Extracting Key Patterns in Text Data (LSA via SVD in Julia)

Presenter: Celestin HAGENIMANA

Supervisor: Prof. Franck Kalala Mutombo

African Institute of Mathematical Sciences
AIMS Rwanda

October 30, 2025

Outline

Introduction

To extract latent patterns and semantic relationships from text, we use Singular Value Decomposition (SVD), a method commonly known as Latent Semantic Analysis (LSA).

A corpus can be represented as a term–document matrix X.

- Applying SVD gives $X = U\Sigma V^T$
- U maps terms to latent topics
- Σ contains singular values representing topic importance
- V maps documents into the latent topic space

SVD helps with dimensionality reduction, noise filtering, and uncovering latent semantic features[1].

Problem Statement

Given a corpus of textual documents, there is a need to analyze and compare their content in a meaningful way by capturing underlying semantic patterns. The goal is to create a clear and reproducible method that converts text into numerical form, allowing visualization and comparison of document similarities.

- Focus on identifying meaningful patterns in textual datasets.

Key Concepts in Text Transformation

Term Frequency(TF) and Inverse Document Frequency(IDF)

Term Frequency (TF)

$$\mathrm{TF}_{i,j} = \frac{\text{Count of term } i \text{ in document } j}{\text{Total Terms in document } j}$$

Inverse Document Frequency (IDF)

$$\mathrm{IDF}_i = \log\left(\frac{\text{Total number of documents in corpus}}{\text{Number of documents containing term } i}\right)$$

- High IDF $\rightarrow$ term is rare (more unique)
- Low IDF $\rightarrow$ term is common (less useful)

Term Frequency–Inverse Document Frequency (TF–IDF)

$$\mathrm{TFIDF}_{i,j} = \mathrm{TF}_{i,j} \times \mathrm{IDF}_i$$

Interpretation of TF–IDF

- High TF–IDF $\rightarrow$ word is important in this document and not common in other documents.
- Low TF–IDF $\rightarrow$ word is either rare in the document or very common across documents[2].

Methodology

The following steps outline the process to extract latent semantic patterns from a corpus of textual documents:

1. Construct a matrix $X = TF - IDF$ (Terms x Documents) from the set of documents.
2. Perform SVD on matrix X and keep the top k components: $X \approx U_k \Sigma_k V_k^T$
3. Interpret the top k latent topics by examining the largest coefficients in each column of U_k.
4. Represent each document in the latent space as columns of $V_k \Sigma_k$.
5. Compute cosine similarity between document vectors to measure semantic similarity.
6. Project new queries into the latent space for semantic search.

Text Data Processing

Corpus — Example Documents

- **Doc 1:** "Radar detects aircraft and ships. It uses signal processing to identify and track moving targets."
- **Doc 2:** "Image processing techniques enhance image quality, reduce noise, and extract visual features."
- **Doc 3:** "Advanced radar signal processing improves object detection and tracking accuracy. It helps in defense systems."
- **Doc 4:** "Machine learning models improve recognition and classification in computer vision applications."
- **Doc 5:** "Aircraft detection radar system uses antennas to classify aircraft and vessels on the sea and air."
- **Doc 6:** "Image recognition with deep learning detects objects, patterns, and faces in digital photos."

Preprocessing Steps

1. Convert text to lowercase.
2. Remove punctuation and non-word characters.
3. Split text into tokens (words).
4. Normalize term counts per document (compute TF),compute IDF and construct TF-IDF matrix).

Latent Semantic Analysis(LSA)

- Compute SVD of the TF–IDF matrix: $X = U\Sigma V^T$.
- Keep top-k singular values and vectors ($k = 3$ used here).
- Document embedding in latent space: columns of $\Sigma_k V_k^T$.
- LSA groups terms that co-occur across documents, revealing latent topics.

Cosine Similarity and Comparison

Cosine similarity:

$$\cos(a, b) = \frac{a \cdot b}{\|a\|\|b\|}$$

- Compute cosine on TF–IDF vectors
- Compute cosine on LSA vectors
- Compare: LSA often reveals higher-level topic links not obvious from TF–IDF.

