UNIVERSITÉ DES SCIENCES ET TECHNOLOGIE HOUARI BOUMEDIENE

FACULTÉ D'INFORMATIQUE - MASTER 2 SII

RECHERCHE D'INFORMATION
INFORMATION RETRIEVAL

CHAPITRE 5: Modèle LSI - Latent Semantic Indexing

27 octobre 2025

I. INTRODUCTION

Problématique des modèles basés sur les termes

- Les modèles de RI classiques (booléen, vectoriel…) dépendent des mots exacts.
- Si un document ne contient pas les mêmes termes que la requête, il ne sera pas retrouvé, même s'il traite du même sujet (ex. *voiture / automobile,* élève/étudiant, livre/ouvrage, …etc.).

Objectif de LSI

Le modèle LSI : Latent Semantic Indexing (Latent → Hidden, Semantic → Relationships between words, Indexing → Information retrieval), propose une solution pour cette problématique, en regroupant les termes sémantiquement reliés (co-occurrence), dans un même concept et faire une recherche par concept.

I. INTRODUCTION

Ce modèle propose alors :

- Une indexation par concept selon la co-occurrence entre termes
- Un calcul de similarité par concept entre document/requête
- Cherche à comprendre **la sémantique latente (Sémantique cachée-Hidden)** des mots à travers leurs co-occurrences dans les documents.

II. PRINCIPE GÉNÉRAL

- LSI est une approche vectorielle
- Exploite les co-occurrences pour Identifier les **concepts latents (caché)** entre termes pour identifier les relations sémantiques.
- Réduire la dimension du vocabulaire (espace des termes), en regroupant les termes co-occurrents (similaires) dans les mêmes dimensions
- Les documents et les requêtes sont alors représentés dans espace plus réduit, composé de concepts de haut niveau (plutôt que de termes).

II. PRINCIPE GÉNÉRAL

MATRICE W–TERME x DOCUMENT

- La matrice **W** ou **TERME x DOCUMENT** est la base de tous les calculs du modèle LSI

- Elle représente le contenu brut de la collection

Term	d_1	d_2	d_3	d_4
t_1	1			
t_2	1	1		
t_3	1	1		
t_4	1			
t_5	3			
t_6	1	1	1	1
t_7	1			
t_8	1	1		
t_9	4	1	3	2
t_{10}	1			
t_{11}	1	2	1	1
t_{12}	2	1		
t_{13}	1	2		
t_{14}	1	2	1	
t_{15}	3	2	1	1
t_{16}	1			
t_{17}	1			
t_{18}				
tM	…	…	….	…

- Les co-occurrences de termes permettent de repérer des corrélations.

II. PRINCIPE GÉNÉRAL

COMMENT DÉTECTER CES CONCEPTS LATENTS

- L'idée est donc d'observer **comment les termes co-apparaissent/co-occurrent dans les documents**.
- Deux termes qui apparaissent souvent dans les mêmes documents sont **corrélés** → ils partagent probablement un même sens ou contexte. Ces corrélations sont contenues dans la **matrice terme-document.**
- La traduction mathématique de cela est de factoriser la matrice **terme-document , "décomposer"** la matrice en plusieurs couches d'information.

III. DE LA MATRICE TERME x DOCUMENT AU CONCEPT : SVD

Afin de trouver les concepts basés sur les co-occurrences entre termes, le modèle LSI utilise la technique de la décomposition d'une matrice en valeurs singulières (**SVD : Singular Value Decomposition**).

Avant de voir le détail du modèle LSI, il faut bien comprendre la technique SVD.

La technique SVD se base sur la notion des valeurs propres et des vecteurs propres d'une matrice.

III.1. RAPPEL MATHEMATIQUE: Transposé, Déterminant et inverse

1) Transposé d'une matrice

La transposée d'une matrice A, notée $\boldsymbol{A^T}$,s'obtient en **inversant les lignes et les colonnes** :

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \Rightarrow A^T = \begin{pmatrix} a & c \\ b & d \end{pmatrix}$$

Exemple :

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \Rightarrow A^T = \begin{pmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{pmatrix}$$

III.1. RAPPEL MATHEMATIQUE: Transposé, Déterminant et inverse

2) Déterminant d'une matrice

Soit une matrice carrée **A nxn**, son **déterminant** est noté par | $\boldsymbol{A}$ | ou $\mathbf{det}(\mathbf{A})$

On peut le calculer en **développant selon une ligne ou une colonne** :

$$\det(A) = \sum_{j=1}^{n} (-1)^{i+j} \, a_{ij} \, \det(A_{ij})$$

où :

- i: ligne choisie pour le développement
- a_{ij} : élément de la matrice sur la ligne i, colonne j
- A_{ij} : la sous-matrice obtenue en **supprimant** la ligne i et la colonne j
- $(-1)^{i+j}$ **facteur de signe** (positif ou négatif selon la position de (a_{ij})

III.1. RAPPEL MATHEMATIQUE: Transposé, Déterminant et inverse

2) Déterminant d'une matrice

Exemple 1: **Soit :** $A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$, Calculons **det(A)**

Prenons par exemple **la première ligne** (on aurait pu prendre n'importe laquelle).

