

Lecture 4c: Metabolic Networks and the Stoichiometric Matrix

Lecturer: Jeffrey Varner

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications.*

In this lecture, we will discuss the following topics:

- **Metabolism and metabolic networks:** A metabolic network is the complete set of metabolic (chemical) processes determining a cell's biochemical state. It encompasses all the chemical reactions associated with metabolism, i.e., the breakdown of raw materials such as sugars (catabolism) and the production of macromolecules, e.g., proteins, lipids, etc (anabolism).
- **A stoichiometric matrix** is a mathematical representation of a metabolic network that encodes the relationships between reactants and products in a metabolic network, where rows correspond to different metabolites, contained in the set \mathcal{M} , and columns correspond to reactions, contained in the set \mathcal{R} . Thus, the stoichiometric matrix is a $\mathbf{S} \in \mathbb{R}^{|\mathcal{M}| \times |\mathcal{R}|}$ matrix holding the stoichiometric coefficients $\sigma_{ij} \in \mathbf{S}$ for $i = 1, 2, \dots, |\mathcal{M}|$ and $j = 1, 2, \dots, |\mathcal{R}|$.
- **Structural analysis of \mathbf{S} :** Structural analysis of the stoichiometric matrix involves examining its connectivity distribution and using tools such as eigendecomposition to explore the network's fundamental pathway structures and other topological properties. These type of analyses give us some more insight into the structure of the network (and perhaps some indication of importance of particular metabolites of reactions).

1 Introduction

In this lecture we'll introduce metabolic networks and the stoichiometric matrix, which is a fundamental tool in systems biology for modeling and analyzing biochemical reaction systems. Metabolic networks are complex systems of enzyme catalyzed biochemical reactions that convert nutrients into energy and building blocks for cellular components, or products of interest. A metabolic network encompasses all the chemical reactions associated with metabolism, i.e., the breakdown of raw materials such as sugars (catabolism) and the production of macromolecules, e.g., DNA, RNA, proteins, lipids, etc (anabolism). These networks are curated for thousands of organisms and are available in various online databases (1–3).

These chemical reaction networks can be mathematically represented using stoichiometric matrices. A stoichiometric matrix provides a systematic way to represent the flow of material through various biochemical reactions, where rows correspond to chemical species (metabolites) and columns correspond to reactions. This matrix is essential for constraint-based modeling and flux balance analysis, allowing researchers to analyze and predict metabolic behavior under different conditions. Today, we'll introduce the stoichiometric matrix and discuss a few of its properties.

2 Stoichiometric Matrix

Suppose we have a set of chemical reactions \mathcal{R} involving the chemical species (metabolite) set \mathcal{M} occurring in some volume V . This can be a physical volume, such as a test tube, or a biological volume, such as a cell

or even a logical volume, such as a compartment in a model. We can represent this metabolic network using the stoichiometric matrix \mathbf{S} (Defn 1):

Definition 1 (Stoichiometric Matrix). *The stoichiometric matrix is a $\mathbf{S} \in \mathbb{R}^{|\mathcal{M}| \times |\mathcal{R}|}$ matrix that holds the stoichiometric coefficients $\sigma_{ij} \in \mathbf{S}$ such that:*

- $\sigma_{ij} > 0$: Chemical species i is produced by reaction j . Species i is a product of reaction j .
- $\sigma_{ij} = 0$: Chemical species i is not connected with reaction j
- $\sigma_{ij} < 0$: Chemical species i is consumed by reaction j . Species i is a reactant of reaction j .

The stoichiometric matrix is a digital representation of the metabolic network, and it can be used to analyze the structure of the network, as well as to estimate operational values, such as the metabolic fluxes.

The stoichiometric matrix \mathbf{S} is the digital representation of the metabolic network, and it can be used to analyze some of the structural properties of the network. There is a large body of literature on the analysis of the structure of metabolic networks using various decomposition approaches of stoichiometric matrices. Let's look at a few of these, starting with the connectivity of the network, the rank of the stoichiometric matrix, and finally the decomposition using singular value decomposition (SVD) (4).

