Système de Recommandation

Basé sur la Factorisation de Matrices

Binh Minh TRAN

Février 2025

Mentore : Prof. Célia Da Costa Pereira

Maître de Conférences (PhD, HDR), Université Côte d'Azur

STID - SPARKS (I3S Lab)

Un Projet Personnel

GitHub : github.com/RecSys-Movies

Table des matières

1 Introduction

Ce projet se concentre sur la construction et l'exploration d'un système de recommandation en utilisant la Factorisation de Matrices afin d'améliorer la précision des prédictions concernant les préférences des utilisateurs. En appliquant la Factorisation de Matrices, où les utilisateurs et les films sont modélisés sous forme de vecteurs de caractéristiques, j'optimise leur alignement grâce à un processus d'entraînement et à l'optimisation d'une fonction de perte. Je combine des recherches existantes avec mes propres idées et explore des moyens de les développer davantage. Les résultats finaux montrent une amélioration de la précision dans la prédiction des films susceptibles d'intéresser les utilisateurs. De plus, j'ai essayé d'utiliser l'ACP (Analyse en Composantes Principales) pour réduire la dimension de la matrice, dans le but de diminuer le temps de calcul tout en minimisant l'augmentation de l'erreur et en tirant quelques observations.

2 Notation

Notation	Description
$r(i,j)$	scalaire ; $= 1$ si l'utilisateur j a noté le film $i, = 0$ sinon
$y(i,j)$	scalaire ; note donnée par l'utilisateur j pour le film i (définie si $r(i,j) = 1$)
$\mathbf{w}^{(j)}$	vecteur ; vecteur de caractéristiques de l'utilisateur j
$\mathbf{x}^{(i)}$	vecteur ; vecteur de caractéristiques du film i
$b^{(i,j)}$	scalaire ; paramètre de biais pour le film i et l'utilisateur j
$b_u^{(j)}$	scalaire ; paramètre de biais pour l'utilisateur j
$b_m^{(i)}$	scalaire ; paramètre de biais pour le film i
b_u	vecteur ; paramètres de biais pour l'utilisateur j
b_m	vecteur ; paramètres de biais pour le film i
n_u	nombre d'utilisateurs
n_m	nombre de films
k	nombre de caractéristiques
$\mathbf{X}$	matrice des vecteurs $\mathbf{x}^{(i)}$
$\mathbf{W}$	matrice des vecteurs $\mathbf{w}^{(j)}$
$\mathbf{B}$	matrice des paramètres de biais $b^{(i,j)}$
$\mathbf{R}$	matrice des éléments $r(i,j)$
$\mathbf{Y}$	matrice des éléments $y(i,j)$
$\hat{\mathbf{Y}}$	matrice estimée de $\mathbf{Y}$
$\mathbf{Ynorm}$	matrice des éléments normalisés de $y(i,j)$

3 Factorisation de Matrices

La Factorisation de Matrices est une méthode utilisée dans le Filtrage Collaboratif [1], une technique qui prédit automatiquement l'intérêt d'un utilisateur actif en recueillant des informations de notation provenant d'autres utilisateurs ou d'éléments similaires. Son objectif est d'apprendre les facteurs latents qui capturent les relations entre les utilisateurs et les éléments (par exemple, les films). Dans cette approche, chaque facteur latent représente une caractéristique cachée influençant la manière dont les utilisateurs notent les films. Pour les films, chaque facteur peut correspondre à des caractéristiques telles que le genre, l'ambiance ou le style, qui contribuent à la note. Pour les utilisateurs, chaque facteur représente leur préférence pour les caractéristiques correspondantes dans les films qu'ils notent favorablement.

Imaginons que nous ayons une matrice de notation Y de dimension n_m films $\times$ n_u utilisateurs où $y(i,j)$ est la note donnée par l'utilisateur j pour le film i. Les facteurs latents ne sont pas les vecteurs $y(i)$ ou $y(j)$ dans la matrice Y ; ils sont définis par deux matrices, X et W, où :

- La matrice des caractéristiques des films, de dimension $n_m \times k$ (films $\times$ caractéristiques), contient les vecteurs de caractéristiques des films $x^{(i)}$.

- La matrice des préférences des utilisateurs, de dimension $k \times n_u$ (caractéristiques $\times$ utilisateurs), contient les vecteurs de caractéristiques des utilisateurs $w^{(j)}$.

Le produit scalaire de X et W représente la collaboration entre les caractéristiques des films et celles des utilisateurs, capturant les facteurs latents qui influencent la manière dont les utilisateurs notent les films. Cela aboutit à la matrice de notation approximée $\hat{Y}$, avec l'espérance $E(\hat{Y}) = Y$, ce qui signifie que les notes prédites doivent être aussi proches que possible des notes réelles.

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}^{(0)} \\ \mathbf{x}^{(1)} \\ \vdots \\ \mathbf{x}^{(n_m-1)} \end{bmatrix} \cdot \mathbf{W} = \begin{bmatrix} \mathbf{w}^{(0)} & \mathbf{w}^{(1)} & \cdots & \mathbf{w}^{(n_u-1)} \end{bmatrix}$$

$$\hat{\mathbf{Y}} = \begin{bmatrix} \mathbf{x}^{(0)} \cdot \mathbf{w}^{(0)} & \cdots & \mathbf{x}^{(0)} \cdot \mathbf{w}^{(n_u-1)} \\ \mathbf{x}^{(1)} \cdot \mathbf{w}^{(0)} & \cdots & \mathbf{x}^{(1)} \cdot \mathbf{w}^{(n_u-1)} \\ \vdots & & \vdots \\ \mathbf{x}^{(n_m-1)} \cdot \mathbf{w}^{(0)} & \cdots & \mathbf{x}^{(n_m-1)} \cdot \mathbf{w}^{(n_u-1)} \end{bmatrix}$$

Notez que nous pouvons également minimiser la distance entre la matrice de notation approximée $\hat{Y}$ et la matrice de notation Y sans utiliser les matrices X et W en initialisant aléatoirement $\hat{Y}$ pour adapter le modèle. Cependant, cette approche augmente considérablement le temps de calcul en raison du grand nombre de valeurs qui doivent être calculées simultanément à mesure que la taille de la matrice augmente. En revanche, le produit scalaire entre X et W permet au modèle de converger plus rapidement.