Formule du développement :

$$\det(A) = a_{11}(-1)^{1+1}\det(A_{11}) + a_{12}(-1)^{1+2}\det(A_{12})$$

A11 : on supprime la **1re ligne** et la **1re colonne**, det(A11)=a22
A12 : on supprime la **1re ligne** et la **2e colonne**, det(A12)=a21

Donc:

$$\det(A) = a_{11}(+1)a_{22} + a_{12}(-1)a_{21}$$

$$\boxed{\det(A) = a_{11}a_{22} - a_{12}a_{21}}$$

III.1. RAPPEL MATHEMATIQUE: Transposé, Déterminant et inverse

2) Déterminant d'une matrice

Exemple numérique :

$$A = \begin{pmatrix} 2 & 3 \\ 4 & 5 \end{pmatrix}$$

Alors :

$$\det(A) = 2 \times 5 - 3 \times 4 = 10 - 12 = -2$$

III.1. RAPPEL MATHEMATIQUE: Transposé, Déterminant et inverse

2) Déterminant d'une matrice

Exemple 3:

Soit : $A \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$, Calculons **det(A)**

Prenons par exemple **la première ligne**

Formule du développement :

$$\det(A) = (-1)^{1+1} a_{11} \det(A_{11}) + (-1)^{1+2} a_{12} \det(A_{12}) + (-1)^{1+3} a_{13} \det(A_{13})$$

$$\det(A) = (+1)a_{11} \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} + (-1)a_{12} \begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + (+1)a_{13} \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix}$$

III.1. RAPPEL MATHEMATIQUE: Transposé, Déterminant et inverse

2) Déterminant d'une matrice

Exemple numérique : $A = \begin{pmatrix} 1 & 2 & 3 \\ 0 & 4 & 5 \\ 1 & 0 & 6 \end{pmatrix}$

$$\det(A) = (-1)^{1+1}(1)\det\begin{pmatrix} 4 & 5 \\ 0 & 6 \end{pmatrix} + (-1)^{1+2}(2)\det\begin{pmatrix} 0 & 5 \\ 1 & 6 \end{pmatrix} + (-1)^{1+3}(3)\det\begin{pmatrix} 0 & 4 \\ 1 & 0 \end{pmatrix}$$

$$= (+1)(1)(24) + (-1)(2)(-5) + (+1)(3)(-4)$$

$$\det(A) = 24 + 10 - 12 = 22$$

III.1. RAPPEL MATHEMATIQUE: Transposé, Déterminant et inverse

3) Inverse d'une matrice diagonal

L'inverse d'une matrice diagonal A, notée A^{-1}, Si det(A) $\neq 0$ alors A est **inversible** et son inverse est simple et direct : il suffit d'inverser chaque élément de la diagonale.

Exemple: $A = \begin{bmatrix} a_{11} & 0 & 0 \\ 0 & a_{22} & 0 \\ 0 & 0 & a_{33} \end{bmatrix} \Longrightarrow A^{-1} = \begin{bmatrix} \frac{1}{a_{11}} & 0 & 0 \\ 0 & \frac{1}{a_{22}} & 0 \\ 0 & 0 & \frac{1}{a_{33}} \end{bmatrix}$

Ce cas de diagonale est beaucoup plus simple que la formule générale de calcul de matrice inverse.

$$A^{-1} = \frac{1}{\det(A)} \operatorname{adj}(A)$$

III.2. RAPPEL MATHEMATIQUE: Vecteur Propres et Valeurs propres

Eigenvectors and Eigenvalues

Pour une matrice carrée **S** m×m, il existe un **vecteur propre** $\mathbf{v} \in \mathbb{R}^m \neq \mathbf{0}$

et une **valeur propre scalaire** $\lambda \in \mathbb{R}$, tel que: $\mathbf{Sv} = \lambda\mathbf{v}$

Vecteur propre *Valeurs propre*

Exemple

$$\begin{pmatrix} 6 & -2 \\ 4 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \begin{pmatrix} 2 \\ 4 \end{pmatrix} = 2 \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

Matrice carrée S **Vecteur propre v** **Valeur propre λ**

Rappel : Multiplication matrice × vecteur, Pour calculer $\boldsymbol{Sv}$, on fait le **produit ligne par colonne**

$$\boldsymbol{S}\mathbf{v} = \begin{pmatrix} 6 & -2 \\ 4 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \begin{pmatrix} 6 \times 1 + (-2) \times 2 \\ 4 \times 1 + 0 \times 2 \end{pmatrix} = \begin{pmatrix} 2 \\ 4 \end{pmatrix} = 2 \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

III.2. RAPPEL MATHEMATIQUE: Vecteur Propres et Valeurs propres

Nombre de valeurs propres et de vecteurs propres possible

On cherche les couples (λ, v) tels que : $\boldsymbol{Sv = \lambda v}$

Cela revient à résoudre l'équation : $\boldsymbol{(S - \lambda I)v = 0}$

Pour que cette équation ait une **solution non nulle** v, il faut que:**det($\boldsymbol{S - \lambda I}$)=0**

Le **déterminant** | $S - \lambda I$ | est une **expression polynomiale en λ**. Si S est de taille $m \times m$, ce **polynôme est de degré m**. Ces solutions sont les **valeurs propres λ_1, λ_2, …, λ_m**. Et pour chaque valeur propre λ_i ,on peut trouver un **vecteur propre** $v_i$associé.

Un système à m équations en λ peut avoir au plus m solutions distinctes. Alors pour une matrice mxm on peut avoir m valeurs propres et m vecteurs propres.

