

Lecture 2a: Eigendecomposition of Data and Systems

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

Topics

- **Eigendecomposition** is a fundamental concept in linear algebra that decomposes a matrix into its constituent parts, the eigenvectors and eigenvalues. Eigendecomposition is widely used in many areas of mathematics, engineering, and physics to analyze data and systems.
- **QR decomposition** is a factorization of a matrix into an orthogonal matrix and an upper triangular matrix. The QR decomposition is useful for solving linear systems of equations, and for computing the eigenvalues and eigenvectors of a matrix (using the QR iteration algorithm), among other applications.
- **Gram-Schmidt orthogonalization** is a procedure that generates a set of mutually orthogonal vectors starting from a set of linearly independent vectors. The Gram-Schmidt orthogonalization procedure is used to compute the QR decomposition of a matrix.

1 Introduction

The eigendecomposition of a matrix is a fundamental concept in linear algebra. In this lecture, we will discuss the eigendecomposition of a matrix, and how it can be used to analyze data and systems in unsupervised machine learning. Eigendecomposition allows us to decompose a matrix into its constituent parts, the eigenvectors and eigenvalues, which can be used to understand the structure of the data or the system represented by the matrix.

- **Solution of Linear Differential Equations:** Eigenvectors form a set of linearly independent solutions, while eigenvalues determine the stability of these solutions.
- **Structural Analysis:** Eigenvalues and eigenvectors describe the structural properties of a matrix or a graph. For example, a structure's natural frequencies and vibration modes, e.g., of a building or a bridge.
- **Singular Value Decomposition (SVD):** SVD is commonly used in data analysis, computer vision, image processing, etc, to find the most important features of the dataset.

2 Eigendecomposition

Suppose we have a real 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:

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

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.

2.1 Interpretation of Eigendecomposition

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}_* into the machine, and we get back the same eigenvector \mathbf{v}_* multiplied by a scalar, the eigenvalue λ_* . 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, if the matrix \mathbf{A} is symmetric, and all entries are positive, then all the eigenvalues will be real-valued, and the eigenvectors will be orthogonal (handy properties, as we shall soon see). Finally, another interpretation is that eigenvectors represent the most critical directions in the data or system, and eigenvalues (or functions of them) represent their importance.

2.2 Symmetric Real Matrices

The eigendecomposition of a symmetric real matrix $\mathbf{A} \in \mathbb{R}^{m \times m}$ has some special properties. First, the eigenvalues of a symmetric real matrix are real-valued. Next, the eigenvectors of a symmetric real matrix are orthogonal, i.e., $\langle v_i, v_j \rangle = 0$. This means that the eigenvectors of a real-symmetric matrix form an orthogonal basis for the space spanned by the matrix \mathbf{A} . Further, we can make that basis orthonormal by normalizing the eigenvectors, i.e., $\hat{v}_i = v_i / \|v_i\|$ where $\|v_i\|$ is the norm (length) of the eigenvector v_i .

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 algorithm, a more robust method to find all a matrix's eigenvalues and eigenvectors.

3.1 QR decomposition

The QR 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 (1, 2), factors a 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 $\mathbf{R} \in \mathbb{R}^{n \times m}$:

$$\mathbf{A} = \mathbf{QR} \quad (3)$$

where $\mathbf{Q}^T = \mathbf{Q}^{-1}$. The QR decomposition is useful for solving linear systems of equations, computing the eigenvalues and eigenvectors of a matrix, and for finding the least squares solution of an overdetermined system of equations.

Gram-Schmidt Orthogonalization

To compute the QR decomposition of a matrix, we can use the Gram-Schmidt process. In principle, Gram-Schmidt orthogonalization generates a set of mutually orthogonal vectors $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n$ starting from a set of linearly independent vectors $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$ using the procedure:

$$\mathbf{q}_k = \mathbf{x}_k - \sum_{i=1}^{k-1} \frac{\langle \mathbf{q}_i, \mathbf{x}_k \rangle}{\langle \mathbf{q}_i, \mathbf{q}_i \rangle} \mathbf{q}_i, \quad k = 1, \dots, n \quad (4)$$

where $\langle \cdot, \cdot \rangle$ denotes an inner product. To compute the columns of \mathbf{Q} we set $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$ equal the columns of \mathbf{A} and follow the Gram-Schmidt procedure. The computation of \mathbf{R} is made simple by exploiting the orthogonality of \mathbf{Q} , i.e., $\mathbf{Q}^{-1} = \mathbf{Q}^T$ which yields:

$$\mathbf{R} = \mathbf{Q}^T \mathbf{A} \quad (5)$$

The computation of the QR decomposition using the Gram-Schmidt orthogonalization procedure is shown in Algorithm 1.