De plus, en pratique, certains utilisateurs peuvent avoir des opinions extrêmes, positives ou négatives, sur un film, même sans l'avoir vu, ou certains films peuvent être plus fortement notés en raison de la présence d'acteurs populaires. Ce sont des exemples de biais. Pour améliorer la précision des prédictions, il est essentiel d'incorporer un terme de biais [2], ce qui conduit à l'expression affinée suivante :

$$\hat{Y} = X \cdot W + b_u^{(j)} + b_m^{(i)} + \mu$$

où $b_u^{(j)}$ est le biais de l'utilisateur - les écarts observés de l'utilisateur j, et $b_m^{(i)}$ est le biais du film - les écarts observés du film i. Par exemple, le film i a tendance à recevoir $b_m^{(i)}$ de plus ou de moins par rapport aux autres films, et l'utilisateur j a tendance à attribuer en moyenne $b_u^{(j)}$ de plus ou de moins pour tous ses films par rapport aux autres utilisateurs. μ représente la moyenne de toutes les notes attribuées.

$$\hat{\mathbf{Y}} = \begin{bmatrix} \mathbf{x}^{(0)} \cdot \mathbf{w}^{(0)} + b_u^{(0)} + b_m^{(0)} + \mu & \cdots & \mathbf{x}^{(0)} \cdot \mathbf{w}^{(n_u-1)} + b_u^{(n_u-1)} + b_m^{(0)} + \mu \\ \mathbf{x}^{(1)} \cdot \mathbf{w}^{(0)} + b_u^{(0)} + b_m^{(1)} + \mu & \cdots & \mathbf{x}^{(1)} \cdot \mathbf{w}^{(n_u-1)} + b_u^{(n_u-1)} + b_m^{(1)} + \mu \\ \vdots & & \vdots \\ \mathbf{x}^{(n_m-1)} \cdot \mathbf{w}^{(0)} + b_u^{(0)} + b_m^{(n_m-1)} + \mu & \cdots & \mathbf{x}^{(n_m-1)} \cdot \mathbf{w}^{(n_u-1)} + b_u^{(n_u-1)} + b_m^{(n_m-1)} + \mu \end{bmatrix}$$

Ensuite, pour apprendre les vecteurs de caractéristiques $x^{(i)}$ et $w^{(j)}$ ou les termes de biais, l'équation d'erreur carrée régularisée sur l'ensemble des notations connues [2] est :

$$\min_{\mathbf{x},\mathbf{w},\mathbf{b_u},\mathbf{b_m}} \frac{1}{2} \sum_{(i,j)\in S} \left(\mathbf{x}^{(i)} \cdot \mathbf{w}^{(j)} + \mathbf{b_u}^{(j)} + \mathbf{b_m}^{(i)} + \mu - y(i,j)\right)^2 + \underbrace{\frac{\lambda}{2}\left(||\mathbf{w}^{(j)}||^2 + ||\mathbf{x}^{(i)}||^2 + (\mathbf{b_u}^{(j)})^2 + (\mathbf{b_m}^{(i)})^2\right)}_{\text{régularisation}} \tag{1}$$

Où S est l'ensemble de toutes les paires utilisateur-élément (i, j) avec des notations réelles.

Ensuite, nous apprenons les paramètres en utilisant la descente de gradient.

$$w^{(j)} = w^{(j)} - \alpha \cdot \frac{\partial \mathcal{L}}{\partial w^{(j)}} \tag{2}$$

$$x^{(i)} = x^{(i)} - \alpha \cdot \frac{\partial \mathcal{L}}{\partial x^{(i)}} \tag{3}$$

$$b_m^{(i)} = b_m^{(i)} - \alpha \cdot \frac{\partial \mathcal{L}}{\partial b_m^{(i)}} \tag{4}$$

$$b_u^{(j)} = b_u^{(j)} - \alpha \cdot \frac{\partial \mathcal{L}}{\partial b_u^{(j)}} \tag{5}$$

Où :

- $\frac{\partial \mathcal{L}}{\partial w^{(j)}} = \sum_{(i,j)\in S} \left(\left(\mathbf{x}^{(i)} \cdot \mathbf{w}^{(j)} + b_u^{(j)} + b_m^{(i)}\right) - y(i,j)\right) \cdot \mathbf{x}^{(i)} + \lambda w^{(j)}$ représente la fonction de perte dérivée associée au paramètre $w^{(j)}$.
- $\frac{\partial \mathcal{L}}{\partial x^{(i)}} = \sum_{(i,j)\in S} \left(\left(\mathbf{x}^{(i)} \cdot \mathbf{w}^{(j)} + b_u^{(j)} + b_m^{(i)}\right) - y(i,j)\right) \cdot \mathbf{w}^{(j)} + \lambda x^{(i)}$ représente la fonction de perte dérivée associée au paramètre $x^{(i)}$.
- $\frac{\partial \mathcal{L}}{\partial b_m^{(i)}} = \sum_{(i,j)\in S} \left(\left(\mathbf{x}^{(i)} \cdot \mathbf{w}^{(j)} + b_u^{(j)} + b_m^{(i)}\right) - y(i,j)\right) + \lambda b_m^{(i)}$ représente la fonction de perte dérivée associée au paramètre $b_m^{(i)}$.
- $\frac{\partial \mathcal{L}}{\partial b_u^{(j)}} = \sum_{(i,j)\in S} \left(\left(\mathbf{x}^{(i)} \cdot \mathbf{w}^{(j)} + b_u^{(j)} + b_m^{(i)}\right) - y(i,j)\right) + \lambda b_u^{(j)}$ représente la fonction de perte dérivée associée au paramètre $b_u^{(j)}$.
- α est un taux d'apprentissage (adam-optimizer)
- $\mathcal{L}$ est la fonction de perte 1.

4 Méthodologie

4.1 Jeu de données

Dans ce projet, deux ensembles de données sont utilisés : l'ensemble de données des notations et l'ensemble de données de la liste des films. Ces ensembles de données ont été créés par MovieLens le 26 septembre 2018, avec des données provenant de 610 utilisateurs et plus de 100 000 notations. Ils font partie de la collection des derniers ensembles de données MovieLens.

4.2 Prétraitement des données

Matrice Y : Ce projet traite les données de l'ensemble de données des notations pour construire une matrice de dimensions 9724 films par 610 utilisateurs, notée Y, représentant les notations des utilisateurs j pour chaque film i. La matrice Y est remplie avec toutes les notations

disponibles, et toutes les valeurs manquantes (NAs) sont remplacées par 0. Il convient de noter que, dans la plupart des applications, Y est une matrice creuse, car seule une petite fraction de ses éléments est non nulle [3]. Dans l'ensemble de données de ce projet, environ 1,7% des éléments sont non nuls. Les notations dans Y varient de 0,5 à 5, et toutes les valeurs en dehors de cette plage sont réinitialisées à 0, ce qui indique une notation non attribuée.

L'objectif est de prédire les cellules ayant une valeur de 0, qui représentent les notations non attribuées.

