

Méthodes hybrides basées sur les décompositions SVD et EVD

Nahid Emad

Résumé

Ce document donne un aperçu des méthodes SVD et EVD et leurs applications ainsi que leurs liens avec l'ACP.

Table des matières

1	Approximation d'une fonction	1
1.1	La méthode des moindres carrés	2
1.2	La méthode des Snapshots	2
1.3	La décomposition SVD d'une matrice	3
1.4	La SVD et la EVD d'une matrice	3
2	Algorithmes de calcul de la SVD	4
3	Applications	4
3.1	La méthode Fiedler	4
3.2	La méthode Fiedler étendue	4
3.3	SVD au lieu de eigen-décomposition	5
3.4	Petits exemples et l'interprétation géométrique	5

1 Approximation d'une fonction

Soit $u(x, t)$ une fonction spatio-temporelle (variant en temps et en espace). Il s'agit d'approcher cette fonction par une combinaison linéaire finie des fonctions $\phi_1(x), \dots, \phi_K(x)$:

$$u(x, t) \simeq \sum_{k=1}^K a_k(t) \phi_k(x) \quad (1)$$

On souhaite que cette approximation devienne exacte quand K tend vers ∞ . Mais évidemment l'approximation (1) n'est pas unique. On essaie donc de chercher la meilleure. Nous verrons qu'une solution à ce problème est la décomposition SVD.

Pour cela, on peut considérer que les fonctions de base $\phi_k(x)$ sont connues a priori comme des polynômes de Chebychev ou de Legendre, ou des fonctions trigonométriques, etc. Une autre approche est de considérer $\phi_k(x)$ inhérente à la fonction u . Celle-ci nous conduira à la méthode de décomposition en EVD.

En supposant ces fonctions connues, le problème est de déterminer les coefficients de projection temporelle $a_k(t)$. Supposons que le produit scalaire canonique pour la description d'orthogonalité de la base $\phi_1(x), \dots, \phi_K(x)$ est défini par :

$$(\phi_k(x), \phi_j(x)) = \int_{\Omega} \phi_k(x) \phi_j(x) dx = \delta_{k,j} \quad (2)$$

où $\delta_{k,j}$ est le symbol de Kronecker ($\delta_{k,j}$ est 1 si $k = j$, 0 sinon). On a donc :

$$a_{\ell}(t) = (u(x, t), \phi_{\ell}(x)) = \int_{\Omega} u(x, t) \phi_{\ell}(x) dx \quad (3)$$

On remarque que pour une suite de fonctions orthonormées $\phi_1(x), \dots, \phi_K(x)$, le coefficient $a_{\ell}(t)$ ne dépend que de la fonction ϕ de degré ℓ ($\phi_{\ell}(x)$). On remarque aussi que *l'orthogonalité de ces fonctions est une propriété très importante*.

Alors, le problème posé est : comment déterminer une telle suite de fonctions orthonormées $\phi_1(x), \dots, \phi_K(x)$ de telle sorte que l'approximation (1) soit la meilleure possible au sens des moindres carrées. Pour cela, on suppose qu'on connaît la valeur de $u(x, t)$ en M_e localisations spatiales x_1, \dots, x_{M_e} (ex : les nœuds d'un maillage) et chacun en N_t instants différents.

1.1 La méthode des moindres carrés

La méthode des moindres carrés (due à Gausse et à Legendre), permet de comparer des données expérimentales, généralement entachées d'erreurs de mesure, à un modèle mathématique (voir ci-dessus) censé décrire ces données. Elle permet, entre autres, de minimiser l'impact des erreurs expérimentales en *ajoutant certaines informations* dans le processus de mesure. Cette méthode permet de sélectionner une fonction ϕ parmi une famille de fonction possibles représentant le mieux les données expérimentales. Elle définit cette fonction comme celle qui minimise la somme quadratique des **déviations** des mesures de u à des instants t_i aux prédictions $\sum_{k=1}^K a_k(t_i) \phi_k(x)$. Si, par exemple, nous disposons de N_t mesures $u(x, t_i)$ (pour $i = 1, \dots, N_t$), les fonction optimales au sens de la méthode des moindres carrés sont celles qui minimisent la quantité :

$$\sum_{i=1}^{N_t} R_i^2(x) = \sum_{i=1}^{N_t} \left(u(x, t_i) - \sum_{k=1}^K a_k(t_i) \phi_k(x) \right)^2 = \sum_{i=1}^{N_t} \left(u(x, t_i) - \sum_{k=1}^K (u(x, t_i), \phi_k(x)) \phi_k(x) \right)^2 \quad (4)$$

où $R_i(x)$ sont les résidus du modèle. Autrement dit, on cherche des fonctions $\phi_1(x), \dots, \phi_K(x)$ avec $K \leq N_t$ tel que :

$$\min_{\phi_k} \left(\sum_{i=1}^{N_t} R_i^2(x) \right) \quad (5)$$

1.2 La méthode des Snapshots

Supposons que les réalisations $u(x, t_1), \dots, u(x, t_{N_t})$ soient rangées dans la matrice suivante :

$$A = \begin{pmatrix} u(x_1, t_1) & u(x_1, t_2) & \dots & u(x_1, t_{N_t-1}) & u(x_1, t_{N_t}) \\ u(x_2, t_1) & u(x_2, t_2) & \dots & u(x_2, t_{N_t-1}) & u(x_2, t_{N_t}) \\ \vdots & \vdots & & \vdots & \vdots \\ u(x_{M_e}, t_1) & u(x_{M_e}, t_2) & \dots & u(x_{M_e}, t_{N_t-1}) & u(x_{M_e}, t_{N_t}) \end{pmatrix} \quad (6)$$

