

Week 2 Lecture Brief

Solving Linear Equations

This lecture provides a detailed explanation of solving linear systems of equations where the number of equations may differ from the number of variables. Key points include:

1. **More Equations than Variables (Overdetermined System):**
 - Inconsistent equations can be solved using **least squares**, minimizing the sum of squared errors.
 - The formula used is $x = (A^T A)^{-1} A^T b$.
 - Example: When solving an overdetermined system, if direct solutions don't exist, least squares can find an approximate solution minimizing errors.
2. **More Variables than Equations (Underdetermined System):**
 - Infinite solutions exist; a unique solution can be selected by **minimizing** $x^T x$, which chooses the solution closest to the origin.
 - The concept of **minimum norm solution** is used to find the optimal solution.
3. **Singular or Non-Full Rank Systems:**
 - The **Moore-Penrose pseudoinverse** is introduced to generalize the solution for all cases, including rectangular matrices or systems without a unique solution.
4. **Implementation in R:**
 - You can use the **pseudo-inverse** function `ginv(A)` from the `MASS` package to compute solutions, especially when handling systems with inconsistent or infinite solutions.

This approach covers all types of systems, giving consistent and interpretable results through optimization techniques and pseudoinverses.

Solving Linear Equations (Continued)

The lecture outlines methods for solving systems of linear equations, focusing on cases where:

1. **More equations than variables (overdetermined system):** The optimization approach finds a least squares solution, minimizing the sum of squared errors. This is useful when the system has no exact solution.
 - Example 1: An inconsistent system results in an approximate solution via optimization.
 - Example 2: A system with consistent equations provides an exact solution.
2. **More variables than equations (underdetermined system):** Infinite solutions are possible. To find a unique solution, we use the optimization technique that minimizes

$x^T x x^T x x^T x$, effectively choosing the solution closest to the origin (minimum norm solution).

3. **Singular cases and generalization:** The lecture discusses the Moore-Penrose pseudoinverse to handle all cases (square, overdetermined, and underdetermined), which can be computed using techniques like singular value decomposition.

In R, the generalized inverse can be calculated using the `g-inverse` function to handle these cases. The interpretation of solutions (least square, minimum norm) is key to understanding the outcomes.

Linear Algebra - Distance, Hyperplanes and Halfspaces, Eigenvalues, Eigenvectors

Vectors and Basis Vectors

Vectors as Points in Space

Vectors can be seen as points in a multi-dimensional space, where each component corresponds to a coordinate axis. The Euclidean distance between a vector and the origin is given by:

$$\sqrt{x_1^2 + x_2^2 + \dots + x_n^2}$$

Vectors Between Points

The vector between two points x_1 and x_2 is defined as $x_2 - x_1$. The length of this vector is:

$$\sqrt{(x_2 - x_1)^T (x_2 - x_1)}$$

Unit vectors are vectors with magnitude 1 in a particular direction. Any vector \vec{a} can be written as $|\vec{a}| \hat{a}$, where \hat{a} is the unit vector in the direction of \vec{a} .

Orthogonal Vectors

Two vectors are orthogonal if their dot product is 0. Orthonormal vectors are orthogonal vectors that also have a magnitude of 1.

Basis Vectors

Basis vectors are a set of linearly independent vectors that span the entire space. Any vector in the space can be written as a unique linear combination of the basis vectors.

vectors are not unique, the number of vectors in each basis set must be the same for a given space.

For example, in \mathbb{R}^4 , the vectors $(1,2,3,4)$ and $(4,1,2,3)$ form a basis, and any vector in \mathbb{R}^4 can be written as a linear combination of these two vectors.