2.1 Connectivity

The connectivity of a metabolic network, and the connectivity distribution, gives us information about the network's structure, e.g., the number of reactions that are connected to a given metabolite, or the number of metabolites that are connected to a given reaction, etc. We compute the connectivity of a metabolic network using the binary stoichiometric matrix $\bar{\mathbf{S}}$ (Defn. 2).

Definition 2 (Binary stoichiometric Matrix). *Suppose we have a stoichiometric matrix \mathbf{S} , the binary stoichiometric matrix $\bar{\mathbf{S}}$ is a matrix with the same dimensions as \mathbf{S} , where the elements are binary, i.e., $\bar{\sigma}_{ij} = 1$ if $\sigma_{ij} \neq 0$, and $\bar{\sigma}_{ij} = 0$ if $\sigma_{ij} = 0$. Matrix products of $\bar{\mathbf{S}}$ with its transpose $\bar{\mathbf{S}}^T$ (or vice-versa) give us connectivity information about the network.*

Metabolite connectivity

The metabolite connectivity matrix, defined as $\mathbf{C}_m \equiv \bar{\mathbf{S}}\bar{\mathbf{S}}^T$, is an $|\mathcal{M}| \times |\mathcal{M}|$ symmetric array with the following features: The elements along the central diagonal $c_{ii} \in \mathbf{C}_m$ are the total number of reactions a particular metabolite participates in. However, because we removed the directionality when we computed the binary stoichiometric matrix, we have no information about whether the participation is a reactant or product. The off-diagonal elements $c_{ij} \in \mathbf{C}_m$ where $i \neq j$ describe how many reactions metabolite i has in common with metabolite j , i.e., the number of joint reactions for the pair.

Reaction connectivity

The reaction connectivity matrix, defined as $\mathbf{C}_r \equiv \bar{\mathbf{S}}^T\bar{\mathbf{S}}$, is an $|\mathcal{R}| \times |\mathcal{R}|$ symmetric array with the following features. The elements along the central diagonal $c_{ii} \in \mathbf{C}_r$ are the total number of reactants and products of a particular reaction. However, because we removed the directionality when we computed the binary stoichiometric matrix, we have no information about the number of reactants or products, just the total participation number for a reaction. The off-diagonal elements of the reaction matrix $c_{ij} \in \mathbf{C}_r$ where $i \neq j$ describe how many metabolites are shared between reaction i and j , i.e., the number of joint metabolites for the pair.

3 Eigendecomposition

In addition to looking at the connectivity of the network, we can also analyze the structure of the network using eigendecomposition of the stoichiometric matrix. However, the stoichiometric matrix is not square, so we cannot use the standard eigendecomposition directly. However, let's introduce eigendecomposition, and then transform the stoichiometric matrix into a square matrix that we can analyze using eigendecomposition.

3.1 Eigenvalues and Eigenvectors

Suppose we have a **square** matrix $\mathbf{A} \in \mathbb{R}^{m \times m}$ which could be a measurement dataset, e.g., the columns of \mathbf{A} represent feature vectors $\mathbf{x}_1, \dots, \mathbf{x}_m$ or an adjacency array in a graph with m nodes, etc. Eigenvalue-eigenvector problems involve finding a set of scalar values $\{\lambda_1, \dots, \lambda_m\}$ called eigenvalues and a set of linearly independent vectors $\{\mathbf{v}_1, \dots, \mathbf{v}_m\}$ called eigenvectors such that:

$$\mathbf{A} \cdot \mathbf{v}_j = \lambda_j \mathbf{v}_j \quad j = 1, 2, \dots, m \quad (1)$$

where $\mathbf{v} \in \mathbb{C}^m$ and $\lambda \in \mathbb{C}$. We can put the eigenvalues and eigenvectors together in matrix-vector form, which gives us an interesting matrix decomposition:

$$\mathbf{A} = \mathbf{V} \cdot \text{diag}(\lambda) \cdot \mathbf{V}^{-1} \quad (2)$$

where \mathbf{V} denotes the matrix of eigenvectors, where the eigenvectors form the columns of the matrix \mathbf{V} , $\text{diag}(\lambda)$ denotes a diagonal matrix with the eigenvalues along the main diagonal, and \mathbf{V}^{-1} denotes the inverse of the eigenvalue matrix. The eigendecomposition of a matrix is a powerful tool that can be used to analyze data and systems in many areas of mathematics, engineering, and physics. Let's discuss some of the interpretations of eigendecomposition and data reduction, which is an interesting application of eigendecomposition. Then, we will discuss how to compute the eigendecomposition of a matrix.

