

(Texte public)

Résumé : On s'intéresse à l'utilisation de méthodes d'analyse numérique matricielle dans le cadre de la gestion de bases de données bibliographiques.

Mots clefs : Algèbre linéaire. Éléments propres de matrices. Moindres carrés.

- *Il est rappelé que le jury n'exige pas une compréhension exhaustive du texte. Vous êtes laissé(e) libre d'organiser votre discussion comme vous l'entendez. Des suggestions de développement, largement indépendantes les unes des autres, vous sont proposées en fin de texte. Vous n'êtes pas tenu(e) de les suivre. Il vous est conseillé de mettre en lumière vos connaissances à partir du fil conducteur constitué par le texte. Le jury appréciera que la discussion soit accompagnée d'exemples traités sur ordinateur.*

On considère une famille $(D_j)_{1 \leq j \leq N_d}$ de documents et une famille $(T_i)_{1 \leq i \leq N_t}$ de termes (ou mots-clés) permettant d'identifier le contenu des documents. Le but de ce texte est l'étude d'un modèle mathématique permettant d'extraire des informations pertinentes de cette collection de documents et notamment de proposer un outil de recherche par mots-clés parmi ceux-ci. Considérons deux exemples où l'ensemble des termes choisi est l'ensemble de tous les mots d'une langue donnée, le nombre de tels termes est de l'ordre de $N_t \sim 300000$. Si l'ensemble de documents est constitué de l'ensemble des articles d'une encyclopédie, on est dans la situation où $N_d \ll N_t$ mais si c'est l'ensemble des pages Web disponibles sur Internet alors on a $N_d \gg N_t$. Les deux situations sont extrêmes mais on voit que la taille des problèmes considérés rend inopérante toute tentative de traitement manuel des informations. On étudie dans ce texte un modèle mathématique (empirique) permettant de traiter toute cette masse d'information de façon automatique et si possible efficace.

1. Le modèle dit *espace vectoriel*

1.1. Présentation

Ce modèle consiste à représenter chacun des N_d documents de la collection comme un élément d_j de l'espace vectoriel \mathbb{R}^{N_t} . La i -ème composante d_{ij} du vecteur d_j représente une certaine mesure de l'importance du terme T_i dans le document D_j . Par souci de simplicité, nous choisirons dans ce texte de prendre $d_{ij} = 1$ ou 0 selon que le terme T_i apparaît ou non dans le document D_j .

Si on met côte-à-côte les N_d vecteurs colonnes représentant chacun des documents de la collection, on obtient une matrice D (appelée la matrice *termes-documents*) de taille $N_t \times N_d$. Toutes les informations nécessaires au traitement des données sont donc contenues dans cette grande matrice.

1.2. Réponse à une requête

Quand un utilisateur fournit une requête (sous la forme d'une liste de termes), on doit être capable de lui retourner une liste de documents correspondant le plus fidèlement possible à sa recherche. La requête est représentée comme un vecteur colonne $q \in \mathbb{R}^{N_t}$ dont les coefficients valent 1 ou 0 selon que le terme correspondant est présent ou non dans la requête de l'utilisateur.

On propose de mesurer la pertinence du document D_j par rapport à la requête q par la valeur du cosinus de l'angle entre le vecteur d_j et le vecteur requête q dans l'espace \mathbb{R}^{N_t} . Soit

$$(1) \quad s_j(q) = \frac{\langle q, d_j \rangle}{\|q\|_2 \|d_j\|_2},$$

où $\langle \cdot, \cdot \rangle$ désigne le produit scalaire usuel sur \mathbb{R}^{N_t} et $\|\cdot\|_2$ la norme associée. Plus $s_j(q)$ est proche de 1, plus le document D_j et la requête q se correspondent. Cette méthode permet ainsi d'attribuer à tous les documents un score $s_j(q)$ relatif à q compris entre 0 et 1 et ainsi de retourner une liste des documents classés selon leur adéquation à la requête de l'utilisateur. En général, on décide de ne conserver dans cette liste que les documents dont le score de pertinence est supérieur à un seuil fixé à l'avance (on prendra ce seuil égal à 0.8 dans les applications numériques).