IV. DECOMPOISITION D'UNE MATRICE

Une **décomposition de matrice** (ou **factorisation de matrice**) consiste à exprimer une matrice S comme un **produit de matrices plus simples**, afin de mieux comprendre sa structure ou ses propriétés afin de trouver les informations importantes. Ex. $S = A \times B \times C$

Pourquoi décomposer une matrice ?

- Comprendre comment une matrice transforme les vecteurs (étirements, rotations, etc.)
- Identifier les directions principales avec les variations maximales (utiles dans la PCA et la SVD)
- Découvrir des structures et relation cachées dans les données, la décomposition aide à **révéler ces relations**.(Ex: regrouper les textes qui parlent du même sujet.)

IV. DECOMPOISITION D'UNE MATRICE

Soit S une matrice carrée réelle de taille $M \times M$, possédant M vecteurs propres linéairement indépendants.

Alors il existe une décomposition : $S = U\Lambda U^{-1}$

où : * U= matrice dont les **colonnes sont les vecteurs propres (Eigenvectors)** de S

* Λ= matrice Lambda **diagonale** dont les éléments diagonaux sont les **valeurs propres (Eigenvalues)** de S, ordonnées décroissantes.

$$\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_m), \quad \lambda_i \geq \lambda_{i+1}$$

Cela signifie que toute matrice carrée peut être réécrite en fonction de ses vecteurs propres et valeurs propres.

- Chaque vecteur propre u_i indique une direction dans l'espace.
- Chaque valeur propre λ_i indique combien la matrice étire ou contracte le vecteur dans cette direction.

IV. DECOMPOSITION EN VALEURS SINGULIÈRE- SINGULAR VALUE DECOMPOSITION – SVD

Cette idée est généralisée aux matrices non carrées comme les matrices terme-document (plus de termes que de documents).

Comment:

Construire des matrices carrées à partir de S

On fabrique deux nouvelles matrices :

$$S^T S \text{ (de taille } N \times N) \text{ et } S\ S^T \text{ (de taille } M \times M)$$

Ces deux matrices **sont carrées**, donc on peut leur appliquer la théorie des valeurs propres / vecteurs propres (La décomposition).

IV. DECOMPOSITION EN VALEURS SINGULIÈRE- SINGULAR VALUE DECOMPOSITION – SVD

On a donc besoin d'une **généralisation** → **(SVD)**. Pour une matrice A(M×N), il existe une décomposition (Singular Value Décomposition) tel que:

$$A = U \Sigma V^T$$

{MxN} {MxM} {MxN} {NxN}

Avec :

- **U** → Les colonnes de U sont des **vecteurs propres** de $A\ A^T$
- **V** → Les colonnes de V sont **les vecteurs propres** de $A^T A$
  Les **valeurs propres** de ces deux matrices sont les **mêmes** : $\lambda_1, \lambda_2, \ldots, \lambda_r$
- **Σ** → Matrice **diagonale** contenant les **valeurs singulières** (réelles, positives et triées par ordre décroissant)

