A VECTOR-SPACE AND PROBABILISTIC INTERPRETATION OF LEARNING-TO-RANK IN MODERN SEARCH SYSTEMS

Acknowledgement

I extend my sincere gratitude to all those who supported me throughout this project.

I am deeply thankful to **Dr. Pratap Saha**, Department of Mathematics, for their invaluable guidance and encouragement. Their insights into connecting pure mathematical theory with applied domains shaped the direction of this work. Their patience in reviewing multiple drafts and providing constructive feedback was instrumental in maintaining the rigor and clarity of this project.

I express my profound respect and gratitude to **Prof. Mrityunjay Kumar Dubey** (Retd.), whose teachings during my earlier years laid the foundation for my mathematical thinking. Though he retired recently, his influence on my approach to rigorous problem-solving and mathematical exposition remains evident throughout this work.

I express my appreciation to the faculty members of the Department of Mathematics for providing a strong foundation in linear algebra and probability theory during my undergraduate studies. The courses on vector spaces, inner product theory, and statistical inference directly enabled the mathematical treatment presented in this work.

I am particularly grateful to my cousin brother **Sonu Kumar** for his constant motivation and invaluable assistance in finding resources related to the mathematical aspects of this project. His encouragement during challenging phases of research and writing kept me focused and determined.

I extend special thanks to **Mohit Durge** for inspiring me to explore the intersection of search systems and machine learning. His insights into practical applications of information retrieval motivated me to bridge pure mathematics with real-world computational problems.

I acknowledge the developers and contributors of open-source libraries including scikit-learn, NumPy, and Matplotlib, which facilitated the experimental component of this project. The availability of standard datasets like the Cranfield Collection enabled reproducible evaluation of retrieval methods.

Finally, I am grateful to my family for their unwavering support and understanding during the intensive periods of writing and experimentation. Their encouragement sustained my motivation throughout this endeavor.

Abstract

This project explores the mathematical foundations of modern search systems through the lens of vector spaces and probability theory. Information retrieval has evolved from simple keyword matching to sophisticated ranking algorithms that understand relevance in a mathematical framework.

We begin by examining the classical vector space model, where documents and queries are represented as vectors in high-dimensional space. This geometric perspective allows us to measure similarity using inner products and cosine metrics. The model provides an intuitive yet rigorous foundation for understanding how search engines match queries to documents.

Next, we investigate probabilistic models of relevance. These models treat relevance as a random variable and apply Bayes theorem to rank documents by their probability of being relevant. The Okapi BM25 algorithm emerges from this framework, combining term frequency statistics with probabilistic principles. This approach addresses limitations of purely geometric models by incorporating statistical evidence.

The project then examines learning-to-rank methods, which use machine learning to optimize ranking functions. These methods learn from labeled training data to predict relevance scores. We study three main approaches: pointwise methods that treat ranking as regression, pairwise methods that learn from document comparisons, and listwise methods that optimize entire result lists.

Modern search systems use dense vector embeddings to capture semantic meaning beyond exact word matches. We analyze how dimensionality reduction techniques like Principal Component Analysis preserve essential information while making computation tractable. The eigenvalue decomposition underlying PCA reveals the geometric structure of document collections.

Our experimental study compares classical methods like TF-IDF with probabilistic models like BM25 and modern embedding approaches. We evaluate these methods on metrics such as normalized discounted cumulative gain and mean average precision. The results demonstrate strengths and weaknesses of each approach across different query types.

This work bridges pure mathematical theory with practical information retrieval. We provide rigorous derivations of key algorithms while explaining their real-world performance. The project shows how linear algebra and probability theory form the backbone of modern search technology, offering insights valuable for both mathematical understanding and system design.

CHAPTER 1: INTRODUCTION AND THEORETICAL FOUNDATIONS

1.1 Introduction & Objectives

1.1.1 Information Retrieval vs Database Systems

Information retrieval and database systems solve fundamentally different problems despite both dealing with data storage and access. Database systems handle structured data with well-defined schemas. They respond to precise queries with exact matches. For example, a query like "SELECT * FROM students WHERE age > 20" returns all records meeting the exact condition. There is no ambiguity in what constitutes a correct answer.

Information retrieval systems deal with unstructured text data. They interpret natural language queries and return ranked results based on relevance. A user searching for "machine learning algorithms" does not expect exact string matches. Instead, they want documents that discuss the topic meaningfully. The system must understand synonyms, context, and semantic relationships. It must rank thousands of potentially relevant documents by estimated usefulness.

Aspect	**Information Retrieval (IR)**	**Database Systems (DB)**
Primary Goal	Retrieve **relevant** information	Retrieve **exact** data
Query Type	Keywords, natural language, vague queries	Structured queries (SQL)
Matching	Approximate, similarity-based	Exact match
Result Ordering	Ranked by relevance score	Unordered or explicitly ordered
Data Structure	Unstructured / semi-structured text	Structured tables with schema
Relevance	Probabilistic / graded	Boolean (true/false)
User Intent	Implicit and uncertain	Explicit and well-defined
Typical Use	Search engines, document search, ML	Transactions, records management
Failure Mode	Returns less relevant results	Returns empty or exact results only

Table 1.1 Information Retrieval vs Database Systems

1.1.2 Evolution of Search Systems (1960s to Present)

The history of search systems reflects increasing mathematical sophistication. In the 1960s, researchers developed the first automated retrieval systems using Boolean logic. These systems treated documents as sets of keywords. Queries combined terms with AND, OR, and NOT operators. The SMART system at Cornell University pioneered vector space representations in the 1970s.

The 1980s brought probabilistic models that treated relevance as a statistical phenomenon. Researchers applied Bayes theorem to estimate relevance probabilities. The Binary Independence Model provided theoretical foundations. These ideas culminated in the Okapi BM25 algorithm in the 1990s, which remains influential today.

The web's emergence in the 1990s created new challenges. PageRank introduced link analysis as a relevance signal. Commercial search engines combined multiple ranking factors. The 2000s saw machine learning enter the field. Learning-to-rank methods trained models on human relevance judgments. They optimized ranking functions using gradient descent.

Modern systems use deep neural networks and transformer architectures. Dense vector embeddings capture semantic meaning. Large language models generate contextual representations. Yet classical mathematical foundations remain essential. Understanding cosine similarity, probability theory, and optimization principles is crucial for advancing search technology. This evolution demonstrates how mathematical rigor drives practical progress.

1.1.3 Keyword-Based Retrieval and Its Limitations

Early search systems relied on exact keyword matching. A document was relevant if it contained query terms. The system counted term occurrences and returned documents exceeding a threshold. This approach had severe limitations that motivated more sophisticated models.

The vocabulary mismatch problem occurs when relevant documents use different words than the query. A search for "automobile" misses documents discussing "car" or "vehicle." Synonyms, related terms, and paraphrases create false negatives. No amount of keyword counting solves this semantic gap.

Keyword matching ignores term importance. Common words like "the" or "is" appear frequently but carry little meaning. Rare technical terms are more discriminative. Simple term counting treats all words equally. This produces poor ranking quality.

Word order and syntax are lost in keyword representations. The queries "dog bites man" and "man bites dog" have identical keyword sets but opposite meanings. Context and relationships between terms matter for understanding intent.

These limitations drove researchers toward vector space models and probabilistic frameworks. Mathematical modeling provides tools to address semantic gaps, weight terms appropriately, and capture relationships between concepts. The subsequent sections develop these solutions rigorously.

1.1.4 Relevance as a Latent Variable

Relevance is not directly observable. We cannot measure it like length or temperature. A document's relevance exists in the user's mind, shaped by their information need and context. This makes relevance a latent variable, a theoretical construct inferred from observable signals.

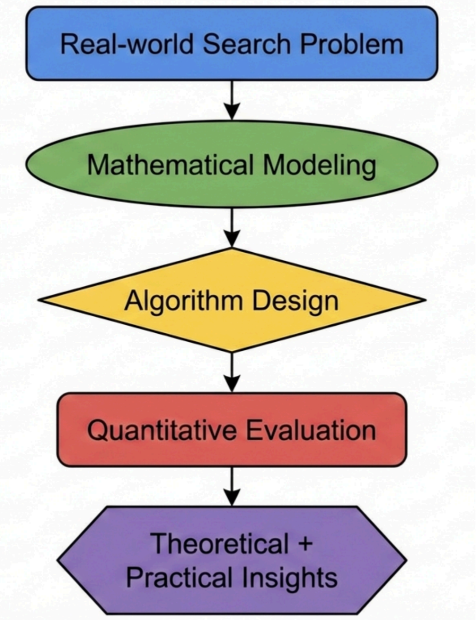
Different users judge relevance differently for the same query and document. A computer science student and a journalist searching "neural networks" seek different information. Relevance depends on expertise, task, and preferences. This subjectivity complicates mathematical modeling.

Treating relevance as a latent variable allows rigorous mathematical treatment. We model it as a random variable with unknown distribution. Observable signals become evidence for inferring this hidden state. Probabilistic models estimate relevance probabilities. Machine learning methods predict relevance scores from features.