Dans le but d'explorer le système de recommandation, j'ai ajouté un utilisateur supplémentaire [Moi] qui a noté tous les films que j'ai considérés comme mes préférences dans l'ensemble de données des films. Ainsi, la dimension de Y est maintenant de 9724 **x** 611.

Matrice R : Cette matrice est basée sur la matrice Y, incluant 1 pour les notations qui ont été attribuées et 0 sinon.

Matrice B : Dans la pratique, certains utilisateurs peuvent avoir des avis extrêmes positifs ou négatifs sur un film, même sans l'avoir vu, ou certains films peuvent être plus fortement notés en raison de la présence d'acteurs populaires. Ce sont des exemples de biais. L'intégration du biais dans le processus d'apprentissage du modèle est cruciale pour capturer et expliquer avec précision les caractéristiques sous-jacentes.

Le biais pour l'utilisateur j est calculé comme suit :

$$b_u^{(j)} = \bar{y}(j) - \mu$$

Où :
- μ représente la moyenne de toutes les notations attribuées.
- $\bar{y}(j)$ est la notation moyenne donnée par l'utilisateur j, calculée comme suit :

$$\bar{y}(j) = \frac{\sum_{i=0}^{n_m-1} y(i,j) \cdot r(i,j)}{\sum_{i=0}^{n_m-1} r(i,j)}$$

Le biais pour le film i est calculé comme suit :

$$b_m^{(i)} = \bar{y}(i) - \mu$$

Où :
- μ représente la moyenne de toutes les notations attribuées.
- $\bar{y}(i)$ est la notation moyenne du film i.

Le biais pour le film i donné par l'utilisateur j est calculé comme suit :

$$b^{(i,j)} = b_m^{(i)} + b_u^{(j)}$$

4.3 Sélection initiale du modèle

Il est important de noter que dans ce contexte, la matrice $\hat{Y}$ ne contient plus de cellules avec une valeur de 0, qui représentaient les notations non attribuées, car ces dernières ont été remplacées par des notations prédites. De plus, les notations prédites peuvent dépasser la valeur maximale de 5 sur l'échelle de notation.

L'objectif est de minimiser la distance entre $\hat{Y}$ et Y. Pour ce faire, une fonction de perte $\mathcal{L}$ est utilisée pour quantifier cette distance. Cette fonction de perte est spécifiquement conçue pour pénaliser les paramètres du modèle, facilitant ainsi une réduction efficace de l'erreur globale.

La fonction de perte $\mathcal{L}$ est calculée comme présenté dans [2], mais spécifiquement pour ce projet :

$$\mathcal{L}(\mathbf{x^{(i)}}, \mathbf{w^{(j)}}, \mathbf{b_u}^{(j)}, \mathbf{b_m}^{(i)}) = \frac{1}{2} \sum_{(i,j):r(i,j)=1} \left(\mathbf{x}^{(i)} \cdot \mathbf{w}^{(j)} + \mathbf{b_u}^{(j)} + \mathbf{b_m}^{(i)} + \mu - y(i,j) \right)^2 + \underbrace{\frac{\lambda}{2} \left(||\mathbf{w}^{(j)}||^2 + ||\mathbf{x}^{(i)}||^2 + (\mathbf{b_u}^{(j)})^2 + (\mathbf{b_m}^{(i)})^2 \right)}_{\text{régularisation}} \tag{6}$$

Où :

- La notation $(i,j) : r(i,j) = 1$ représente l'ensemble des notations observées, indiquant que la cellule correspondante dans la matrice des notations R a une valeur de 1. Il est important d'exclure les cellules où $r(i,j) = 0$ afin d'empêcher le modèle d'ajuster les paramètres pour ajuster les notations prédites à des valeurs indéfinies (NAs).

4.4 Ajustement du modèle

Après l'exécution du modèle initial, il a été observé que la valeur du coût résultant était relativement élevée. Par conséquent, des ajustements du biais sont effectués pour déterminer si cette modification peut réduire le coût total. Plus précisément, l'ajustement du modèle contient un seul terme de biais, qui est une combinaison de deux termes de biais, $b_u^{(j)}$ et $b_m^{(i)}$.

Fondamentalement, nous avons la matrice de biais B, qui inclut le biais $b^{(i,j)}$ de l'utilisateur j pour le film i. Ce biais est défini comme suit :

$$b^{(i,j)} = b_m^{(i)} + b_u^{(j)}$$

Où $b_m^{(i)}$ représente le biais du film i, et $b_u^{(j)}$ représente le biais de l'utilisateur j.

La matrice des notations prédites est maintenant définie comme suit :

$$\hat{\mathbf{Y}} = X \cdot W + B$$

$$\hat{\mathbf{Y}} = \begin{bmatrix} \mathbf{x}^{(0)} \cdot \mathbf{w}^{(0)} + b^{(0,0)} + \mu & \cdots & \mathbf{x}^{(0)} \cdot \mathbf{w}^{(n_u-1)} + b^{(0,n_u-1)} + \mu \\ \mathbf{x}^{(1)} \cdot \mathbf{w}^{(0)} + b^{(1,0)} + \mu & \cdots & \mathbf{x}^{(1)} \cdot \mathbf{w}^{(n_u-1)} + b^{(1,n_u-1)} + \mu \\ \vdots & & \vdots \\ \mathbf{x}^{(n_m-1)} \cdot \mathbf{w}^{(0)} + b^{(n_m-1,0)} + \mu & \cdots & \mathbf{x}^{(n_m-1)} \cdot \mathbf{w}^{(n_u-1)} + b^{(n_m-1,n_u-1)} + \mu \end{bmatrix}$$

Puis, l'équation de l'erreur au carré pour cet ajustement est la suivante :

$$\mathcal{L}(\mathbf{x^{(i)}}, \mathbf{w^{(j)}}, \mathbf{b}^{(i,j)}) = \frac{1}{2} \sum_{(i,j):r(i,j)=1} \left(\mathbf{x}^{(i)} \cdot \mathbf{w}^{(j)} + \mathbf{b}^{(i,j)} + \mu - y(i,j) \right)^2 + \underbrace{\frac{\lambda}{2} \left(||\mathbf{w}^{(j)}||^2 + ||\mathbf{x}^{(i)}||^2 + (\mathbf{b}^{(i,j)})^2 \right)}_{\text{régularisation}} \tag{7}$$

4.4.1 Ajustements supplémentaires

Tout d'abord, j'ai décidé de ne pas inclure μ (le facteur fixe dans le modèle) dans le terme de prédiction, comme indiqué dans l'équation 1. Lorsque μ est inclus, la moyenne de toutes les notations attribuées, μ, domine les notations prédites, ce qui fait que le terme de biais et le produit scalaire de X et W prennent un rôle secondaire. Cela peut empêcher le modèle d'apprendre efficacement les interactions et les biais, ce qui conduit à une convergence plus lente. Je souhaite que le modèle se concentre davantage sur les interactions entre les utilisateurs et les films, donc j'ai retiré μ pour permettre au modèle de mieux capturer les facteurs latents. Puis,

$$\hat{Y} = X \cdot W + b_u^{(j)} + b_m^{(i)}$$

Maintenant, le système ne comprend plus le terme fixe μ, qui jouait auparavant un rôle important dans la prédiction des évaluations. Au lieu de cela, le système apprendra à partir des paramètres d'entraînement X, W et des biais. Un point clé est que je ne veux pas que le modèle ajuste directement la matrice des évaluations Y, mais plutôt la matrice des évaluations normalisées Y_{norm}, afin de se concentrer sur les facteurs relatifs plutôt que sur les facteurs absolus. Par exemple, en raison de la popularité de certains films, certaines évaluations peuvent être plus élevées que la moyenne. Concrètement, nous visons à normaliser les évaluations réelles lorsque $r(i, j) = 1$.

