# Extraction of linearly independent rows and dependency coefficients in datasets

*Raphaël* Sasportas[1]

[1] ECAM-EPMI, Cergy, France

**Abstract.** This paper presents a constructive method for extracting a maximal set of linearly independent rows from a dataset matrix and computing explicit dependency coefficients for dependent rows. The proposed approach incrementally builds an independent subset by preserving linear independence at each step and subsequently derives the coefficient matrix expressing dependent rows as linear combinations of independent rows using a closed-form analytical expression based on the Gram matrix. Unlike classical rank-revealing techniques such as Gaussian elimination or singular value decomposition, which primarily identify independent components, the method directly provides explicit reconstruction relationships between dependent and independent rows. This enables structural analysis of redundancy within datasets and facilitates dimensionality reduction and feature selection. The mathematical formulation, algorithmic procedure, and numerical considerations are presented, demonstrating the effectiveness and interpretability of the approach.

## 1 Introduction

In many scientific and engineering applications, data are naturally represented in matrix form, where each row corresponds to an observation and each column to a feature [8]. Such datasets often contain redundant information, as some rows may be expressed as linear combinations of others. Identifying a maximal subset of linearly independent rows is therefore a fundamental problem in linear algebra and data analysis [8]. This task plays an important role in dimensionality reduction, feature selection, data compression, and numerical stability assessment.

Classical linear algebra provides several tools to address linear dependence, such as Gaussian elimination, QR decomposition, and singular value decomposition (SVD) [1]. These methods allow the rank of a matrix to be determined and reveal linear dependence relationships. However, they are not always designed to explicitly extract a subset of independent rows while simultaneously providing the coefficients that express dependent rows as linear combinations of this subset in a constructive and interpretable manner. In many practical applications, especially in data analysis contexts, such explicit extraction is essential.

This paper addresses the problem of extracting linearly independent rows from a dataset and determining the corresponding dependency coefficients. We propose a constructive algorithm, called DataSelect, which iteratively identifies independent rows and expresses dependent rows as linear combinations of previously selected rows. The method provides

both the independent subset and the associated dependency structure, making it particularly suitable for data analysis and numerical applications.
The remainder of the paper is organized as follows. Section 2 presents the mathematical formulation of the problem. Section 3 describes the proposed algorithm. Section 4 illustrates the method with a numerical examples. Section 5 discusses numerical considerations, and Section 6 concludes the paper.

## 2 Mathematical formulation

Let $A \in \mathbb{R}^{NxP}$ be a real matrix representing a dataset, where each row corresponds to an observation and each column to a feature. The rows of $A$ are denoted by

$$A = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \end{pmatrix}, a_i \in \mathbb{R}^{1xP}, i = 1, \dots, N$$

The rows of $A$ span a vector subspace of $\mathbb{R}^P$ [4] whose dimension is equal to the rank of the matrix:

$$rank(A) = r.$$

The rows of a matrix form a basis of the row space when they are linearly independent [3].
The objective is to extract a maximal subset of linearly independent rows of $A$. Let $Y \in \mathbb{R}^{r \times P}$ denote the matrix formed by these independent rows, and let $X \in \mathbb{R}^{(N-r) \times P}$ denote the matrix formed by the remaining rows, which are linearly dependent on the rows of $Y$.

Since the rows of $X$ belong to the row space of $Y$, there exists a coefficient matrix $Z \in \mathbb{R}^{r \times (N-r)}$ such that

$$X = Z^{\top} Y.$$

This follows from the fundamental properties of vector spaces and linear combinations [4] and expresses each dependent row as a linear combination of the independent rows [3].
Each column of $Z$ contains the coefficients expressing one dependent row as a linear combination of the rows of $Y$ [4].
The problem addressed in this work is therefore to determine the matrix $Y$, whose rows form a maximal linearly independent subset of the dataset, and to compute explicitly the matrix $Z$, which characterizes the linear dependency relationships between the rows of $A$.