Et les **valeurs singulières** de A sont : $\sigma_i = \sqrt{\lambda_i}$

Donc $\Sigma = diag(\sigma_1, \sigma_2, . , .. \sigma_r)$

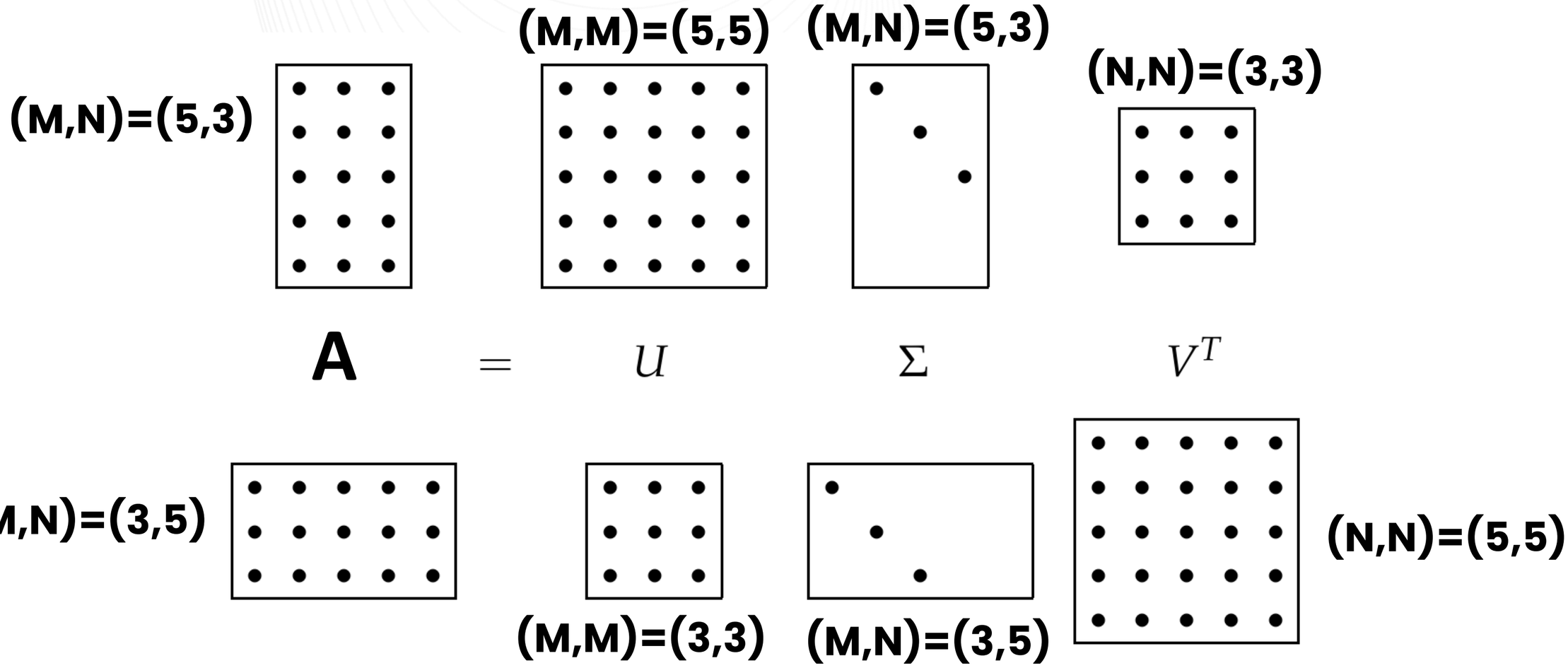
IV. DECOMPOSITION EN VALEURS SINGULIÈRE- SINGULAR VALUE DECOMPOSITION – SVD

La "diagonale" de Σ contient les valeurs singulières de A.

- Les valeurs singulières sont réelles et toujours positives ou nulles.
- La partie supérieure de la diagonale de Σ contient les valeurs singulières strictement positives.
  - Leur nombre r correspond au rang de A. (Le rang d'une matrice est donc révélé par le nombre de valeurs singulières non nulles).
  - Chaque valeur singulière est égale aux racines carrées positives des valeurs propres de $A\ A^T$.
  - La partie inférieure de la diagonale contient les (n - r) valeurs singulières nulles.

IV. DECOMPOSITION EN VALEURS SINGULIÈRE- SINGULAR VALUE DECOMPOSITION - SVD

ILLUSTRATION DE LA SVD



IV. DECOMPOSITION EN VALEURS SINGULIÈRE- SINGULAR VALUE DECOMPOSITION – SVD

SVD Réduite (ou Tronquée)

- On garde uniquement les **k plus grandes valeurs singulières**.
- Cela permet de réduire la taille des matrices tout en préservant l'essentiel de l'information.
- La matrice Σ devient de taille **k×k**.
- La matrice U devient **M×k** et la matrice V^T devient **k×N**.
- On obtient une approximation de A notée $\mathbf{A_k = U_k \Sigma_k V_k^T}$.
- Plus **k** est petit → réduction plus forte, mais perte d'information possible.
- Si **k = rang(A)**, on retrouve exactement la matrice **A**.
- La SVD réduite est utilisée pour la réduction de dimension et la compression de données.

V. LSI

V.1. SVD pour le LSI

- Base mathématique de la LSI : décomposition par valeur singulière de la matrice terme-document
- SVD identifie un ensemble utile de vecteurs colonnes couvrant le même espace de vecteurs associés à la représentation des documents
- SVD décompose la matrice **TERME x DOCUMENT W** en 3 matrices: $\boldsymbol{W = T \times S \times D}$

Ou :

T : matrice des **termes**, ses colonnes sont les **vecteurs propres de** WW^T
→décrit comment les **termes sont corrélés** entre eux
→ chaque colonne correspond à un **axe sémantique** (concept)

S : matrice **diagonale** des **valeurs singulières**
→ chaque valeur indique **l'importance** du concept latent associé (plus $\sigma_i$est grand, plus le concept associé à la i^e colonne de T et de D est important.)
→ plus elle est grande, plus le concept explique de variance dans les données

D : matrice des **documents**, ses colonnes sont **vecteurs propres de** W^TW
→décrit comment chaque document se projette sur les **concepts** trouvés

V. LSI

V.2. SVD Réduite pour le LSI

- Après la décomposition $W = T \times S \times D$, on ne conserve que les k plus grandes valeurs singulières (les plus significatives).
- On obtient alors une approximation réduite :

$$W_k \approx T_k \times S_k \times D_k$$

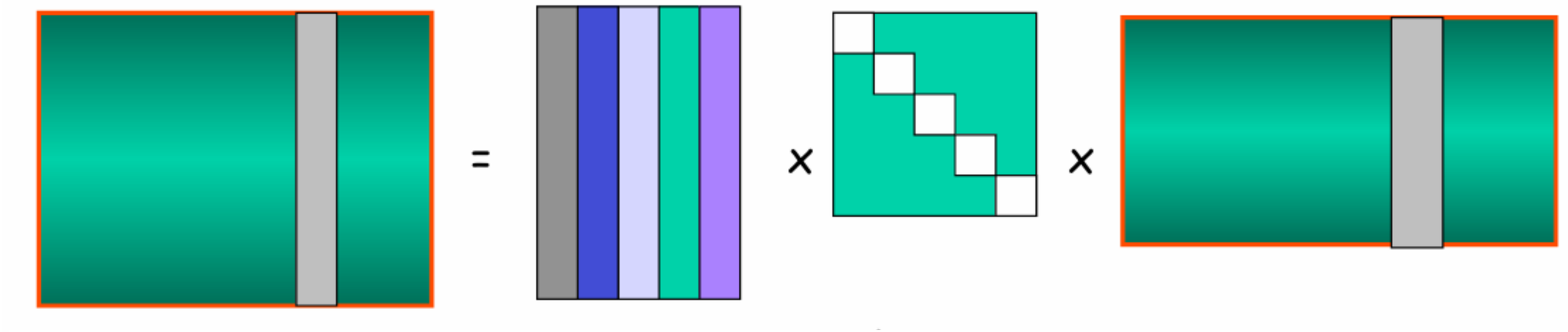
où :