Eigenvectors represent fundamental directions of the matrix \mathbf{A} . For the linear transformation defined by a matrix \mathbf{A} , eigenvectors are the only vectors that do not change direction during the transformation. Thus, if we think about the matrix \mathbf{A} as a machine, we put the eigenvector \mathbf{v}_\star into the machine, and we get back the same eigenvector \mathbf{v}_\star multiplied by a scalar, the eigenvalue λ_\star . On the other hand, eigenvalues are scale factors for their eigenvector. An eigenvalue is a scalar that indicates how much a corresponding eigenvector is stretched or compressed during a linear transformation represented by the matrix \mathbf{A} . We can use the eigendecomposition to diagonalize the matrix \mathbf{A} , i.e., transform the matrix into a diagonal form where the eigenvalues lie along the main diagonal. To see this, solve the eigendecomposition for the $\text{diag}(\lambda) = \mathbf{V}^{-1} \cdot \mathbf{A} \cdot \mathbf{V}$. We can also use the eigenvalues to classify a matrix \mathbf{A} as positive (semi)definite or negative (semi)definite (which will be handy later). Further, suppose the matrix \mathbf{A} is symmetric, and all entries are positive. In that case, all the eigenvalues will be real-valued, and the eigenvectors will be orthogonal (super handy properties, as we shall soon see). Finally, eigenvectors represent the most critical directions in the data or system, and eigenvalues (or functions of them) represent their importance. However, the eigendecomposition is not always possible, e.g., for non-square matrices or matrices that are not diagonalizable (which may be the case with repeated eigenvalues). Finally, it may seem rare to encounter a square symmetric real-valued matrix in practice, e.g., stoichiometric matrices are (often) not square or symmetric; actually, this is not the case in engineering systems and data. Let's dig into the properties of symmetric real-valued matrices and introduce the covariance matrix, a common example of a symmetric real-valued matrix we will encounter in many applications.

3.2 Symmetric Real Matrices

The eigendecomposition of a symmetric real matrix $\mathbf{A} \in \mathbb{R}^{m \times m}$ has some unique properties. First, all the eigenvalues $\{\lambda_1, \lambda_2, \dots, \lambda_m\}$ of the matrix \mathbf{A} are real-valued. Next, the eigenvectors $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m\}$ of the matrix \mathbf{A} are orthogonal, i.e., $\langle \mathbf{v}_i, \mathbf{v}_j \rangle = 0$ for $i \neq j$. Finally, the (normalized) eigenvectors $\mathbf{v}_j / \|\mathbf{v}_j\|$ of a symmetric real-valued matrix form an orthonormal basis for the space spanned by the matrix \mathbf{A} such that:

$$\langle \hat{\mathbf{v}}_i, \hat{\mathbf{v}}_j \rangle = \delta_{ij} \quad \text{for } i, j \in \mathbf{A} \quad (3)$$

where δ_{ij} is the Kronecker delta function. The eigendecomposition of a symmetric real-valued matrix is a powerful tool that can be used to analyze data and systems in many areas of mathematics, engineering, and physics. For example, eigenvectors of a real-symmetric matrix form an orthogonal (orthonormal) basis for the space spanned by the matrix \mathbf{A} . Thus, any vector $\mathbf{x} \in \mathbb{R}^m$ in that space, i.e., a solution vector can be expressed as a linear combination of the eigenvectors of the matrix \mathbf{A} i.e., $\mathbf{x} = \sum_{i=1}^m c_i \mathbf{v}_i$, where c_i are the coefficients of the linear combination. Further, the eigenvectors of a symmetric real-valued matrix can be used to reduce the dimensionality of a dataset (which we will discuss later).