Algorithm 1 QR decomposition using Classical Gram-Schmidt Orthogonalization

```

1: Input: Matrix  $\mathbf{A} \in \mathbb{R}^{n \times m}$ 
2: Output: Orthogonal matrix  $\mathbf{Q} \in \mathbb{R}^{n \times n}$  and upper triangular matrix  $\mathbf{R} \in \mathbb{R}^{n \times m}$ 
3:  $n, m \leftarrow$  dimensions of  $\mathbf{A}$ 
4:  $\mathbf{Q} \leftarrow$  zeros matrix of size  $n \times m$ 
5:  $\mathbf{Q}(:, 1) \leftarrow \mathbf{A}(:, 1) / \|\mathbf{A}(:, 1)\|$ 

6: for  $k = 2$  to  $m$  do
7:    $\mathbf{Q}(:, k) \leftarrow \mathbf{A}(:, k)$ 
8:   for  $j = 1$  to  $k - 1$  do
9:      $\mathbf{Q}(:, k) \leftarrow \mathbf{Q}(:, k) - \langle \mathbf{Q}(:, j), \mathbf{A}(:, k) \rangle \mathbf{Q}(:, j)$ 
10:  end for
11:   $\mathbf{Q}(:, k) \leftarrow \mathbf{Q}(:, k) / \|\mathbf{Q}(:, k)\|$ 
12: end for
13:  $\mathbf{R} \leftarrow \mathbf{Q}^T \mathbf{A}$ 
```

The cost of the Gram-Schmidt orthogonalization procedure is $\mathcal{O}(nm^2)$ floating point operations (flops) for an $n \times m$ matrix \mathbf{A} (3). However, in practice, the Gram-Schmidt orthogonalization procedure as shown *will often fail to generate mutually orthogonal vectors* because of rounding errors; we will address this issue subsequently when we discuss the *Modified Gram-Schmidt Orthogonalization* procedure which is shown in Algorithm 2.

Algorithm 2 QR decomposition using Modified Gram-Schmidt Orthogonalization

-
- 1: **Input:** Matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$
 - 2: **Output:** Orthogonal matrix $\mathbf{Q} \in \mathbb{R}^{n \times n}$ and upper triangular matrix $\mathbf{R} \in \mathbb{R}^{n \times m}$
 - 3: $n, m \leftarrow$ dimensions of \mathbf{A}
 - 4: $\mathbf{Q} \leftarrow$ zeros matrix of size $n \times m$
-

Aside: Solution of Linear Systems of Equations

In addition to computing the eigendecomposition of a matrix, another handy applications of QR decomposition is to solve the linear systems of equations $\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$. Once \mathbf{A} has been factored into the product of \mathbf{Q} and \mathbf{R} :

$$\mathbf{QR} \cdot \mathbf{x} = \mathbf{b} \quad (6)$$

we multiply both sides by \mathbf{Q}^T :

$$\mathbf{R} \cdot \mathbf{x} = \mathbf{Q}^T \cdot \mathbf{b} \quad (7)$$

where $\mathbf{Q}^T \cdot \mathbf{Q} = \mathbf{I}$. Because \mathbf{R} is an upper triangular matrix, we can solve the linear system of equations using back substitution. Back substitution is an algorithm for solving a system of linear equations whose matrix is upper triangular. Suppose we have an $n \times n$ system ($n \geq 2$) of equations which is upper triangular and non-singular of the form:

$$\mathbf{U}\mathbf{x} = \mathbf{b}$$

where \mathbf{U} is an upper triangular matrix and \mathbf{b} is a column vector. Then, the solution of to this system is given by:

$$\begin{aligned} x_n &= \frac{b_n}{u_{nn}} \\ x_i &= \frac{1}{u_{ii}} \left(b_i - \sum_{j=i+1}^n u_{ij}x_j \right) \quad i = n-1, \dots, 1 \end{aligned}$$

where $u_{ii} \neq 0$. Back substitution requires $\mathcal{O}(n^2)$ floating point operations (flops).

3.2 The QR Iteration Algorithm

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 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 :

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

where \mathbf{Q}_k is an orthogonal matrix, i.e., $\mathbf{Q}_k^T = \mathbf{Q}_k^{-1}$ 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.

4 Summary and Conclusions

In this lecture, we discussed the eigendecomposition of a matrix and how it can be used to analyze data and systems in the context of unsupervised machine learning. Eigendecomposition gives us a way to decompose a matrix into its constituent parts, the eigenvectors and eigenvalues, which can be used to understand the structure of the data or the system represented by the matrix. We introduced the QR decomposition, which is a factorization of a matrix into an orthogonal matrix and an upper triangular matrix. We introduced the Gram-Schmidt orthogonalization procedure, which is used to compute the QR decomposition of a matrix. Gram-Schmidt orthogonalization generates a set of mutually orthogonal vectors starting from a set of linearly independent vectors. However, the Gram-Schmidt orthogonalization procedure may fail to generate mutually orthogonal vectors because of rounding errors. To fix this issue, we introduced the Modified Gram-Schmidt orthogonalization procedure, which is a more stable version of the Gram-Schmidt orthogonalization procedure. Finally, we introduced the QR iteration algorithm, which is an iterative method that computes the eigenvalues and eigenvectors of a matrix. The QR iteration algorithm estimates the eigenvalues and eigenvectors of a square matrix by repeatedly applying the QR decomposition to the matrix.

References

1. 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.
2. Francis JGF. The QR Transformation—Part 2. The Computer Journal. 1962;4(4):332–345. doi:10.1093/comjnl/4.4.332.
3. Golub GH, van Loan CF. Matrix Computations. 4th ed. JHU Press; 2013. Available from: <http://www.cs.cornell.edu/cv/GVL4/golubandvanloan.htm>.