in \{1, \ldots, n\}) \\ &\qquad \mathbf{B}_k = [\mathbf{x}_{j(1)}, \ldots, \mathbf{x}_{j(k)}, \mathbf{e}_{k+1}, \ldots, \mathbf{e}_n]^T,\end{aligned} \tag{2.4}$$

where $j(k)$ is the index of the feature vector $\mathbf{x}_{j(k)}$ inserted to the basis $\mathbf{B}_{k-1}$ (2.4) during the k-th stage.

The k-th basis $\mathbf{B}_k$ (2.4) is composed of k feature vectors $\mathbf{x}_{j(l)}$ ($l = 1, \ldots, k$) and $n - k$ unit vectors $\mathbf{e}_l$ ($l = k + 1, \ldots, n$).

The selected feature vectors $\mathbf{x}_{j(i)}$ ($i = 1, \ldots, k$) and $n - k$ unit vectors $\mathbf{e}_i$ ($i = k + 1, \ldots, n$) constituting the rows of the nonsingular matrix $\mathbf{B}_k$ (2.4) form the *base* of the n-dimensional feature space $\mathbf{F}[n]$. The bases $\mathbf{B}_k$ (2.4) are ranked in the below sequence:

$$\mathbf{B}_0, \mathbf{B}_1, \ldots, \mathbf{B}_{n-1}, \mathbf{B}_n. \tag{2.5}$$

The first matrix $\mathbf{B}_0$ in this sequence is equal to the unit matrix $\mathbf{I}$ ($\mathbf{B}_0 = I$), and the k-th base matrix $\mathbf{B}_k$ is composed of k feature vectors $\mathbf{x}_{j(k)}$ ($j(k) \in \{1, \ldots, n\}$) and $n - k$ unit vectors $\mathbf{e}_i$ ($i = k + 1, \ldots, n$). The last matrix $\mathbf{B}_n = [\mathbf{x}_{j(1)}, \ldots, \mathbf{x}_{j(n)}]^T$ in the sequence (2.5) is composed of n feature vectors $\mathbf{x}_j$ ($j = 1, \ldots, n$).

The Gauss-Jordan transformation makes it possible to generate the following sequence of the inverted matrices $\mathbf{B}_k^{-1}$ obtained from the non-singular matrices $\mathbf{B}_k$ (2.4) [6]:

$$\mathbf{B}_0^{-1}, \mathbf{B}_1^{-1}, \ldots, \mathbf{B}_{n-1}^{-1}, \mathbf{B}_n^{-1}. \tag{2.6}$$

The k-th inverted matrix $\mathbf{B}_k^{-1}$ can be represented in the following manner:

$$(\forall k \in \{1, \ldots, n\})\ \mathbf{B}_k^{-1} = [\mathbf{r}_1(k), \ldots, \mathbf{r}_n(k)], \tag{2.7}$$

where the symbol $\mathbf{r}_i(k)$ stands for the i-th column of the k-th inverse matrix $\mathbf{B}_k^{-1}$.

3 Gauss-Jordan Vector Transformation

The sequence of the inverse matrices $\mathbf{B}_k^{-1}$ (2.6) results from the multistage process of the matrices $\mathbf{B}_k$ (2.5) transformations. During the k-th stage the matrix $\mathbf{B}_k$ (2.4) is transformed into the basis $\mathbf{B}_{k+1}$:

$$(\forall k \in \{1,\ldots,n-1\})\ \mathbf{B}_k \to \mathbf{B}_{k+1}. \tag{3.1}$$

The basis $\mathbf{B}_{k+1}$ is obtained as the result of replacing the $(k+1)$-th unit vector $\mathbf{e}_{k+1}$ in the matrix $\mathbf{B}_k$ (2.4) by the $j(k+1)$-th feature vector $\mathbf{x}_{j(k+1)}$. In accordance with the Gauss-Jordan vector transformation the replacement of the unit vector $\mathbf{e}_k$ by the feature vector $\mathbf{x}_{j(k+1)}$ (3.1) causes the following modifications of the columns $\mathbf{r}_i(k)$ of the inverse matrix $\mathbf{B}_k^{-1}$ (2.7) [7]:

$$\begin{aligned}
&(\forall k \in \{0,\ldots,n-1\})\\
&\qquad \mathbf{r}_{k+1}(k+1) = (1/\mathbf{r}_{k+1}(k)^T\mathbf{x}_{j(k+1)})\mathbf{r}_{k+1}(k)\\
&\text{and } (\forall i \neq k)\\
&\mathbf{r}_i(k+1) = \mathbf{r}_i(k) - (\mathbf{r}_i(k)^T\mathbf{x}_{j(k+1)})\mathbf{r}_{k+1}(k+1) =\\
&= \mathbf{r}_i(k) - (\mathbf{r}_i(k)^T\mathbf{x}_{j(k+1)}/\mathbf{r}_{k+1}(k)^T\mathbf{x}_{j(k+1)})\mathbf{r}_{k+1}(k),
\end{aligned} \tag{3.2}$$

where $j(k+1)$ is the index of the feature vector $\mathbf{x}_{j(k+1)}$ inserted to the basis $\mathbf{B}_k$ (2.4).

Remark 3.1. The Gauss-Jordan transformation (3.2) linked to the replacement of the unit vector $\mathbf{e}_{k+1}$ by the feature vector $\mathbf{x}_{j(k+1)}$ cannot be performed when the below condition is met:

$$\mathbf{r}_{k+1}(k)^T\mathbf{x}_{j(k+1)} = 0. \tag{3.3}$$

The condition (3.3) would result in the division by zero in the equation (3.2).