TF–IDF vs LSA Heatmaps

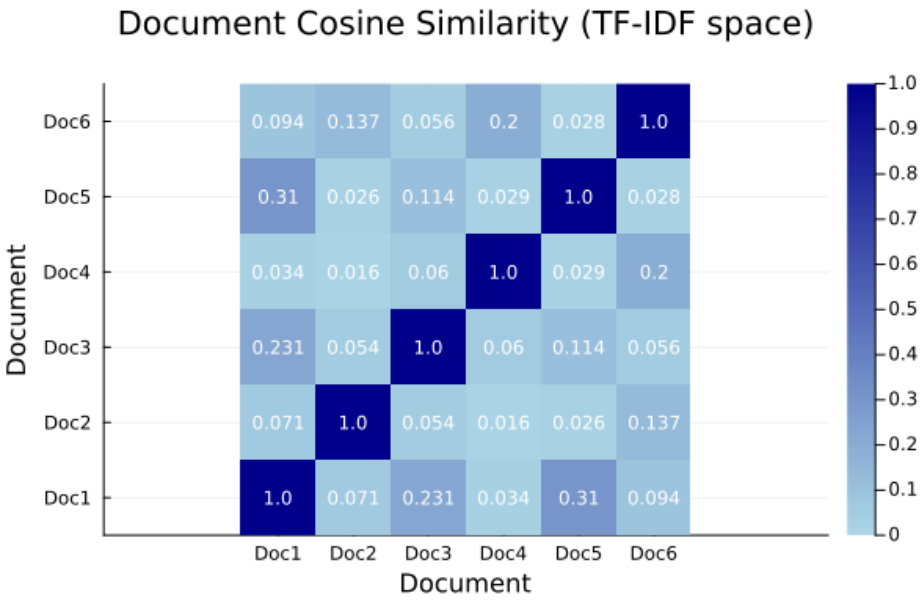

Figure 1: TF-IDF Heatmap

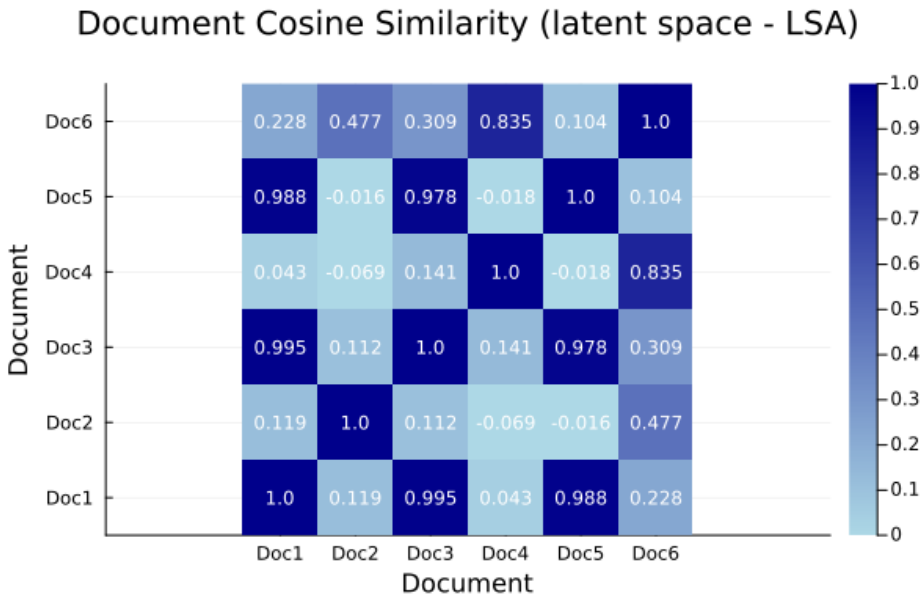

Figure 2: LSA Heatmap

Query Projection and Discussion

Example query: "Radar and object detection"

- Compute query TF $\rightarrow$ TF–IDF (using same idf vector).
- Project to latent: $q_{latent} = \Sigma_k^{-1} U_k^T q_{tfidf}$.
- Compute cosine with each document's latent vector.

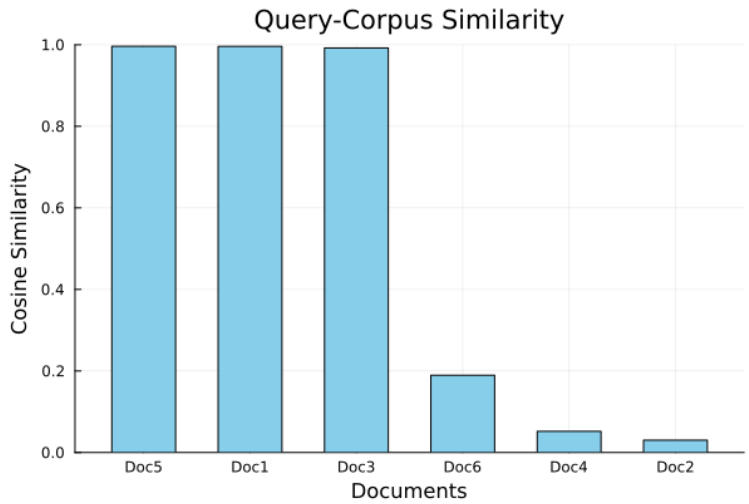

Figure 3: Bar-plot showing similarity between query and corpus's documents

Conclusion

- By utilizing TF–IDF and SVD, LSA uncovers hidden patterns in data, reducing complexity and making it easier to interpret.
- TF-IDF matrix,X: 57×6 dimensions, 6 singular values.
- $X \approx U_k \Sigma_k V_k^T$: 3×6 dimensions, 3 singular values($k = 3$).
- LSA enables clearer comparisons and understanding of documents.
- LSA reveals meaningful relationships between documents, reducing noise for better analysis.

References

Wikipedia Contributors.
Singular value decomposition.
https://en.wikipedia.org/wiki/Singular_value_decomposition, 2025.
Accessed: 2025-10-30.

GeeksforGeeks.
Understanding tf-idf (term frequency–inverse document frequency).
https://www.geeksforgeeks.org/machine-learning/understanding-tf-idf-term-frequency-inverse-document-frequ 2025.
Accessed: 2025-10-29.

Thank you for your time and attention!