The latent variable perspective connects search to broader statistical inference. We apply tools from factor analysis, graphical models, and Bayesian statistics. This framework handles uncertainty and multiple sources of evidence systematically. It provides a principled foundation for ranking algorithms.

1.1.5 Motivation for Mathematical Modeling

Intuition alone cannot scale to modern search systems handling billions of queries. Mathematical models provide precision, speed, and reliability. They allow formal analysis of ranking functions, error bounds, and uncertainty. Vector space, probabilistic, and learning models share foundations in linear algebra and optimization, enabling unified understanding. Mathematical metrics such as precision, recall, and NDCG enable objective evaluation. Analysis guides scalability through efficient approximations and optimization. Overall, mathematical modeling replaces ad-hoc design with principled, reproducible, and transferable solutions.



Flowchart: Motivation Pipeline

1.1.6 Role of Vector Spaces in Machine Learning

Vector spaces provide the geometric foundation for modern machine learning. Every data point becomes a vector in some high-dimensional space. This representation enables powerful mathematical tools from linear algebra.

In information retrieval, documents and queries are vectors. Each dimension corresponds to a term in the vocabulary. A document's vector encodes which terms it contains and how often. This geometric view transforms text into points in space. Similarity becomes measurable through distance and angles.

The inner product captures relatedness between vectors. It combines magnitude and direction information. When normalized, it yields cosine similarity, a standard metric in text analysis. Two documents are similar if their vectors point in similar directions. This geometric intuition makes abstract similarity concrete.

Linear transformations modify vector representations. Dimensionality reduction projects high-dimensional data onto lower-dimensional subspaces. This preserves essential structure while discarding noise. Principal Component Analysis finds optimal projections by solving an eigenvalue problem. The mathematical theory guarantees variance preservation.

Machine learning algorithms exploit vector space structure. Classification finds separating hyperplanes. Clustering groups nearby vectors. Regression fits functions in vector spaces. Neural networks compose linear transformations with nonlinearities. All rely on vector space properties like linearity and metric structure.

This project emphasizes how vector space axioms underpin practical algorithms. We derive key results rigorously from first principles. This demonstrates that sophisticated search systems rest on solid mathematical foundations.

1.1.7 Role of Probability Theory in Ranking

Probability theory handles uncertainty inherent in relevance judgments. We cannot know with certainty whether a document satisfies a user's need. Probabilistic models quantify this uncertainty through probability distributions.

The Probability Ranking Principle states that optimal ranking orders documents by decreasing probability of relevance. This theorem provides a normative goal for retrieval systems. It reduces ranking to probability estimation. If we can estimate relevance probabilities accurately, we can rank optimally.

Bayes theorem connects observable evidence to latent relevance. We observe term frequencies, document lengths, and other features. Bayes theorem inverts conditional probabilities, letting us infer relevance from these observations. This provides a principled inference framework.

Statistical language models treat text generation as a stochastic process. Documents are samples from probability distributions over words. Query likelihood measures how probable a query is under a document's language model. This probabilistic perspective avoids arbitrary weighting schemes.

Maximum likelihood estimation fits model parameters to data. Given relevance judgments, we find parameters maximizing the likelihood of observed labels. This objective function has clear statistical meaning. Optimization methods find parameters that best explain training data.

Probabilistic models enable principled handling of missing data and noise. They quantify confidence through posterior distributions. They combine multiple sources of evidence through probability rules. This project develops these ideas rigorously, showing how probability theory solves core retrieval problems.

1.1.8 Problem Statement

This project addresses the question: How do mathematical frameworks of vector spaces and probability theory enable effective document retrieval and ranking?

We decompose this into specific problems. First, how do we represent unstructured text mathematically while preserving semantic information? Vector space models map documents to points in Euclidean space. But which vector representation is optimal? How do we handle high dimensionality?

Second, how do we define and measure relevance objectively? Relevance is subjective and context-dependent. Probabilistic models treat it as a random variable. But what assumptions are necessary? How do we estimate relevance probabilities from limited data?

Third, how do we learn ranking functions from examples? Machine learning methods optimize ranking quality directly. But which loss functions are appropriate? How do pointwise, pairwise, and listwise approaches compare? What are their theoretical guarantees?

Fourth, how do classical methods like TF-IDF and BM25 relate to modern embedding-based approaches? Are they fundamentally different or variations on common principles? Can we derive them from unified mathematical foundations?

Finally, how do we evaluate ranking quality? Metrics like normalized discounted cumulative gain quantify performance. But what properties should good metrics satisfy? How do we account for position bias and user behavior?

This project provides rigorous mathematical answers to these questions, bridging pure theory with practical algorithm design.

1.1.9 Research Objectives and Scope

The primary objective is to develop a rigorous mathematical treatment of information retrieval that connects classical theory with modern practice. We aim to derive key algorithms from first principles rather than present them as given recipes.

Specifically, we will formalize vector space representations axiomatically. We will prove properties of inner products and norms relevant to similarity measurement. We will derive cosine similarity and show its geometric interpretation. We will analyze dimensionality reduction through eigenvalue decomposition.

We will develop probabilistic models of relevance from measure-theoretic foundations. We will state and prove the Probability Ranking Principle. We will derive the BM25 formula from probabilistic assumptions. We will analyze maximum likelihood estimation for language models.

We will formalize learning-to-rank as an optimization problem. We will present pointwise, pairwise, and listwise approaches with their loss functions. We will discuss convergence properties and sample complexity. We will connect these methods to statistical learning theory.

The scope includes experimental validation on a constructed dataset. We will compare TF-IDF, BM25, and embedding methods empirically. We will measure performance using standard metrics. We will analyze failure modes and discuss practical considerations.

This project excludes neural ranking models and deep learning methods to maintain focus on foundational mathematics. We also exclude distributed systems and implementation details. The emphasis is on mathematical principles that remain relevant regardless of computational platform.

1.1.10 Thesis Organization

This thesis follows a theory-to-practice structure. Each chapter builds on previous mathematical foundations while moving toward practical applications.

Chapter 1 establishes motivation and mathematical preliminaries. It covers linear algebra and probability theory rigorously. These sections prove theorems and develop intuition through geometric interpretations. Readers gain tools needed for subsequent chapters.

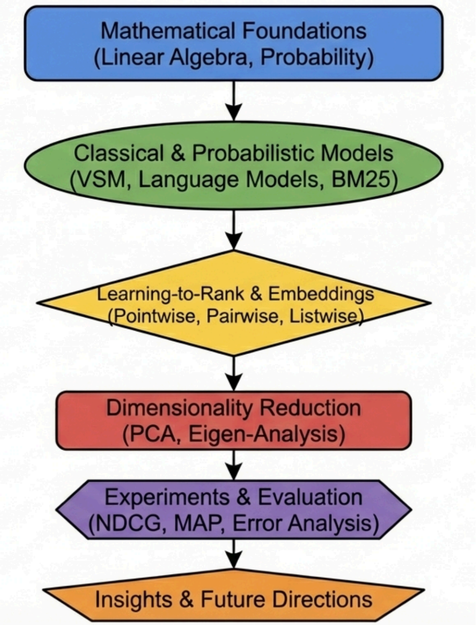
Chapter 2 presents classical and probabilistic retrieval models. It derives the vector space model from inner product spaces. It develops probabilistic models from Bayes theorem. It shows how BM25 emerges from the Binary Independence Model. This chapter connects abstract mathematics to concrete algorithms.

Chapter 3 analyzes data and discusses results. It presents learning-to-rank methods and modern embeddings. It describes dimensionality reduction techniques with mathematical justification. It reports experimental findings and compares different approaches. This chapter demonstrates how theory predicts empirical performance.

The conclusion synthesizes insights from theory and experiments. It discusses limitations and future directions. It reflects on how mathematical rigor enhances our understanding of search systems.

Annexures provide detailed derivations, additional visualizations, and experimental data. They allow interested readers to verify calculations and explore details without interrupting main narrative flow.

This organization serves pure mathematics students entering applied domains. It maintains mathematical rigor while showing relevance to real-world problems. Each section answers the question: How does this theorem enable better search?


Mathematical Foundations
(Linear Algebra, Probability)
Classical & Probabilistic Models
(VSM, Language Models, BM25)
Learning-to-Rank & Embeddings
(Pointwise, Pairwise, Listwise)
Dimensionality Reduction
(PCA, Eigen-Analysis)
Experiments & Evaluation
(NDCG, MAP, Error Analysis)
Insights & Future Directions

Flowchart 1.2 Structure of the Thesis and Knowledge Flow

1.2 Mathematical Foundations

1.2.1 Linear Algebra Foundations

The foundation of linear algebra rests on the algebraic structure called a vector space. This structure formalizes the notion of objects that can be added together and scaled by numbers.