Remark 3.2. The k-th column of the inverse matrix $\mathbf{B}_k^{-1} = [\mathbf{r}_1(k),\ldots,\mathbf{r}_n(k)]$ (2.6) is the vector $\mathbf{r}_k(k) = [r_{k,1}(k),\ldots,r_{k,n}(k)]^T$ with the last $n-k$ components $r_{k,l}(k)$ equal to zero:

$$\begin{aligned}
&(\forall k \in \{0,1,\ldots,n-1\})\ (\forall l \in \{k+1,\ldots,n\})\\
&\qquad\qquad\qquad r_{k,l}(k) = 0.
\end{aligned} \tag{3.4}$$

The above property results directly from the matrix inverse equations (2.3). The k-th column $\mathbf{r}_k(k)$ of the inverse matrix $\mathbf{B}_k^{-1}$ (2.7) is perpendicular to all vectors $\mathbf{x}_{j(l)}$ belonging to to the basis $\mathbf{B}_k$ (2.4) except the vector $\mathbf{x}_{j(k)}$ [6]:

$$\begin{aligned}
&(\forall k \in \{1,\ldots,n-1\})\ (\forall l \in \{1,\ldots,k-1\})\\
&\qquad\qquad\qquad \mathbf{r}_k(k)^T\mathbf{x}_{j(l)} = 0.
\end{aligned}$$

Given the conditions (3.4), the above equations can be represented as follows:

$$\begin{aligned}
&(\forall k \in \{1,\ldots,n-1\})\ (\forall l \in \{1,\ldots,k-1\})\\
&\qquad\qquad\qquad \mathbf{r}[k]^T\mathbf{x}_{j(l)}[k] = 0,
\end{aligned} \tag{3.5}$$

where the symbol $\mathbf{r}_k[k]$ means the k-th column $\mathbf{r}_k(k) = [r_{k,1}(k),\ldots,r_{k,n}(k)]^T$ of the inverse matrix $\mathbf{B}_k^{-1}$ (2.7) after reducing the last $n-k$ components $r_{k,i}(k)$:

$$\begin{aligned}
&(\forall k \in \{1,\ldots,n-1\})\\
&\qquad\qquad \mathbf{r}_k[k] = [r_{k,1}(k),\ldots,r_{k,k}(k)]^T.
\end{aligned}$$

Similarly, the symbol $\mathbf{x}_j[k] = [x_{j,1}, \ldots, x_{j,k}]^T$ means the reduced vector obtained from the feature vector $\mathbf{x}_j = [x_{j,1}, \ldots, x_{j,n}]^T$ after reducing the last $n - k$ components $x_{j,i}$:

$$(\forall j \in \{1, \ldots, n\}) \quad \mathbf{x}_j[k] = [x_{j,1}, \ldots, x_{j,k}]^T. \tag{3.6}$$

During step k we try to replace the $(k+1)$-th unit vector $\mathbf{e}_{k+1}$ in the matrix $\mathbf{B}_k$ (2.4) by the $j(k+1)$-th feature vector $\mathbf{x}_{j(k+1)}$. In accordance with the vectoral Gauss-Jordan transformation (3.2) such replacement is impossible if the condition (3.3) appears.

Lemma 3.3. If the reduced vector $\mathbf{x}_{j(k+1)}[k]$ (3.6) during the k-th step is a linear combination of the basis *reduced* vectors $\mathbf{x}_{j(i)}[k]$ with $i \leq k$, then the condition $\mathbf{r}_k(k)^T\mathbf{x}_{j(k+1)} = 0$ (3.3) appears, when:

$$\mathbf{x}_{j(k+1)}[k] = \alpha_{j(k+1),1}\mathbf{x}_{j(1)}[k] + \ldots + \alpha_{j(k+1),k}\mathbf{x}_{j(k)}[k], \tag{3.7}$$

where $(\forall i \in \{1, \ldots, k\})$ $\alpha_{j(k+1),i} \in R^1$.

The lemma can be directly proved by using the equations (3.5).

Remark 3.4. The collinearity condition (3.7) which appears during the k-th step may disappear during the $(k+1)$-th step.

The disappearance of the collinearity condition (3.7) may result from taking into account the additional component $x_{j(k+1),k+1}$ of the reduced feature vector $\mathbf{x}_{j(k+1)}[k+1]$.

The condition (3.3) provides a possibility to block the insertion of almost linearly dependent vectors $\mathbf{x}_{j(k+1)}[k+1]$ (3.7) into the nonsingular matrix $\mathbf{B}_k$ (2.4).

4 Matrix Inversion through Basis Exchange

Simplex algorithm from linear programming is defined by the *exit criterion*, *entry criterion* and the *stop criterion* [8]. Any basis exchange algorithm can also be defined by such criteria. The basis exchange algorithm oriented to matrices inversion is defined by the following criteria:

1. *exit criterion*
   The unit vector $\mathbf{e}_{k+1}$ leaves the basis $\mathbf{B}_k$ (2.4) during the step k.
2. *entry criterion*
   The feature vector $\mathbf{x}_{j(k+1)}$ which enters the basis $\mathbf{B}_k$ (2.4) during the step k has the smallest index $j(k+1)$ among all such vectors $\mathbf{x}_j$ which fulfill the below collinearity condition (3.3):
   $$|\mathbf{r}_{k+1}(k)^T\mathbf{x}_{j(k+1)}| \geq \epsilon \tag{4.1}$$
   where ϵ is a small, positive parameter ($\epsilon > 0$).
3. *stop criterion*
   The algorithm is stopped during the step k ($k \leq n$) if no vector $\mathbf{x}_{j(k+1)}$ ($\mathbf{x}_{j(k+1)} \notin \mathbf{B}_k$) can be inserted into the basis $\mathbf{B}_k$ (2.4) in accordance with the condition (4.1). The choice of the value of the parameter ϵ in the *entry condition* (3.5) gives the possibility to control the level of *ill-conditioning* of the inverted matrices $\mathbf{B}_k$ (2.7) Fulfillment of the *stop criterion* during the k-th step ($k < n$) means that the matrix $\mathbf{X}$ (2.1) is not reversible at the acceptable ϵ-level of ill-conditioning (4.1).

Properties of the basis exchange algorithm defined by the above criteria can be analyzed by using the *convex and piecewise linear* (*CPL*) criterion functions.