Covariance Matrix

The covariance matrix of a dataset is an example of a symmetric real matrix that we will encounter in various applications. The covariance matrix is a square matrix that summarizes the variance and covariance of the features in the dataset. Suppose we have a dataset $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ where each $\mathbf{x}_i \in \mathbb{R}^m$ is a feature vector. The covariance of feature vectors i and j , denoted as $\text{cov}(\mathbf{x}_i, \mathbf{x}_j)$, is an $\Sigma \in \mathbb{R}^{n \times n}$ real-valued symmetric matrix $\Sigma \in \mathbb{R}^{n \times n}$ with elements:

$$\Sigma_{ij} = \text{cov}(\mathbf{x}_i, \mathbf{x}_j) = \sigma_i \sigma_j \rho_{ij} \quad \text{for } i, j \in \mathcal{D} \quad (4)$$

where σ_i denote the standard deviation of the feature vector \mathbf{x}_i , σ_j denote the standard deviation of the feature vector \mathbf{x}_j , and ρ_{ij} denotes the correlation between features i and j in the dataset \mathcal{D} . The correlation is given by:

$$\rho_{ij} = \frac{\mathbb{E}(\mathbf{x}_i - \mu_i) \cdot \mathbb{E}(\mathbf{x}_j - \mu_j)}{\sigma_i \sigma_j} \quad \text{for } i, j \in \mathcal{D} \quad (5)$$

where $\mathbb{E}(\cdot)$ denotes the expected value, and μ_i denotes the mean of the feature vector \mathbf{x}_i . The diagonal elements of the covariance matrix $\Sigma_{ii} \in \Sigma$ are the variances of features i , while the off-diagonal elements $\Sigma_{ij} \in \Sigma$ for $i \neq j$ measure the relationship between features i and j in the dataset \mathcal{D} . Interestingly, we can use the eigendecomposition of the covariance matrix for data reduction, i.e., factor the dataset \mathcal{D} into a set of weighted (increasingly important) features. For example, if $n \gg 2$, we can use the eigendecomposition of the covariance to reduce the dimensionality of the dataset \mathcal{D} to 2 or 3 dimensions, which can be visualized. We can also use the eigendecomposition of the covariance to find the most important features in the dataset, which can be used for clustering, classification, or other machine-learning tasks. This idea underlies the principal component analysis (PCA) algorithm, which is widely used to analyze biological and clinical dataset.

Kernel Matrix

The kernel matrix is another example of a symmetric real-valued matrix that we will encounter in various applications. The elements of the kernel matrix are computed using a kernel function. Kernel functions are mathematical tools in machine learning that enable algorithms to operate in high-dimensional spaces without explicitly computing the coordinates of those spaces. Kernel functions are used in various machine learning algorithms, including Support Vector Machines (SVMs), kernelized regression, and kernelized clustering.

Kernel functions have a few different interpretations. For example, kernel functions can be considered similarity measures, quantifying the similarity between pairs of data points in a high-dimensional space. They also implicitly map data into a high-dimensional space, where the data becomes linearly separable (which is helpful for classification algorithms). Thus, kernel functions are powerful tools that we will use for many machine learning applications.

A kernel function $k : \mathbb{R}^* \times \mathbb{R}^* \rightarrow \mathbb{R}$ is a function that takes a pair of vectors $\mathbf{v}_i \in \mathbb{R}^*$ and $\mathbf{v}_j \in \mathbb{R}^*$ as arguments, e.g., a pair of feature vectors, a feature vector and a parameter vector, or any two vectors of compatible size and computes a scalar value that represents the similarity (in some sense) between the two vector arguments. For example, the linear kernel function computes the dot product between two vectors:

$$k(\mathbf{v}_i, \mathbf{v}_j) = \mathbf{v}_i^\top \mathbf{v}_j \quad (6)$$

On the other hand, a polynomial kernel is defined as:

$$k_d(\mathbf{v}_i, \mathbf{v}_j) = (1 + \mathbf{v}_i^\top \mathbf{v}_j)^d \quad (7)$$

where d is the degree of the polynomial. The radial basis function (RBF) kernel is defined as:

$$k_\gamma(\mathbf{v}_i, \mathbf{v}_j) = \exp(-\gamma \|\mathbf{v}_i - \mathbf{v}_j\|_2^2) \quad (8)$$

where γ is a scaling factor, and $\|\cdot\|_2^2$ is the squared Euclidean norm; If we define γ as $1/2\sigma^2$, the RBF kernel looks like a Gaussian function, without the normalization constant. Of course, not all functions are kernel functions (Defn. 3).