La matrice normalisée Y est calculée comme suit :

$$Y_{\text{norm}}(i, j) = (y(i, j) - \bar{y}(i)) \cdot r(i, j)$$

Où :
- $\bar{y}(i)$ est la note moyenne du film i.

$$\bar{y}(i) = \frac{\sum_{j=0}^{n_u - 1} y(i, j) \cdot r(i, j)}{\sum_{j=0}^{n_u - 1} r(i, j)}$$

L'objectif est de minimiser la distance entre $\hat{Y}$ et Y_{norm}. Pour ce faire, une fonction de perte $\mathcal{L}$ est utilisée pour quantifier cette distance. Cette fonction de perte est spécifiquement conçue pour pénaliser les paramètres du modèle, facilitant ainsi une réduction efficace de l'erreur globale.

$$\begin{aligned}\mathcal{L}(\mathbf{x^{(i)}}, \mathbf{w^{(j)}}, \mathbf{b_u^{(j)}}, \mathbf{b_m^{(i)}}) = \frac{1}{2} \sum_{(i,j):r(i,j)=1} \Big(\mathbf{x}^{(i)} \cdot \mathbf{w}^{(j)} + \mathbf{b_u}^{(j)} + \mathbf{b_m}^{(i)} - y_{\text{norm}}(i, j)\Big)^2 \\ + \underbrace{\frac{\lambda}{2} \left(||\mathbf{w}^{(j)}||^2 + ||\mathbf{x}^{(i)}||^2 + (\mathbf{b_u}^{(j)})^2 + (\mathbf{b_m}^{(i)})^2 \right)}_{\text{régularisation}}\end{aligned} \tag{8}$$

Où :
- L'évaluation normalisée dans la matrice **Ynorm** est désignée par $y_{\text{norm}}(i, j)$.

Cet ajustement supplémentaire est appelé **Approche 1**.

Approche 1 : Cette approche implique quatre paramètres d'entraînement : X, W, b_u et b_f. Tout d'abord, le modèle "entraîne partiellement" ces paramètres, ce qui signifie que seulement X et W sont initialisés de manière aléatoire, tandis que b_u et b_f sont initialisés à l'aide de valeurs pré-calculées de 4.2. Je pense que les biais pré-calculés peuvent aider le modèle à converger

plus rapidement. Je fais référence à cela comme l'approche **1.B**. Ensuite, le modèle "entraîne complètement" ces quatre paramètres, ce qui signifie qu'ils sont initialisés de manière aléatoire ; je fais référence à cela comme l'approche **1.A**.

Approche 2 : Cette approche implique trois paramètres d'entraînement : X, W et B. Tout d'abord, le modèle "entraîne partiellement" ces paramètres, ce qui signifie que seulement X et W sont initialisés de manière aléatoire, tandis que B est initialisé à l'aide de valeurs pré-calculées de 4.2. Je pense que les biais pré-calculés peuvent aider le modèle à converger plus rapidement. Je fais référence à cela comme l'approche **2.B**. Ensuite, le modèle "entraîne complètement" ces trois paramètres, ce qui signifie qu'ils sont initialisés de manière aléatoire ; je fais référence à cela comme l'approche **2.A**.

Rappelez-vous que B est la matrice des biais où $b^{(i,j)}$ est la combinaison de $b_m^{(i)}$ et $b_u^{(j)}$. Par conséquent, la fonction de perte est calculée comme suit :

$$\begin{aligned}\mathcal{L}(\mathbf{w}^{(j)}, \mathbf{x}^{(i)}, \mathbf{b}^{(i,j)}) =& \frac{1}{2} \sum_{(i,j):r(i,j)=1} \left(\mathbf{x}^{(i)} \cdot \mathbf{w}^{(j)} + \mathbf{b}^{(i,j)} - y_{\text{norm}}(i,j)\right)^2 \\ &+ \underbrace{\frac{\lambda}{2}\left(||\mathbf{w}^{(j)}||^2 + ||\mathbf{x}^{(i)}||^2 + (\mathbf{b}^{(i,j)})^2\right)}_{\text{régularisation}} \end{aligned} \tag{9}$$

Les mises à jour par descente de gradient sont maintenant définies par les Équations 3 et 4, avec une équation d'ajustement supplémentaire pour le paramètre $\mathbf{b}^{(i,j)}$ calculée comme suit :

$$b^{(i,j)} = b^{(i,j)} - \alpha \cdot \frac{\partial \mathcal{L}}{\partial b^{(i,j)}}$$

Où :
- $\mathcal{L}$ est la fonction de perte donnée par l'Équation 9

4.5 Évaluations Prédites

La matrice des évaluations prédites pour le modèle initial et le modèle ajusté est simplement la matrice $\hat{Y}$,

$$\textbf{Matrice des évaluations prédites} = \widehat{Y}$$

qui correspond à chaque modèle. Cependant, cela n'est pas le cas pour le modèle avec ajustements supplémentaires.

Pour le modèle avec ajustements supplémentaires 4.4.1, après avoir obtenu la matrice $\widehat{Y}$, nous devons revenir à la forme originale des évaluations en ajoutant la note moyenne de chaque film, qui avait été soustraite dans le but de normaliser 4.4.1. Il existe un petit problème avec les nouveaux utilisateurs qui n'ont pas fourni d'évaluations. Par conséquent, ces utilisateurs ne sont pas inclus dans le processus de normalisation, et leurs évaluations prédites pour le film i sont remplacées par la note moyenne du film i. En conséquence, la matrice des évaluations prédites est calculée comme suit :

$$\textbf{Matrice des évaluations prédites} = \widehat{Y} + \overline{Y_i}$$

Où :

- $\widehat{Y}$: représente la matrice des évaluations estimées de Y. La méthode pour calculer $\widehat{Y}$ dépend des considérations spécifiques prises en compte. Cette matrice contient les évaluations prédites $\hat{y}(i,j)$ pour le film i donné par l'utilisateur j.