1.3. Quelques défauts du modèle et une solution : la réduction de rang

Le premier défaut du modèle précédent est purement d'ordre pratique : il s'agit de la taille gigantesque des calculs qu'il est nécessaire de mettre en œuvre pour chaque requête, ce qui rend la méthode assez difficilement applicable de façon efficace.

D'autres problèmes soulevés par ce modèle sont davantage liés à la modélisation choisie :

- Il y a bien souvent de nombreuses *redondances* dans les informations contenues dans la matrice D : par exemple, si la base de données contient plusieurs éditions d'un même livre, ou plusieurs pages Web d'un même site portant sur les mêmes sujets. On doit pouvoir tirer profit de l'existence de ces redondances pour simplifier les calculs.
- *Synonymie* : plusieurs termes peuvent avoir exactement le même sens ou au moins des sens très voisins (*voiture* et *automobile* par exemple). On voudrait pouvoir extraire automatiquement ce type d'informations de la matrice du système afin de ne pas oublier de documents sémantiquement pertinents pour la requête utilisateur.
- On rencontre aussi bien souvent le problème de la *polysémie*, c'est-à-dire d'un terme qui peut avoir plusieurs sens très différents en fonction du contexte dans lequel il est utilisé. Encore une fois, un bon modèle devrait être capable de détecter de telles situations et de proposer à l'utilisateur une recherche précisée selon le contexte.

On propose donc ici une approche mathématique dont on peut constater sur des cas réels qu'elle permet (**sans qu'on sache le démontrer précisément!**) de corriger partiellement les défauts de modélisation évoqués ci-dessus. Par ailleurs, la méthode proposée bien qu'initialement coûteuse, permettra de réduire les temps de calcul à chaque nouvelle requête utilisateur.

L'idée est de remplacer la matrice D par une matrice D_k de rang k « petit » et la plus proche possible de la matrice D . La philosophie sous-jacente est que la réduction au rang k revient à déterminer « les k groupements de termes les plus significatifs » mais aussi « les k groupements de documents les plus significatifs » dans la collection étudiée. Notons bien que le terme de *groupement* désigne ici un élément de l'espace des termes \mathbb{R}^{N_t} ou de l'espace des documents \mathbb{R}^{N_d} . Il s'agit donc d'un objet essentiellement conceptuel.

2. Décomposition en valeurs singulières d'une matrice

2.1. Aspects théoriques

L'outil mathématique utilisé pour effectuer la réduction de rang évoquée précédemment est la *décomposition en valeurs singulières* (ou encore **DVS**) d'une matrice. Voyons de quoi il s'agit.

Théorème 1. Soit A une matrice de taille $n \times m$ et de rang r . Il existe une unique famille de réels strictement positifs $\sigma_1 \geq \dots \geq \sigma_r > 0$ et un couple de matrices orthogonales $(U, V) \in O_n(\mathbb{R}) \times O_m(\mathbb{R})$ telles que

$$(2) \quad A = U \Sigma^t V,$$

où Σ est la matrice de taille $n \times m$ dont les r premiers éléments diagonaux sont les σ_i et tous les autres éléments sont nuls.

Les σ_i sont appelées les *valeurs singulières* de la matrice A . Ce sont également les racines carrées des valeurs propres non nulles de ${}^t A A$ et de $A {}^t A$. Par ailleurs, les r premières colonnes de U et de V sont des vecteurs propres de ces deux matrices.

La propriété de cette décomposition matricielle qui nous intéresse ici est la suivante.

