Soft Cosine Measure

Implementation Notes

Vít Novotný

Faculty of Informatics, Masaryk University, Brno, Czechia

Introduction

The standard bag-of-words vector space model (VSM) represents documents as real vectors. Documents are expressed in a basis where each basis vector corresponds to a single term, and each coordinate corresponds to the frequency of a term in a document. Consider the documents

d_1 = "When Antony found **Julius Caesar** dead", and
d_2 = "I did enact **Julius Caesar**: I was killed i' the Capitol"

represented in a basis $\{\boldsymbol{\alpha}_i\}_{i=1}^{14}$ of $\mathbb{R}^{14}$, where the basis vectors correspond to the terms in the order of first appearance. Then the corresponding document vectors $\mathbf{v}_1$, and $\mathbf{v}_2$ would have the following coordinates in $\boldsymbol{\alpha}$:

$$(\mathbf{v}_1)_{\boldsymbol{\alpha}} = [1\,1\,1\,\mathbf{1}\,\mathbf{1}\,1\,0\,0\,0\,0\,0\,0\,0\,0]^\mathsf{T}, \text{and}$$
$$(\mathbf{v}_2)_{\boldsymbol{\alpha}} = [0\,0\,0\,\mathbf{1}\,\mathbf{1}\,0\,2\,1\,1\,1\,1\,1\,1\,1]^\mathsf{T}.$$

Assuming $\boldsymbol{\alpha}$ is orthonormal, we can take the inner product of the ℓ^2-normalized vectors $\mathbf{v}_1$, and $\mathbf{v}_2$ to measure the cosine of the angle (i.e. the *cosine similarity*) between the documents d_1, and d_2:

$$\langle \mathbf{v}_1/\|\mathbf{v}_1\|, \mathbf{v}_2/\|\mathbf{v}_2\| \rangle = \frac{\big((\mathbf{v}_1)_{\boldsymbol{\alpha}}\big)^\mathsf{T}(\mathbf{v}_2)_{\boldsymbol{\alpha}}}{\sqrt{\big((\mathbf{v}_1)_{\boldsymbol{\alpha}}\big)^\mathsf{T}(\mathbf{v}_1)_{\boldsymbol{\alpha}}}\sqrt{\big((\mathbf{v}_2)_{\boldsymbol{\alpha}}\big)^\mathsf{T}(\mathbf{v}_2)_{\boldsymbol{\alpha}}}} \approx 0.23.$$

Intuitively, this underestimates the true similarity between d_1, and d_2. Assuming $\boldsymbol{\alpha}$ is orthogonal but not orthonormal, and that the terms Julius, and Caesar are twice as important as the other terms, we can construct a diagonal change-of-basis matrix $\mathbf{W} = (w_{ij})$ from $\boldsymbol{\alpha}$ to an orthonormal basis $\boldsymbol{\beta}$, where w_{ii} corresponds to the importance of a term i. This brings us closer to the true similarity:

$$(\mathbf{v}_1)_{\boldsymbol{\beta}} = \mathbf{W}(\mathbf{v}_1)_{\boldsymbol{\alpha}} = [1\,1\,1\,\mathbf{2}\,\mathbf{2}\,1\,0\,0\,0\,0\,0\,0\,0\,0]^\mathsf{T},$$
$$(\mathbf{v}_2)_{\boldsymbol{\beta}} = \mathbf{W}(\mathbf{v}_2)_{\boldsymbol{\alpha}} = [0\,0\,0\,\mathbf{2}\,\mathbf{2}\,0\,2\,1\,1\,1\,1\,1\,1\,1]^\mathsf{T}, \text{ and}$$
$$\langle \mathbf{v}_1/\|\mathbf{v}_1\|, \mathbf{v}_2/\|\mathbf{v}_2\| \rangle = \frac{\big(\mathbf{W}(\mathbf{v}_1)_{\boldsymbol{\alpha}}\big)^\mathsf{T}\mathbf{W}(\mathbf{v}_2)_{\boldsymbol{\alpha}}}{\sqrt{\big(\mathbf{W}(\mathbf{v}_1)_{\boldsymbol{\alpha}}\big)^\mathsf{T}\mathbf{W}(\mathbf{v}_1)_{\boldsymbol{\alpha}}}\sqrt{\big(\mathbf{W}(\mathbf{v}_2)_{\boldsymbol{\alpha}}\big)^\mathsf{T}\mathbf{W}(\mathbf{v}_2)_{\boldsymbol{\alpha}}}} \approx 0.53.$$

Since we assume that the bases $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are orthogonal, the terms dead and killed contribute nothing to the cosine similarity despite the clear synonymy, because $\langle \boldsymbol{\beta}_{\text{dead}}, \boldsymbol{\beta}_{\text{killed}} \rangle = 0$. In general, the VSM will underestimate the true similarity between documents that carry the same meaning but use different terminology.