- $\overline{Y_i}$ représente la matrice des évaluations moyennes des films, où $\overline{y_i}$ désigne la note moyenne du film i.

4.6 Analyse en Composantes Principales - ACP

L'objectif n'est pas d'utiliser l'ensemble du jeu de données pour l'analyse, mais de se concentrer sur ses composants les plus importants - ses composantes principales. En réduisant la taille d'une grande matrice, on fait face à un compromis : le modèle peut perdre une petite part de précision, mais le temps de calcul est considérablement réduit, ce qui nous permet de traiter des ensembles de données plus volumineux de manière plus efficace. De plus, après réduction, la matrice devient plus dense, car les informations essentielles sont concentrées dans quelques vecteurs compacts, plutôt que d'être dispersées avec beaucoup de valeurs nulles dans la matrice.

Dans ce projet, la matrice a une taille de 9724 films x 611 utilisateurs. Ici, je traite les 611 utilisateurs comme des caractéristiques pour chaque film. Ces caractéristiques représentent les "préférences" ou "tendances" de la manière dont le film est évalué. Ainsi, la matrice des caractéristiques des films est représentée par une matrice de dimension 611 x 611.

Étape 1 : Standardiser les données pour avoir une échelle de valeurs équivalente. La standardisation est réalisée en divisant une valeur de caractéristique par la valeur moyenne de toutes les caractéristiques, de sorte que les données seront centrées autour de 0.

Étape 2 : Calculer la matrice de covariance pour déterminer la corrélation de chaque caractéristique dans les données.

Étape 3 : Calculer les vecteurs propres et les valeurs propres à partir de la matrice de covariance, puis les trier par ordre de valeur décroissante.

Étape 4 : L'objectif est de sélectionner les composantes principales en considérant les vecteurs propres correspondant aux plus grandes ou aux plus significatives valeurs propres.

Dans ce projet, le modèle initial et les principaux ajustements du modèle sont conçus pour s'ajuster de manière rapprochée à la matrice des évaluations Y, tandis que les ajustements supplémentaires dans l'Approche 1 et l'Approche 2 visent à ajuster le modèle à la matrice des évaluations normalisées Y_{norm}. Ainsi, l'ACP doit être appliquée à trois matrices : Y, Y_{norm} et B.

Il est important de noter que l'ACP ne peut pas être appliquée à l'ajustement supplémentaire dans l'Approche 1.A. En effet, dans cette approche, les biais pour les utilisateurs $b_u^{(j)}$ et pour les films $b_m^{(i)}$ sont pré-calculés afin d'accélérer la convergence. Le problème réside dans le fait que ces deux matrices de biais ont des dimensions de 1 x 611 et 9724 x 1, respectivement, ce qui les rend incompatibles pour l'ACP.

La matrice des composantes principales $P1$ pour la matrice des évaluations Y a des dimensions $611 \times t$, où 611 représente le nombre de caractéristiques (utilisateurs) et t est le nombre de composantes principales. En revanche, les matrices des composantes principales $P2$ et $P3$ appliquées à la matrice des évaluations normalisées Y_{norm} et à la matrice B (utilisée dans les ajustements supplémentaires), respectivement, ont toutes les deux des dimensions $611 \times v$, où v désigne le nombre de composantes principales.

Étape 5 : Calculer les nouvelles matrices de dimension :

$$\mathbf{New\,Y} = Y \cdot P1, \quad \mathbf{New\,Y_{norm}} = Y_{norm} \cdot P2, \quad \mathbf{New\,B} = B \cdot P3.$$

Ici, $\mathbf{New\,Y}$ a des dimensions $9724 \times t$, tandis que $\mathbf{New\,Y_{norm}}$ et $\mathbf{New\,B}$ ont des dimensions $9724 \times v$.

Étape 6 : Pour le modèle initial et ses principaux ajustements, j'ajuste le modèle pour minimiser la différence :

$$\widehat{\mathbf{New}Y} - \mathbf{New\,Y}.$$

De même, pour les ajustements supplémentaires (suivant l'Approche 1 et l'Approche 2), j'ajuste le modèle pour minimiser :

$$\widehat{\mathbf{New}Y_{\mathbf{norm}}} - \mathbf{New\,Y_{norm}}.$$

Étape 7 : Projeter les matrices d'évaluations approximées, obtenues par réduction de dimension, vers l'espace dimensionnel d'origine (Reconstruction) :

$$\hat{Y}_{init} = \widehat{\mathbf{New}Y} \cdot P1^T, \quad \hat{Y}_{add} = \widehat{\mathbf{New}Y}_{norm} \cdot P2^T.$$

Bien que $\hat{Y}_{init}$ et $\hat{Y}_{add}$ partagent les mêmes dimensions, elles diffèrent par la notation, car $\hat{Y}_{add}$ n'est pas la matrice de prédiction finale.

Étape 8 : La matrice de prédictions pour le modèle initial (et ses principaux ajustements) est donnée par $\hat{Y}_{init}$,

$$\mathbf{Matrice\ des\ évaluations\ prédites} = \hat{Y}_{init},$$

tandis que la prédiction finale pour le Modèle avec Ajustements Supplémentaires est

$$\mathbf{Matrice\ des\ évaluations\ prédites} = \hat{Y}_{add} + \overline{Y_i},$$

où $\overline{Y_i}$ représente la matrice des évaluations moyennes des films 4.5.

5 Résultats

5.1 Métriques d'Évaluation

Les métriques d'évaluation pour la recommandation dans ce projet comprennent :

1. RMSE [4] : cet indicateur est particulièrement utile car il fournit une valeur unique pour indiquer à quel point les prédictions du modèle correspondent aux données observées. Plus le RMSE est faible, plus les prédictions sont proches des valeurs réelles, et mieux le modèle fonctionne. Cependant, le RMSE est sensible aux valeurs aberrantes en raison de l'élévation au carré des erreurs, ce qui signifie que les grandes erreurs affectent de manière disproportionnée le score final.

$$RMSE = \sqrt{\frac{1}{S} \sum_{(i,j) \in S} (\hat{y}(i,j) - y(i,j))^2} \tag{10}$$

Où :

- S est l'ensemble de toutes les paires utilisateur-élément (i, j) avec des évaluations réelles.

Nous pouvons également calculer le Coefficient de Variation du RMSE pour évaluer la variation des données par rapport à la moyenne des évaluations.

$$CV(RMSE) = \frac{\text{RMSE}}{\mu} \cdot 100 \tag{11}$$

Où :

- μ représente la moyenne de toutes les évaluations attribuées.