## 3 Proposed algorithm

The proposed algorithm, called *DataSelect*, constructively extracts a maximal subset of linearly independent rows from the dataset matrix $A \in \mathbb{R}^{N \times P}$ and computes the dependency coefficient matrix $Z$.
The algorithm proceeds sequentially. The first row of $A$ is placed in the matrix $Y$, which initially contains one row. Each subsequent row $a_i \in \mathbb{R}^{1 \times P}$ is then tested for linear independence with respect to the current rows of $Y$.
To perform this test, a temporary matrix $\tilde{Y}$ is constructed by appending the candidate row to $Y$:

$$\tilde{Y} = \begin{pmatrix} Y \\ a_i \end{pmatrix}.$$

The Gram matrix associated with $\tilde{Y}$ is then computed:

$$G = \tilde{Y}\tilde{Y}^{\top}.$$

The Gram matrix provides a fundamental characterization of linear independence and matrix rank [5,7] and is a classical tool for analyzing linear independence and numerical properties of matrices [1].

In exact arithmetic, the row $a_i$ is linearly independent of the rows of $Y$ if and only if the matrix $G$ is non-singular. In practice, due to finite precision arithmetic, independence is determined using a numerical tolerance $\varepsilon > 0$. The row $a_i$ is classified as linearly dependent if

$$\min\big(\mathrm{eig}(G)\big) \leq \varepsilon,$$

and is appended to the matrix $X$. Otherwise, it is appended to the matrix $Y$. This eigenvalue-based criterion is standard in numerical linear algebra [2].

This process is repeated until all rows of $A$ have been examined. At termination, the matrix $Y$ contains a maximal set of linearly independent rows, and $X$ contains the remaining dependent rows.
The dependency coefficient matrix $Z \in \mathbb{R}^{r\times(N-r)}$ is then computed using the closed-form expression

$$Z = (YY^{\top})^{-1}YX^{\top}.$$

By construction, the dependent rows satisfy the fundamental relationship :

$$X = Z^{\top}Y.$$

The DataSelect algorithm therefore provides both a maximal linearly independent subset of the dataset and explicit coefficients describing the linear dependency relationships between the rows of $A$.

### 3.1 Algorithm 1: DataSelect (row selection and dependency coefficients)

**Input:** $A \in \mathbb{R}^{N\times P}$, tolerance $\varepsilon > 0$

**Output:** $X$ (dependent rows), $Y$ (selected rows), $Z$ (dependency coefficients)

1. Initialize $Y \leftarrow [a_1]$, $X \leftarrow \emptyset$.
2. For $i = 2$ to $N$:
   1. $\tilde{Y} \leftarrow \begin{pmatrix} Y \\ a_i \end{pmatrix}$
   2. $G \leftarrow \tilde{Y}\tilde{Y}^{\top}$
   3. Compute eigenvalues $\{\lambda_j\}$ of $G$
   4. If $\min_j \lambda_j \leq \varepsilon$, then append $a_i$ to $X$; else append $a_i$ to $Y$.
3. Compute
$$Z \leftarrow XY^{\top}(YY^{\top})^{-1}.$$
4. Return $(X, Y, Z)$.

## 4 Numerical example

In this section, we illustrate the behavior of the DataSelect algorithm on three matrices representing distinct structural cases: a low-rank matrix, a random matrix, and a full-rank structured matrix. Eigenvalue-based criteria are commonly used to detect numerical rank and linear dependence [2].