In this paper, we further develop the soft VSM described by Sidorov et al. (2014), which does not assume $\boldsymbol{\alpha}$ is orthogonal and which achieved state-of-the-art results on the question answering (QA) task at SemEval 2017. We restate the definition of the soft VSM, we prove a tighter lower worst-case time complexity bound of $\mathcal{O}(n^3)$ for an orthonormalization problem, and we discuss practical implementation in vector databases and inverted indices. We conclude by summarizing our results and suggesting future work.

Computational Complexity

In this section, we restate the definition of the soft VSM as described by Sidorov et al. (2014). We then prove a tighter lower worst-case time complexity bound for computing a change-of-basis matrix to an orthonormal basis. We also prove that under some assumptions, the inner product is a linear-time operation.

Definition 3.1. Let $\mathbb{R}^n$ be the real n-space over $\mathbb{R}$ equipped with the bilinear inner product $\langle \cdot, \cdot \rangle$. Let $\{\boldsymbol{\alpha}_i\}_{i=1}^n$ be the basis of $\mathbb{R}^n$ in which we express our vectors. Let $\mathbf{W}_{\boldsymbol{\alpha}} = (w_{ij})$ be a diagonal change-of-basis matrix from $\boldsymbol{\alpha}$ to a normalized basis $\{\boldsymbol{\beta}_i\}_{i=1}^n$ of $\mathbb{R}^n$, i.e. $\langle \boldsymbol{\beta}_i, \boldsymbol{\beta}_j \rangle \in [-1, 1], \langle \boldsymbol{\beta}_i, \boldsymbol{\beta}_i \rangle = 1$. Let $\mathbf{S}_{\boldsymbol{\beta}} = (s_{ij})$ be the metric matrix of $\mathbb{R}^n$ w.r.t. $\boldsymbol{\beta}$, i.e. $s_{ij} = \langle \boldsymbol{\beta}_i, \boldsymbol{\beta}_j \rangle$. Then $(\mathbb{R}^n, \mathbf{W}_{\boldsymbol{\alpha}}, \mathbf{S}_{\boldsymbol{\beta}})$ is a *soft* VSM.

Theorem 3.2. *Let $G = (\mathbb{R}^n, \mathbf{W}_{\boldsymbol{\alpha}}, \mathbf{S}_{\boldsymbol{\beta}})$ be a soft* VSM. *Then a change-of-basis matrix $\mathbf{E}$ from the basis $\boldsymbol{\beta}$ to an orthonormal basis of $\mathbb{R}^n$ can be computed in time $\mathcal{O}(n^3)$.*

Proof. By definition, $\mathbf{S} = \mathbf{E}\mathbf{E}^\mathsf{T}$ for any change-of-basis matrix $\mathbf{E}$ from the basis $\boldsymbol{\beta}$ to an orthonormal basis. Since $\mathbf{S}$ contains inner products of linearly independent vectors $\boldsymbol{\beta}$, it is Gramian and positive definite. The Gramianness of $\mathbf{S}$ also implies its symmetry. Therefore, a lower triangular $\mathbf{E}$ is uniquely determined by the Cholesky factorization of the symmetric positive-definite $\mathbf{S}$, which we can compute in time $\mathcal{O}(n^3)$. □

Remark. See **Table 1** for an experimental comparison.

Lemma 3.3. *Let $G = (\mathbb{R}^n, \mathbf{W}_{\boldsymbol{\alpha}}, \mathbf{S}_{\boldsymbol{\beta}})$ be a soft* VSM. *Let $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$. Then $\langle \mathbf{x}, \mathbf{y} \rangle = (\mathbf{W}(\mathbf{x})_{\boldsymbol{\alpha}})^\mathsf{T}\mathbf{S}\mathbf{W}(\mathbf{y})_{\boldsymbol{\alpha}}$.*

Proof. Let $\mathbf{E}$ be the change-of-basis matrix from the basis $\boldsymbol{\beta}$ to an orthonormal basis $\boldsymbol{\gamma}$ of $\mathbb{R}^n$. Then:

$$\begin{aligned}\langle \mathbf{x}, \mathbf{y} \rangle &= \big((\mathbf{x})_{\boldsymbol{\gamma}}\big)^\mathsf{T}(\mathbf{y})_{\boldsymbol{\gamma}} = \big(\mathbf{E}(\mathbf{x})_{\boldsymbol{\beta}}\big)^\mathsf{T}\mathbf{E}(\mathbf{y})_{\boldsymbol{\beta}} = \big(\mathbf{E}\mathbf{W}(\mathbf{x})_{\boldsymbol{\alpha}}\big)^\mathsf{T}\mathbf{E}\mathbf{W}(\mathbf{y})_{\boldsymbol{\alpha}} \\ &= \left(\sum_{i=1}^n (\boldsymbol{\alpha}_i)_{\boldsymbol{\gamma}} \cdot w_{ii} \cdot (x_i)_{\boldsymbol{\alpha}}\right) \cdot \left(\sum_{j=1}^n (\boldsymbol{\alpha}_j)_{\boldsymbol{\gamma}} \cdot w_{jj} \cdot (y_j)_{\boldsymbol{\alpha}}\right) \\ &= \sum_{i=1}^n \sum_{j=1}^n w_{ii} \cdot (x_i)_{\boldsymbol{\alpha}} \cdot \langle \boldsymbol{\alpha}_i, \boldsymbol{\alpha}_j \rangle \cdot w_{jj} \cdot (y_j)_{\boldsymbol{\alpha}} \\ &= \sum_{i=1}^n \sum_{j=1}^n w_{ii} \cdot (x_i)_{\boldsymbol{\alpha}} \cdot s_{ij} \cdot w_{jj} \cdot (y_j)_{\boldsymbol{\alpha}} = \big(\mathbf{W}(\mathbf{x})_{\boldsymbol{\alpha}}\big)^\mathsf{T}\mathbf{S}\mathbf{W}(\mathbf{y})_{\boldsymbol{\alpha}}. \quad \square\end{aligned}$$

Remark. From here, we can directly derive the cosine of the angle between $\mathbf{x}$ and $\mathbf{y}$ (i.e. what Sidorov et al. (2014) call the SCM) as follows:

$$\langle \mathbf{x}/\|\mathbf{x}\|, \mathbf{y}/\|\mathbf{y}\| \rangle = \frac{\big(\mathbf{W}(\mathbf{x})_{\boldsymbol{\alpha}}\big)^\mathsf{T}\mathbf{S}\mathbf{W}(\mathbf{y})_{\boldsymbol{\alpha}}}{\sqrt{\big(\mathbf{W}(\mathbf{x})_{\boldsymbol{\alpha}}\big)^\mathsf{T}\mathbf{S}\mathbf{W}(\mathbf{x})_{\boldsymbol{\alpha}}}\sqrt{\big(\mathbf{W}(\mathbf{y})_{\boldsymbol{\alpha}}\big)^\mathsf{T}\mathbf{S}\mathbf{W}(\mathbf{y})_{\boldsymbol{\alpha}}}}.$$

The SCM is actually the starting point for Charlet and Damnati (2017), who propose matrices $\mathbf{S}$ that are not necessarily metric. If, like them, we are only interested in computing the SCM, then we only require that the square roots remain real, i.e. that $\mathbf{x} \neq 0 \implies (\mathbf{W}(\mathbf{x})_{\boldsymbol{\alpha}})^\mathsf{T}\mathbf{S}\mathbf{W}(\mathbf{x})_{\boldsymbol{\alpha}} \geq 0$. For arbitrary $\mathbf{x} \in \mathbb{R}^n$, this holds iff $\mathbf{S}$ is positive semi-definite. However, since the coordinates $(\mathbf{x})_{\boldsymbol{\alpha}}$ correspond to non-negative term frequencies, it is sufficient that $\mathbf{W}$ and $\mathbf{S}$ are non-negative as well. If we are only interested in computing the inner product, then $\mathbf{S}$ can be arbitrary.

Table 1: The real time to compute a change-of-basis matrix $\mathbf{E}$ from a dense matrix $\mathbf{S}$ averaged over 100 iterations. We used two Intel Xeon E5-2650 v2 processors to evaluate the $\mathcal{O}(n^3)$ Cholesky factorization from NumPy 1.14.3, and the $\mathcal{O}(n^4)$ iterated Gaussian elimination from LAPACK. For $n > 1000$, only sparse $\mathbf{S}$ seem practical.

n terms	Algorithm	Real computation time
100	Cholesky factorization	0.0006 sec (0.606 ms)
100	Gaussian elimination	0.0529 sec (52.893 ms)
500	Cholesky factorization	0.0086 sec (8.640 ms)
500	Gaussian elimination	22.7361 sec (22.736 sec)
1000	Cholesky factorization	0.0304 sec (30.378 ms)
1000	Gaussian elimination	354.2746 sec (5.905 min)