Definition 3. (*Valid Kernel Function*) A function $k : \mathbb{R}^* \times \mathbb{R}^* \rightarrow \mathbb{R}$ is a valid kernel function if and only if the Kernel matrix \mathbf{K} is positive semidefinite for all possible choices of the data points \mathbf{v}_i , where $K_{ij} = k(\mathbf{v}_i, \mathbf{v}_j)$. This is equivalent to saying that all eigenvalues of the Kernel matrix \mathbf{K} are non-negative. Further, for all real value vectors \mathbf{x} , the Kernel matrix \mathbf{K} must satisfy $\mathbf{x}^\top \mathbf{K} \mathbf{x} \geq 0$.

Kernel functions can also be combined to create more complex kernel functions using the concept of kernel composition. For example, the sum of two valid kernel functions is also a valid kernel function. The product of two valid kernel functions is also a valid kernel function. Multiplying a kernel function by a scalar is also a valid kernel function, etc. See the CS 4780 Lecture Notes (Fall 2018) for more details on kernel composition.

3.3 Computing the Eigendecomposition of a Matrix

There are several techniques to compute the eigendecomposition of a matrix. The most common method is power iteration, an iterative algorithm that finds the eigenvector corresponding to the largest eigenvalue of a matrix. The power iteration method is simple and easy to implement. Still, it may not converge to the desired eigenvector if the matrix is ill-conditioned or has multiple eigenvalues of the same magnitude. Alternatively, we can use the QR iteration algorithm to find all the eigenvalues and eigenvectors. We'll only briefly discuss the QR iteration algorithm here, but it is a powerful method for computing the eigendecomposition of a matrix.

Power Iteration

The power iteration method is an iterative algorithm to compute the largest eigenvalue and its corresponding eigenvector of a square (real) matrix; we'll consider only real-valued matrices here, but this approach can also be used for matrices with complex entries.

The power iteration method consists of two phases:

- **Phase 1: Eigenvector:** Suppose we have a real-valued square diagonalizable matrix $\mathbf{A} \in \mathbb{R}^{m \times m}$ whose eigenvalues have the property $|\lambda_1| \geq |\lambda_2| \cdots \geq |\lambda_m|$. Then, the eigenvector $\mathbf{v}_1 \in \mathbb{C}^m$ which corresponds to the largest eigenvalue $\lambda_1 \in \mathbb{C}$ can be (iteratively) estimated as:

$$\mathbf{v}_1^{(k+1)} = \frac{\mathbf{A}\mathbf{v}_1^{(k)}}{\|\mathbf{A}\mathbf{v}_1^{(k)}\|} \quad k = 0, 1, 2 \dots \quad (9)$$

where $\|\star\|$ denotes the L2 (Euclidean) vector norm. The power iteration method converges to a value for the eigenvector as $k \rightarrow \infty$ when a few properties are true, namely, $|\lambda_1|/|\lambda_2| < 1$ (which is unknown beforehand), and we pick an appropriate initial guess for \mathbf{v}_1 (in our case, a random vector will work)