Linear Algebra - Distance, Hyperplanes and Halfspaces, Eigenvalues, Eigenvectors (Continued 1)

In this lecture, we've explored some key concepts of linear algebra relevant to data science:

1. **Equations in Space:** A single linear equation in two dimensions represents a line, and two equations represent a point where two lines intersect. This concept extends to higher dimensions: one equation in three-dimensional space represents a plane, two represent a line, and three equations can represent a point.
2. **Geometric Interpretation:** A linear equation of the form $\mathbf{n}^T \mathbf{x} + b = 0$ represents a hyperplane, where \mathbf{n} is a normal vector perpendicular to this hyperplane. In 2D, this describes a line, and in 3D, a plane. The normal vector plays a critical role in understanding the orientation of these geometric objects.
3. **Projections:** Projection is the process of finding the closest point on a subspace (like a plane) to a given point. In data science, projecting high-dimensional data onto a lower-dimensional space is crucial, especially in techniques like Principal Component Analysis (PCA). Mathematically, projection involves finding the best approximation of a vector using basis vectors of the subspace.
4. **Orthogonal Vectors and Projections:** When projecting onto a plane defined by two orthogonal vectors, the formula for projection simplifies due to the orthogonality of the basis vectors. This helps us efficiently represent vectors in lower dimensions.
5. **General Projections:** Even when basis vectors are not orthogonal, we can project vectors onto a subspace spanned by linearly independent vectors, using matrix operations to compute the projection.

These concepts are foundational to many data science algorithms, and mastering them will help you understand how data is represented, transformed, and analyzed in high-dimensional spaces.

Linear Algebra - Distance, Hyperplanes and Halfspaces, Eigenvalues, Eigenvectors (Continued 2)

Key Concepts in Linear Algebra for Data Science:

1. **Hyperplanes:** A hyperplane is a geometric entity whose dimension is one less than that of its surrounding space. In 3D space, a hyperplane is a 2D plane, and in 2D space, it becomes a 1D line. A hyperplane can be described by the equation:

$$X_1n_1 + X_2n_2 + \cdots + X_nn_n + b = 0$$

When the hyperplane passes through the origin, it becomes a subspace.

2. **Half-spaces:** A half-space is one side of a hyperplane. In a classification context, these are used to separate data points into different classes. For example, in a binary classification problem, a hyperplane can separate data points into two groups, each in a different half-space.
3. **Classification Using Hyperplanes:** In data science, hyperplanes help in classification by distinguishing between different data points. For example, you might classify data points based on whether they lie in one half-space or the other.
4. **Eigenvalues and Eigenvectors:** These are essential concepts in linear algebra. For a square matrix A , an eigenvector x and corresponding eigenvalue λ satisfy the equation:

$$Ax = \lambda x$$

5. The eigenvector's direction remains unchanged after applying A , while the eigenvalue scales the vector. Eigenvalues help in data compression, noise reduction, and dimensionality reduction.

Linear Algebra - Distance, Hyperplanes and Halfspaces, Eigenvalues, Eigenvectors (Continued 3)

This lecture highlights the key connections between eigenvalues, eigenvectors, and symmetric matrices in the context of data science:

1. **Eigenvalue-Eigenvector Equation:** The equation $Ax = \lambda x$ leads to a characteristic polynomial of degree n , implying real or complex eigenvalues. However, for symmetric matrices ($A = A^T$), eigenvalues are always real, and eigenvectors are also real.
2. **Distinct Eigenvalues:** If all eigenvalues are distinct, the matrix has n linearly independent eigenvectors. In contrast, repeated eigenvalues may yield fewer independent eigenvectors, though symmetric matrices always guarantee n independent eigenvectors, even if eigenvalues repeat.
3. **Symmetric Matrices in Data Science:** Covariance matrices and other forms like AA^T or $A^T A$ are symmetric and feature prominently in data science. These matrices have real, non-negative eigenvalues (≥ 0).
4. **Null Space and Column Space:** Eigenvectors corresponding to zero eigenvalues lie in the null space of matrix A , while eigenvectors corresponding to non-zero eigenvalues span the column space.
5. **Full Rank and Eigenvalues:** If none of the eigenvalues are zero, the matrix is full-rank, implying no non-trivial null space.

The lecture stresses the importance of these concepts in data science, particularly for algorithms like Principal Component Analysis (PCA), which heavily relies on eigenvalue-eigenvector connections.