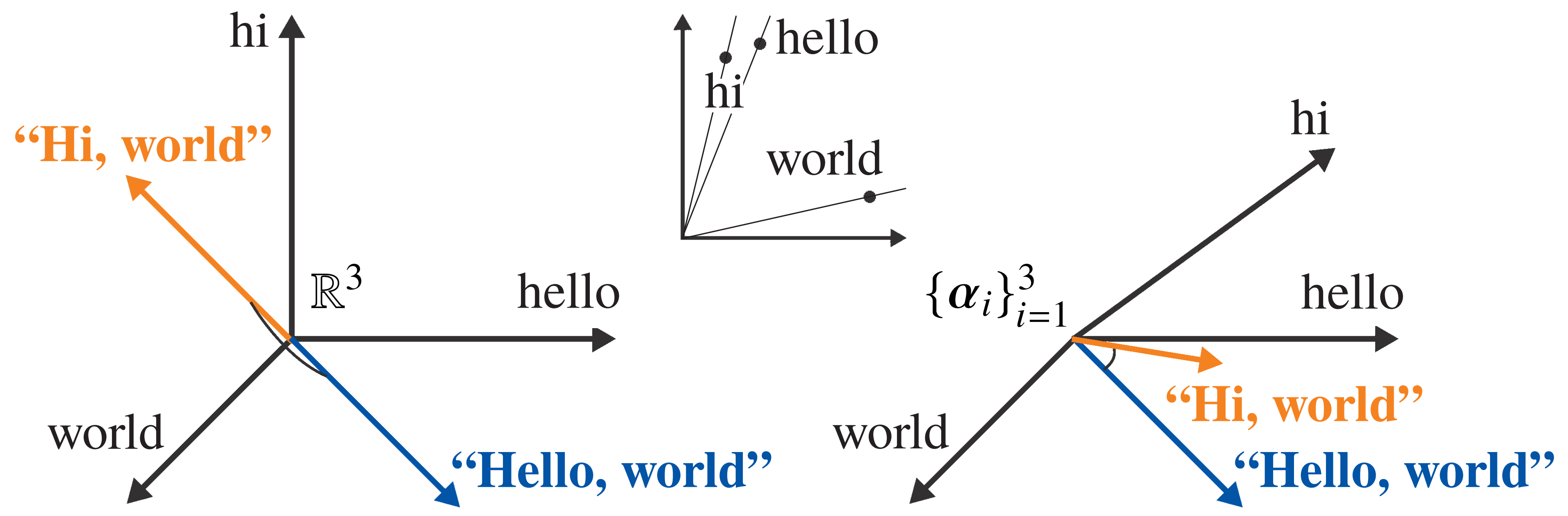

Figure 1: Vectors of documents d_3 = "Hi, world", and d_4 = "Hello, world" assuming the coordinates are in an orthonormal basis (left), and in a non-orthogonal basis $\{\boldsymbol{\alpha}_i\}_{i=1}^3$ (right). The inner product between the basis vectors $\boldsymbol{\alpha}_{\text{hi}}, \boldsymbol{\alpha}_{\text{hello}}$, and $\boldsymbol{\alpha}_{\text{world}}$ is derived from the cosine similarity of word2vec word embeddings (middle).

Theorem 3.4. *Let $G = (\mathbb{R}^n, \mathbf{W}_{\boldsymbol{\alpha}}, \mathbf{S}_{\boldsymbol{\beta}})$ be a soft* VSM *such that no column of $\mathbf{S}$ contains more than C non-zero elements, where C is a constant. Let $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ and let m be the number of non-zero elements in $(\mathbf{x})_{\boldsymbol{\beta}}$. Then $\langle \mathbf{x}, \mathbf{y} \rangle$ can be computed in time $\mathcal{O}(m)$.*

Proof. Assume that $(\mathbf{x})_{\boldsymbol{\alpha}}, (\mathbf{y})_{\boldsymbol{\alpha}}$, and $\mathbf{S}$ are represented by data structures with constant-time column access and non-zero element traversal, e.g. compressed sparse column (CSC) matrices. Further assume $\mathbf{W}$ is represented by an array containing the main diagonal of $\mathbf{W}$. Then **Algorithm 1** computes $\big(\mathbf{W}(\mathbf{x})_{\boldsymbol{\alpha}}\big)^\mathsf{T}\mathbf{S}\mathbf{W}(\mathbf{y})_{\boldsymbol{\alpha}}$ in time $\mathcal{O}(m)$, which by **Lemma 3.3**, corresponds to $\langle \mathbf{x}, \mathbf{y} \rangle$. □

Algorithm 1 The inner product of $\mathbf{x}$ and $\mathbf{y}$

```
r ← 0
for each i such that $(x_i)_\alpha$ is non-zero do                ▹ = m iterations
    for each j such that $s_{ij}$ is non-zero do                   ▹ ≤ C iterations
        $r \leftarrow r + w_{ii} \cdot (x_i)_\alpha \cdot s_{ij} \cdot w_{jj} \cdot (y_j)_\alpha$
return r
```

Remark. Similarly, we can show that if a column of $\mathbf{S}$ contains C non-zero elements on average, $\langle \mathbf{x}, \mathbf{y} \rangle$ has the average-case time complexity of $\mathcal{O}(m)$. Note also that most information retrieval systems impose a limit on the length of a query document. Therefore, m is usually bounded by a constant and $\mathcal{O}(m) = \mathcal{O}(1)$.