5 The Inversion Criterion Function

The convex and piecewise linear (*CPL*) criterion functions are used, among others, for the purpose of examining the linear separability of data sets or for extracting collinear patterns [7]. Similar *CPL* criterion functions can also be useful in examining the nonsingularity of high-dimensional matrices $\mathbf{X}$ (2.1). The following penalty functions $\varphi_j(\mathbf{w})$ can be used for this purpose [5]:

$$(\forall j \in \{1, \ldots, n\})$$
$$\varphi(\mathbf{w}) = |1 - \mathbf{x}_j^T\mathbf{w}| = \begin{matrix} 1 - \mathbf{x}_j^T\mathbf{w} \;\; \text{if } \mathbf{x}_j^T\mathbf{w} \leq 1 \\ \\ \mathbf{x}_j^T\mathbf{w} - 1 \;\; \text{if } \mathbf{x}_j^T\mathbf{w} > 1 \end{matrix} \tag{5.1}$$

where $\mathbf{w} = [w_1, \ldots, w_n]^T$ is the parameter (*weight*) vector ($w \in R^n$).

The *inversion criterion function* $\mathbf{\Phi}_{inv}(\mathbf{w})$ is the sum of the penalty functions $\varphi_j(\mathbf{w})$:

$$\mathbf{\Phi}_{\mathbf{inv}}(\mathbf{w}) = \sum_{j=1,\ldots,n} \varphi_j(\mathbf{w}). \tag{5.2}$$

Two types of the dual hyperplanes h_j^1 and h_i^0 in the n-dimensional parameter space R^n are introduced in order to explore the properties of the inversion criterion function $\mathbf{\Phi}_{inv}(\mathbf{w})$ (5.2) [7]. Each of n feature vectors $\mathbf{x}_j$ (2.1) defines the below dual hyperplane h_j^1:

$$(\forall j \in \{1, \ldots, n\}) \; h_j^1 = \{\mathbf{w} : \mathbf{x}_j^T\mathbf{w} = 1\}. \tag{5.3}$$

Similarly, each of n unit vectors $\mathbf{e}_i = [0, \ldots, 1, \ldots, 0]^T$ defines the hyperplane h_i^0:

$$(\forall i \in \{1, \ldots, n\}) \; h_i^0 = \{\mathbf{w} : \mathbf{e}_i^T\mathbf{w} = 0\} = \{\mathbf{w} : w_i = 0\}, \tag{5.4}$$

where $\mathbf{w} = [w_1, \ldots, w_n]^T \in R^n$.

Let us consider the k-th subset $\mathbf{S}_k$ of n linearly independent feature vectors $\mathbf{x}_j$ (2.1) and unit vectors $\mathbf{e}_i$:

$$\mathbf{S}_k = \{\mathbf{x}_j : j \in J_k\} \cup \{\mathbf{e}_i : i \in I_k\}. \tag{5.5}$$

The set $\mathbf{S}_k$ is composed of k feature vectors $\mathbf{x}_j$ ($j \in J_k$) and $n - k$ unit vectors $\mathbf{e}_i$ ($i \in I_k$).

The vertex $\mathbf{w}_k$ of the rank r_k is defined as the intersection point of the r_k hyperplanes h_j^1 (5.3) where $j \in J_k$, and the $n - r_k$ hyperplanes h_i^0 (5.4) which are determined by the unit vectors $\mathbf{e}_i$ ($i \in I_k$) from the subset $\mathbf{S}_k$ (5.5).

The below linear equations are fulfilled in the vertex $\mathbf{w}_k$ of the rank r_k:

$$(\forall j \in J_k) \;\; \mathbf{w}_k^T\mathbf{x}_j = 1 \tag{5.6}$$

and

$$(\forall i \in I_k) \;\; \mathbf{w}_k^T\mathbf{e}_i = 0. \tag{5.7}$$

The equations (5.6) and (5.7) can also be represented by using the matrix $\mathbf{B}_k$ (2.4):

$$\mathbf{B}_k\mathbf{w}_k = \mathbf{1'} = [1, , 1, 0, , 0]^T$$

and (3.7)

$$\mathbf{w}_k = \mathbf{B}_k^{-1}\mathbf{1'} = \mathbf{r}_1(k) + \ldots + \mathbf{r}_k(k). \tag{5.8}$$

Remark 5.1. The vertex $\mathbf{w}_k$ (5.8) of the rank r_k can be represented as the below vector $\mathbf{w}_k = [w_{k,1}, \ldots, w_{k,n}]^T$ with the last $n - r_k$ components $w_{k,i}$ equal to zero:

$$\mathbf{w}_k = [w_{k,1}, \ldots, w_{k,r_k}, 0, \ldots, 0]^T. \tag{5.9}$$

The components $w_{k,i}$ of the vertex $\mathbf{w}_k$ (5.8) equal to zero are linked to the $n - r_k$ unit vectors $\mathbf{e}_i$ $(i \in I_k)$ (5.7) in the basis $\mathbf{B}_k$ (2.4).

The inversion criterion function $\mathbf{\Phi}_{inv}(\mathbf{w})$ (5.2) is convex and piecewise linear (*CPL*) and their minimum can be located in one of the vertices $\mathbf{w}_k$ (5.9) [7].

$$(\exists \mathbf{w}_k^*)(\forall \mathbf{w})\mathbf{\Phi}_{inv}(\mathbf{w}) \geq \mathbf{\Phi}_{inv}(\mathbf{w}_k^*) = \mathbf{\Phi}_{inv}^* \geq 0. \tag{5.10}$$

The basis exchange algorithms allow efficient finding the optimal vertex $\mathbf{w}_k^*$ constituting the minimal value $\mathbf{\Phi}_{inv}(\mathbf{w}_k^*)$ (5.10) even in the case of large matrices $\mathbf{X}$ (2.1) [7].

Remark 5.2. The minimal value $\mathbf{\Phi}_{inv}(\mathbf{w}_k^*)$ of the *CPL* criterion function
$\mathbf{\Phi}_{inv}(\mathbf{w})$ (5.2) is equal to zero if and only if each of n dual hyperplanes h_j^1 (5.3) passes through the optimal vertex $\mathbf{w}_k^*$ (5.10):

$$(\mathbf{\Phi}_{inv}(\mathbf{w}_k^*) = 0) \Leftrightarrow ((\forall j \in \{1, \ldots, n\})(\mathbf{w}_k^*)^T \mathbf{x}_j = 1). \tag{5.11}$$

The above property results directly from the definition (5.2) of the penalty functions $\varphi_j(\mathbf{w})$ (5.1). If the relation (5.11) holds, then each of penalty functions $\varphi_j(\mathbf{w})$ (5.1) is equal to zero in the optimal vertex $\mathbf{w}_k^*$.