Definition 1.2.1 (Vector Space):
A vector space is a set V along with an addition on V and a scalar multiplication on V such that the following properties hold:

1. **Commutativity:**
   u + v = v + u for all u, v ∈ V.
2. **Associativity:**
   (u + v) + w = u + (v + w) and (ab)v = a(bv) for all u, v, w ∈ V and all a, b ∈ F.
3. **Additive Identity:**
   There exists an element 0 ∈ V such that v + 0 = v for all v ∈ V.
4. **Additive Inverse:**
   For every v ∈ V, there exists w ∈ V such that v + w = 0.
5. **Multiplicative Identity:**
   1v = v for all v ∈ V.
6. **Distributive Properties:**
   a(u + v) = au + av and (a + b)v = av + bv for all a, b ∈ F and all u, v ∈ V.
   (Axler, 2026, Definition 1.20)

In information retrieval, documents are represented as vectors in $\mathbb{R}^n$, where n is the vocabulary size. Each coordinate corresponds to a term weight such as TF-IDF. Coordinate-wise addition and scalar multiplication satisfy the vector space axioms, allowing ranking functions to be expressed using linear algebraic operations such as inner products and norms.

Definition 1.2.2 (Subspace):
A subset U of V is called a subspace of V if U is also a vector space with the same additive identity, addition, and scalar multiplication as on V.
(Axler, 2026, Definition 1.33)

Theorem 1.2.3 (Conditions for a Subspace):
A subset U of V is a subspace of V if and only if U satisfies the following three conditions:

- **Additive Identity:** $0 \in U$
- **Closed under Addition:** $u, w \in U$ implies $u + w \in U$
- **Closed under Scalar Multiplication:** $a \in F$ and $u \in U$ implies $au \in U$
  (Axler, 2026, Result 1.34)

Linear Independence, Span, and Basis Vectors

We now develop concepts that describe how vectors generate and structure vector spaces.

Definition 1.2.4 (Linear Combination):
A linear combination of a list $v_1, ..., v_m$ of vectors in V is a vector of the form
$a_1v_1 + ... + a_mv_m$, where $a_1, ..., a_m \in F$.
(Axler, 2026, Definition 2.2)

Linear combinations describe all vectors obtainable by scaling and adding vectors from a given list. This construction appears throughout linear algebra and machine learning.

Definition 1.2.5 (Span):
The set of all linear combinations of a list of vectors $v_1, ..., v_m$ in V is called the span of $v_1, ..., v_m$, denoted by $\text{span}(v_1, ..., v_m)$. In other words,
$\text{span}(v_1, ..., v_m) = \{a_1v_1 + ... + a_mv_m : a_1, ..., a_m \in F\}$.
The span of the empty list () is defined to be $\{0\}$.
(Axler, 2026, Definition 2.4)

The span represents all vectors reachable through linear combinations. Geometrically, the span of a single nonzero vector is a line through the origin. The span of two non-collinear vectors is a plane. The span grows as we add independent directions.

Theorem 1.2.6 (Span is the Smallest Containing Subspace):
The span of a list of vectors in V is the smallest subspace of V containing all vectors in the list.
(Axler, 2026, Result 2.6)

This theorem confirms that span behaves as expected. Any subspace containing our vectors must contain all their linear combinations. The span is exactly this set and nothing more.

Linear independence identifies minimal spanning sets.

Definition 1.2.7 (Linearly Independent): A list $v_1, ..., v_m$ of vectors in V is called linearly independent if the only choice of $a_1, ..., a_m \in F$ that makes $a_1v_1 + ... + a_mv_m = 0$ is $a_1 = ... = a_m = 0$.

The empty list () is also declared to be linearly independent.
(Axler, 2026, Definition 2.15)

Linear independence means no vector in the list can be expressed as a linear combination of the others. Each vector contributes a new direction. In contrast, linearly dependent lists contain redundancy.

For document vectors, linear independence indicates that documents cover distinct topics. Dependent documents provide overlapping information. Dimensionality reduction techniques remove dependencies to create efficient representations.

Definition 1.2.8 (Basis): A basis of V is a list of vectors in V that is linearly independent and spans V.
(Axler, 2026, Definition 2.26)

A basis provides coordinates for the vector space. Every vector has a unique representation as a linear combination of basis vectors. The coefficients in this representation are the coordinates. Different bases give different coordinate systems for the same space.

In information retrieval, the standard basis uses individual terms as basis vectors. A document's coordinates are its term frequencies. Alternative bases can reveal latent struct ure. Topic models find bases where basis vectors represent topics rather than terms. This change of basis can improve retrieval quality.

Inner Product Spaces

Inner products generalize the dot product from Euclidean space. They provide a notion of angle and length in abstract vector spaces.

Definition 1.2.9 (Inner Product): An inner product on V is a function that takes each ordered pair (u, v) of elements of V to a number $\langle u, v\rangle \in F$ and has the following properties:

- **Positivity:** $\langle v, v\rangle \geq 0$ for all $v \in V$
- **Definiteness:** $\langle v, v\rangle = 0$ if and only if $v = 0$
- **Additivity in First Slot:** $\langle u + v, w\rangle = \langle u, w\rangle + \langle v, w\rangle$ for all $u, v, w \in V$
- **Homogeneity in First Slot:** $\langle av, w\rangle = a\langle v, w\rangle$ for all $a \in F$ and all $v, w \in V$

- **Conjugate Symmetry:** ⟨u, v⟩ = ⟨v, u⟩ *for all u, v ∈ V*
  (Axler, 2026, Definition 6.3)*

For real vector spaces, conjugate symmetry reduces to symmetry: ⟨u, v⟩ = ⟨v, u⟩. This makes the inner product symmetric and bilinear. These properties enable geometric interpretations.

The standard inner product on R^n is the dot product:
$\langle x, y \rangle = x_1 y_1 + x_2 y_2 + \ldots + x_\mathrm{n} y_\mathrm{n}$

Norms and Metrics

Definition 1.2.10 (Norm): For v ∈ V, the norm of v, denoted ||v||, is defined by ||v|| = √⟨v, v⟩.
(Axler, 2026, Definition 6.7)

Theorem 1.2.11 (Cauchy-Schwarz Inequality):
Suppose u, v ∈ V. Then
|⟨u, v⟩| ≤ ||u|| ||v||.
This inequality is an equality if and only if one of u, v is a scalar multiple of the other.
(Axler, 2026, Result 6.14)

This fundamental inequality states that the inner product cannot exceed the product of norms. Equality occurs when vectors are parallel. The proof uses positivity of the inner product applied to carefully chosen vectors.

The Cauchy-Schwarz inequality enables defining angles between vectors. Since $|\langle u, v \rangle| \leq ||u||||v||$,
we have $-1 \leq \langle u, v \rangle / (||u||||v||) \leq 1$ for nonzero vectors.
This ratio lies in the valid range for cosine.

Theorem 1.2.12 (Triangle Inequality):
Suppose u, v ∈ V. Then
||u + v|| ≤ ||u|| + ||v||.
This inequality is an equality if and only if one of u, v is a nonnegative real multiple of the other.
(Axler, 2026, Result 6.17)

Geometrically, this states that the direct path between two points is shorter than any detour. The sum u + v connects the origin to the endpoint of v placed at the tip of u. The inequality says this direct distance does not exceed the sum of individual distances.

These inequalities are essential for proving convergence and bounding errors in numerical algorithms. They ensure that norms behave as expected for length measurements.

Cosine Similarity Derivation from Inner Product

Cosine similarity is the primary similarity measure in information retrieval. We derive it from the inner product.

For nonzero vectors u and v, the Cauchy-Schwarz inequality guarantees
$-1 \leq \langle u, v\rangle/(||u||||v||) \leq 1.$
This ratio has the same range as the cosine function on $[0, \pi]$. We define the angle between vectors using this observation.

Definition 1.2.13 (Angle Between Vectors):
The angle θ between two nonzero vectors u, v ∈ V is defined to be
$\theta = arccos(\langle u, v\rangle/(||u||||v||)),$
where the motivation for this definition comes from the geometric interpretation in R².
(Axler, 2026, Exercise 15 and page 193)

This definition extends the familiar notion of angle from R² to arbitrary inner product spaces. In R², the dot product formula $u \cdot v = ||u||||v||cos\theta$ relates inner product to angle. Our definition ensures this relationship holds in all inner product spaces.

Cosine similarity is the cosine of this angle:
$cos\theta = \langle u, v\rangle/(||u||. ||v||)$

This normalized inner product removes length effects. It measures only directional similarity. Two documents with identical term distributions but different lengths have cosine similarity 1.

Cosine similarity ranges from -1 to 1. A value of 1 means vectors point in the same direction. A value of 0 means orthogonal vectors. A value of -1 means opposite directions. For document vectors with nonnegative entries, cosine similarity ranges from 0 to 1.

The geometric interpretation is clear. Similar documents point in similar directions in term space. The angle between their vectors is small, yielding large cosine. Dissimilar documents point in different directions, yielding small cosine.

This derivation shows cosine similarity follows naturally from inner product structure. It is not an arbitrary choice but the geometrically meaningful way to measure directional similarity in inner product spaces.