Since we are usually interested in the inner products of all document pairs in two corpora (e.g. one containing queries and the other actual documents), we can achieve significant speed improvements with vector processors by computing $(\mathbf{W}\mathbf{X})^\mathsf{T}\mathbf{S}\mathbf{W}\mathbf{Y}$, where $\mathbf{X}$, and $\mathbf{Y}$ are *corpus matrices* containing the coordinates of document vectors in the basis $\boldsymbol{\alpha}$ as columns. To compute the SCM, we first need to normalize the document vectors by performing an entrywise division of every column in $\mathbf{X}$ by diag $\sqrt{(\mathbf{W}\mathbf{X})^\mathsf{T}\mathbf{S}\mathbf{W}\mathbf{X}} = \sqrt{(\mathbf{W}\mathbf{X})^\mathsf{T}\mathbf{S} \circ (\mathbf{W}\mathbf{X})^\mathsf{T}}$, where $\circ$ denotes entrywise product. $\mathbf{Y}$ is normalized analogously.

There are several strategies for making no column of $\mathbf{S}$ contain more than C non-zero elements. If we do not require that $\mathbf{S}$ is metric (e.g. because we only wish to compute the inner product, or the SCM), a simple strategy is to start with an empty matrix, and to insert the $C - 1$ largest elements and the diagonal element from every column of $\mathbf{S}$. However, the resulting matrix will likely be asymmetric, which makes the inner product formula asymmetric as well. We can regain symmetry by always inserting an element s_{ij} together with the element s_{ji} and only if this does not make the column j contain more than C non-zero elements. This strategy is greedy, since later columns contain non-zero elements inserted by earlier columns. Our preliminary experiments suggest that processing colums that correspond to increasingly frequent terms performs best on the task of Charlet and Damnati (2017). Finally, by limiting the sum of all non-diagonal elements in a column to be less than one, we can make $\mathbf{S}$ strictly diagonally dominant and therefore positive definite, which enables us to compute $\mathbf{E}$ through Cholesky factorization.

Implementation in Vector Databases and Inverted Indices

In this section, we present coordinate transformations for retrieving nearest document vectors from general-purpose vector databases such as Annoy, or Faiss. We also discuss the implementation in the inverted indices of text search engines such as Apache Lucene, or Elasticsearch.

Remark. With a vector database, we can transform document vectors to an orthonormal basis $\boldsymbol{\gamma}$. In the transformed coordinates, the dot product $((\mathbf{x})_{\boldsymbol{\gamma}})^\mathsf{T}(\mathbf{y})_{\boldsymbol{\gamma}}$ corresponds to the inner product $\langle \mathbf{x}, \mathbf{y} \rangle$ and the cosine similarity corresponds to the cosine of an angle $\langle \mathbf{x}/\|\mathbf{x}\|, \mathbf{y}/\|\mathbf{y}\| \rangle$ (i.e. the soft cosine measure). A vector database that supports nearest neighbor search according to either the dot product, or the cosine similarity will therefore retrieve vectors expressed in $\boldsymbol{\gamma}$ according to either the inner product, or the soft cosine measure. We can compute a change-of-basis matrix $\mathbf{E}$ of order n in time $\mathcal{O}(n^3)$ by **Theorem 3.2** and use it to transform every vector $\mathbf{x} \in \mathbb{R}^n$ to $\boldsymbol{\gamma}$ by computing $\mathbf{E}\mathbf{W}(\mathbf{x})_{\boldsymbol{\alpha}}$. However, this approach requires that $\mathbf{S}$ is symmetric positive-definite and that we recompute $\mathbf{E}$, and reindex the vector database each time $\mathbf{S}$ has changed. We will now discuss transformations that do not require $\mathbf{E}$ and for which a non-negative $\mathbf{S}$ is sufficient as discussed in the remark for **Lemma 3.3**.

Theorem 4.1. *Let $G = (\mathbb{R}^n, \mathbf{W}_{\boldsymbol{\alpha}}, \mathbf{S}_{\boldsymbol{\beta}})$ be a soft* VSM. *Let $\mathbf{x}, \mathbf{x}', \mathbf{y} \in \mathbb{R}^n$ such that $(\mathbf{x}')_{\boldsymbol{\beta}} = \mathbf{S}^\mathsf{T}(\mathbf{x})_{\boldsymbol{\beta}}$. Then $\langle \mathbf{x}, \mathbf{y} \rangle = ((\mathbf{x}')_{\boldsymbol{\beta}})^\mathsf{T}(\mathbf{y})_{\boldsymbol{\beta}}$.*

Proof. $\big((\mathbf{x}')_{\boldsymbol{\beta}}\big)^\mathsf{T}(\mathbf{y})_{\boldsymbol{\beta}} = \big((\mathbf{x})_{\boldsymbol{\beta}}\big)^\mathsf{T}\mathbf{S}(\mathbf{y})_{\boldsymbol{\beta}} = \langle \mathbf{x}, \mathbf{y} \rangle$ from **Lemma 3.3**. □

Remark. By transforming a query vector $\mathbf{x}$ into $(\mathbf{x}')_{\boldsymbol{\beta}}$, we can retrieve documents according to the inner product in vector databases that only support nearest neighbor search according to the dot product. Note that we do not introduce $\mathbf{S}$ into $(\mathbf{y})_{\boldsymbol{\beta}}$, which allows us to change $\mathbf{S}$ without changing the documents in a vector database and that $\mathbf{S}$ can be arbitrary as discussed in the remark for **Lemma 3.3**.