Remark 5.3. The optimal vertex $\mathbf{w}_n^*$ (5.10) of the rank n can be linked to the final basis $\mathbf{B}_n = [\mathbf{x}_{j(1)}, \ldots, \mathbf{x}_{j(n)}]^T$ (2.4) composed of n linearly independent vectors $\mathbf{x}_{j(k)}$ $(k \in \{1, \ldots, n\})$. In this case, the vector $\mathbf{w}_k^*$ (5.10) is the sum of n columns $\mathbf{r}_i(n)$ of the inverse matrix $\mathbf{B}_n^{-1} = [\mathbf{r}_1(n), \ldots, \mathbf{r}_n(n)]$ (2.7):

$$\mathbf{w}_n^* = \mathbf{B}_n^{-1}\mathbf{1} = \mathbf{r}_1(n) + \ldots + \mathbf{r}_n(n). \tag{5.12}$$

Remark 5.4. If all the unit vectors $\mathbf{e}_k$ $(k = 1, \ldots, n)$ in the matrix $\mathbf{I} = [\mathbf{e}_1, \ldots, \mathbf{e}_n]$ have been replaced in accordance with the Gauss-Jordan equations (3.2) by the feature vectors $\mathbf{x}_k$, then the matrix $\mathbf{X}^{-1}$ (2.2) is equal to the inverted basis $\mathbf{B}_n^{-1}$ (2.7) $(\mathbf{X}^{-1} = \mathbf{B}_n^{-1})$.

The above remark specifies the sufficient conditions for the equality $\mathbf{X}^{-1} = \mathbf{B}_n^{-1}$ (2.7). It was assumed in this remark, that the j-th unit vectors $\mathbf{e}_j$ was always replaced by the j-th feature vector $\mathbf{x}_j$ $(j = 1, \ldots, n)$. The collinearity condition (3.3) can cause a situation when the final matrix $\mathbf{B}_n = [\mathbf{x}_{j(1)}, \ldots, \mathbf{x}_{j(n)}]^T$ (2.7) in the sequence (2.5) will be different from the data matrix $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^T$ (2.1). In this case, the equality $\mathbf{X}^{-1} = (\mathbf{B}'_n)^{-1}$ can be reached by changing the order of the rows $\mathbf{x}_{j(i)}^T$ in the final basis $\mathbf{B}_n$ (2.7) and the columns $\mathbf{r}_i(n)$ in the inverted matrix $\mathbf{B}_n^{-1} = [\mathbf{r}_1(n), \ldots, \mathbf{r}_n(n)]$ (2.7).

Changing the order of the rows $\mathbf{x}_{j(i)}^T$ in the matrix $\mathbf{B}_n = [\mathbf{x}_{j(1)}, \ldots, \mathbf{x}_{j(n)}]^T$ should be accompanied by such changing of the columns $\mathbf{r}_i(n)$ order in the inverse matrix $\mathbf{B}_n^{-1}$ (2.7) that the inverse equation $\mathbf{B}_n\mathbf{B}_n^{-1} = \mathbf{I}$ is preserved:

$$\begin{gathered}(\forall i \in \{1, \ldots, n\})\ \ \mathbf{x}_{j(i)}^T \mathbf{r}_i(n) = 1 \\ \text{and } (\forall i' \in \{1, \ldots, n; i' \neq i\})\ \ \mathbf{x}_{j(i)}^T \mathbf{r}_{i'}(n) = 0\end{gathered}$$

Remark 5.5. The equation $\mathbf{B}_n\mathbf{B}_n^{-1} = \mathbf{I}$ is preserved during the replacement of the $j(i)$-th vector $\mathbf{x}_{j(i)}$ in the matrix $\mathbf{B}_n = [\mathbf{x}_{j(1)}, \ldots, \mathbf{x}_{j(n)}]^T$ (2.4) to the i-th row of this matrix if it is accompanied by the replacement of the $j(i)$-th column of the inverse matrix $\mathbf{B}_n^{-1} = [\mathbf{r}_1(n), \ldots, \mathbf{r}_n(n)]$ (2.6) to the i-th column.

The replacement of the rows $\mathbf{x}_{j(i)}^T$ in the matrix $\mathbf{B}_n = [\mathbf{x}_{j(1)}, \ldots, \mathbf{x}_{j(n)}]^T$ (2.4) and the columns $\mathbf{r}_i(n)$ in the inverse matrix $\mathbf{B}_n^{-1} = [\mathbf{r}_1(n), \ldots, \mathbf{r}_n(n)]$ (2.6) in accordance with the *Remark 8* allows to find the inversed matrix $\mathbf{X}^{-1}$ (2.2) on the basis of the matrix $\mathbf{B}_n^{-1}$.

Theorem 5.6. The matrix $\mathbf{X}$ (2.1) is reversible ($\mathbf{X}^{-1}$ exists) if and only if the minimal value $\mathbf{\Phi}_{inv}(\mathbf{w}_k^*)$ (5.10) of the inversion criterion function $\mathbf{\Phi}_{inv}(\mathbf{w})$ (5.2) is equal to zero and the rank r_k (*Definition 1*) of the optimal vertex $\mathbf{w}_k^*$ (5.10) is equal to n ($r_k = n$).