Orthogonality and Orthogonal Projections

Orthogonality generalizes perpendicularity to abstract vector spaces.

Definition 1.2.14 (Orthogonal):
Two vectors u, v are called orthogonal if ⟨u, v⟩ = 0.
(Axler, 2026, Definition 6.10)

Definition 1.2.15 (Orthogonal Complement): If U is a subset of V, then the orthogonal complement of U, denoted U⊥, is the set of all vectors in V that are orthogonal to every vector in U:
U⊥ = {v ∈ V : ⟨v, u⟩ = 0 for all u ∈ U}
(Axler, 2026, Definition 6.46)

Theorem 1.2.16 (Basic Properties of Orthogonal Complement):

Suppose U is a finite-dimensional subspace of V. Then:

a) U⊥ is a subspace of V.

b) $0 \perp = V$.

c) $V \perp = 0$.

d) $U \cap U \perp = 0$.

e) If U ⊂ W, then W⊥ ⊂ U⊥.

(Axler, 2026, Result 6.47)

Theorem 1.2.17 (Direct Sum of Subspace and Orthogonal Complement): Suppose U is a finite-dimensional subspace of V. Then V = U ⊕ U⊥.

(Axler, 2026, Result 6.50)

Definition 1.2.18 (Orthogonal Projection):

Suppose U is a finite-dimensional subspace of V. The orthogonal projection of V onto U is the operator $P_U \in L(V)$ defined as follows: For v ∈ V, write v = u + w, where u ∈ U and w ∈ U⊥. Then $P_U v = u$.

(Axler, 2026, Definition 6.55)

In dimensionality reduction, we project high-dimensional document vectors onto low-dimensional subspaces. The projection preserves components along principal directions while discarding noise. This reduces computation while maintaining essential structure.

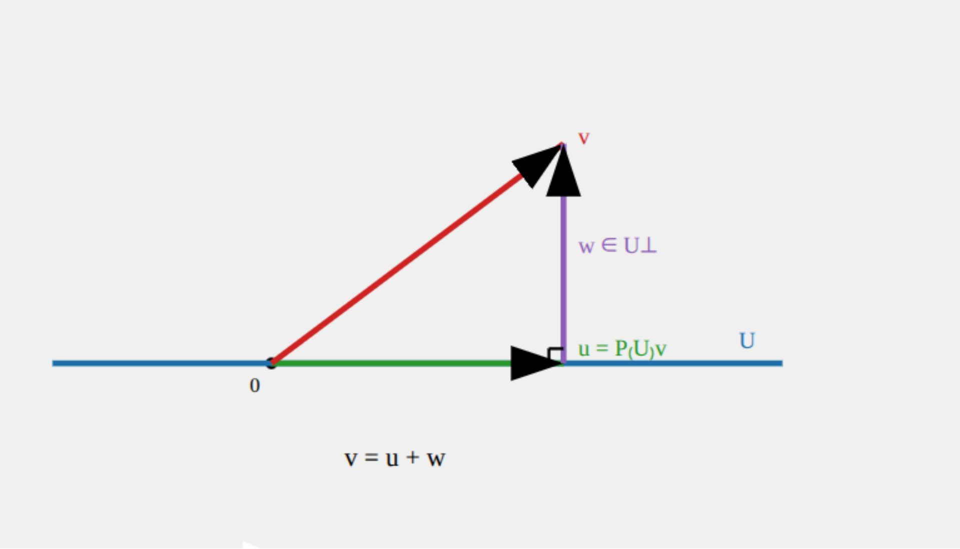


Figure 1.1: Geometric illustration of orthogonal projection showing vector v decomposed into components u ∈ U and w ∈ U⊥, with $P_U v = u$

Definition 1.2.19 (Eigenvalue): Suppose T ∈ L(V). A number λ ∈ F is called an eigenvalue of T if there exists v ∈ V such that v ≠ 0 and Tv = λv.

(Axler, 2026, Definition 5.5)

An eigenvalue is a scaling factor for which the operator acts as pure scaling on some nonzero vector. The operator does not change the direction, only the magnitude.

Definition 1.2.20 (Eigenvector): Suppose T ∈ L(V) and λ is an eigenvalue of T. A vector v ∈ V is called an eigenvector of T corresponding to λ if v ≠ 0 and Tv = λv.

(Axler, 2026, Definition 5.8)

Eigenvectors are the special directions along which the operator acts as scaling. Finding eigenvectors identifies the natural coordinate system for understanding T.

Definition 1.2.21 (Characteristic Polynomial): Suppose V is a complex vector space and $T \in L(V)$. Let λ_1, ..., λ_m denote the distinct eigenvalues of T, with multiplicities d_1, ..., d_m. The polynomial $(z - \lambda_1)^(d_1) \cdot \ldots \cdot (z - \lambda_m)^(d_m)$ is called the characteristic polynomial of T.
(Axler, 2026, Definition 8.26)

Theorem 1.2.22 (Conditions Equivalent to Diagonalizability): Suppose V is finite-dimensional and T ∈ L(V). Let λ_1, ..., λ_m denote the distinct eigenvalues of T. Then the following are equivalent:
a) T is diagonalizable.
b) V has a basis consisting of eigenvectors of T.
c) V = E(λ_1, T) ⊕ ... ⊕ E(λ_m, T).
d) dim V = dim E(λ_1, T) + ... + dim E(λ_m, T).
(Axler, 2026, Result 5.55)

This theorem provides multiple characterizations of diagonalizability. Condition (b) states that diagonalizability means existence of an eigenvector basis. In this basis, the operator acts as coordinate-wise scaling. Matrix representations become diagonal matrices with eigenvalues on the diagonal.

Diagonalization simplifies operator analysis. Powers of diagonal matrices are easy to compute. This enables efficient algorithms for matrix exponentiation and spectral methods.

Matrix Decomposition
Matrix decompositions factor matrices into products of simpler matrices. These factorizations reveal structure and enable efficient computation.
The spectral theorem guarantees diagonalizability for important operator classes.

Theorem 1.2.23 (Real Spectral Theorem): Suppose F = R and T ∈ L(V). Then the following are equivalent:
a) T is self-adjoint.
b) V has an orthonormal basis consisting of eigenvectors of T.
c) T has a diagonal matrix with respect to some orthonormal basis of V.
(Axler, 2026, Result 7.29)

Self-adjoint operators satisfy T = *T, where T* is the adjoint. For matrices, this means the matrix equals its transpose. Such operators always have real eigenvalues and orthogonal eigenvectors. We can construct an orthonormal eigenvector basis.

Theorem 1.2.24 (Complex Spectral Theorem): Suppose F = C and T ∈ L(V). Then the following are equivalent:
a) T is normal.
b) V has an orthonormal basis consisting of eigenvectors of T.
c) T has a diagonal matrix with respect to some orthonormal basis of V.
(Axler, 2026, Result 7.31)

Normal operators satisfy TT = *TT*. This includes self-adjoint, unitary, and other important operator classes. The spectral theorem guarantees orthonormal eigenvector bases for all normal operators.

These decomposition results are fundamental in data analysis. Covariance matrices in statistics are self-adjoint. The spectral theorem enables principal component analysis. Document-term matrices can be analyzed through singular value decomposition, which relates to spectral decomposition of associated symmetric matrices.

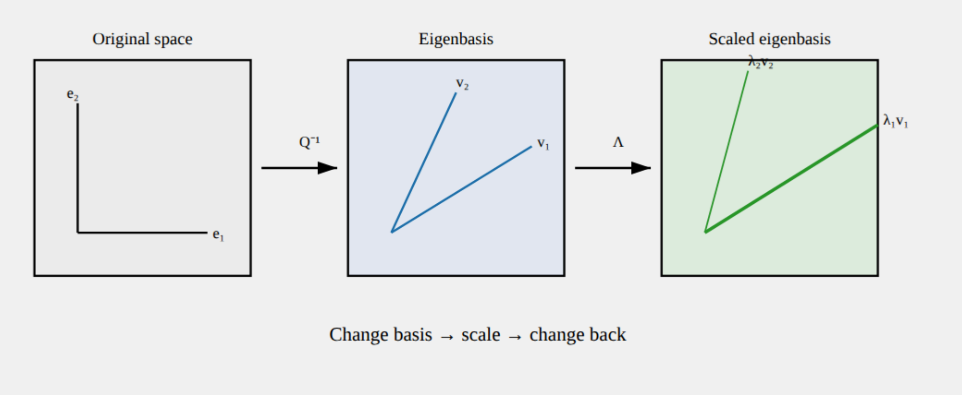


Figure 1.2: Matrix decomposition illustration showing $T = Q\Lambda Q^{(} - 1)$ where Q contains eigenvectors as columns and Λ is diagonal with eigenvalues

Geometric Interpretations

The algebraic definitions gain intuitive meaning through geometric interpretation.