2. MAE [4] : cet indicateur fournit une interprétation intuitive de l'erreur du modèle, représentant la quantité moyenne par laquelle les prédictions diffèrent des valeurs réelles, dans la même unité que les données. Un MAE plus faible indique une meilleure précision prédictive. Comme le MAE ne met pas les erreurs au carré, les grandes déviations (valeurs aberrantes) n'ont pas autant d'impact que dans le RMSE. Cela rend le MAE plus robuste en présence de valeurs extrêmes. Le MAE est une métrique plus simple et plus robuste, qui se concentre sur la magnitude absolue des erreurs. Il traite toutes les déviations de manière égale.

$$MAE = \frac{1}{S} \sum_{(i,j) \in S} |\hat{y}(i,j) - y(i,j)| \tag{12}$$

Le Coefficient de Variation pour le MAE est également pris en compte pour mesurer la variation relative des erreurs absolues par rapport à la moyenne des valeurs observées, fournissant une mesure normalisée de la constance des erreurs de prédiction.

$$CV(MAE) = \frac{\text{MAE}}{\mu} \cdot 100 \tag{13}$$

3. Coefficient de Corrélation du Produit-Moment de Pearson [4,5] : cet indicateur est utilisé pour déterminer à quel point deux variables sont corrélées. Il est sensible aux valeurs aberrantes, car des valeurs extrêmes peuvent distordre considérablement la corrélation.

$$PCC = \frac{\sum_{(i,j) \in S} \left(\hat{y}(i,j) - \overline{\hat{y}}\right) \left(y(i,j) - \bar{y}\right)}{\sqrt{\sum_{(i,j) \in S} \left(\hat{y}(i,j) - \overline{\hat{y}}\right)^2} \sqrt{\sum_{(i,j) \in S} \left(y(i,j) - \bar{y}\right)^2}} \tag{14}$$

Où :

- $\overline{\hat{y}}$ est la moyenne de toutes les évaluations prédites $\hat{y}(i,j)$, correspondant aux évaluations réelles $y(i,j)$ non nulles.
- $\bar{y}$ est la moyenne de toutes les évaluations réelles $y(i,j)$.

5.2 RMSE et MAE

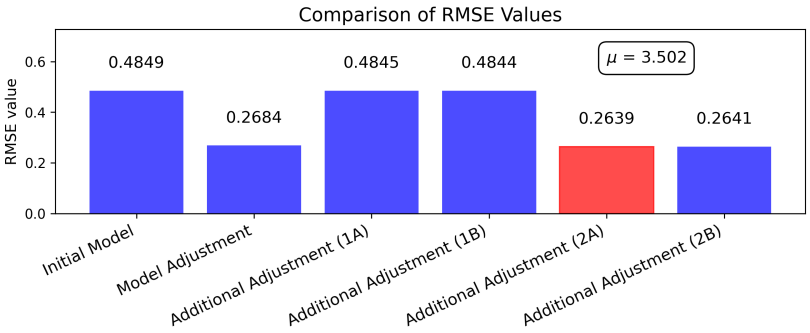

FIGURE 1 – Comparaison des valeurs de RMSE

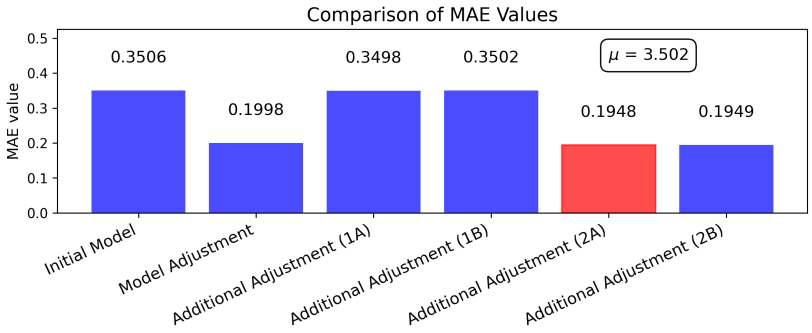

FIGURE 2 – Comparaison des valeurs de MAE

Évidemment, dans Figure 1 et Figure 2, trois modèles – le modèle principal ajusté et les deux scénarios du modèle d'ajustement supplémentaire dans l'Approche 2 – génèrent les valeurs les plus basses de RMSE et MAE. Ces trois modèles entraînent seulement trois paramètres, à savoir X, W et B, par rapport aux quatre paramètres utilisés dans les autres modèles. De plus, au sein de chaque groupe de modèles — ceux avec trois paramètres et ceux avec quatre paramètres — le modèle qui intègre le terme fixe μ (c'est-à-dire, le modèle principal ajusté dans le groupe à trois paramètres et le modèle initial dans le groupe à quatre paramètres) présente les RMSE et MAE les plus élevés de son groupe. Comme prévu, les modèles avec ajustements supplémentaires se sont avérés plus efficaces. Nous pouvons également conclure que l'Approche **2.B** du modèle d'ajustement supplémentaire offre la meilleure performance, avec moins de valeurs aberrantes dans ses prédictions et la plus faible amplitude d'erreur. Avec un MAE de seulement 0.2, les prédictions sont très proches des évaluations réelles.

Par exemple, si le modèle prédit une note de 4, la note réelle se situe généralement dans une fourchette d'environ 3.8 à 4.2. Cette petite marge d'erreur indique une grande précision, ce qui signifie que le modèle produit systématiquement des prédictions presque identiques aux valeurs réelles. De plus, en entraînant seulement trois paramètres, le modèle est plus simple et moins sujet à l'overfitting, ce qui contribue à ses bonnes performances. Dans l'ensemble, ces résultats suggèrent que la conception plus simple de ces modèles, en particulier l'Approche **2.B**, est très efficace pour capturer les motifs sous-jacents dans les données tout en minimisant les erreurs de prédiction.

De plus, il est important de noter que dans les Approches 1 et 2, il existe deux catégories de paramètres : les paramètres entièrement entraînés (Approches **1.B**, **2.B**) et les paramètres partiellement entraînés (Approches **1.A**, **2.A**). La catégorie des paramètres partiellement entraînés est moins efficace que celle des paramètres entièrement entraînés, bien que la différence de performance ne soit pas significative. Nous pouvons également conclure que les paramètres partiellement entraînés n'aident pas le modèle à converger plus rapidement que les paramètres entièrement entraînés, car ces derniers permettent au modèle de trouver la direction optimale dès le départ et de converger plus rapidement.