Proof. If the optimal vertex $\mathbf{w}_k^*$ (5.10) has the rank r_k equal to n ($r_k = n$), then the basis $\mathbf{B}_n$ (2.4) linked to this vertex is composed of n vectors $\mathbf{x}_{j(i)}$ (*Definition 1*). The last matrix The last matrix $\mathbf{B}_n = [\mathbf{x}_{j(1)}, \ldots, \mathbf{x}_{j(n)}]^T$ in the sequence (2.5) is composed of linearly independent vectors $\mathbf{x}_{j(k)}$ ($k = 1, \ldots, n$). In this case, the inverse matrix $\mathbf{X}^{-1}$ (2.2) can be obtained from the matrix $\mathbf{B}_n^{-1} = [\mathbf{r}_1(n), \ldots, \mathbf{r}_n(n)]$ (2.6) through the replacement of the columns $\mathbf{r}_i(n)$ (*Remark 8*). If the matrix $\mathbf{X}^{-1}$ (2.2) exists, then the vector $\mathbf{w}_n^*$ (5.10) is the sum (5.12) of n columns $\mathbf{r}_i(n)$ (2.6) and the rank r_k of this vertex is equal to n ($r_k = n$) □. □

The sequence of the bases $\mathbf{B}_k$ (2.5) is stopped at the stage k when there is no vector $\mathbf{x}_{j(k+1)}$ ($\mathbf{x}_{j(k+1)} \notin \mathbf{B}_{r_k}$) that could be inserted into matrix $\mathbf{B}_k = [\mathbf{x}_{j(1)}, \ldots, \mathbf{x}_{j(k)}, \mathbf{e}_{k+1}, \ldots, \mathbf{e}_n]^T$ (2.4) in accordance with the condition $|\mathbf{r}_{k+1}(k)^T\mathbf{x}_{j(k+1)}| \geq \epsilon$ (4.1). Such situation occurs in the k vertex $\mathbf{w}_k = [w_{k,1}, \ldots, w_{k,k}, 0, \ldots, 0]^T$ (5.9) when each reduced, non-basis vector $\mathbf{x}_{j(k+1)}[k]$ ($\mathbf{x}_{j(k+1)} \notin \mathbf{B}_k$) is a linear combination (3.7) of the basis vectors $\mathbf{x}_{j(i)}[k+1]$ (3.6) with $i \leq k$ [6]:

$$(\forall \mathbf{x}_{j(k+1)} \notin \mathbf{B}_k)\mathbf{x}_{j(k+1)}[k+1] = \\ \alpha_{j(k+1),1}\mathbf{x}_{j(1)}[k+1] + \ldots + \alpha_{j(k+1),k}\mathbf{x}_{j(k)}[k+1], \tag{5.13}$$

where $(\forall i \in \{1, \ldots, k\})\alpha_{j(k+1),i} \in R^1$.

Let us regard the parameters $\alpha_{j(k+1),i}$ (5.13) which for some reduced feature vector $\mathbf{x}_{j(k+1)}[k+1]$ (3.6) fulfill the below *standardizing condition* [5]:

$$\alpha_{j(k+1),1} + \ldots + \alpha_{j(k+1),k} = 1. \tag{5.14}$$

Lemma 5.7. If the parameters $\alpha_{j(k+1),i}$ (5.13) fulfill the standardizing condition (5.14) for some feature vector $\mathbf{x}_{j(k+1)}$ then the dual hyperplane $h_{j(k+1)}^1$ (5.3) defined by this vector passes through the vertex $\mathbf{w}_k$ (5.8) [5]:

$$\mathbf{w}_k^T\mathbf{x}_{j(k+1)} = 1.$$

The vertex $\mathbf{w}_k$ (5.8) of the rank k is *degenerated* if the number n_k of dual hyperplanes $h_{j(k+1)}^1$ (5.3) passing through this vertex is greater than k ($n_k > k$). The *degree of degeneration* of this vertex is defined as $d_k = n_k - k$.

Theorem 5.8. The minimal value $\mathbf{\Phi}_{inv}(\mathbf{w}_k^*)$ (5.10) of the inversion criterion function $\mathbf{\Phi}_{inv}(\mathbf{w})$ (5.2) is equal to zero in the degenerated vertex $\mathbf{w}_{r_k}$ (5.8) of the rank k ($k < n$) if the *degree of degeneration* of this vertex is equal to $d_k = n - k$.

Proof. The optimal vertex $\mathbf{w}_k^*$ (5.10) constitutes the minimal value $\mathbf{\Phi}_{inv}(\mathbf{w}_\mathbf{k}^*)$ of the criterion function
$\mathbf{\Phi}_{inv}(\mathbf{w})$ (5.2). The minimal value $\mathbf{\Phi}_{inv}(\mathbf{w}_\mathbf{k}^*)$ is equal to zero ($\mathbf{\Phi}_{inv}(\mathbf{w}_\mathbf{k}^*) = 0$) if each of the n dual hyperplanes h_j^1 (5.3) passes through the vertex $\mathbf{w}_\mathbf{k}^*$ (5.10). The optimal vertex $\mathbf{w}_k^*$ (5.10) has the rank $r_k = k$. It means that the vertex $\mathbf{w}_k^*$ is located on the k hyperplanes h_j^1 $(\forall_{j \in J_k})$ (5.3), and on the $n - k$ the hyperplanes h_i^0 $(\forall_{i \in I_k})$ (5.4). As a result $d_k = n - k = n - r_k$. □. □

The minimization of the criterion function $\mathbf{\Phi}_{inv}(\mathbf{w})$ (5.2) can be a useful tool in examining the matrix $\mathbf{X}$ (2.1) singularity. As it results from the *Theorem 1*, the matrix $\mathbf{X}$ (2.1) is nonsingular if the minimal value $\mathbf{\Phi}_{inv}(\mathbf{w}_k^*)$ (5.10) of the inversion criterion function $\mathbf{\Phi}_{inv}(\mathbf{w})$ (5.2) is equal to zero and the rank r_k of the optimal vertex $\mathbf{w}_k^*$ is equal to n. If the optimal vertex $\mathbf{w}_k^*$ (5.10) has the rank r_k less than n $(r_k < n)$, then the sequence of the bases $\mathbf{B}_k$ (2.5) is stopped at such base $\mathbf{B}_{r_k}$ (2.4) which contains r_k feature vectors $\mathbf{x}_{j(i)}(i = 1, \ldots, r_k)$ and $n - r_k$ unit vectors $\mathbf{e}_i$ $(i = r_k + 1, \ldots, n)$. In this case, the matrix $\mathbf{X}$ (2.1) singular. We infer, on the basis of the *Theorem 2*, that the maximal nonsingular submatrix $\mathbf{X}'$ $(\mathbf{X}' \subset \mathbf{X})$ can be extracted from the matrix $\mathbf{X}$ (2.1) under the condition that $\mathbf{\Phi}_{inv}(\mathbf{w}_k^*) = 0$ (5.10). The submatrix $\mathbf{X}'$ extracted in this case by neglecting $n - r_k$ rows and the same number of columns in the matrix $\mathbf{X}$ (2.1) selected according to zero components $w_{k,i}^*$ $(w_{k,i}^* = 0)$ of the optimal vertex $\mathbf{w}_k^*$) (5.10) [5].