Theorem 4.2. *Let $G = (\mathbb{R}^n, \mathbf{W}_{\boldsymbol{\alpha}}, \mathbf{S}_{\boldsymbol{\beta}})$ be a soft* VSM. *Let $\mathbf{x}, \mathbf{x}', \mathbf{y}, \mathbf{y}', \mathbf{z}, \mathbf{z}' \in \mathbb{R}^n$ such that $\mathbf{x}, \mathbf{y}, \mathbf{z} \neq 0, (\mathbf{x}')_{\boldsymbol{\beta}} = \mathbf{S}^\mathsf{T}(\mathbf{x})_{\boldsymbol{\beta}}, (\mathbf{y}')_{\boldsymbol{\beta}} = \frac{(\mathbf{y})_{\boldsymbol{\beta}}}{\sqrt{((\mathbf{y})_{\boldsymbol{\beta}})^\mathsf{T}\mathbf{S}(\mathbf{y})_{\boldsymbol{\beta}}}}$, and $(\mathbf{z}')_{\boldsymbol{\beta}} = \frac{(\mathbf{z})_{\boldsymbol{\beta}}}{\sqrt{((\mathbf{z})_{\boldsymbol{\beta}})^\mathsf{T}\mathbf{S}(\mathbf{z})_{\boldsymbol{\beta}}}}$. Then $\langle \mathbf{x}/\|\mathbf{x}\|, \mathbf{y}/\|\mathbf{y}\| \rangle \leq \langle \mathbf{x}/\|\mathbf{x}\|, \mathbf{z}/\|\mathbf{z}\| \rangle$ iff $\big((\mathbf{x}')_{\boldsymbol{\beta}}\big)^\mathsf{T}(\mathbf{y}')_{\boldsymbol{\beta}} \leq \big((\mathbf{x}')_{\boldsymbol{\beta}}\big)^\mathsf{T}(\mathbf{z}')_{\boldsymbol{\beta}}$.*

Proof. $\big((\mathbf{x}')_{\boldsymbol{\beta}}\big)^\mathsf{T}(\mathbf{y}')_{\boldsymbol{\beta}} = \frac{((\mathbf{x})_{\boldsymbol{\beta}})^\mathsf{T}\mathbf{S}(\mathbf{y})_{\boldsymbol{\beta}}}{\sqrt{((\mathbf{y})_{\boldsymbol{\beta}})^\mathsf{T}\mathbf{S}(\mathbf{y})_{\boldsymbol{\beta}}}}$. By **Lemma 3.3**, this equals $\langle \mathbf{x}/\|\mathbf{x}\|, \mathbf{y}/\|\mathbf{y}\| \rangle$ except for the missing term $\sqrt{\big((\mathbf{x})_{\boldsymbol{\beta}}\big)^\mathsf{T}\mathbf{S}(\mathbf{x})_{\boldsymbol{\beta}}}$ in the divisor. The term is constant in both $\langle \mathbf{x}/\|\mathbf{x}\|, \mathbf{y}/\|\mathbf{y}\| \rangle$, and $\langle \mathbf{x}/\|\mathbf{x}\|, \mathbf{z}/\|\mathbf{z}\| \rangle$, so ordering is preserved. □

Remark. By transforming a query vector $\mathbf{x}$ into $(\mathbf{x}')_{\boldsymbol{\beta}}$ and document vectors $\mathbf{y}$ into $(\mathbf{y}')_{\boldsymbol{\beta}}$, we can retrieve documents according to the SCM in vector databases that only support nearest neighbor search according to the dot product.

Theorem 4.3. *Let $G = (\mathbb{R}^n, \mathbf{W}_{\boldsymbol{\alpha}}, \mathbf{S}_{\boldsymbol{\beta}})$ be a soft* VSM *s.t. $\mathbf{S}_{\boldsymbol{\beta}}$ is non-negative. Let $\mathbf{x}, \mathbf{y}, \mathbf{y}', \mathbf{z}, \mathbf{z}' \in \mathbb{R}^n$, and $\mathbf{x}', \mathbf{y}'', \mathbf{z}'' \in \mathbb{R}^{n+1}$ s.t. $\mathbf{x} \neq 0, \mathbf{y}, \mathbf{z} > 0$,*

$$(\mathbf{x}')_{\boldsymbol{\beta}'} = \left[\frac{\mathbf{S}^\mathsf{T}(\mathbf{x})_{\boldsymbol{\beta}}}{\sqrt{\big(\mathbf{S}^\mathsf{T}(\mathbf{x})_{\boldsymbol{\beta}}\big)^\mathsf{T}\mathbf{S}^\mathsf{T}(\mathbf{x})_{\boldsymbol{\beta}}}} \quad 0\right]^\mathsf{T}, (\mathbf{y}')_{\boldsymbol{\beta}} = \frac{(\mathbf{y})_{\boldsymbol{\beta}}}{\sqrt{\big((\mathbf{y})_{\boldsymbol{\beta}}\big)^\mathsf{T}\mathbf{S}(\mathbf{y})_{\boldsymbol{\beta}}}},$$