- T_k → matrice **(t × k)** : termes exprimés selon les **k concepts principaux**
- S_k → matrice **(k × k)** : valeurs singulières conservées
- D_k → matrice **(k × d)** : documents exprimés dans le **même espace conceptuel**

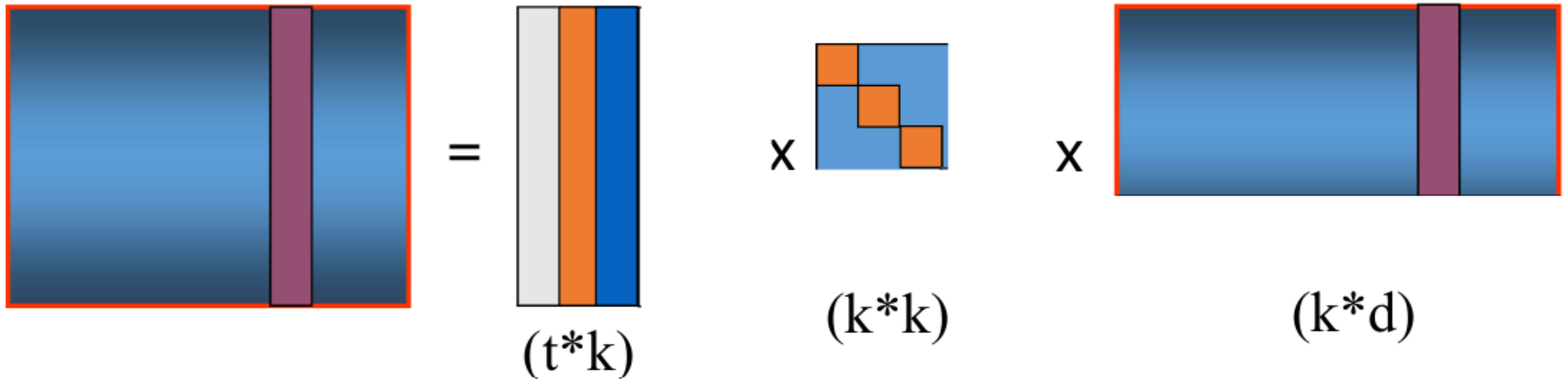
- Les colonnes de D_k sont les documents exprimés selon les concepts latents.

V. LSI

SVD Vs. SVD réduite



W = T x S x D



V. LSI

V.3. Matrice de Passage ou fonction de projection sémantique

Problème :

- Après la SVD, les documents sont déjà représentés dans l'**espace des concepts**.
- Mais les **requêtes** sont encore dans l'**espace des termes (mots)**.

On ne peut donc pas comparer directement les requêtes avec les documents.

Solution :

- La transformation de l'espace des termes vers l'espace des concepts se fait par :

$$M = T_k \times S_k^{-1}$$

V. LSI

V.4. Étapes de construction de l'espace sémantique par SVD

1) Calculer la SVD de la matrice terme document
2) Sélectionner les k premières valeurs singulières de la matrice S
3) Garder les colonnes correspondantes dans les matrices T et D pour obtenir :

$$T_k,\ S_k,\ D_k$$

4) La matrice D_k représente les vecteurs documents dans le nouvel espace M.
5) Projeter les requêtes dans le même espace grâce à la fonction de changement de repère : $M = T_k\ S_k^{-1}$

V. LSI

V.5. Représentation de la requête pour le modèle LSI

Pour évaluer une requête Q (vecteur de mots), on la projette dans le même espace latent que les documents en utilisant la matrice de passage vers l'espace de concept M :

$$Q_{new} = Q^T \cdot M = Q^T \cdot T \cdot S^{-1}$$

où :

- Q^T =vecteur de la requête dans l'espace des termes
- $M = T \cdot S^{-1}$ =**fonction de passage** vers l'espace des concepts

Notre objectif est donc de représenter la requête non plus comme une simple combinaison de mots, mais comme une **combinaison de concepts latents**.