Vector spaces are geometric objects. Vectors are points or arrows in space. Addition corresponds to placing arrows tip-to-tail. Scalar multiplication stretches or shrinks arrows. Subspaces are lines, planes, or higher-dimensional analogs passing through the origin.

Span has clear geometric meaning. The span of one vector is a line through the origin. The span of two non-collinear vectors is a plane. Each additional independent vector adds a dimension. The span fills out a flat subspace.

Linear independence means geometric independence. Independent vectors point in different directions. No vector lies in the span of the others. Dependent vectors exhibit geometric redundancy.

Bases provide coordinate systems. Given a basis, every vector has unique coordinates. Different bases give different views of the same geometric object. Orthonormal bases aligned with principal axes simplify analysis.

Inner products measure angles and lengths. The inner product of orthogonal vectors is zero, matching perpendicularity. The Cauchy-Schwarz inequality has geometric content: projecting one vector onto another cannot exceed the vector's length.

Eigenspaces reveal invariant directions. An operator stretches space along eigenvector directions. Non-eigenvector directions get rotated or skewed. Diagonalizable operators have enough eigenvectors to span the space. The operator acts as pure stretching in eigenvector coordinates.

These geometric pictures guide intuition and suggest algorithmic approaches. Abstract algebra becomes concrete through visualization. This interplay between algebra and geometry makes linear algebra powerful for both theory and applications.

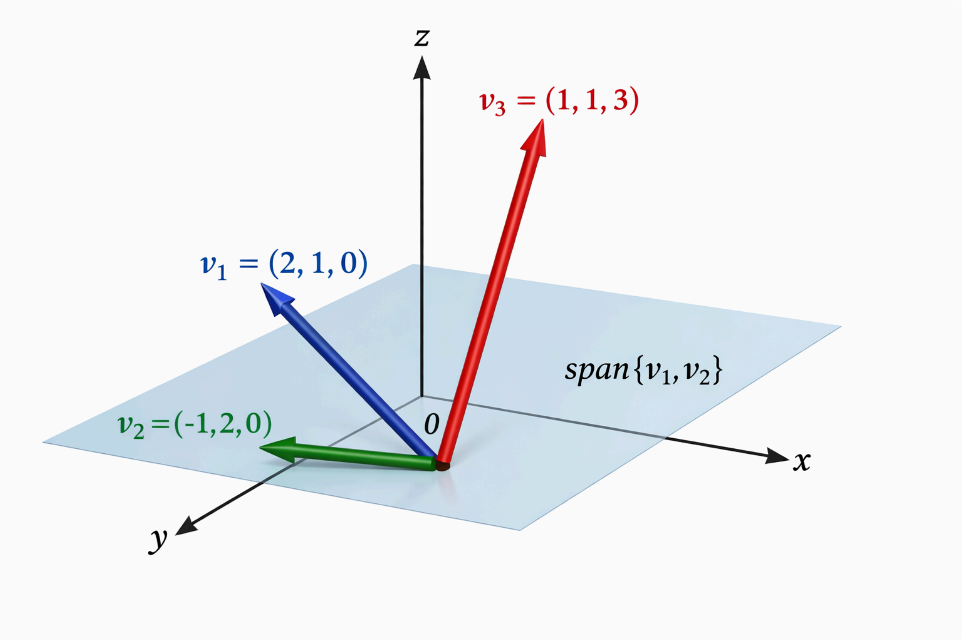


Figure 1.3: 3D visualization showing span of two vectors as a plane, with a third independent vector extending into 3D space

1.2.2 Probability Theory

Probability Spaces and Sigma-Algebras

Probability theory provides a rigorous framework for reasoning under uncertainty. We begin with the foundational structures.

Definition 1.2.25 (Sample Space): The set of all possible outcomes of an experiment is called the sample space of the experiment and is denoted by S.
(Ross, 2010, Section 2.2)

The sample space captures all possibilities. For document retrieval, an outcome might represent whether a document is relevant to a query. The sample space contains all possible relevance configurations.

Definition 1.2.26 (Event): Any subset E of the sample space is known as an event.
(Ross, 2010, Section 2.2)

Events are collections of outcomes we wish to assign probabilities. The event "document is relevant" is a subset of all possible relevance states.

Axioms of Probability: For any event E in sample space S:

1. $0 \leq P(E) \leq 1$
2. $P(S) = 1$
3. For any sequence of mutually exclusive events $E_1, E_2, \ldots,$

$$P\left(\bigcup_{i=1}^{\infty} E_i\right) = \sum_{i=1}^{\infty} P(E_i)$$

(Ross, 2010, Section 2.4)

These axioms formalize probability as a measure. Axiom 1 bounds probabilities between zero and one. Axiom 2 states that something must occur. Axiom 3 ensures probabilities of disjoint events add. All probability theory derives from these three axioms.

Random Variables and Measurability

Definition 1.2.27 (Random Variable): A random variable is a real-valued function defined on the sample space.
(Ross, 2010, Chapter 4, Section 4.1)

Random variables map outcomes to numbers. In information retrieval, we model relevance as a random variable taking values 0 (not relevant) or 1 (relevant). Term frequencies can also be modeled as random variables.

Conditional Probability and Independence

Conditional probability updates beliefs given new information.

Definition 1.2.28 (Conditional Probability): If $P(F) > 0$, then

$$P(E|F) = \frac{P(EF)}{P(F)}$$

(Ross, 2010, Definition 3.1)

This formula computes the probability of E given that F occurred. We restrict attention to outcomes where F holds and renormalize. In search, $P(\text{relevant}|\text{clicked})$ represents the probability a document is relevant given the user clicked it.

Definition 1.2.29 (Independence): Two events E and F are said to be independent if

$$P(EF) = P(E)P(F)$$

(Ross, 2010, Definition 3.2)

Independent events provide no information about each other. Knowing one occurred does not change the probability of the other. Document relevance and user location might be independent, while relevance and query terms are clearly dependent.

Bayes Theorem

Bayes theorem inverts conditional probabilities, enabling inference from evidence.

Theorem 1.2.30 (Bayes Formula): Let $E_1, E_2, \ldots, E_n$ be mutually exclusive events whose union is the sample space S. Then

$$P(E_j|F) = \frac{P(F|E_j)P(E_j)}{\sum_{i=1}^{n} P(F|E_i)P(E_i)}$$

(Ross, 2010, Proposition 3.1)

This theorem is fundamental to probabilistic information retrieval. We observe features F (query terms appearing in document). We want to infer relevance E_j. Bayes theorem relates $P(\text{relevant}|\text{terms})$ to $P(\text{terms}|\text{relevant})$, which is often easier to estimate. The denominator normalizes over all possible relevance states.

The prior $P(E_j)$ represents our belief before seeing evidence. The likelihood $P(F|E_j)$ describes how likely the evidence is under each hypothesis. The posterior $P(E_j|F)$ updates our belief after observing F. This framework underlies probabilistic ranking models.

Expectation as Linear Functional

Theorem 1.2.31 (Linearity of Expectation): The expected value of a sum of random variables is equal to the sum of the expected values.

$$E\left[\sum_{i=1}^{n} X_i\right] = \sum_{i=1}^{n} E[X_i]$$

(Ross, 2010, Chapter 7, Proposition 2.1)

This linearity holds even when random variables are dependent. Expectation behaves as a linear functional on the space of random variables. This property simplifies calculations and enables decomposition of complex quantities into simpler components.

Variance, Covariance, and Correlation

Definition 1.2.32 (Variance): If X is a random variable with mean μ, then the variance of X, denoted by $\text{Var}(X)$, is defined by

$$\text{Var}(X) = E[(X - \mu)^2]$$

(Ross, 2010, Chapter 4, Definition 4.2)

Variance measures spread around the mean. It quantifies uncertainty in the random variable's value.

Definition 1.2.33 (Covariance): The covariance of X and Y, denoted by $\text{Cov}(X, Y)$, is defined by

$$\text{Cov}(X, Y) = E[(X - E[X])(Y - E[Y])]$$

(Ross, 2010, Chapter 7, Section 7.4)

Covariance measures how two random variables vary together. Positive covariance means they tend to increase together. Negative covariance means one increases when the other decreases. Zero covariance indicates no linear relationship.

Definition 1.2.34 (Correlation): The correlation of two random variables X and Y, denoted by $\rho(X, Y)$, is defined by

$$\rho(X, Y) = \frac{\text{Cov}(X, Y)}{\sqrt{\text{Var}(X)\text{Var}(Y)}}$$

(Ross, 2010, Chapter 7, Section 7.4)

Correlation normalizes covariance by the product of standard deviations. It ranges from -1 to 1. Values near 1 indicate strong positive linear relationship. Values near -1 indicate strong negative relationship. Values near 0 indicate weak linear relationship. In feature analysis, correlation identifies redundant features that provide similar information.

Joint, Marginal, and Conditional Distributions

Definition 1.2.35 (Joint Distribution): The joint cumulative probability distribution function of X and Y is defined by

$$F(a,b) = PX \leq a, Y \leq b, \quad -\infty < a, b < \infty$$

(Ross, 2010, Chapter 6, Section 6.1)

Joint distributions describe multiple random variables simultaneously. They capture dependencies between variables.