$$(\mathbf{y}'')_{\boldsymbol{\beta}'} = \left[\big((\mathbf{y}')_{\boldsymbol{\beta}}\big)^\mathsf{T} \quad \sqrt{1 - \big((\mathbf{y}')_{\boldsymbol{\beta}}\big)^\mathsf{T}(\mathbf{y}')_{\boldsymbol{\beta}}}\right]^\mathsf{T}, (\mathbf{z}')_{\boldsymbol{\beta}} = \frac{(\mathbf{z})_{\boldsymbol{\beta}}}{\sqrt{\big((\mathbf{z})_{\boldsymbol{\beta}}\big)^\mathsf{T}\mathbf{S}(\mathbf{z})_{\boldsymbol{\beta}}}}, \text{ and}$$

$$(\mathbf{z}'')_{\boldsymbol{\beta}'} = \left[\big((\mathbf{z}')_{\boldsymbol{\beta}}\big)^\mathsf{T} \quad \sqrt{1 - \big((\mathbf{z}')_{\boldsymbol{\beta}}\big)^\mathsf{T}(\mathbf{z}')_{\boldsymbol{\beta}}}\right]^\mathsf{T}, \text{ where } \boldsymbol{\beta}' = \boldsymbol{\beta} \cup \{[0 \ldots 0\ 1]^\mathsf{T} \in \mathbb{R}^{n+1}\}.$$

Then $\langle \mathbf{x}/\|\mathbf{x}\|, \mathbf{y}/\|\mathbf{y}\| \rangle \leq \langle \mathbf{x}/\|\mathbf{x}\|, \mathbf{z}/\|\mathbf{z}\| \rangle$ iff

$$\frac{\big((\mathbf{x}')_{\boldsymbol{\beta}'}\big)^\mathsf{T}(\mathbf{y}'')_{\boldsymbol{\beta}'}}{\sqrt{\big((\mathbf{x}')_{\boldsymbol{\beta}'}\big)^\mathsf{T}(\mathbf{x}')_{\boldsymbol{\beta}'}}\sqrt{\big((\mathbf{y}'')_{\boldsymbol{\beta}'}\big)^\mathsf{T}(\mathbf{y}'')_{\boldsymbol{\beta}'}}} \leq \frac{\big((\mathbf{x}')_{\boldsymbol{\beta}'}\big)^\mathsf{T}(\mathbf{z}'')_{\boldsymbol{\beta}'}}{\sqrt{\big((\mathbf{x}')_{\boldsymbol{\beta}'}\big)^\mathsf{T}(\mathbf{x}')_{\boldsymbol{\beta}'}}\sqrt{\big((\mathbf{z}'')_{\boldsymbol{\beta}'}\big)^\mathsf{T}(\mathbf{z}'')_{\boldsymbol{\beta}'}}}.$$

Proof. $\big((\mathbf{x}')_{\boldsymbol{\beta}'}\big)^\mathsf{T}(\mathbf{x}')_{\boldsymbol{\beta}'} = 1$. Since $\mathbf{S}$ is non-negative, and $(\mathbf{y})_{\boldsymbol{\beta}} > 0$, $\sqrt{\big((\mathbf{y})_{\boldsymbol{\beta}}\big)^\mathsf{T}\mathbf{S}(\mathbf{y})_{\boldsymbol{\beta}}} \geq \sqrt{\big((\mathbf{y})_{\boldsymbol{\beta}}\big)^\mathsf{T}(\mathbf{y})_{\boldsymbol{\beta}}}$ and therefore $\big((\mathbf{y}')_{\boldsymbol{\beta}'}\big)^\mathsf{T}(\mathbf{y}')_{\boldsymbol{\beta}'} \leq 1$, and $\big((\mathbf{y}'')_{\boldsymbol{\beta}'}\big)^\mathsf{T}(\mathbf{y}'')_{\boldsymbol{\beta}'} = 1$. Therefore:

$$\frac{\big((\mathbf{x}')_{\boldsymbol{\beta}'}\big)^\mathsf{T}(\mathbf{y}'')_{\boldsymbol{\beta}'}}{\sqrt{\big((\mathbf{x}')_{\boldsymbol{\beta}'}\big)^\mathsf{T}(\mathbf{x}')_{\boldsymbol{\beta}'}}\sqrt{\big((\mathbf{y}'')_{\boldsymbol{\beta}'}\big)^\mathsf{T}(\mathbf{y}'')_{\boldsymbol{\beta}'}}} = \big((\mathbf{x}')_{\boldsymbol{\beta}'}\big)^\mathsf{T}(\mathbf{y}'')_{\boldsymbol{\beta}'}$$
$$= \frac{\big((\mathbf{x})_{\boldsymbol{\beta}}\big)^\mathsf{T}\mathbf{S}(\mathbf{y})_{\boldsymbol{\beta}}}{\sqrt{\big(\mathbf{S}^\mathsf{T}(\mathbf{x})_{\boldsymbol{\beta}}\big)^\mathsf{T}\mathbf{S}^\mathsf{T}(\mathbf{x})_{\boldsymbol{\beta}}}\sqrt{\big((\mathbf{y})_{\boldsymbol{\beta}}\big)^\mathsf{T}\mathbf{S}(\mathbf{y})_{\boldsymbol{\beta}}}}.$$