### 4.1 Example 1: Low-rank matrix

We first consider the matrix $A \in \mathbb{R}^{8\times 3}$:

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \\ 13 & 14 & 15 \\ 16 & 17 & 18 \\ 19 & 20 & 21 \\ 22 & 23 & 24 \end{pmatrix}.$$

The rows of this matrix belong to a two-dimensional subspace. Applying the DataSelect algorithm produces the matrix of independent rows

$$Y = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix},$$

and the matrix of dependent rows

$$X = \begin{pmatrix} 7 & 8 & 9 \\ 10 & 11 & 12 \\ 13 & 14 & 15 \\ 16 & 17 & 18 \\ 19 & 20 & 21 \\ 22 & 23 & 24 \end{pmatrix}.$$

The algorithm also computes the dependency coefficient matrix:

$$Z = \begin{pmatrix} -1 & -2 & -3 & -4 & -5 & -6 \\ 2 & 3 & 4 & 5 & 6 & 7 \end{pmatrix}.$$

By construction, the dependent rows satisfy the fundamental relationship:

$$X = Z^{\top} Y.$$

For example, the first dependent row satisfies

$$(7,8,9) = -1\,(1,2,3) + 2\,(4,5,6),$$

and the second dependent row satisfies

$$(10,11,12) = -2\,(1,2,3) + 3\,(4,5,6).$$

This example confirms that the DataSelect algorithm correctly extracts a maximal set of linearly independent rows and computes explicit dependency coefficients for all remaining rows.

### 4.2 Example 2: Random matrix

We next consider a randomly generated matrix $A \in \mathbb{R}^{12\times 5}$ obtained using MATLAB:
A = randi([-10,10],12,5);

$$A = \begin{pmatrix} -5 & 0 & 9 & 5 & -10 \\ 4 & 4 & -3 & 5 & -3 \\ 3 & 8 & -6 & -3 & -7 \\ -7 & 10 & -5 & 1 & 6 \\ -8 & 1 & 2 & -9 & -4 \\ 0 & -8 & -1 & -9 & 1 \\ 10 & -7 & -3 & 1 & -7 \\ -3 & -5 & 7 & 6 & 2 \\ 2 & 7 & 2 & 9 & -5 \\ -6 & -5 & 1 & -8 & 3 \\ 5 & 7 & 9 & 1 & 4 \\ -5 & -5 & -4 & -1 & 5 \end{pmatrix}.$$

Applying the DataSelect algorithm produces the matrix of independent rows

$$Y = \begin{pmatrix} -5 & 0 & 9 & 5 & -10 \\ 4 & 4 & -3 & 5 & -3 \\ 3 & 8 & -6 & -3 & -7 \\ -7 & 10 & -5 & 1 & 6 \\ -8 & 1 & 2 & -9 & -4 \end{pmatrix}$$

and the dependency coefficient matrix

$$Z \in \mathbb{R}^{5\times 7}.$$

By construction, the dependent rows satisfy the fundamental relationship

$$X = Z^{\top}Y.$$

Since $Y$ is square and non-singular, its rows form a basis of the row space of $A$ [3], confirming that the rank of the matrix is equal to $r = 5$.
This example illustrates the behavior of the algorithm in a general setting and confirms its ability to identify a maximal linearly independent subset.

### 4.3 Example 3: Hadamard matrix

We consider a Hadamard matrix constructed recursively from the initial matrix

$$H_1 = (1).$$

The Hadamard matrices are then defined recursively by

$$H_{2n} = \begin{pmatrix} H_n & H_n \\ H_n & -H_n \end{pmatrix}.$$

Using this construction, we obtain the matrix $H \in \mathbb{R}^{8\times 8}$.

This matrix satisfies the orthogonality property

$$HH^{\top} = 8I,$$

which implies that its rows are linearly independent.

Applying the DataSelect algorithm produces

$$Y = H$$

and

$$X = 0.$$

This confirms that the algorithm correctly detects that the matrix is full rank and preserves all independent rows. The matrix $Y$ therefore forms a basis of the row space of $H$ [3].
This example demonstrates that the DataSelect algorithm behaves correctly in the full-rank case. Similar techniques are widely used in data analysis, dimensionality reduction, and pattern recognition [8].