Definition 1.2.36 (Marginal Distribution): The individual probability mass functions of X and Y are easily obtained from the joint probability mass function by

$$PX = x_i = \sum_j P(x_i, y_j)$$

(Ross, 2010, Chapter 6, Section 6.1)

Marginal distributions recover individual variable distributions from joint distributions by summing over other variables.

Definition 1.2.37 (Conditional Distribution): If X and Y have a joint probability mass function $p(x,y)$, then the conditional probability mass function of X, given that $Y = y$, is defined by

$$p_{X|Y}(x|y) = PX = x|Y = y = \frac{p(x,y)}{p_Y(y)}$$

(Ross, 2010, Chapter 6, Section 6.3)

Conditional distributions describe one variable's behavior given knowledge of another. In ranking, we model term frequencies conditional on document relevance. Relevant documents exhibit different term distributions than non-relevant documents.

Maximum Likelihood Estimation

Maximum likelihood estimation chooses parameters that make observed data most probable. Given data D and model with parameter θ, the likelihood is $L(\theta) = P(D|\theta)$. The maximum likelihood estimate is

$$\hat{\theta} = \arg\max_{\theta} L(\theta)$$

This principle underlies parameter fitting in language models and learning-to-rank systems. We observe query-document pairs with relevance labels. We find parameters maximizing the probability of these observations. Under appropriate conditions, maximum likelihood estimators are consistent and asymptotically normal, providing theoretical justification for their use.

CHAPTER 2: Classical and Probabilistic Search Models

This chapter presents foundational retrieval models based on set theory, vector spaces, and probability theory.

References: Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to Information Retrieval. Cambridge University Press. Chapters 1, 6, 11, 12.

2.1 Classical Information Retrieval Models

2.1.1 Boolean Retrieval Model

The Boolean model represents the earliest formal approach to information retrieval. Documents and queries are sets of terms combined using logical operators.

Set-Theoretic Foundations: A document collection $D = d_1, d_2, \ldots, d_N$ contains documents represented as term sets. Each document $d_i \subseteq V$ where V is the vocabulary. Boolean queries combine terms with AND, OR, and NOT operators.

For terms $t_1, t_2 \in V$:

- t_1 AND t_2: Returns documents containing both terms
- t_1 OR t_2: Returns documents containing either term
- NOT t_1: Returns documents not containing the term

These correspond to set intersection, union, and complement operations.

Inverted Index Structure: Efficient retrieval requires an inverted index mapping terms to document lists. For term t, the posting list $\text{postings}(t) = d_i : t \in d_i$ contains all documents with that term. Query processing becomes set operations on posting lists.

Limitations: The Boolean model provides no ranking. All matching documents are equally relevant. It requires exact term matches, missing synonyms and related concepts. Users must formulate precise queries with correct operators. These limitations motivated vector space and probabilistic models.

2.1.2 Vector Space Model and TF-IDF

The vector space model represents documents and queries as vectors in high-dimensional space, enabling similarity-based ranking.

Document Representation: With vocabulary size $|V|$, each document becomes a vector $\mathbf{d} = (w_{1,d}, w_{2,d}, \ldots, w_{|V|,d})$ where $w_{i,d}$ is the weight of term i in document d. Queries are similarly represented as vectors $\mathbf{q}$.

Term Frequency (TF): Raw term frequency counts occurrences: $\text{tf}_{t,d}$ = number of times term t appears in document d. Logarithmic scaling prevents dominance by high-frequency terms:

$$\text{tf}_{t,d} = \{1 + \log(\text{tf}_{t,d}) \quad \text{if } \text{tf}_{t,d} > 0\ 0 \quad \text{otherwise}$$

Inverse Document Frequency (IDF): Rare terms are more discriminative than common terms. IDF quantifies term rarity:

$$\text{idf}_t = \log \frac{N}{\text{df}_t}$$

where N is total documents and df_t is the number of documents containing term t. Terms in all documents have $\text{idf}_t = 0$. Rare terms have high IDF values.

TF-IDF Weighting: Combining frequency and rarity produces:

$$w_{t,d} = \text{tf}_{t,d} \times \text{idf}_t = (1 + \log \text{tf}_{t,d}) \times \log \frac{N}{\text{df}_t}$$

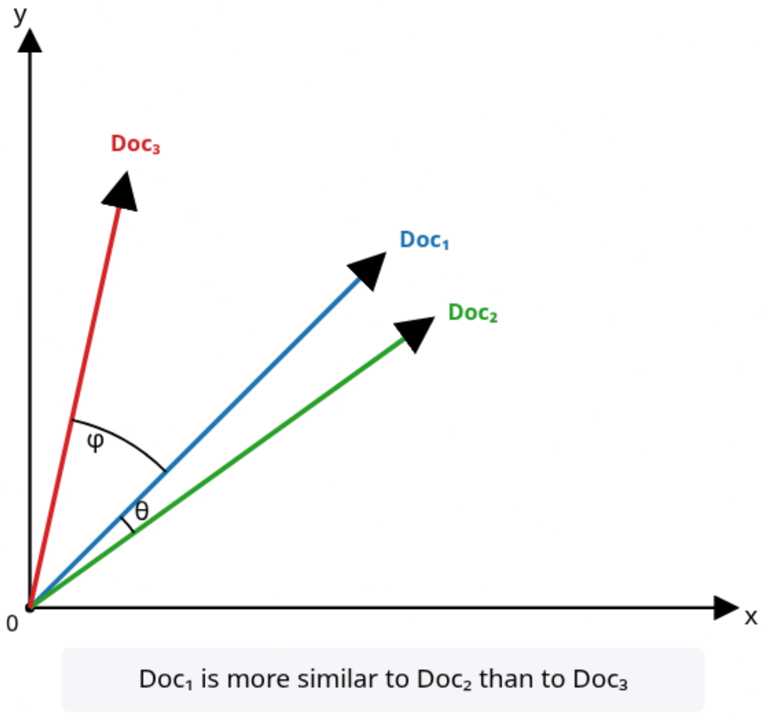
This balances local importance (within document) with global rarity (across collection).

Cosine Similarity: Documents are ranked by cosine of the angle between document and query vectors:

$$\text{sim}(\mathbf{d}, \mathbf{q}) = \frac{\mathbf{d} \cdot \mathbf{q}}{|\mathbf{d}||\mathbf{q}|} = \frac{\sum_{i=1}^{|V|} w_{i,d} \times w_{i,q}}{\sqrt{\sum_{i=1}^{|V|} w_{i,d}^2} \times \sqrt{\sum_{i=1}^{|V|} w_{i,q}^2}}$$

Cosine similarity ranges from 0 (no shared terms) to 1 (identical term distributions). This normalization removes document length effects.

Geometric Interpretation: The vector space provides geometric intuition. Similar documents cluster as their vectors point in similar directions. The angle between vectors measures topical relatedness independent of document length.



Doc₁ is more similar to Doc₂ than to Doc₃

Figure 2.1: Cosine Similarity

Vocabulary Mismatch Problem: Despite improvements over Boolean retrieval, VSM suffers from vocabulary mismatch. Documents using synonyms or paraphrases receive zero similarity despite relevance. A query for "automobile" misses documents about "car". This lexical gap motivated probabilistic and semantic approaches.

2.2 Probabilistic Models of Relevance

References: Manning et al. (2008), Chapter 11; Croft, B., Metzler, D., & Strohman, T. (2009). Search Engines: Information Retrieval in Practice. Chapter 7.

2.2.1 Foundational Concepts

Probabilistic information retrieval models relevance as a random variable and ranks documents by relevance probability.

Binary Relevance Assumption: Relevance $R \in 0, 1$ is modeled as binary: relevant (1) or not relevant (0). For query q and document d, we estimate $P(R = 1|q, d)$.

Probability Ranking Principle (PRP): Ranking documents by decreasing $P(R = 1|d, q)$ maximizes retrieval effectiveness under independence assumptions. This theorem reduces retrieval to probability estimation.

Odds Ratio Formulation: Working with odds simplifies derivations:

$$O(R = 1|d, q) = \frac{P(R = 1|d, q)}{P(R = 0|d, q)}$$

Log-odds yield additive scoring functions convenient for term independence models.

2.2.2 Statistical Language Models

Language models estimate the probability that a document generates the query.

Query Likelihood Model: Rank documents by $P(q|d)$, the probability document d's language model generates query q. Assuming term independence:

$$P(q|d) = \prod_{t \in q} P(t|d)$$

The maximum likelihood estimate is:

$$P(t|d) = \frac{\mathrm{tf}_{t,d}}{\sum_{t' \in d} \mathrm{tf}_{t',d}}$$

Documents with zero probability for any query term receive score zero, preventing meaningful ranking.