Théorème 2. Soit A une matrice comme au théorème précédent et $k \leq r$ un entier. On définit une matrice A_k par

$$(3) \quad A_k = U \Sigma_k {}^t V,$$

où $U \Sigma {}^t V$ est une DVS de A et Σ_k la matrice obtenue à partir de Σ en ne conservant que les k premiers éléments diagonaux et en annulant les autres. Alors cette matrice vérifie

$$(4) \quad \|A - A_k\|_2 = \inf_{B, \text{rang}(B)=k} \|A - B\|_2.$$

Dans ce théorème la norme matricielle utilisée est définie par $\|M\|_2 = \sup_{\|x\|_2=1} \|Mx\|_2$.

Autrement dit, il existe une meilleure approximation de rang k de la matrice A au sens de la norme euclidienne et celle-ci peut s'obtenir à partir de la DVS de la matrice.

2.2. Préliminaire : décomposition QR d'une matrice bidiagonale inférieure

Toute matrice carrée A peut s'écrire comme un produit $A = QR$ d'une matrice orthogonale Q et d'une matrice triangulaire supérieure à coefficients diagonaux positifs R . Dans le cas où

A est inversible, cela revient à appliquer la méthode d'orthonormalisation de Gram-Schmidt à la base de \mathbb{R}^n formée des vecteurs colonnes de A , et une telle décomposition QR est alors unique.

Dans le cas particulier où A est bidiagonale inférieure (ses seuls coefficients non nuls sont sur et au dessous de la diagonale), le calcul de sa décomposition QR peut s'effectuer en $O(n)$ opérations en cherchant la matrice Q sous la forme d'un produit $Q = C_1 \dots C_{n-1}$, chaque C_i étant la matrice (dite de Givens) représentant une rotation d'angle θ_i (bien choisi) dans le plan engendré par les i -ème et $i + 1$ -ème vecteurs de la base canonique de \mathbb{R}^n .

2.3. Aspects algorithmiques de la DVS

Il existe plusieurs méthodes de calcul de la DVS d'une matrice. Nous en proposons une relativement simple mais pas nécessairement la plus efficace. L'idée est d'éviter de calculer les produits matriciels $A^t A$ et ${}^t A A$ très coûteux en temps de calcul et en place mémoire.

Pour simplifier les notations, on suppose dorénavant que $n \geq m$.

• ÉTAPE 1. Bidiagonalisation :

On se propose de ramener le calcul de la DVS d'une matrice A quelconque à celui d'une matrice bidiagonale. Considérons pour cela l'algorithme suivant :

Choisir $v_1 \in \mathbb{R}^m$ **unitaire quelconque et poser** $\beta_0 = 0$, $u_0 = 0 \in \mathbb{R}^n$

Pour $i = 1$ **jusqu'à** $m - 1$, **faire**

$$\begin{aligned} \tilde{u}_i &= Av_i - \beta_{i-1}u_{i-1}, & \alpha_i &= \|\tilde{u}_i\|_2, & u_i &= \frac{\tilde{u}_i}{\alpha_i}, \\ \tilde{v}_{i+1} &= {}^t Au_i - \alpha_i v_i, & \beta_i &= \|\tilde{v}_{i+1}\|_2, & v_{i+1} &= \frac{\tilde{v}_{i+1}}{\beta_i}. \end{aligned}$$

Poser $\tilde{u}_m = Av_m - \beta_{m-1}u_{m-1}$, **calculer** $\alpha_m = \|\tilde{u}_m\|_2$ **puis** $u_m = \frac{\tilde{u}_m}{\alpha_m}$.