6 Matrix Inversion Based on the Gauss-Jordan Transformation

We have assumed a natural order of the of unit vectors $\mathbf{e}_k$ replacement in the proposed multistage procedure. During the k-th stage, the unit vectors $\mathbf{e}_k$ is replaced by the $j(k)$-th feature vector $\mathbf{x}_{j(k)}$. During the first stage $(k = 1)$ the unit vector $\mathbf{e}_1$ in the matrix $\mathbf{B}_0 = \mathbf{B}_0^{-1} = \mathbf{I} = [\mathbf{e}_1, \ldots, \mathbf{e}_n]$ is replaced by the feature vector $\mathbf{x}_{j(1)} = [x_{j(1),1}, \ldots, x_{j(1),n}]^T$, the nonsingular matrix $\mathbf{B}_1 = [\mathbf{x}_{j(1)}, \mathbf{e}_2, \ldots, \mathbf{e}_n]^T$ (2.5) appears, and:

$$\mathbf{B}_1^{-1} = [\mathbf{r}_1(1), \ldots, \mathbf{r}_n(1)]. \tag{6.1}$$

The columns $\mathbf{r}_i(1)$ of the matrix $\mathbf{B}_1^{-1}$ (6.1) are determined by the Gauss-Jordan transformation (3.2):

$$\mathbf{r}_1(1) = (1/\mathbf{e}_1^T\mathbf{x}_{j(1)})\mathbf{e}_1 = (1/\mathbf{x}_{j(1),1})\mathbf{e}_1$$

and

$$\begin{aligned} &(\forall i \in \{2, \ldots, n\}) \\ &\quad \mathbf{r}_i(1) = \mathbf{e}_i - (\mathbf{r}_i(1)^T\mathbf{x}_{j(1)})\mathbf{r}_1(1) = \\ &\quad\quad \mathbf{e}_i - (\mathbf{e}_i^T\mathbf{x}_{j(1)}/\mathbf{e}_1^T\mathbf{x}_{j(1)})\mathbf{e}_1 = \\ &\quad\quad \mathbf{e}_i - (x_{j(1),i}/x_{j(1),1})\mathbf{e}_1. \end{aligned}$$

Remark 6.1. The feature vector $\mathbf{x}_{j(1)} = [x_{j(1),1}, \ldots, x_{j(1),n}]^T$ enters the basis $\mathbf{B}_0 = [\mathbf{e}_1, \ldots, \mathbf{e}_n]^T$ if $x_{j(1),1} \geq \epsilon$ (4.1).

During the second stage $(k = 2)$ the unit vector $\mathbf{e}_2$ in the matrix $\mathbf{B}_1 = [\mathbf{x}_{j(1)}, \mathbf{e}_2, \ldots, \mathbf{e}_n]^T$ is replaced by the feature vector $\mathbf{x}_{j(2)} = [x_{j(2),1}, \ldots, x_{j(2),n}]^T$ and the below basis $\mathbf{B}_2$ appears:

$$\mathbf{B}_2 = [\mathbf{x}_{j(1)}, \mathbf{x}_{j(2)}, \mathbf{e}_3 \ldots, \mathbf{e}_n]^T.$$

The columns $\mathbf{r}_i(2)$ $(i = 1, \ldots, n)$ of the inverse matrix $\mathbf{B_2}^{-1} = [\mathbf{r}_1(2), \ldots, \mathbf{r_n}(2)]$ can be computed in the following manner (3.2):

$$\begin{aligned}\mathbf{r}_2(2) = [1/\mathbf{r}_2(1)^T\mathbf{x}_{j(2)}]\mathbf{r}_2(1) =\\ =[1/(\mathbf{e}_2 - (x_{j(1),2}/x_{j(1),1})\mathbf{e}_1)^T\mathbf{x}_{j(2)}]\\ (\mathbf{e}_2 - (x_{j(1),2}/x_{j(1),1})\mathbf{e}_1) =\\ =[1/(x_{j(2),2} - (x_{j(1),2}/x_{j(1),1})x_{j(2),1})]\\ (\mathbf{e}_2 - (x_{j(1),2}/x_{j(1),1})\mathbf{e}_1),\end{aligned}$$

$$\begin{aligned}\mathbf{r}_1(2) = \mathbf{r}_1(1) - \mathbf{r}_1(1)^T\mathbf{x}_{j(2)}\mathbf{r_2}(2) =\\ = [1/(\mathbf{e}_2 - (x_{j(1),2}/x_{j(1),1})\mathbf{e}_1)^T\mathbf{x}_{j(2)}]\\ (\mathbf{e}_2 - (x_{j(1),2}/x_{j(1),1})\mathbf{e}_1) =\\ = [1/(x_{j(2),2} - (x_{j(1),2}/x_{j(1),1})x_{j(2),1})]\\ (\mathbf{e}_2 - (x_{j(1),2}/x_{j(1),1})\mathbf{e}_1),\end{aligned}$$

and

$$\begin{aligned}(\forall i \in \{3, \ldots, n\})\\ \mathbf{r}_i(2) = \mathbf{r}_i(1) - (\mathbf{r}_i(1)^T\mathbf{x}_{j(2)})\mathbf{r}_2(2) =\\ \mathbf{e}_i - (x_{j(1),i}/x_{j(1),1})\mathbf{e}_1 - (x_{j(2),1}/x_{j(1),1})\\ [1/(x_{j(2),2} - (x_{j(1),2}/x_{j(1),1})x_{j(2),1})]\\ (\mathbf{e}_2 - (x_{j(1),2}/x_{j(1),1})\mathbf{e}_1).\end{aligned}$$