Loss of Approach 2.A in PCA space	**Loss of Approach 2.B in PCA space**
183192.3	17796.3

TABLE 1 – Comparison of Loss between Approach 2.A and 2.B (PCA)

Lorsque j'applique la PCA au modèle, le RMSE et le MAE augmentent comme le montrent les deux graphiques ci-dessus Figure 3 et Figure 4, ce qui était attendu. Cependant, nous pouvons observer que maintenant le modèle avec un ajustement supplémentaire dans l'Approche 2.A

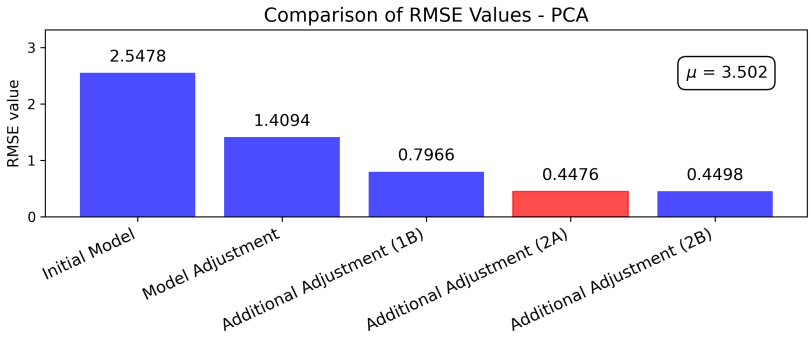

FIGURE 3 – Comparison of RMSE Values - PCA

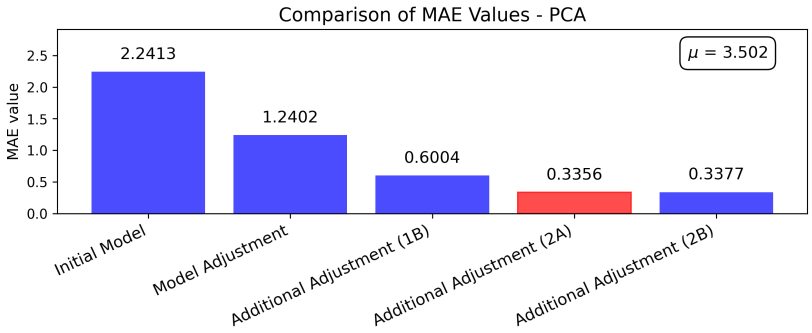

FIGURE 4 – Comparison of MAE Values - PCA

performe mieux que le précédent modèle ayant les meilleures performances, l'Approche 2.B, lorsque la PCA n'est pas utilisée. Il est à noter que le modèle avec ajustement supplémentaire dans l'Approche 2.A doit réduire la dimensionnalité des deux matrices Y_{norm} et B pour adapter le modèle, tandis que le Modèle 2.B réduit uniquement Y_{norm}, et la matrice de biais B est initialisée de manière aléatoire.

De plus, nous pouvons observer que le modèle initial et sa version ajustée ont des RMSE et MAE nettement plus élevés que les autres modèles. Une explication possible de cela est que ces deux modèles, sans PCA, incluent le terme fixe μ (la note moyenne de toutes les notes attribuées). Cependant, lors de l'application de la PCA, le terme fixe μ n'est plus explicitement pris en compte, car dans l'espace réduit, toutes les informations sont compressées, rendant floues les notes réelles attribuées. Cela entraîne une difficulté accrue dans l'optimisation du modèle.

Une autre chose que j'ai remarquée lors de l'application de la PCA aux deux modèles les plus performants (2.A et 2.B) : la valeur de la perte de l'Approche 2.A est significativement plus élevée que celle de l'Approche 2.B (Table 1), ce qui est considéré comme nettement moins précis. Cependant, lorsque je reconstruis l'espace dimensionnel d'origine, le RMSE et le MAE de l'Approche 2.A sont plus faibles que ceux de l'Approche 2.B, qui est considérée comme plus précise. Ce phénomène peut être expliqué comme suit : le Modèle 2.B apprend "trop bien" dans l'espace PCA car tous les paramètres sont initialisés de manière aléatoire. Ces paramètres s'ajustent bien avec la matrice de notation normalisée $\mathbf{New}Y_{\text{norm}}$ dans l'espace PCA, donc après reconstruction, avec la quantité d'informations revenant, le modèle dans l'espace dimensionnel d'origine ne s'ajuste pas aussi bien qu'il ne l'était. En revanche, le Modèle 2.A apprend "moins bien" la matrice de notation normalisée $\mathbf{New}Y_{\text{norm}}$ dans l'espace PCA, donc lorsqu'il revient dans l'espace d'origine, il peut performer bien mieux.

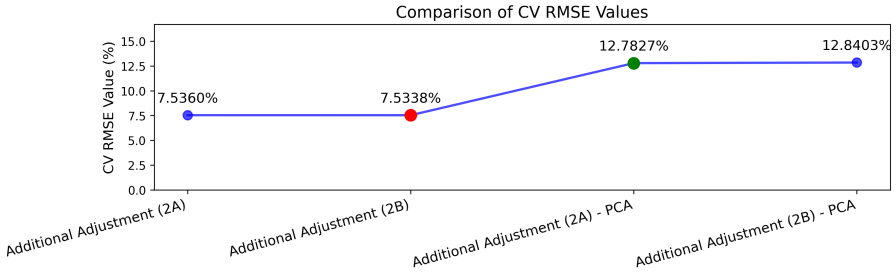

FIGURE 5 – Comparaison des valeurs de CV(RMSE)

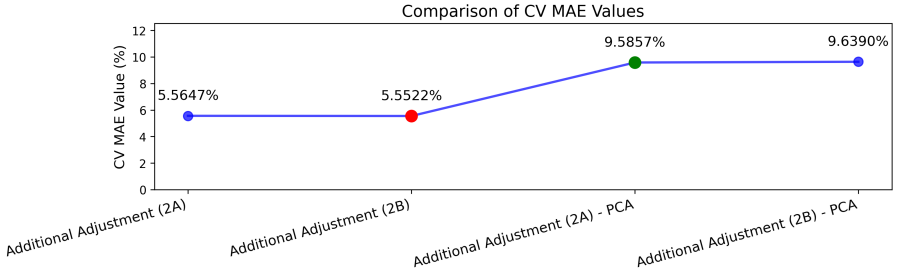

FIGURE 6 – Comparaison des valeurs de CV(MAE)

Selon Figure 5 et Figure 6, examinons deux autres indicateurs : CV(RMSE) et CV(MAE). Évidemment, on prévoyait que les erreurs des modèles sans PCA seraient plus petites. Cependant, il est à noter que les valeurs de CV(RMSE) et de CV(MAE) des modèles sans PCA sont inférieures de 5% par rapport aux valeurs des modèles avec PCA. Cette différence de 5% peut s'expliquer par le fait que, dans la première partie de ce projet, j'ai seulement sélectionné les vecteurs propres qui expliquent 95% de la variance totale pour construire la matrice de composantes principales.