By **Lemma 3.3**, this equals $\langle \mathbf{x}/\|\mathbf{x}\|, \mathbf{y}/\|\mathbf{y}\| \rangle$ except for the missing term $\sqrt{\big((\mathbf{x})_{\boldsymbol{\beta}}\big)^\mathsf{T}\mathbf{S}(\mathbf{x})_{\boldsymbol{\beta}}}$, and the extra term $\sqrt{\big(\mathbf{S}^\mathsf{T}(\mathbf{x})_{\boldsymbol{\beta}}\big)^\mathsf{T}\mathbf{S}^\mathsf{T}(\mathbf{x})_{\boldsymbol{\beta}}}$ in the divisor. These are constant in $\langle \mathbf{x}/\|\mathbf{x}\|, \mathbf{y}/\|\mathbf{y}\| \rangle$, and $\langle \mathbf{x}/\|\mathbf{x}\|, \mathbf{z}/\|\mathbf{z}\| \rangle$, so ordering is preserved. □

Remark. By transforming a query vector $\mathbf{x}$ into $(\mathbf{x}')_{\boldsymbol{\beta}'}$ and document vectors $\mathbf{y}$ into $(\mathbf{y}'')_{\boldsymbol{\beta}'}$, we can retrieve documents according to the SCM in vector databases that only support nearest neighbor search according to the cosine similarity.

Whereas most vector databases are designed for storing low-dimensional and dense vector coordinates, document vectors have the dimension n, which can be in the millions for real-world corpora such as the English Wikipedia. Apart from that, a document contains only a small fraction of the terms in the vocabulary, which makes the coordinates extremely sparse. Therefore, the coordinates need to be converted to a dense low-dimensional representation, using e.g. the latent semantic analysis (LSA), before they are stored in a database or used for queries.

Unlike vector databases, inverted-index-based search engines are built around a data structure called the *inverted index*, which maps each term in our vocabulary to a list of documents (a *posting*) containing the term. Documents in a posting are sorted by a common criterion. The search engine tokenizes a text query into terms, retrieves postings for the query terms, and then traverses the postings, computing similarity between the query and the documents.

We can directly replace the search engine's document similarity formula with the formula for the inner product from **Lemma 3.3**, or the formula for the SCM. After this straightforward change, the system will still only retrieve documents that have at least one term in common with the query. Therefore, we first need to *expand* the query vector $\mathbf{x}$ by computing $((\mathbf{x})_{\boldsymbol{\beta}})^\mathsf{T}\mathbf{S}$ and retrieving postings for all terms corresponding to the nonzero coordinates in the expanded vector. The expected number of these terms is $\mathcal{O}(mC)$, where m is the number of non-zero elements in $(\mathbf{x})_{\boldsymbol{\alpha}}$, and C is the maximum number of non-zero elements in any column of $\mathbf{S}$. Assuming m and C are bounded by a constant, $\mathcal{O}(mC) = \mathcal{O}(1)$.

Conclusion and Future Work

In this paper, we examined the soft vector space model (VSM) of Sidorov et al. (2014). We restated the definition, we proved a tighter lower time complexity bound of $\mathcal{O}(n^3)$ for a related orthonormalization problem, and we showed how the inner product, and the soft cosine measure between document vectors can be efficiently computed in general-purpose vector databases, in the inverted indices of text search engines, and in other applications. To complement this paper, we also provided an implementation of the SCM to Gensim,[a] a free open-source natural language processing library.

In our remarks for **Theorem 3.4**, we discuss strategies for making no column of matrix $\mathbf{S}$ contain more than C non-zero elements. Future research will evaluate their performance on the semantic text similarity task with public datasets. Various choices of the matrix $\mathbf{S}$ based on word embeddings, Levenshtein distance, thesauri, and statistical regression as well as metric matrices from previous work will also be evaluated both amongst themselves and against other document similarity measures such as the LDA, LSA, and WMD.

[a] See https://github.com/RaRe-Technologies/gensim/, pull requests 1827, and 2016.

Acknowledgements

We gratefully acknowledge the support by TAČR under the Omega program, project TD03000295.

Bibliography

NOVOTNÝ, Vít. Implementation Notes for the Soft Cosine Measure. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM '18)*. Torino, Italy: Association for Computing Machinery, 2018. 4 pp. ISBN 978-1-4503-6014-2. doi:10.1145/3269206.3269317.