According to the proposed method, the inverse matrix $\mathbf{X}^{-1}$ (2.2) can be obtained through the n stages of the computations of the inverted matrices $\mathbf{B}_k^{-1}$ (2.6) with n rows $\mathbf{q}_i(k)$:

$$(\forall k \in \{1, \ldots, n\})\mathbf{B}_k^{-1} = [\mathbf{q}_1(k), \ldots, \mathbf{q}_n(k)]^T. \tag{6.2}$$

The rows $\mathbf{q}_i(k)$ of the inverted matrices $\mathbf{B}_k^{-1}$ (6.2) can be treated as the *computational layers* $(i = 1, \ldots, n)$. When calculating the inverse matrix $\mathbf{B}_k^{-1}$ (6.2) most calculations are performed in the first layer $\mathbf{q}_1(k)$. The first layer $\mathbf{q}_1(k)$ has to be computed n times. Least calculations must be performed in the last layer $\mathbf{q}_n(k)$. The last layer $\mathbf{q}_n(k)$ has to be computed only once.

7 Numerical Example

Experimental verification of the method presented in the paper has been done with a code written in Python using the *NumPy* library [10]. The random matrix of the size 1000x1000 was inverted and the results validated via multiplication by original data to test if they give the identity matrices. The random matrix of the 1000x1000 were inverted in 6 seconds on the personal computer (Intel Core i7-740QM processor with 8GB RAM).

The complete process of matrix inversion during the successive k $(k = 5)$ stages is demonstrated on the example of the following matrix $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_5]^T$ (2.1) with five rows $\mathbf{x}_j$ (Table 1).

Each inverse matrix $\mathbf{B}_k^{-1} = [\mathbf{r}_1(k), \ldots, \mathbf{r}_n(k)]$ (2.6) is composed of five columns $\mathbf{r}_j(k)$.

$$\mathbf{X} = \begin{bmatrix} 1 & -3 & 0 & -1 & 0 \\ 0 & 0 & -2 & 0 & 3 \\ 2 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & -4 & 0 \\ 5 & 0 & -5 & 0 & 6 \end{bmatrix}$$

$$\mathbf{X}^{-1} = \begin{bmatrix} 0 & 0 & 1/2 & 0 & 0 \\ -1/4 & 0 & 1/8 & 1/16 & 0 \\ 0 & 2 & 2.5 & 0 & -1 \\ -1/4 & 0 & 1/8 & -3/16 & 0 \\ 0 & 5/3 & 5/3 & 0 & -2/3 \end{bmatrix}$$

Table 1. The complete process of matrix inversion during the successive stages.

k	$\mathbf{B}_k$	$\mathbf{B}_k^{-1}$	Remarks
0	$\begin{bmatrix} 1&0&0&0&0 \\ 0&1&0&0&0 \\ 0&0&1&0&0 \\ 0&0&0&1&0 \\ 0&0&0&0&1 \end{bmatrix}$	$\begin{bmatrix} 1&0&0&0&0 \\ 0&1&0&0&0 \\ 0&0&1&0&0 \\ 0&0&0&1&0 \\ 0&0&0&0&1 \end{bmatrix}$	The process of the matrix $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_5]^T$ inversion starts with the unit matrix $\mathbf{B}_0 = \mathbf{I} = [\mathbf{e}_1, \ldots, \mathbf{e}_n]$ (2.5)
1	$\begin{bmatrix} 1&-3&0&-1&0 \\ 0&1&0&0&0 \\ 0&0&1&0&0 \\ 0&0&0&1&0 \\ 0&0&0&0&1 \end{bmatrix}$	$\begin{bmatrix} 1&3&0&1&0 \\ 0&1&0&0&0 \\ 0&0&1&0&0 \\ 0&0&0&1&0 \\ 0&0&0&0&1 \end{bmatrix}$	The first feature vector $\mathbf{x}_1 = [1, -3, 0, -1, 0]^T$ enters the base $\mathbf{B}_1$.
2	$\begin{bmatrix} 1&-3&0&-1&0 \\ 2&0&0&0&0 \\ 0&0&1&0&0 \\ 0&0&0&1&0 \\ 0&0&0&0&1 \end{bmatrix}$	$\begin{bmatrix} 0&3&0.5&0&0 \\ -1/3&1/6&0&-1/3&0 \\ 0&0&1&0&0 \\ 0&0&0&1&0 \\ 0&0&0&0&1 \end{bmatrix}$	Third feature vector $\mathbf{x}_3 = [2, 0, 0, 0, 0]^T$ enters the base $\mathbf{B}_2$, because $\mathbf{x}_2^T \mathbf{r}_2(2) = 0$ and $\mathbf{x}_3^T \mathbf{r}_2(2) \neq 0$
3	$\begin{bmatrix} 1&-3&0&-1&0 \\ 2&0&0&0&0 \\ 0&0&-2&0&3 \\ 0&0&0&1&0 \\ 0&0&0&0&1 \end{bmatrix}$	$\begin{bmatrix} 0&1/2&0&0&0 \\ -1/3&1/6&0&-1/3&0 \\ 0&0&-1/2&0&1.5 \\ 0&0&0&1&0 \\ 0&0&0&0&1 \end{bmatrix}$	Second feature vector $\mathbf{x}_2 = [0, 0, -2, 0, 3]^T$ enters the base $\mathbf{B}_3$, as $\mathbf{x}_2^T \mathbf{r}_3(3) \neq 0$
4	$\begin{bmatrix} 1&-3&0&-1&0 \\ 0&0&-2&0&3 \\ 2&0&0&0&0 \\ 0&4&0&-4&0 \\ 0&0&0&0&1 \end{bmatrix}$	$\begin{bmatrix} 1/2&0&0&0& \\ -1/4&1/8&0&1/16&0 \\ 0&0&-1/2&0&1.5 \\ -1/4&1/8&0&-3/16&0 \\ 0&0&0&0&1 \end{bmatrix}$	Fourth feature vector $\mathbf{x}_4 = [0, 4, 0, -4, 0]^T$ enters the base $\mathbf{B}_4$.
5	$\begin{bmatrix} 1&-3&0&-1&0 \\ 0&0&-2&0&3 \\ 2&0&0&0&0 \\ 0&4&0&-4&0 \\ 5&0&-5&0&6 \end{bmatrix}$	$\begin{bmatrix} 0&1/2&0&0&0 \\ -1/4&1/8&0&1/16&0 \\ 0&2.5&2&0&-1 \\ -1/4&1/8&0&-3/16&0 \\ 0&5/3&5/3&0&-2/3 \end{bmatrix}$	Fifth feature vector $\mathbf{x}_5 = [5, 0, -5, 0, 6]^T$ enters the base $\mathbf{B}_5$. $\mathbf{B}_5^{-1}$ is not equal to $\mathbf{X}^{-1}$ because the columns $\mathbf{r}_j(5)$ have to be reordered