## 5 Numerical considerations

The DataSelect algorithm relies on testing the linear independence of candidate rows using the Gram matrix

$$G = \tilde{Y}\tilde{Y}^{\top}.$$

In exact arithmetic, linear dependence corresponds to the singularity of G. The properties of Gram matrices are well established in matrix analysis [5, 7]. However, in finite precision arithmetic, numerical errors must be taken into account. Therefore, a row is classified as

linearly dependent when the smallest eigenvalue of $G$ is below a prescribed tolerance $\varepsilon > 0$, that is,

$$\min\big(\mathrm{eig}(G)\big) \leq \varepsilon.$$

In the present implementation, the tolerance is set to $\varepsilon = 10^{-5}$. The use of numerical tolerance to detect rank deficiency is standard practice in numerical linear algebra [6]. This value provides a practical compromise between numerical stability and sensitivity in detecting linear dependence. A tolerance that is too small may lead to instability due to floating-point rounding errors, while a tolerance that is too large may incorrectly classify independent rows as dependent.
The computation of the dependency coefficient matrix

$$Z = (YY^{\top})^{-1}YX^{\top}$$

requires the inversion of the matrix $YY^{\top}$. This type of formulation is standard in numerical linear algebra [5]. Since the rows of $Y$ are linearly independent by construction, the matrix $YY^{\top}$ is symmetric positive definite and therefore invertible. This guarantees the existence and numerical stability of the coefficient matrix $Z$.
From a computational perspective, the dominant cost of the algorithm arises from the repeated evaluation of eigenvalues of Gram matrices of size at most $r \times r$, where $r$ is the rank of the dataset. The overall computational complexity is therefore of order

$$\mathcal{O}(Nr^2),$$

which remains efficient when the intrinsic dimension $r$ is small compared to the number of rows $N$.
Finally, the constructive nature of the algorithm ensures that the dependency coefficients are computed explicitly, making the method particularly suitable for applications in data analysis, feature selection, and dimensionality reduction.

## 6 Conclusion

This paper introduced the DataSelect algorithm, a constructive method for extracting a maximal subset of linearly independent rows from a dataset matrix and explicitly computing the associated dependency coefficients. The method partitions the dataset into two matrices: one containing the selected independent rows and another containing the dependent rows, together with a coefficient matrix that fully characterizes the linear dependency relationships. Unlike classical approaches that focus primarily on determining the rank or performing matrix decompositions, the proposed algorithm directly identifies independent rows while providing explicit linear representations of dependent rows. This constructive approach makes the method particularly suitable for data analysis applications where interpretability and explicit dependency relationships are required.
The numerical example illustrates the effectiveness of the algorithm in correctly identifying the intrinsic dimension of the dataset and computing the corresponding dependency coefficients. The numerical considerations show that the method is stable under standard floating-point arithmetic when an appropriate tolerance is used.
Future work may include optimizing the independence test using alternative numerical techniques, such as QR factorization or incremental orthogonalization, and extending the method to large-scale datasets and applications in feature selection, dimensionality reduction, and data compression.

## References

1. G. H. Golub and C. F. Van Loan, Matrix Computations, 4th ed., Johns Hopkins University Press, Baltimore, 2013.
2. L. N. Trefethen and D. Bau III, Numerical Linear Algebra, SIAM, Philadelphia, 1997.
3. G. Strang, Linear Algebra and Its Applications, 4th ed., Brooks/Cole, 2006.
4. K. Hoffman and R. Kunze, Linear Algebra, 2nd ed., Prentice-Hall, 1971.
5. S. Boyd and L. Vandenberghe, Introduction to Applied Linear Algebra: Vectors, Matrices, and Least Squares, Cambridge University Press, 2018.
6. A. Quarteroni, R. Sacco, and F. Saleri, Numerical Mathematics, 2nd ed., Springer, 2007.
7. C. D. Meyer, Matrix Analysis and Applied Linear Algebra, SIAM, 2000.
8. L. Eldén, Matrix Methods in Data Mining and Pattern Recognition, SIAM, 2007.