Nous pouvons confirmer clairement que le modèle sans PCA présente des erreurs stables et moins fluctuantes.

5.3 Coefficient de corrélation du produit-moment de Pearson - PCC

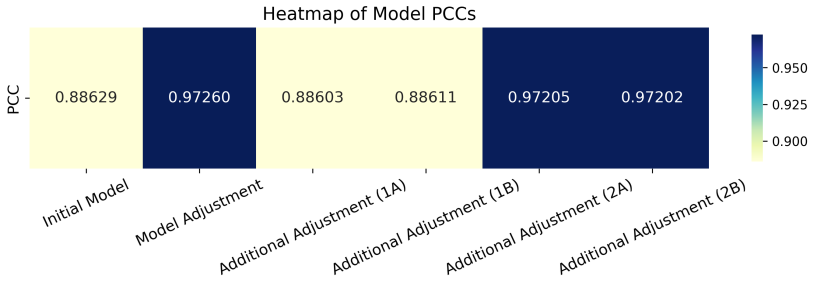

FIGURE 7 – Carte thermique des PCC des modèles

La carte thermique (Figure 7) montre que l'ajustement principal du modèle et l'ajustement supplémentaire dans l'Approche 2 obtiennent des valeurs de PCC très élevées (environ 0.973), ce qui signifie qu'ils ont une forte corrélation avec les notes réelles. En revanche, le modèle initial et les ajustements supplémentaires dans l'Approche 1 génèrent des valeurs de PCC plus faibles (environ 0.885), indiquant une performance moins efficace. Dans l'ensemble, cela suggère que l'Approche 2 est plus efficace pour améliorer la précision des prédictions.

L'Approche **2.B**, qui présente la meilleure corrélation avec les notes réelles, est compréhensible étant donné que ce modèle a également les meilleurs RMSE et MAE.

6 Discussion

1. Le problème de la division des données pour l'entraînement : le modèle apprend les caractéristiques latentes à partir des données d'interaction utilisateur-élément. Cependant, si certains utilisateurs ou éléments dans l'ensemble de test/sondage n'apparaissent pas dans l'ensemble d'entraînement, le modèle n'a pas de connaissances préalables (c'est-à-dire, de facteurs latents) à leur sujet. Ce manque d'informations préalables entraîne une mauvaise performance dans la prédiction des préférences, ce qui constitue une manifestation classique du problème de démarrage à froid. Par exemple, l'ensemble de données comprend 10 utilisateurs et 50 films. Le diviser en un ensemble d'entraînement (7 utilisateurs, 35 films) et un ensemble de test (3 utilisateurs, 15 films) pose des défis importants. Le modèle entraîné sur les paires utilisateur-élément dans l'ensemble d'entraînement ne peut pas prédire les interactions pour les paires utilisateur-élément dans l'ensemble de test en raison de l'absence d'exposition préalable.
De plus, une autre raison pour laquelle nous ne pouvons pas utiliser le modèle entraîné sur l'ensemble d'entraînement pour prédire l'ensemble de test est que la formule pour trouver la matrice approximative de Y : $\hat{Y} = X \cdot W + B$, où W, X, et B sont aléatoires pour minimiser la fonction de perte.

2. Un problème clé survient lors de l'application de la PCA : lorsque nous réduisons la dimensionnalité de la matrice de notations et ajustons le modèle dans l'espace de dimension inférieure, les notations réelles n'existent plus sous leur forme originale. Au lieu de cela, la matrice de notations de dimension réduite sert d'approximation de la matrice d'origine. Par conséquent, lors du calcul de la fonction de perte dans l'espace réduit, nous ajustons involontairement le modèle en utilisant toutes les notations, y compris celles qui n'ont pas été attribuées. Après la reconstruction de l'espace original, cela peut entraîner une augmentation des valeurs de RMSE et de MAE.
Ainsi, l'utilisation de la PCA pour réduire la dimensionnalité des matrices dans ce cas n'est pas idéale, car elle entraîne la perte d'informations concernant les notations réelles dans l'espace PCA. Une approche plus favorable consiste à appliquer directement des modèles de factorisation de matrice, en veillant à ce que la fonction de perte soit calculée uniquement sur les notations observées. Selon Koren, Bell, & Volinsky (2009) [2], les méthodes de factorisation de matrice dans les systèmes de recommandation sont largement utilisées et efficaces en pratique, car elles optimisent explicitement le modèle uniquement en fonction des notations réelles.

7 Conclusion

En conclusion, ce rapport présente un projet personnel visant à explorer comment les systèmes de recommandation fonctionnent avec la factorisation de matrice. Ce n'est pas une étude de recherche formelle, mais plutôt un ensemble d'observations et d'ajustements apportés aux modèles existants dans le but d'améliorer leurs performances.

Parmi tous les modèles évalués dans ce projet, l'Approche 2.B a obtenu le plus faible RMSE et MAE, ainsi que le plus haut PCC, ce qui indique que les notations prédites sont fortement corrélées avec les notations réelles. Cependant, son amélioration par rapport à l'Approche 2.A est seulement marginale en termes de métriques d'évaluation.

De plus, j'ai tenté d'appliquer la PCA pour optimiser la fonction de perte dans un espace de dimension réduite afin d'accélérer la convergence, puis de la reconstruire dans l'espace original.

Cependant, la perte d'informations concernant les notations réelles lors de la réduction de la dimensionnalité a rendu difficile la convergence correcte du modèle et a conduit à une optimisation dans une direction incorrecte.

Il est donc préférable d'appliquer directement la factorisation de matrice dans la dimension d'origine, où les notations réelles sont explicitement préservées. Cependant, à mesure que l'ensemble de données devient plus grand et plus sparse, le coût computationnel, le temps et la qualité des prédictions deviennent des défis importants. Certaines méthodes de factorisation de matrice directe, telles que la SVD, peuvent être appliquées. Alternativement, des techniques de réduction de dimension telles que t-SNE ou les autoencodeurs peuvent aider à atténuer ces contraintes computationnelles tout en préservant les informations essentielles.

Références

[1] J Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. Collaborative filtering recommender systems. In *The adaptive web : methods and strategies of web personalization*, pages 291–324. Springer, 2007.

[2] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8) :30–37, 2009.

[3] L. Lü, M. Medo, et al. Recommender systems. *Physics Reports*, 519(1) :20–21, 2012.

[4] L. Lü, M. Medo, et al. Recommender systems. *Physics Reports*, 519(1) :10–11, 2012.

[5] J.L. Rodgers and W.A. Nicewander. Thirteen ways to look at the correlation coefficient. *The American Statistician*, 42 :59–66, 1988.