Contents

Chapter 1

Projections

The projection of a vector $\mathbf{a}$ onto another vector $\mathbf{b}$ is a vector that lies along $\mathbf{b}$ and represents the component of $\mathbf{a}$ in the direction of $\mathbf{b}$. This projection can be computed using the dot product of the two vectors.

Given vectors $\mathbf{a}$ and $\mathbf{b}$, the projection of $\mathbf{a}$ onto $\mathbf{b}$, denoted as $\mathbf{proj}_{\mathbf{b}}(\mathbf{a})$, can be calculated as follows:

$$\mathbf{proj}_{\mathbf{b}}(\mathbf{a}) = \left(\frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{b}\|^2}\right)\mathbf{b}$$

where $\cdot$ denotes the dot product and $|\mathbf{b}|$ represents the magnitude (or length) of vector $\mathbf{b}$.

The numerator $\mathbf{a} \cdot \mathbf{b}$ measures the extent to which $\mathbf{a}$ and $\mathbf{b}$ are aligned with each other. Dividing this by $|\mathbf{b}|^2$ scales the projection to ensure it represents the correct length along $\mathbf{b}$.

Finally, multiplying the scaled value with $\mathbf{b}$ gives us the projection vector itself.

In summary, the dot product is used to determine the alignment between two vectors, and by appropriately scaling one vector and multiplying it with the other vector, we can obtain the projection of one vector onto the other.

Chapter 2

The Gram-Schmidt Process

The Gram-Schmidt process is a method for orthonormalizing a set of vectors in an inner product space, most commonly the Euclidean space $\mathbb{R}^n$. The process takes a finite, linearly independent set $S = \{v_1, v_2, \ldots, v_k\}$ for $k \leq n$, and generates an orthogonal set $S' = \{u_1, u_2, \ldots, u_k\}$ that spans the same k-dimensional subspace of $\mathbb{R}^n$ as S.

2.1 Process

Given a set of vectors $S = \{v_1, v_2, \ldots, v_k\}$, the Gram-Schmidt process is as follows:

1. Let $u_1 = v_1$.
2. For $j = 2, 3, \ldots, k$:
   (a) Let $w_j = v_j - \sum_{i=1}^{j-1} \frac{\langle v_j, u_i \rangle}{\langle u_i, u_i \rangle} u_i$
   (b) Let $u_j = w_j$

Here, $\langle ., . \rangle$ denotes the inner product.

2.2 Orthonormalization

The set of vectors $S' = \{u_1, u_2, \ldots, u_k\}$ obtained from the process above is orthogonal, but not necessarily orthonormal. To form an orthonormal set, we simply need to normalize each vector u_i to unit length. That is, $u_i' = \frac{u_i}{\|u_i\|}$, where $\|.\|$ denotes the norm (or length) of a vector.

Chapter 3

Eigenvectors and Eigenvalues

In linear algebra, eigenvalues and eigenvectors are a way of breaking down matrices that can simplify many calculations and enable us to understand various properties of the matrix. They are widely used in physics and engineering for stability analysis, vibration analysis, and many other applications.

3.1 Definition

Given a square matrix A, a non-zero vector v is an eigenvector of A if multiplying A by v results in a scalar multiple of v, i.e.,

$$Av = \lambda v \tag{3.1}$$

where λ is a scalar known as the eigenvalue corresponding to the eigenvector v.

3.2 Finding Eigenvalues and Eigenvectors

The eigenvalues of a matrix A can be found by solving the characteristic equation given by:

$$\det(A - \lambda I) = 0 \tag{3.2}$$

where $\det(.)$ denotes the determinant, I is the identity matrix of the same size as A, and λ is a scalar.

Once the eigenvalues are found, the corresponding eigenvectors can be found by plugging each eigenvalue back into the equation $Av = \lambda v$, and solving for v.

3.3 Example

For a 2×2 matrix $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$, the characteristic equation is given by:

$$(a - \lambda)(d - \lambda) - bc = 0 \tag{3.3}$$

Solving this equation will give the eigenvalues. Substituting each eigenvalue back into the equation $Av = \lambda v$ will give the corresponding eigenvectors.

Chapter 4

Singular Value Decomposition

Singular Value Decomposition (SVD) is a powerful mathematical technique widely used in numerical linear algebra, data analysis, and machine learning. It provides a way to break down a matrix into simpler, constituent parts, making it easier to calculate and understand.

4.1 Definition

For any $m \times n$ matrix A, the singular value decomposition is given by

$$A = U\Sigma V^T \tag{4.1}$$

where:

- U is an $m \times m$ matrix, and its columns are the eigenvectors of AA^T. These are the left singular vectors of A.

- V is an $n \times n$ matrix, and its columns are the eigenvectors of A^TA. These are the right singular vectors of A.
- Σ is an $m \times n$ diagonal matrix, and its non-zero elements are the square roots of the eigenvalues of both A^TA and AA^T. These are the singular values of A.

It's worth noting that the singular values along the diagonal of Σ are arranged in descending order, and U and V are orthogonal matrices, meaning $U^TU = I$ and $V^TV = I$.

4.2 Applications of SVD

SVD has numerous applications:

- It's used in machine learning and data science to perform dimensionality reduction, particularly through a technique known as Principal Component Analysis (PCA).
- In numerical linear algebra, SVD is used to solve linear equations and compute matrix inverses in a more numerically stable way.
- It's used in image compression, where low-rank approximations of an image matrix provide a compressed version of the original image.

4.3 Doing SVD with numpy

```python
import numpy as np

# Initialize a 3x3 matrix
A = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])

# Perform singular value decomposition
U, S, VT = np.linalg.svd(A)

print("U:\n", U)
print("S:\n", S)
print("VT:\n", VT)

# To check if the decomposition is correct we can rebuild the original matrix:
S = np.diag(S)

A_rebuilt = U.dot(S.dot(VT))

print("Rebuilt A:\n", A_rebuilt)
```

Appendix A

Answers to Exercises

Index