Si $n > m$, **compléter** $(u_i)_{1 \leq i \leq m}$ **en une base orthonormée.**

On a alors le résultat suivant :

Proposition 1. *Si tous les coefficients α_i et β_i sont non nuls alors les matrices $\tilde{U} = (u_1 \dots u_n)$ et $\tilde{V} = (v_1 \dots v_m)$ données par l'algorithme ci-dessus sont orthogonales et de plus, on a*

$$(5) \quad \tilde{U} A^t \tilde{V} = \begin{pmatrix} \alpha_1 & \beta_1 & 0 & \dots & \dots \\ 0 & \alpha_2 & \beta_2 & 0 & \dots \\ \vdots & \ddots & \ddots & \ddots & \ddots \\ \vdots & & \ddots & \alpha_{m-1} & \beta_{m-1} \\ \vdots & & & \ddots & \alpha_m \\ 0 & \dots & \dots & \dots & 0 \\ \vdots & & & & \vdots \end{pmatrix}$$

• ÉTAPE 2. Calcul de la DVS d'une matrice B bidiagonale :

L'étape précédente a ramené le calcul de la DVS d'une matrice rectangulaire quelconque au cas d'une matrice bidiagonale B supérieure donnée par le membre de droite de (5). On voit aisément que les $n - m$ dernières lignes de B ne jouent aucun rôle dans la suite. On se ramène donc dorénavant au cas où B est bidiagonale carrée (i.e. $m = n$).

Plusieurs méthodes différentes peuvent s'envisager pour cette étape, l'idée étant de profiter de la structure bidiagonale de B pour simplifier les calculs. La méthode que l'on se propose d'étudier dans ce texte est la suivante.

Poser $B_1 = B$

Jusqu'à convergence, faire :

Calculer la décomposition QR de ${}^t B_k = Q_k R_k$.

Calculer la décomposition QR de ${}^t R_k = \tilde{Q}_k B_{k+1}$.

S'arrêter quand les coefficients sur-diagonaux de B_{k+1} sont assez petits.

La matrice B_{k+1} finale contient des approximations des valeurs singulières de B et les matrices orthogonales U et V correspondantes s'obtiennent à partir des matrices $(Q_i)_{1 \leq i \leq k}$ et $(\tilde{Q}_i)_{1 \leq i \leq k}$. On peut prouver que l'algorithme ci-dessus est formellement équivalent à la méthode QR de calcul des valeurs propres d'une matrice appliquée à la matrice $B^t B$, mais ne nécessite pas son calcul. Chaque étape de l'algorithme s'effectue en $O(n)$ opérations d'après la section 2.2.

3. Retour au modèle *espace vectoriel*: un exemple « concret »

On reprend ici les notations de la première partie. Une fois que l'on a calculé la DVS de la matrice termes-documents $D = U \Sigma^t V$, on obtient sa meilleure approximation de rang k en ne conservant que les k plus grandes valeurs de Σ , d'après le théorème 2. Seules les k premières colonnes des matrices U et V sont nécessaires de sorte que la matrice approchée D_k s'écrit

$$(6) \quad D_k = U_k \tilde{\Sigma}_k {}^t V_k,$$

où $U_k \in M_{n,k}(\mathbb{R})$, $V_k \in M_{m,k}(\mathbb{R})$ sont telles que ${}^t U_k U_k = \text{Id}$, ${}^t V_k V_k = \text{Id}$ et $\tilde{\Sigma}_k$ est la matrice diagonale $k \times k$ contenant les k plus grandes valeurs singulières de D .

Comment doit-on maintenant répondre à une requête utilisateur ? On propose un nouveau score de pertinence défini par

$$(7) \quad \tilde{s}_j(q) = \frac{\langle {}^t U_k q, \tilde{\Sigma}_k {}^t V_k e_j \rangle}{\|{}^t U_k q\|_2 \|\tilde{\Sigma}_k {}^t V_k e_j\|_2},$$

où e_j est le j -ième vecteur de base de \mathbb{R}^{N_d} . Ceci revient *presque* à remplacer les vecteurs colonnes de D par ceux de D_k dans le calcul du score mais sans effectuer le calcul explicite de la matrice D_k . Ce point est important car, même si en général la matrice D est creuse (i.e. avec énormément de coefficients nuls), il n'en est pas de même de la matrice D_k .

Dans les cas réels, les nombres de termes N_t et de documents N_d sont très grands (jusqu'à 10^6) et on prend l'indice k d'approximation du rang égal à quelques centaines.