- **Phase 2: Eigenvalue:** Once we have an estimate for the eigenvector $\hat{\mathbf{v}}_1$, we can estimate the corresponding eigenvalue $\hat{\lambda}_1$ using the Rayleigh quotient. This argument proceeds from the definition of the eigenvalues and eigenvectors. We know, from the definition of eigenvalue-eigenvector pairs, that:

$$\mathbf{A}\hat{\mathbf{v}}_1 - \hat{\lambda}_1\hat{\mathbf{v}}_1 \simeq 0 \quad (10)$$

where we use the \simeq symbol because we don't have the true eigenvector \mathbf{v}_1 , only an estimate of it. To solve this expression for the (estimated) eigenvalue $\hat{\lambda}_1$, we multiply through by the transpose of the eigenvector and solve for the eigenvalue:

$$\hat{\lambda}_1 \simeq \frac{\hat{\mathbf{v}}_1^\top \mathbf{A} \hat{\mathbf{v}}_1}{\hat{\mathbf{v}}_1^\top \hat{\mathbf{v}}_1} = \frac{\langle \mathbf{A}\hat{\mathbf{v}}_1, \hat{\mathbf{v}}_1 \rangle}{\langle \hat{\mathbf{v}}_1, \hat{\mathbf{v}}_1 \rangle} \quad (11)$$

where $\langle \star, \star \rangle$ denotes an inner product. While simple to implement, the power iteration method may exhibit slow convergence, mainly when the largest eigenvalue is close in magnitude to other eigenvalues, i.e., $|\lambda_1|/|\lambda_2| \sim 1$.

The power iteration method has several notable applications. For example, likely the most famous application is the Google PageRank algorithm. Google's PageRank algorithm, which uses power iteration, utilizes the dominant eigenvalue and its corresponding eigenvector to assess the importance of web pages within a network. For a deeper discussion of the PageRank algorithm, see Brezinski and Redivo-Zaglia (5). Psuedo code for the power iteration method is shown in Algorithm 1.

Algorithm 1 Power Iteration Method

```

1: Input: Matrix  $A \in \mathbb{R}^{n \times n}$ , initial vector  $x_0 \in \mathbb{R}^n$ , tolerance  $\epsilon$ , maximum iterations  $N$ 
2: Normalize the initial vector:  $x_0 \leftarrow \frac{x_0}{\|x_0\|}$ 
3: for  $k = 1, 2, \dots, N$  do
4:   Compute the matrix-vector product:  $y_k \leftarrow Ax_{k-1}$ 
5:   Normalize the resulting vector:  $x_k \leftarrow \frac{y_k}{\|y_k\|}$ 
6:   Compute the Rayleigh quotient:  $\lambda_k \leftarrow x_k^\top Ax_k / (x_k^\top x_k)$ 
7:   if  $\|x_k - x_{k-1}\| < \epsilon$  then
8:     break
9:   end if
10: end for
11: Set  $v \leftarrow x_k$  and  $\lambda \leftarrow \lambda_k$ 
12: return  $\lambda, v$ 

```

3.4 The QR Iteration Algorithm

The QR iteration algorithm relies on the QR decomposition of a matrix, which is a factorization of a matrix into an orthogonal matrix and an upper triangular matrix. QR decomposition, originally developed by Francis in the early 1960s (6, 7), factors a (rectangular) matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$ into the product of an orthogonal matrix $\mathbf{Q} \in \mathbb{R}^{n \times n}$ and an upper triangular matrix $\mathbf{R} \in \mathbb{R}^{n \times m}$:

$$\mathbf{A} = \mathbf{Q}\mathbf{R} \quad (12)$$

where $\mathbf{Q}^\top = \mathbf{Q}^{-1}$ (property of an orthogonal matrix). The QR decomposition is helpful in computing a matrix's eigenvalues and eigenvectors and finding the solution to a system of linear algebraic equations. We can compute the \mathbf{Q} of QR decomposition using the Gram-Schmidt process, Householder transformations, or Givens rotations.

To compute the eigendecomposition of a matrix using QR decomposition, we first factorize the matrix \mathbf{A} into the product of \mathbf{Q} and \mathbf{R} , then we use the QR-iteration algorithm. The QR-iteration algorithm is an iterative method that computes the eigenvalues and eigenvectors of a matrix by repeatedly applying the QR decomposition to the matrix. The QR iteration algorithm estimates the eigenvalues and eigenvectors of a square matrix.

