Bag of Words, Inverse Document Frequency & Singular Value Decomposition

From raw text to a compact vector space

Andres Mendez-Vazquez

Cinvestav GDL

August 16, 2025

Outline

Introduction

What is Bag of Words?

- Treat a document as an unordered collection of words.
- Discard grammar, word order, and often stop-words.
- Represent each document as a vector $\mathbf{d} \in \mathbb{R}^{|V|}$ where V is the vocabulary.

$$\mathbf{d} = \begin{bmatrix} \#\text{occurrence of } w_1 \\ \#\text{occurrence of } w_2 \\ \vdots \\ \#\text{occurrence of } w_{|V|} \end{bmatrix} = \underbrace{\begin{pmatrix} 0 \\ 2 \\ \vdots \\ 0 \end{pmatrix}}_{\text{Sparse Vector}}$$

About Frequencies

Term Frequency (TF)

$$\mathrm{tf}_{i,j} = \frac{\mathrm{count}(t_i, d_j)}{\sum_k \mathrm{count}(t_k, d_j)}$$

- Normalizes raw counts by document length.
- Still suffers from *common word bias*.

Inverse Document Frequency (IDF)

Equation

$$\mathrm{idf}_i = \log\left(\frac{N}{1 + \mathrm{df}_i}\right)$$

Where

- $N =$ total number of documents,
- $\mathrm{df}_i =$ number of documents containing term t_i.

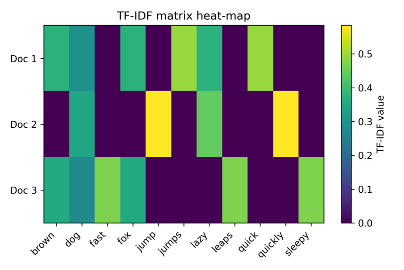
An advantage of this equation is that down-weights ubiquitous words (e.g. "the", "and") and highlights discriminating terms.

TF-IDF

Observation

$$\mathrm{tfidf}_{i,j} = \mathrm{tf}_{i,j} \times \mathrm{idf}_i$$

- Combines local importance (TF) with global rarity (IDF).
- The resulting matrix $\mathbf{X} \in \mathbb{R}^{N\times|V|}$ is often very sparse.



Introduction

Why Reduce Dimensions?

- $|V|$, the vocabulary, can be tens or hundreds of thousands.
- Storage & computation become expensive.
- Many terms are highly correlated $\rightarrow$ redundancy.
- We want a compact, noise-reduced representation.

Singular Value Decomposition

It is a matrix decomposition

$$\mathbf{X} = \mathbf{U}\,\mathbf{\Sigma}\,\mathbf{V}^T$$

- $\mathbf{U} \in \mathbb{R}^{N \times r}$ – basis for the document space.
- $\mathbf{\Sigma} \in \mathbb{R}^{r \times r}$ – singular values (diagonal).
- $\mathbf{V} \in \mathbb{R}^{|V| \times r}$ – basis for the vocabulary space.
  - $r = \mathrm{rank}(\mathbf{X})$ (often $\ll |V|$).

Truncated SVD (Latent Semantic Indexing)

We have that

$$\mathbf{X}_k \approx \mathbf{U}_k \, \mathbf{\Sigma}_k \, \mathbf{V}_k^T$$

- Keep only the top k singular values/vectors.
- k is a hyper-parameter (e.g. 100–300).
- Resulting document vectors: $\mathbf{d}_j^{(k)} = \mathbf{U}_k[j,:]\mathbf{\Sigma}_k$.
- Captures *latent topics* (hence "Latent Semantic Analysis").

Benefits of SVD on TF-IDF

Several of Them

- **Noise filtering**: small singular values often correspond to noise.
- **Synonym/Polysemy resolution**: different words sharing similar contexts merge.
- **Speed**: cosine similarity in low-dim space is cheap.
- **Memory**: store $\mathbf{U}_k$ and $\mathbf{V}_k$ instead of huge $\mathbf{X}$.
- **Computational cost**: full SVD is $O(N|V|^2)$. Use sparse or iterative methods (e.g. Lanczos).
- **Interpretability**: reduced dimensions are not directly interpretable as topics.

Full Workflow

Full Workflow

1. **Pre-processing**
   - Tokenisation, lower-casing, punctuation removal,
   - Optional: stop-word removal, stemming/lemmatisation.
2. **Vocabulary construction**
   - Build list of unique terms.
   - Optionally prune very rare/high-frequency terms.
3. **TF-IDF matrix**

$$X_{ij} = \mathrm{tf}_{i,j} \times \mathrm{idf}_i$$

4. **Dimensionality reduction**

$$X_k \approx U_k \Sigma_k V_k^T$$

5. **Down-stream tasks**
   - Document similarity (cosine similarity on rows of $U_k \Sigma_k$).
   - Classification / clustering (use reduced vectors as features).
   - Information retrieval / search.

Conclusion

Observations

- Bag-of-Words + TF-IDF gives a solid baseline representation.
- SVD (Latent Semantic Indexing) compresses this high-dimensional sparse matrix into a dense, semantically meaningful subspace.
- Together they form the foundation of many classical NLP pipelines.

References

- Salton, G. & Buckley, C. "Term-Weighting Approaches in Automatic Text Retrieval." *Information Retrieval* 1991.
- Deerwester, S. et al. "The Anatomy of a Large-Scale Hypertextual Web Search Engine." *ACM SIGIR* 1990.
- Blei, D. M., Ng, A. Y., & Jordan, M. I. "Latent Dirichlet Allocation." *JMLR* 2003.
- scikit-learn documentation: <https://scikit-learn.org>