Smoothing Techniques: Smoothing redistributes probability to unobserved terms. Dirichlet smoothing interpolates document and collection statistics:

$$P(t|d) = \frac{\mathrm{tf}_{t,d} + \mu P(t|C)}{|d| + \mu}$$

where $P(t|C)$ is the collection probability and μ controls smoothing strength. Long documents rely on their statistics. Short documents rely on collection statistics.

2.2.3 Okapi BM25

BM25 is a probabilistic ranking function derived from the Binary Independence Model. It remains highly effective in modern systems.

BM25 Formula: The complete ranking function is:

$$\text{BM25}(d,q) = \sum_{t \in q} \text{IDF}(t) \cdot \frac{\text{tf}_{t,d} \cdot (k_1 + 1)}{\text{tf}_{t,d} + k_1 \cdot \left(1 - b + b \cdot \frac{|d|}{\text{avgdl}}\right)}$$

where:

- $\text{IDF}(t) = \log \frac{N - \text{df}_t + 0.5}{\text{df}_t + 0.5}$ weights term rarity
- $k_1 \in [1.2, 2.0]$ controls term frequency saturation
- $b \in [0, 1]$ controls document length normalization
- $|d|$ is document length, avgdl is average length

Term Saturation: As term frequency increases, contribution saturates at $(k_1 + 1)$. This implements diminishing returns: additional occurrences matter less than initial occurrences.

Document Length Normalization: The denominator penalizes long documents:

$$1 - b + b \cdot \frac{|d|}{\text{avgdl}}$$

With $b = 0$, no normalization. With $b = 1$, full normalization. Standard value $b = 0.75$ provides moderate penalty.

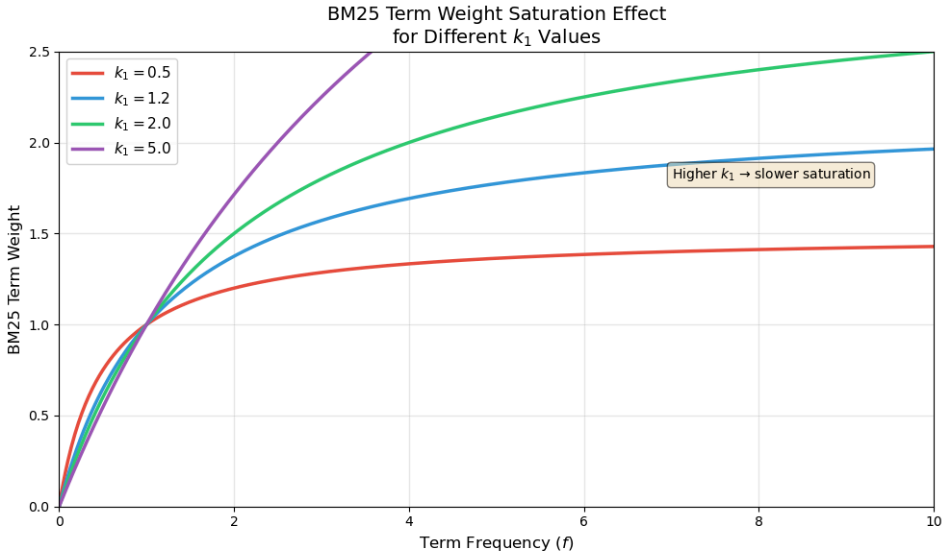


Figure 2.2: BM25 term weight vs term frequency for different k_1 values

Parameter Tuning: Standard values $k_1 = 1.2$ and $b = 0.75$ work well across collections. Optimal values vary with document length distribution and query characteristics. Tuning requires relevance judgments and grid search.

Strengths: BM25 combines theoretical foundations with empirical effectiveness. It performs competitively with modern neural models on many benchmarks. The formula is efficient and interpretable.

Weaknesses: Like classical models, BM25 suffers vocabulary mismatch. It assumes term independence, ignoring phrases and proximity. Parameters require tuning for optimal performance.

CHAPTER 3: DATA ANALYSIS AND RESULT DISCUSSION

This chapter examines supervised learning approaches to ranking and dimensionality reduction techniques.

References: Liu, T.-Y. (2009). Learning to Rank for Information Retrieval. Foundations and Trends in Information Retrieval, 3(3); Manning et al. (2008), Chapter 18.

3.1 Learning-to-Rank Framework

3.1.1 Ranking as Supervised Learning

Learning-to-rank formulates ranking as supervised machine learning. Given training data of queries, documents, and relevance labels, we learn a scoring function.

Supervised Formulation: Represent query-document pairs as feature vectors $\mathbf{x} = \phi(q, d) \in \mathbb{R}^n$. Learn function $f : \mathbb{R}^n \to \mathbb{R}$ predicting relevance scores. Rank documents by $s_i = f(\phi(q, d_i))$.

Feature Representation: Features capture query-document relationships:

- Query-independent: Document length, PageRank, spam score
- Query-dependent: BM25 score, TF-IDF score, term coverage
- Matching features: Exact matches, proximity, synonym matches

Typical feature vectors contain 50-500 features combining multiple signals.

Relevance Labels: Training data consists of queries with judged documents. Labels may be binary (relevant/not relevant) or graded (0-3 scale). Human judgments are expensive but essential for supervised learning.

3.1.2 Ranking Approaches

Learning-to-rank methods divide into pointwise, pairwise, and listwise approaches.

Pointwise Methods: Treat ranking as regression or classification on individual documents. Minimize loss over query-document pairs:

$$\min_f \sum_{(q,d,r)} L(f(\phi(q,d)), r)$$

Simple but ignore relative ordering. Predicting absolute scores is harder than predicting rankings.

Pairwise Methods: Learn from document pairs. For documents with $r_i > r_j$, we want $f(\phi(q,d_i)) > f(\phi(q,d_j))$. Minimize pairwise loss:

$$\min_f \sum_{r_i > r_j} L(f(\phi(q,d_i)) - f(\phi(q,d_j)))$$

RankNet uses logistic loss with neural networks. RankSVM uses hinge loss. These better capture ranking objectives but treat all pairs equally.

Listwise Methods: Optimize metrics defined over entire ranked lists. LambdaMART uses gradient boosted trees with gradients designed to optimize NDCG directly. Listwise methods align with evaluation metrics but are computationally expensive.

3.1.3 Evaluation Metrics

Precision@K: Fraction of top-K results that are relevant:

$$\text{Precision@K} = \frac{|\text{relevant docs in top-K}|}{K}$$

Mean Average Precision (MAP): Averages precision across queries, emphasizing early relevant results.

Mean Reciprocal Rank (MRR): Average reciprocal rank of first relevant document:

$$\text{MRR} = \frac{1}{|Q|} \sum_{q \in Q} \frac{1}{\text{rank}_q}$$

Normalized Discounted Cumulative Gain (NDCG): Measures graded relevance with position discounting:

$$\text{DCG@K} = \sum_{i=1}^{K} \frac{2^{r_i} - 1}{\log_2(i+1)}$$

$$\text{NDCG@K} = \frac{\text{DCG@K}}{\text{IDCG@K}}$$

where IDCG is ideal DCG from perfect ranking. NDCG ranges from 0 to 1.

These metrics prioritize top results, reflecting user behavior of examining only first-page results.

3.2 Dimensionality Reduction and Experimental Results

Motivation

Document vectors in TF-IDF space have very high dimensionality (7,184 terms in our dataset). High-dimensional vectors:

- Increase computation time
- Contain redundant information
- Can introduce noise

Dimensionality reduction projects these vectors into a lower-dimensional subspace while preserving the most important information. This reduces computation and may improve retrieval by filtering low-variance noise.

Principal Component Analysis (PCA)

Purpose: Find orthogonal directions (principal components) that capture maximum variance.
Mathematical Formulation: For document vectors $(\mathbf{d}_1, \dots, \mathbf{d}_N \in \mathbb{R}^{|V|})$, the covariance matrix is:

$$\Sigma = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{d}_i - \bar{\mathbf{d}})(\mathbf{d}_i - \bar{\mathbf{d}})^\top$$

Eigenvectors $(\mathbf{v}_i)$ satisfy:

$$\Sigma \mathbf{v}_i = \lambda_i \mathbf{v}_i$$

where $(\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{|V|})$. The top (k) eigenvectors capture most variance.