The QR-iteration algorithm consists of two phases:

- **Phase 1: Eigenvalues.** Let $\mathbf{A} \in \mathbb{R}^{n \times n}$. Then, starting with $k = 0$, we compute the QR decomposition of the matrix \mathbf{A}_k :

$$\mathbf{A}_{k+1} \leftarrow \mathbf{R}_k \mathbf{Q}_k = \mathbf{Q}_k^\top \mathbf{A}_k \mathbf{Q}_k \quad k = 0, 1, \dots$$

where \mathbf{Q}_k is an orthogonal matrix, i.e., $\mathbf{Q}_k^\top = \mathbf{Q}_k^{-1}$ and \mathbf{R}_k is an upper triangular matrix. As $k \rightarrow \infty$, the matrix \mathbf{A}_k converges to a triangular matrix with the eigenvalues listed along the diagonal.

- **Phase 2: Eigenvectors.** We compute the eigenvectors associated with each eigenvalue by solving the homogenous system of equations:

$$(\mathbf{A} - \lambda_j \mathbf{I}) \cdot \mathbf{v}_j = \mathbf{0}$$

where \mathbf{I} is the identity matrix.

3.5 Singular Value Decomposition (SVD)

Another widely used approach to decompose (rectangular) matrices is the singular value decomposition (SVD) method. Singular value decomposition, originally developed in the 1870s (8) is a matrix factorization technique that is based on the eigendecomposition of a matrix. We can compute the rank (and get some other useful information) about a matrix \mathbf{A} by decomposing the matrix using singular value decomposition into its singular values and singular vectors.

Suppose we have a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$. The SVD of \mathbf{A} is a factorization of the form: $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$, where $\mathbf{U} \in \mathbb{R}^{n \times n}$ and $\mathbf{V} \in \mathbb{R}^{m \times m}$ are orthogonal matrices, i.e., $\mathbf{U} \cdot \mathbf{U}^\top = \mathbf{I}$ and $\mathbf{\Sigma} \in \mathbb{R}^{n \times m}$ is a diagonal matrix containing the singular values $\sigma_i = \Sigma_{ii}$. The matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$ can be decomposed as:

$$\mathbf{A} = \sum_{i=1}^{r_{\mathbf{A}}} \sigma_i \cdot (\mathbf{u}_i \otimes \mathbf{v}_i) \quad (13)$$

where $r_{\mathbf{A}}$ is the rank of matrix \mathbf{A} , and σ_i are the singular values (ordered from largest to smallest) of the matrix \mathbf{A} , and \otimes denotes the outer product. The outer product $\hat{\mathbf{A}}_i = \mathbf{u}_i \otimes \mathbf{v}_i$ is a rank-1 matrix, i.e., a mode of the original matrix \mathbf{A} , with elements:

$$\hat{a}_{jk} = u_j v_k \quad j = 1, 2, \dots, n \text{ and } k = 1, 2, \dots, m \quad (14)$$

where the vectors \mathbf{u}_i and \mathbf{v}_i denote the left (right) singular vectors, respectively, of the matrix \mathbf{A} . Singular value decomposition is a powerful tool for analyzing the structure of a matrix, e.g., for computing properties such as the rank of a matrix, and is used in many applications, including data compression, image processing, and control theory. It is also used for solving linear systems of equations, such as those that arise in linear regression tasks. The SVD is also used in a huge variety of unsupervised learning type applications, e.g., understanding gene expression data (9, 10), the structure of chemical reaction networks (4), in process control applications (11), and analysis of various type of human centered networks (12, 13).

Singular value decomposition is a special sort of eigendecomposition, i.e., decomposition of the matrix products $\mathbf{A}\mathbf{A}^\top$ and $\mathbf{A}^\top\mathbf{A}$. Thus, we could (theoretically) use QR-iteration to compute the SVD, see (14) for a discussion of various (better) approaches to computing the SVD. The columns of \mathbf{U} (left-singular vectors) are the eigenvectors of $\mathbf{A}\mathbf{A}^\top$, while the columns of \mathbf{V} (right singular vectors) are the eigenvectors of $\mathbf{A}^\top\mathbf{A}$. Finally, the singular values σ_i are the square roots of the eigenvalues λ_i , i.e., $\sigma_i = \sqrt{\lambda_i}$ of the matrix product(s) $\mathbf{A}\mathbf{A}^\top$ or $\mathbf{A}^\top\mathbf{A}$. Thus, there is a direct relationship between the eigenvalue decomposition of a matrix \mathbf{A} and its singular value decomposition.