Afin d'illustrer les différents concepts et méthodes proposés plus haut, on propose d'étudier un tout petit exemple. On considère donc une base de données constituée de 12 articles ayant trait à deux domaines thématiques distincts (l'économie et la géologie) et référencés par le biais de 20 termes. La situation est résumée dans le tableau 1. Sur cet exemple, on peut construire à la main la matrice D , calculer sa DVS puis construire son approximation de rang k pour tout k . On pourra par exemple tester la réduction au rang $k = 2$ pour les différentes

TABLE 1. Un exemple de petite collection de documents

DOCUMENTS ÉCONOMIQUES	DOCUMENTS GÉOLOGIQUES
Doc. 1 : chômage, impôts, dépression, commerce	Doc. 8 : bassin, faille, dérive
Doc. 2 : économie, marché, commerce	Doc. 9 : dépression, faille
Doc. 3 : commerce, marché, production	Doc. 10 : bassin, drainage, vallée
Doc. 4 : dépression, liquidation, emploi, redressement	Doc. 11 : dépression, drainage, érosion
Doc. 5 : indemnisation, chômage, liquidation	Doc. 12 : bassin, drainage, volcan
Doc. 6 : commerce, marché, prix, emploi	
Doc. 7 : bénéfiques, indemnisation, chômage	

requêtes suivantes :

$$q_1 = \{\text{dépression, économie}\}, \quad q_2 = \{\text{dépression, commerce}\} \quad \text{et} \quad q_3 = \{\text{emploi}\}.$$

Suggestions pour le développement

- *Soulignons qu'il s'agit d'un menu à la carte et que vous pouvez choisir d'étudier certains points, pas tous, pas nécessairement dans l'ordre, et de façon plus ou moins fouillée. Vous pouvez aussi vous poser d'autres questions que celles indiquées plus bas. Il est très vivement souhaité que vos investigations comportent une partie traitée sur ordinateur et, si possible, des représentations graphiques de vos résultats.*
- Donner les principales étapes de démonstration des théorèmes 1 et/ou 2. Dans le théorème 2, on pourra préciser la valeur de $\|A - A_k\|_2$. Dans quel cas la matrice A_k est-elle la seule à minimiser la valeur de $\|A - B\|_2$ pour B de rang k ?
- Démontrer la proposition 1. Cet algorithme de bidiagonalisation (ÉTAPE 1) s'avère assez sensible aux erreurs d'arrondis. On pourra vérifier que, même sur des matrices de taille modeste (carrées pour simplifier), le défaut d'orthogonalité des matrices \tilde{U} et \tilde{V} à la fin de l'algorithme peut être assez important. Que proposer pour remédier à cela ?
- Commenter et détailler le paragraphe 2.2.
- L'algorithme de calcul de la DVS pour une matrice bidiagonale (ÉTAPE 2) est volontairement peu détaillé. On pourra commenter le lien avec la méthode QR de calcul des valeurs propres d'une matrice, discuter la convergence, etc.
- Imaginer d'autres méthodes de calcul de la DVS d'une matrice bidiagonale, en ramenant par exemple le problème à la diagonalisation d'une ou plusieurs matrices tridiagonales symétriques. On pourra comparer les différentes méthodes entre elles.
- Mettre en œuvre la réduction de rang proposée en exemple dans la section 3 et discuter les résultats numériques obtenus vis-à-vis des questions soulevées dans la section 1.3.
- Commenter la pertinence du modèle *espace vectoriel* présenté au début du texte. On pourra éventuellement proposer des améliorations ou des variantes de ce modèle, par exemple en s'intéressant à une construction plus judicieuse des coefficients de la matrice *termes-documents* D .

Important : Des routines de calcul de la DVS (appelée SVD : *Singular Value Decomposition* en anglais) sont disponibles dans les logiciels à votre disposition, ce qui permet de traiter l'exemple même sans avoir implémenté l'algorithme proposé dans la section 2.