Cette matrice est appelée la matrice des Snapshots. On peut noter que chaque colonne de A représente une réalisation de u à un instant donné

L'indépendance linéaire des données et le rang de A . Le rang de la matrice A est maximal si ...

On peut montrer que la SVD tronquée (SVDT) de la matrice A à l'ordre K permet de résoudre le problème (5).

A développer...

1.3 La décomposition SVD d'une matrice

Soit A une matrice réelles de dimensions $M_e \times N_t$. La SVD de A est une

$$A = U \Sigma V^T \quad (7)$$

où $U = [u_1, u_2, \dots, u_{M_e}] \in \mathbb{R}^{M_e \times M_e}$ et $V = [v_1, v_2, \dots, v_{N_t}] \in \mathbb{R}^{N_t \times N_t}$ sont des matrices orthogonales ($U^T = U^{-1}$, $V^T = V^{-1}$) et $\Sigma \in \mathbb{R}^{r \times r}$ est une matrice diagonale

$$\Sigma = \begin{pmatrix} \delta_1 & & & \\ & \delta_2 & & \\ & & \ddots & \\ & & & \delta_r \end{pmatrix} \quad (8)$$

avec $\delta_1 \geq \delta_2 \geq \dots \geq \delta_r \geq 0$ et $r = \min(M_e, N_t)$. Le rang de la matrice A est égale au nombre de valeurs singulières non-nulles de A . Les r **premières lignes** de V^T et de V sont appelés vecteurs singuliers droit et gauche de A . En général, on ne calcule pas les valeurs et vecteurs singuliers de la matrice A mais plutôt les valeurs et vecteurs propres des matrices AA^T et $A^T A$ (voir la section suivante).

1.4 La SVD et la EVD d'une matrice

Le calcul des matrices U , V et Σ d'une décomposition SVD d'une matrice rectangulaire quelconque $A \in \mathbb{R}^{M_e \times N_t}$ peut être fait à l'aide de la décomposition EVD des matrices carrées $A^T A \in \mathbb{R}^{N_t \times N_t}$ et $AA^T \in \mathbb{R}^{M_e \times M_e}$. En effet, en utilisant la décomposition (13) de A , on a :

$$A^T A = V \Sigma U^T U \Sigma V^T = V \Sigma^2 V^T \quad (9)$$

Or, $A^T A$ (resp. AA^T) étant une matrice hermitienne diagonalisable, il existe une matrice diagonale Λ et une matrice orthogonale des vecteurs propres X (resp. Y) telle que :

$$A^T A = X \Lambda X^T = V \Sigma^2 V^T \quad (10)$$

et

$$AA^T = Y \Lambda Y^T = U \Sigma^2 U^T \quad (11)$$

Par conséquent, $V = X$ (resp. $U = Y$) et $\Sigma^2 = \Lambda$ sont les vecteurs et les valeurs propres de $A^T A$ (resp. AA^T). et donc

$$\Sigma = \begin{pmatrix} \delta_1 & & & \\ & \delta_2 & & \\ & & \ddots & \\ & & & \delta_r \end{pmatrix} = \begin{pmatrix} \sqrt{\lambda_1} & & & \\ & \sqrt{\lambda_2} & & \\ & & \ddots & \\ & & & \sqrt{\lambda_r} \end{pmatrix} \quad (12)$$

Autrement dit, pour obtenir les valeurs et vecteurs (droite et gauche) singuliers d'une matrice rectangulaire A , il suffit de calculer les valeurs et vecteurs propres des matrices symétriques $A^T A$ et AA^T . Lorsque $N_t \ll M_e$, il est plus intéressant de calculer la EVD de la matrice $A^T A$ (Snapshots). Dans le cas contraire, il est préférable de résoudre le problème EVD de la matrice AA^T (méthode usuelle).

2 Algorithmes de calcul de la SVD

3 Applications

Une des applications de la SDV d'une matrice est la compression d'image. En effet, dans une telle applications, on cherche à déterminer une matrice X de même taille que A mais de rang inférieur, minimisant sa distance avec A au sens d'une certaine norme. Une autre application est celle de "clustering". Généralement, la théorie spectrale de graphes permet de faire le partitionnement par la décomposition diagonale d'une matrice carrée (eigen-decomposition). Mais l'"eigen-decomposition" n'existe pas pour les matrices rectangulaires qui représentent la plupart des ensembles de données à partitionner. Dans ce cas, on utilise plutôt une SVD-décomposition qui permet de partitionner efficacement de telles matrices en utilisant les signes des composants de certains vecteurs singuliers (droit ou gauche) de la matrice représentant le graphe.