V. LSI

V.6. Appariement documents/requête par LSI

Le calcul de la similarité entre chaque document et la requête, tous représentés dans le nouvel espace vectoriel M, se fait par la formule:

$$sim = Q_{new} \cdot S^2 \cdot D$$

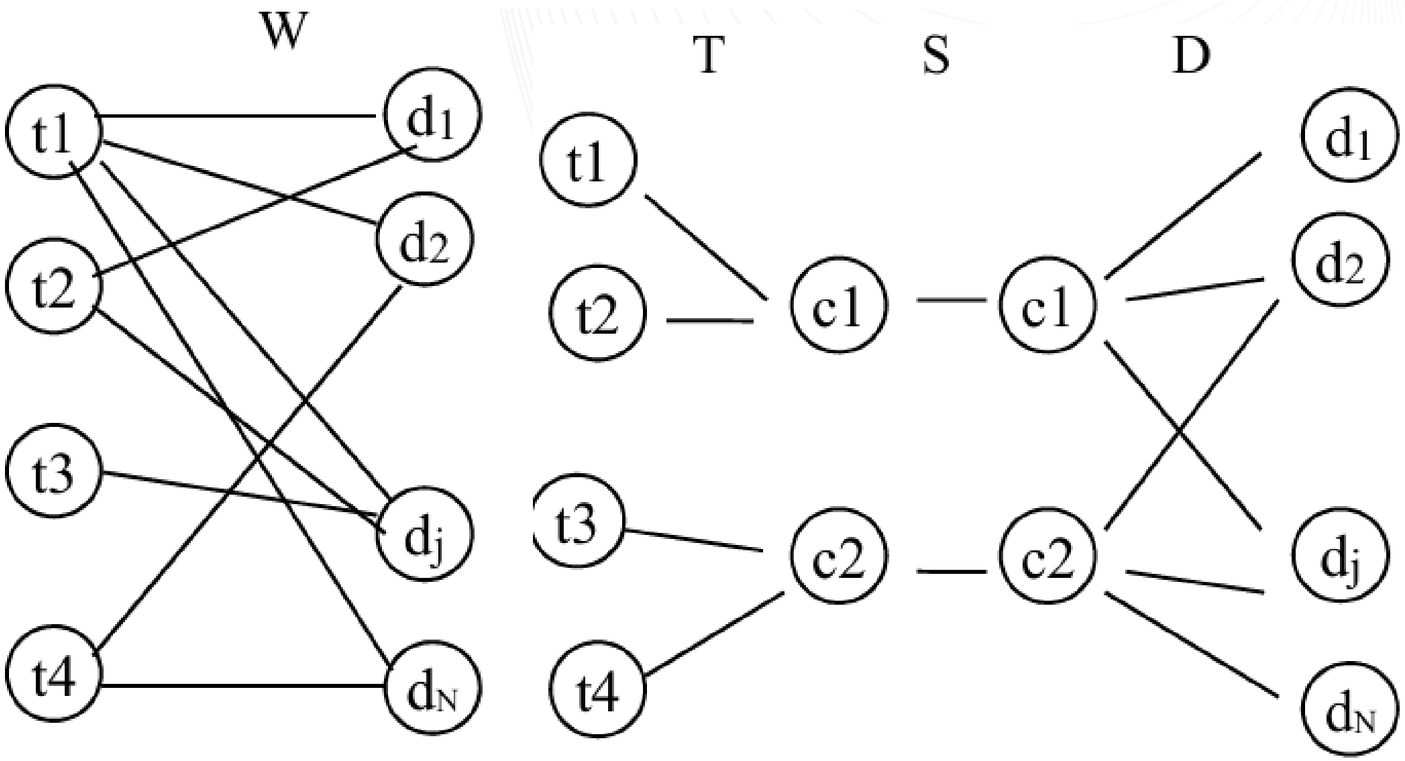
Le résultat “sim” est un vecteur de M cases. Chaque sim[j] représente la similarité entre la requête et le document D_j (RSV(Dj, Q))

- **Ce produit est un calcul vectoriel-matriciel**
- **Deux documents peuvent être proches même sans mots communs explicites.**

V. LSI

Où sont les concepts ? (Interprétation)

$$W_{t\times d} = T_{t\times r} \times S_{r\times r} \times D_{r\times d}$$



Les concepts (ou facteurs latents) se trouvent au centre de la décomposition, entre T et D.

- ✓ Chaque colonne de T indique comment un terme est relié à chaque concept latent.
- ✓ Chaque colonne de D indique comment un document est relié à chaque concept latent.

La matrice S relie les deux mondes (termes ↔ concepts ↔ documents) et pondère l'importance de chaque concept.

VI. Recap SVD

Quand on fait une SVD sur une matrice terme-document W :

$$W = T\,S\,D^T$$

- on cherche à trouver une base orthogonale optimale pour représenter les lignes (termes) et les colonnes (documents) de W.
- Cette base est trouvée à partir des vecteurs propres de deux matrices carrées :

$$W^T W \text{ et } W W^T$$

- Ces deux matrices ont les mêmes valeurs propres non nulles, et leurs vecteurs propres servent à construire :
  - **D → vecteurs propres de $W^T W$ (documents)**
  - **T → vecteurs propres de $W W^T$ (termes)**

Les **valeurs singulières** σ_i sont alors les **racines carrées** des valeurs propres correspondantes :

$$\sigma_i = \sqrt{\lambda_i}$$

les valeurs propres λ_i sont calculées en résolvant le **polynôme caractéristique**

$$\det(W^T W - \lambda I) = 0$$

et ce calcul fait émerger les **directions principales de variance** dans les données.

VI. Recap SVD

- La SVD cherche les axes de variation maximale dans les données — c'est-à-dire les directions où les termes co-apparaissent souvent.
- Chaque vecteur propre (colonne de T ou de D) définit un axe (une direction) orthogonale dans un espace qui correspond à une combinaison linéaire de termes corrélés : c'est un concept latent.
- Les "concepts" ne sont pas visibles comme des mots, mais comme des dimensions abstraites regroupant les termes qui varient ensemble.