4 Summary

In this lecture we introduce the stoichiometric matrix, a digital representation of the metabolic network, and discuss some of its properties. We also introduce the eigendecomposition of a matrix, a powerful tool for analyzing data and systems in many areas of mathematics, engineering, and physics. We discuss the power iteration method for computing the largest eigenvalue/eigenvector pair of a matrix, and the QR iteration algorithm for computing all the eigenvalues and eigenvectors of a matrix. Finally, we introduce the singular value decomposition (SVD) method, a matrix factorization technique that is based on the eigendecomposition of a matrix. These techniques can be applied to analyze the structure of a matrix, compute the rank of a matrix, and solve linear systems of equations, among other applications.

References

1. Kanehisa M, Furumichi M, Sato Y, Matsuura Y, Ishiguro-Watanabe M. KEGG: biological systems database as a model of the real world. *Nucleic Acids Research*. 2024;53(D1):D672–D677. doi:10.1093/nar/gkae909.
2. Karp PD, Billington R, Caspi R, Fulcher CA, Latendresse M, Kothari A, et al. The BioCyc collection of microbial genomes and metabolic pathways. *Brief Bioinform*. 2019;20(4):1085–1093. doi:10.1093/bib/bbx085.
3. Norsigian CJ, Pusarla N, McConn JL, Yurkovich JT, Dräger A, Palsson BO, et al. BiGG Models 2020: multi-strain genome-scale models and expansion across the phylogenetic tree. *Nucleic Acids Research*. 2019;48(D1):D402–D406. doi:10.1093/nar/gkz1054.
4. Famili I, Palsson BO. Systemic metabolic reactions are obtained by singular value decomposition of genome-scale stoichiometric matrices. *J Theor Biol*. 2003;224(1):87–96. doi:10.1016/s0022-5193(03)00146-2.
5. Brezinski C, Redivo-Zaglia M. The PageRank Vector: Properties, Computation, Approximation, and Acceleration. *SIAM Journal on Matrix Analysis and Applications*. 2006;28(2):551–575. doi:10.1137/050626612.
6. Francis JGF. The QR Transformation A Unitary Analogue to the LR Transformation—Part 1. *The Computer Journal*. 1961;4(3):265–271. doi:10.1093/comjnl/4.3.265.
7. Francis JGF. The QR Transformation—Part 2. *The Computer Journal*. 1962;4(4):332–345. doi:10.1093/comjnl/4.4.332.
8. Stewart GW. On the Early History of the Singular Value Decomposition. *SIAM Review*. 1993;35(4):551–566. doi:10.1137/1035134.
9. Alter O, Brown PO, Botstein D. Singular value decomposition for genome-wide expression data processing and modeling. *Proc Natl Acad Sci U S A*. 2000;97(18):10101–10106. doi:10.1073/pnas.97.18.10101.
10. Alter O, Golub GH. Singular value decomposition of genome-scale mRNA lengths distribution reveals asymmetry in RNA gel electrophoresis band broadening. *Proceedings of the National Academy of Sciences*. 2006;103(32):11828–11833. doi:10.1073/pnas.0604756103.
11. Moore C. Application of Singular Value Decomposition to the Design, Analysis, and Control of Industrial Processes. In: 1986 American Control Conference; 1986. p. 643–650.
12. Sastry CS, Rawat S, Pujari AK, Gulati VP. Network traffic analysis using singular value decomposition and multiscale transforms. *Information Sciences*. 2007;177(23):5275–5291. doi:https://doi.org/10.1016/j.ins.2006.07.007.
13. Palacios P, Castro A, Azurdia-Meza C, Estevez C. SVD detection analysis in cognitive mobile radio networks. In: 2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN); 2017. p. 222–224.
14. Cline AK, Dhillon IS. Computation of the Singular Value Decomposition; 2006. Available from: <https://api.semanticscholar.org/CorpusID:124233634>.