Projection:

$$\mathbf{d}_i' = [(\mathbf{d}_i - \bar{\mathbf{d}}) \cdot \mathbf{v}_1, \dots, (\mathbf{d}_i - \bar{\mathbf{d}}) \cdot \mathbf{v}_k]^T$$

Variance Preservation:

$$\text{Variance Preserved} = \frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{|V|} \lambda_i}$$

Impact on Retrieval:

- Low-variance noise is removed

- Co-occurring terms merge into components (partially addressing vocabulary mismatch)
- Some low-variance but relevance-discriminative features may be lost

Dataset Statistics

Metric	Value
Documents	1,400
Queries	225
Vocabulary Size	7,184
TF-IDF Matrix Shape	(1400, 7184)

PCA Variance Analysis

Components (k)	Variance Preserved
20	13.80%
50	23.32%
100	34.04%

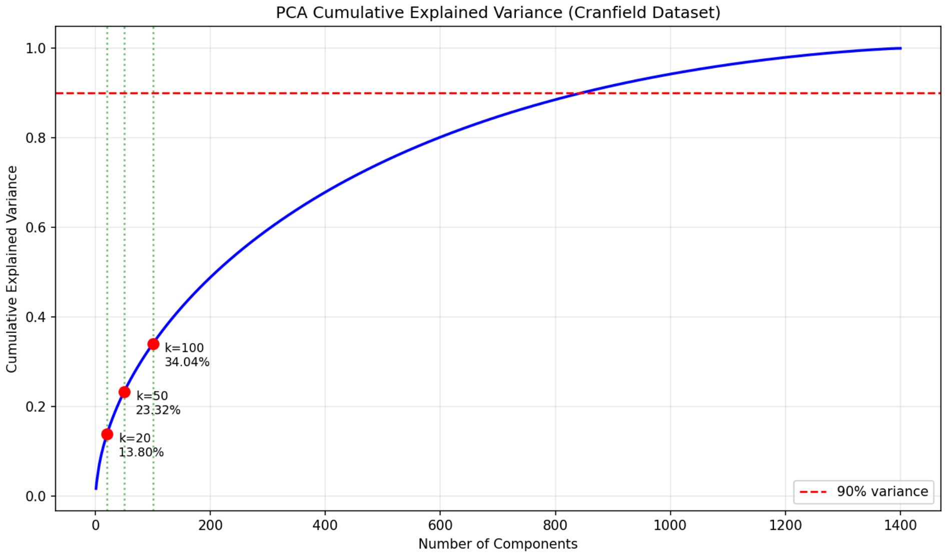


Figure 2.3: Experimental graph

Observations:

- Scree plot Figure 2.3 shows diminishing returns beyond top components.
- ~850 components needed to preserve 90% of variance.
- High-dimensional TF-IDF vectors contain significant redundancy.

Ranking Performance (Precision@10)

Method	Precision@10	Relative Performance
TF-IDF (Full)	0.2173	100% (baseline)
PCA-100	0.2151	99.0%

Analysis:

- Dimensionality reduction from 7,184 → 100 (~98.6% reduction).
- Only 34.04% of variance preserved, yet ~99% of ranking performance retained.
- Top 100 principal components capture the most salient semantic features.

Computational Efficiency:

- Cosine similarity computation ~72x faster (7184/100).
- Storage and memory requirements significantly reduced.

Practical Implications:

- PCA offers a strong trade-off between retrieval accuracy and efficiency.
- Minimal loss in Precision@10 is outweighed by computational gains.
- Dense low-dimensional vectors resemble modern embedding-based retrieval practices.

CONCLUSION

This project examined mathematical foundations of information retrieval, from vector spaces and probability theory to classical models and modern learning-to-rank methods.

Theoretical Contributions: Chapter 1 established vector space axioms, inner product properties, and probability theory fundamentals. We derived cosine similarity from inner products and developed probabilistic inference through Bayes theorem. These foundations enabled rigorous analysis of retrieval models.

Chapter 2 applied these frameworks. The Boolean model's set operations provided exact matching. The vector space model's geometric interpretation explained TF-IDF and cosine ranking. Probabilistic models treated relevance as inference, yielding language models and BM25. Each model connects abstract mathematics to practical algorithms.

Chapter 3 examined learning-to-rank, which optimizes ranking through supervised learning. Pointwise, pairwise, and listwise methods formalize ranking with different loss functions. PCA demonstrated how eigenvalue decomposition enables dimensionality reduction while preserving variance.

Mathematical Insights: Several themes recurred: geometric intuition from vector spaces, probabilistic reasoning under uncertainty, and optimization frameworks for learning. These mathematical structures enable analysis, guide design, and predict performance.

Limitations: Classical models suffer vocabulary mismatch, relying on lexical matching. Binary relevance assumptions oversimplify graded judgments. Term independence ignores important dependencies. These limitations motivate neural models with learned embeddings.

Future Directions: Neural ranking models use transformers like BERT for contextual embeddings addressing semantic gaps. Personalized search adapts to individual users. Multistage ranking balances efficiency and quality through cascades. Fairness considerations prevent bias amplification.

Closing Remarks: This project demonstrated how mathematical rigor enhances understanding of information retrieval. Vector spaces formalize document similarity. Probability theory enables reasoning about relevance. Optimization formalizes learning objectives. These foundations remain relevant despite rapid advances in neural models.

For pure mathematics students, this illustrates how abstract theory drives practical applications. The axioms of vector spaces enable geometric reasoning about text. Probability axioms enable inference under uncertainty. The spectral theorem enables dimensionality reduction. Theory and practice reinforce each other, preparing researchers for future developments in retrieval and beyond.

References & Bibliography

Textbooks and Monographs

Axler, S. (2026). *Linear Algebra Done Right* (4th ed.). Springer International Publishing.

Croft, W. B., Metzler, D., & Strohman, T. (2009). *Search Engines: Information Retrieval in Practice*. Addison-Wesley.

Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press. (2008) Available online at https://nlp.stanford.edu/IR-book/

Ross, S. M. (2010). *A First Course in Probability* (8th ed.). Pearson Education.

Skillicorn, D. B. (2007). *Understanding Complex Datasets: Data Mining with Matrix Decompositions*. Chapman and Hall/CRC.

Research Papers and Articles

Burges, C. J. C. (2010). From RankNet to LambdaRank to LambdaMART: An Overview. *Microsoft Research Technical Report MSR-TR-2010-82*.

Liu, T.-Y. (2009). Learning to Rank for Information Retrieval. *Foundations and Trends in Information Retrieval*, 3(3), 225-331.

Robertson, S. E., & Zaragoza, H. (2009). The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval*, 3(4), 333-389.

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5), 513-523.

Software and Datasets

Cleverdon, C. W. (1967). The Cranfield tests on index language devices. *Aslib Proceedings*, 19(6), 173-194.

Pedregosa, F., et al. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830.

Online Resources

Anthropic. (2026). Claude AI Assistant. https://claude.ai
Python Software Foundation. (2026). Python Programming Language. https://www.python.org

Annexures

Appendix A: Key Mathematical Derivations

A.1 Derivation of Cosine Similarity from Inner Product

Starting from the inner product definition in Chapter 1, for vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$:
The Cauchy-Schwarz inequality states:

$$|\langle \mathbf{u}, \mathbf{v} \rangle| \leq |\mathbf{u}||\mathbf{v}|$$

Dividing both sides by $|\mathbf{u}||\mathbf{v}|$ (for nonzero vectors):

$$\frac{|\langle \mathbf{u}, \mathbf{v} \rangle|}{|\mathbf{u}||\mathbf{v}|} \leq 1$$

This implies:

$$-1 \leq \frac{\langle \mathbf{u}, \mathbf{v} \rangle}{|\mathbf{u}||\mathbf{v}|} \leq 1$$

Since this ratio lies in $[-1, 1]$, it corresponds to the range of the cosine function. We define:

$$\cos\theta = \frac{\langle \mathbf{u}, \mathbf{v} \rangle}{|\mathbf{u}||\mathbf{v}|}$$

For TF-IDF vectors with nonnegative components, $\langle \mathbf{u}, \mathbf{v} \rangle \geq 0$, so cosine similarity ranges from 0 to 1.

A.2 BM25 Parameter Sensitivity Analysis

Standard BM25 parameters are $k_1 = 1.2$ and $b = 0.75$. The effect of varying k_1 on term weight:

For $\mathrm{tf} = 5$, $|d| = \mathrm{avgdl}$, $b = 0.75$:

- $k_1 = 1.0$: weight ≈ 2.27
- $k_1 = 1.2$: weight ≈ 2.44
- $k_1 = 1.5$: weight ≈ 2.67
- $k_1 = 2.0$: weight ≈ 3.00

Higher k_1 increases the impact of term frequency before saturation occurs.

Appendix B: Code Repository

The complete implementation of the project (paper, tools, code, results) is available at:

GitHub Repository: https://github.com/ganeshspeaks/vector-space-probabilistic-ranking

The repository contains:

- Implementation of TF-IDF and BM25 ranking functions
- Metric calculations (Precision@K, NDCG, MAP)
- Dimensionality reduction and visualization
- Main experimental workflow
- Usage instructions and project overview
- Output tables and figures

All code is written in Python 3 and requires only standard scientific computing libraries (NumPy, scikit-learn, Matplotlib). The implementation prioritizes clarity and reproducibility over optimization.

Appendix C: Software Environment:

- Python
- NumPy
- scikit-learn
- Matplotlib
- rank-bm25

Dataset:

- Cranfield Collection
- 1,400 documents
- 225 queries
- Relevance judgments: Binary (relevant/not relevant)

All experiments were conducted on the basic hardware without requiring high-performance computing resources. The modest computational requirements demonstrate the accessibility of classical information retrieval methods.