Exemple concret

Les termes "voiture", "pneu", "moteur" et "essence" apparaissent souvent ensemble :

- Leurs vecteurs dans WW^T auront des covariances élevées.
- le **vecteur propre principal** correspondant (dans T) les combinera fortement,

cette combinaison linéaire devient un **axe latent**, donc un **concept** (ici : "automobile").

Latent Semantic Indexing

Avantages

- Réduit la dimensionnalité des données.
- Capte les relations sémantiques entre termes.
- Gère la synonymie et la polysémie.
- Diminue le bruit dans la matrice terme-document.
- Améliore la recherche même sans mots communs

Inconvénients

- Calcul SVD coûteux pour de grands corpus.
- Concepts latents parfois difficiles à interpréter.
- Choix du paramètre *k* délicat.

EXEMPLE

Soit la collection suivante :

D1: “ t1 t2 t3 t4 t5 t6 t7 ”
D2: “ t8 t2 t9 t10 t5 t6 t11 t9 ”
D3: “ t1 t2 t3 t10 t5 t6 t11 ”

Soit le requête suivante :

Q : “ *t3 t9 t11* ”

La matrice Termes*Documents est comme suit :

Documents Termes	D1	D2	D3
t6	1	1	1
t10	0	1	1
t4	1	0	0
t8	0	1	0
t7	1	0	0
t3	1	0	1
t5	1	1	1
t2	1	1	1
t1	1	0	1
t9	0	2	0
t11	0	1	1

Questions :

1. Décomposer la matrice par SVD
2. Réduire la matrice S (prendre les k valeurs fortes >=2)
3. Mesurer la similarité entre les documents et la requête Q

EXEMPLE

Décomposition de la matrice par SVD et Réduire la matrice S (prendre les k valeurs fortes supérieures ou égales à 2)

T

-0,4201	0,0748
-0,2995	-0,2001
-0,1206	0,2749
-0,1576	-0,3046
-0,1206	0,2749
-0,2626	0,3794
-0,4201	0,0748
-0,4201	0,0748
-0,2626	0,3794
-0,3151	-0,6093
-0,2995	-0,2001

S

4,0909	0
0	2,3616

D

-0,4945	-0,6458	-0,5817
0,6492	-0,7194	-0,2469

Représentation de la requête Q dans l'espace terme

La requête $Q = \{t3, t9, t11\}$ s'écrit comme vecteur terme×1 (ordre des termes idem matrice) :

$$q = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

Calculer sa projection dans l'espace latent en utilisant la formule standard :

$$q_{new} = S^{-1} T^{\top} q \qquad \in \mathbb{R}^2.$$

Réponse : Q dans l'espace de concept

Calcul étape 0 — *calcul de transposee de* T : $T^\top$

On part de la matrice T (les **vecteurs propres des termes**) :

$$T = \begin{bmatrix} t_{1,1} & t_{1,2} \\ t_{2,1} & t_{2,2} \\ \vdots & \vdots \\ t_{11,1} & t_{11,2} \end{bmatrix} = \begin{bmatrix} -0.227821 & -0.184102 \\ -0.188718 & 0.088121 \\ \vdots & \vdots \\ -0.299487 & 0.200092 \end{bmatrix}_{11\times 2}$$

La **transposée** $T^\top$ inverse les lignes et colonnes :

$$T^\top = \begin{bmatrix} t_{1,1} & t_{2,1} & \dots & t_{11,1} \\ t_{1,2} & t_{2,2} & \dots & t_{11,2} \end{bmatrix}_{2\times 11}$$

Ainsi, chaque **colonne** de $T^\top$ correspond à un **terme**, et chaque **ligne** correspond à une **dimension latente** (concept).

Réponse : Q dans l'espace de concept

Calcul étape 1 — $T^\top q$

Par définition $T^\top q$ =somme des lignes de T correspondant aux termes présents dans q (ici lignes 6, 10, 11). On extrait ces lignes (lignes utiles) :

ligne 6 (t3) : $[-0.262561,\ -0.379447]$

ligne 10 (t9): $[-0.315122,\ 0.609295]$

ligne 11 (t11):$[-0.299487,\ 0.200092]$

On additionne composante par composante :

Composante 1 : $(T^\top q)_1 = -0.262561 + (-0.315122) + (-0.299487) = -0.877170$

Composante 2 : $(T^\top q)_2 = -0.379447 + 0.609295 + 0.200092 = 0.429940$

Donc :

$$\boxed{T^\top q \approx \begin{bmatrix} -0.877170 \\ 0.429940 \end{bmatrix}}$$

Réponse : Q dans l'espace de concept

Calcul étape 2 — S^{-1} (Verifier si S est inversible)

S est diagonale → inverse simple :

$$S^{-1} = \begin{bmatrix} 1/4.098872 & 0 \\ 0 & 1/2.361571 \end{bmatrix} \approx \begin{bmatrix} 0.24396956 & 0 \\ 0 & 0.42344693 \end{bmatrix}$$

Calcul étape 3 — projection $q_{new} = S^{-1}(T^{\top}q)$

Calcul composante par composante :

$$\begin{cases} (q_{new})_1 = -0.877170 \times 0.24396956 = -0.214002779 \\ (q_{new})_2 = 0.429940 \times 0.42344693 = 0.182056775 \end{cases}$$

Donc :

$$q_{new} \approx \begin{bmatrix} -0.214002779 \\ 0.182056775 \end{bmatrix}$$

(ce sont les coordonnées de la requête dans l'espace latent de dimension $k = 2$.)