As the order of the feature vectors $\mathbf{x}_j$ entering the base is the following: $[1, 3, 2, 4, 5]$, the final inverse matrix has its columns $\mathbf{r}_j(k)$ to be reordered accordingly – the second and the third ones are swapped.

8 Concluding Remarks

The described multistage procedure of the matrix inversion $\mathbf{X}$ (2.1) is based on the successive replacements (3.1) of the unit vectors $\mathbf{e}_k$ in the matrix $\mathbf{I} = [\mathbf{e}_1, \ldots, \mathbf{e}_n]$ by some feature vector

$\mathbf{x}_{j(k)}$ (5.2). The replacement of the vector $\mathbf{e}_k$ by the feature vector $\mathbf{x}_{j(k)}$ (5.2) causes the modification of the columns $\mathbf{r}_i(k)$ of the k-th inversed basis $\mathbf{B}_k^{-1}$ (2.6). New columns $\mathbf{r}_i(k+1)$ of the inverted basis $\mathbf{B}_k^{-1}$ (2.6) are efficiently computed with the Gauss-Jordan vector transformation (3.2).

The proposed method of inversion could make it possible to increase the size of the inverted matrices. This possibility is based on similarity to the Simplex algorithm of linear programming [11].

It is also expected that the computational efficiency of the new procedure of large matrices inversion will be high. In an attempt to increase the computational efficiency the parallel implementations of the inversion algorithms based on the vector Gauss-Jordan transformation should be examined [4, 9, 12].

The presented stepwise method gives a possibility for a partial inversion of large matrices. A partial inversion of a given matrix means that the inversion procedure is stopped during the k-th stage ($k < n$) before all unit vectors $\mathbf{e}_i$ in the matrix $\mathbf{B}_k$ (2.4) have been replaced by the feature vectors $\mathbf{x}_{j(i)}$. Earlier stopping of the inverting process may be caused by the condition (4.1) protecting against an excessive increase of the matrix ill-conditioning.

Availability of Data and Material

The authors can make available the matrix examples on demand.

Code availability: The authors can make available the Python source code on demand.

Competing Interests

Authors have declared that no competing interests exist.

References

[1] Hand David J, Smyth Padhraic, Mannila Heikki. Principles of data mining. MIT Press; 2001. ISBN: 026208290X.

[2] Johnson Richard A, Wichern DW. Applied multivariate statistical analysis. Prentice-Hall International Editions. Prentice Hall; 1992.
Available: https://books.google.pl/books?id=UYwQAQAAIAAJ
ISBN: 9780130417732, LCCN=91042721.

[3] Duda Richard O, Hart Peter E, Stork David G. Pattern classification. New York; 2001.
Available:https://www.bibsonomy.org/bibtex/25ef4fe4778daaf4b4e56c0d66161e048/flint63

[4] Sharma Girish, Agarwala Abhishek, Bhattacharya Baidurya. A Fast Parallel Gauss Jordan Algorithm for Matrix Inversion Using CUDA. Pergamon Press, Inc. USA. Comput. Struct. 2013;128(7):31-37.
Available:https://doi.org/10.1016/j.compstruc.2013.06.015
DOI : 10.1016/j.compstruc.2013.06.015
ISSN: 0045-7949.

[5] Bobrowski Leon. Discovering Main Vertexical Planes in a Multivariate Data Space by Using CPL Functions. In: Advances in Data Mining. Applications and Theoretical Aspects. Springer International Publishing. 2014;200-213.
ISBN:
978-3-319-08976-8.

[6] Bobrowski Leon. Large matrices inversion using the basis exchange algorithm. Journal of Advances in Mathematics and Computer Science. 2017;1-11.

[7] Bobrowski Leon. Data exploration and linear separability. LAMBERT Academic Publishing; 2019.
ISBN: 978-6139846979.

[8] Simonnard M. Linear programming. International series in management. Prentice-Hall; 1966.
Available: https://books.google.pl/books?id=2eVQAAAAMAAJ

[9] Mamalis Basilis, Pantziou Grammati. Advances in the parallelization of the simplex method. In: Algorithms, Probability, Networks, and Games: Scientific Papers and Essays Dedicated to Paul G. Spirakis on the Occasion of His 60th Birthday. Zaroliagis, Christos and Pantziou, Grammati and Kontogiannis, Spyros, Editor. Springer International Publishing. 2015;281-307.
ISBN: 978-3-319-24024-4, DOI: 10.1007/978-3-319-24024-4_17

[10] Oliphant, Travis E. A guide to NumPy. Trelgol Publishing USA. 2006;1.

[11] Koberstein Achim, Suhl Uwe H. Progress in the Dual Simplex Method for Large Scale LP Problems: Practical Dual Phase 1 Algorithms. Comput. Optim. Appl. 2007;37(1):49-65.
Available: https://doi.org/10.1007/s10589-007-9022-3
ISSN: 0926-6003
DOI: 10.1007/s10589-007-9022-3.

[12] Yang K, Li Y, Xia Y. A parallel method for matrix inversion based on Gauss-Jordan algorithm. Journal of Computational Information Systems. 2013;9:5561-5567.
DOI: 10.12733/jcis6364

Peer-review history:
The peer review history for this paper can be accessed here (Please copy paste the total link in your browser address bar)
https://www.sdiarticle5.com/review-history/83290