La technique de clustering consiste à placer des objets d'une BD dans des groupes des objets similaires. Quand la BD est représentée par une matrice, ce partitionnement consiste à réordonner les colonnes ou les lignes de la matrice afin de placer celles qui se ressemblent ensemble (???). Il existe des centaines de méthodes pour la création de ce type de groupes. Elles pourront être classées dans deux catégories : *clustering* hiérarchique et *clustering* partitionné. La 1ère décompose la BD en des ensembles emboîtés (crée un arbre en partant de la racine ou à l'envers) basés sur certaines métriques de similarité entre les objets-feuilles (comme par exemple des méthodes proches voisins, centroides). Le seconde décompose la BD en sous-ensembles disjoints (comme PDDP-bissection, SVD ou bien méthodes spectrales). PageRank, analyse en composant indépendant, k-means sont aussi parmi les méthodes de cette 2ème catégorie. Nous détaillons ici le *clustering* spectral incluant des algorithmes signed-SVD et Gap-SVD qui nous intéressent tout particulièrement.

3.1 La méthode Fiedler

Cette méthode est à la base de l'utilisation de la SVD pour le clustering. Soit A la matrice adjacente représentant un graphe. La Laplacienne de cette matrice est définie par $L=D-A$ où D est une matrice diagonale dont les éléments diagonaux sont définis par la somme des lignes de A . Cette matrice est réelle symétrique définie positive et sa plus petite valeur propre est zéro. Le signes des composants du vecteur propres associé au second plus petite valeurs propres (**vecteur de Fiedler**) pourront être utilisés pour décomposer l'ensemble en sous-ensemble. Les composants positifs (ou négatifs) correspondent aux nœuds (lignes) appartenant aux même groupe. L'algorithme est **itératif** et à chaque itération, nous devons résoudre un problème de valeur propre, donc très **couteux**. En effet, pour décomposer le graphe en k groupes, il faut faire $k - 1$ itérations (en tout cas pour notre exemple). En plus il n'est applicable qu'aux **matrices carrées et symétriques**. Une solution au problème du coût élevé de cet algorithme est donnée dans la section suivante.

3.2 La méthode Fiedler étendue

Afin de remédier au problème de coût de l'algorithme Fiedler, nous pourrions calculer les vecteurs propres associés aux k plus petites (après la plus petite qui est zéro) valeurs propres. Dans ce cas les composants du même signe ou bien du même séquence de signes définissent un même groupe. Ainsi avec une seule décomposition diagonale de la matrice, nous définissons un nombre k de groupes. Nous avons donc décomposé notre ensemble en k sous-ensemble avec une seule itération de eigen-décomposition et le problème du coût élevé ne se pose plus. Cependant, cette

méthode n'est toujours pas applicable aux matrices rectangulaires (Fiedler étendu ne marche que pour les matrices carrées dont les valeurs et vecteurs propres sont réels— donc symétrique).

3.3 SVD au lieu de eigen-décomposition

Afin de pouvoir résoudre le problème de clustering une BD représentée par une matrice rectangulaire A , nous pourrions utiliser la décomposition en valeurs singulières de A . Soit $A_{m \times n}$ une matrice du rang r dont la décomposition SVD est :

$$A_{m \times n} = U_{m \times r} \Sigma_{r \times r} V_{r \times n}^T \quad (13)$$

Comme nous l'avons vu dans les sections précédentes, la SVD d'une matrice est liée à sa EVD. En effet, les matrices $B = AA^T$ et $C = A^T A$ sont réelles et symétriques avec les valeurs propres s_i^2 et les vecteurs singuliers gauche u_i (resp. droits v_i) sont les vecteurs propres de B (resp. C). Un des 1ers intérêts de la SVD est qu'elle permet de trouver une approximation de A du rang $k < r$. Pour cela, il suffit de calculer les k premiers vecteurs colonnes de U et et vecteurs lignes de V^T et de constituer une approximation de A par $U_{m \times k} \Sigma_{k \times k} V_{k \times n}^T$. On peut montrer que cette approximation de A est optimale et cette méthode est appelée SVDT (pour SVD tronquée). Le problème posé est alors : comment choisir la valeur de k ? On pourrait par exemple choisir k comme la valeur (parmi les SV) qui définit un palier sur les valeurs descendantes des SV.

3.4 Petits exemples et l'interprétation géométrique

Afin de tester la SVD pour le clustering, une méthode serait de définir une matrice de relativement petite taille (par exemple 45 X 24) dont les lignes représentent des mots clés et les colonnes les résultats de recherche Google. Ce tableau 2D est une matrice bloc diagonale avec des données bien ordonnées. On effectue une permutation sur cette matrice pour obtenir A . L'application de la SVD sur A devra nous conduire à la cette matrice **avant** la permutation. Un autre exemple simple pourrait être l'application de la SVD à un nuage de points afin de détecter les axes principaux de ce nuage de points.