EXEMPLE

Calculer la similarité entre les vecteurs documents et la requête Q

$sim = Q_{new} \cdot S^2 \cdot D$

Étape 1 : Calcul de S^2

$$S^2 = \begin{bmatrix} 4.0988^2 & 0 \\ 0 & 2.3615^2 \end{bmatrix} = \begin{bmatrix} 16.8007 & 0 \\ 0 & 5.5768 \end{bmatrix}$$

Étape 2 — calcul de $S^2 \cdot D$ (2*3): On multiplie chaque ligne de D par la valeur diagonale correspondante**.**

Ligne 1 (multipliée par 16.8007) :

$$16.8007522 \times (-0.494467) \approx -8.3074223$$
$$16.8007522 \times (-0.645822) \approx -10.8502880$$
$$16.8007522 \times (-0.581736) \approx -9.7736160$$

Ligne 2 (multipliée par 5.5768510) :

$$5.5768510 \times (-0.649176) \approx -3.6203606$$
$$5.5768510 \times 0.719447 \approx 4.0123046$$
$$5.5768510 \times (-0.246915) \approx -1.3770120$$

Donc

$$S^2 D \approx \begin{bmatrix} -8.3074223 & -10.8502880 & -9.7736160 \\ -3.6203606 & 4.0123046 & -1.3770120 \end{bmatrix}.$$

EXEMPLE

Calculer la similarité entre les vecteurs documents et la requête Q

Étape 3 — produit final $sim = q_{new}^{\top}(S^2 D)$

Pour chaque document j on calcule la combinaison linéaire :

$$\mathrm{sim}_j = (q_{\mathrm{new}})_1 \cdot (S^2 D)_{1,j} \; + \; (q_{\mathrm{new}})_2 \cdot (S^2 D)_{2,j}.$$

Calculs détaillés :

Document D1 (colonne 1)

Document D1 (colonne 1)

$$(S^2 D)_{:,1} = \begin{bmatrix} -8.3074223 \\ -3.6203606 \end{bmatrix}$$

- première composante : $(-0.214002779) \times (-8.3074223) = +1.777789$
- deuxième composante : $0.182056775 \times (-3.6203606) = -0.659164$

$$\mathrm{sim}_1 \approx 1.777789 - 0.659164 = \mathbf{1.118625}$$

EXEMPLE

Calculer la similarité entre les vecteurs documents et la requête Q

Étape 3 — produit final $sim = q_{new}^{\top}(S^2 D)$

Pour chaque document j on calcule la combinaison linéaire :

$$\mathrm{sim}_j = (q_{\mathrm{new}})_1 \cdot (S^2 D)_{1,j} \; + \; (q_{\mathrm{new}})_2 \cdot (S^2 D)_{2,j}.$$

Calculs détaillés :

Document D2 (colonne 2)

$$(S^2 D)_{:,2} = \begin{bmatrix} -10.8502880 \\ 4.0123046 \end{bmatrix}$$

- première composante : $(-0.214002779) \times (-10.8502880) = +2.321962$
- deuxième composante : $0.182056775 \times 4.0123046 = +0.730464$

$$\mathrm{sim}_2 \approx 2.321962 + 0.730464 = \mathbf{3.052426}$$

EXEMPLE

Calculer la similarité entre les vecteurs documents et la requête Q

Étape 3 — produit final $sim = q_{new}^{\top}(S^2D)$

Pour chaque document j on calcule la combinaison linéaire :

$$\mathrm{sim}_j = (q_{\mathrm{new}})_1 \cdot (S^2D)_{1,j} \; + \; (q_{\mathrm{new}})_2 \cdot (S^2D)_{2,j}.$$

Calculs détaillés :

Document D3 (colonne 3)

$$(S^2D)_{:,3} = \begin{bmatrix} -9.7736160 \\ -1.3770120 \end{bmatrix}$$

- première composante : $(-0.214002779) \times (-9.7736160) = +2.091549$
- deuxième composante : $0.182056775 \times (-1.3770120) = -0.250693$

$$\mathrm{sim}_3 \approx 2.091549 - 0.250693 = \mathbf{1.840856}$$

Classement : D2 > D3 > D1

EXERCICE

A “collection” consists of the following “documents”:

d1: Shipment of gold damaged in a fire.

d2: Delivery of silver arrived in a silver truck.

d3: Shipment of gold arrived in a truck.

1. Suppose that we use the term frequency as term weights and query weights.
2. The following document indexing rules are used:
   - stop words were are ignored
   - text was tokenized and lowercased
   - no stemming was used
   - terms were sorted alphabetically

Problem: Use Latent Semantic Indexing (LSI) to rank these documents for the query **q={gold silver truck}**

(taken from Grossman and Frieder’s Information Retrieval, Algorithms and Heuristics)

EXERCICE

QUESTIONS

1) Construct the term-document matrix W and query matrix Q.
2) Decompose W based on SVD decomposition : W=TSD.
3) Reduce S , take k strong values k >=2.
4) Represent the query in the M latent space.
5) Measure similarity cosine between the documents and query, rank them

(taken from Grossman and Frieder’s Information Retrieval, Algorithms and Heuristics)