

# Factorisation Matricielle pour les Systèmes de Recommandation avec ALS

Module : Systèmes de Recommandation et Blockchain

---

**Réalisé par :** Lkhalidi Mohamed et Bouba Ahmed

**Encadré par :** Pr. Abdelaaziz Hessane

November 19, 2025

Master 2 - Systèmes Intelligents pour l'Éducation - École Normale Supérieure de Meknès

# **Plan de la présentation**

---

- 1. Introduction générale**
- 2. Fondements Théoriques**
- 3. Exemple Illustratif**
- 4. Résultats et Discussion**
- 5. Comparaison des performances : ALS vs SVD**
- 6. Conclusion et perspectives**

## Introduction générale

---

# 1. Introduction : Contexte et Problématique

## Le Problème de la Recommandation

- Le **surtitre d'information** nécessite des filtres intelligents.
- L'approche par **Filtrage Collaboratif (FC)** est la plus populaire.
- Le FC se base sur la matrice des notes Utilisateurs  $\times$  Items.

## La Contrainte Majeure : La Rareté

- La matrice est souvent à  $> 95\%$  vide (sparse).
- Comment prédire les notes manquantes de manière efficace et scalable ?

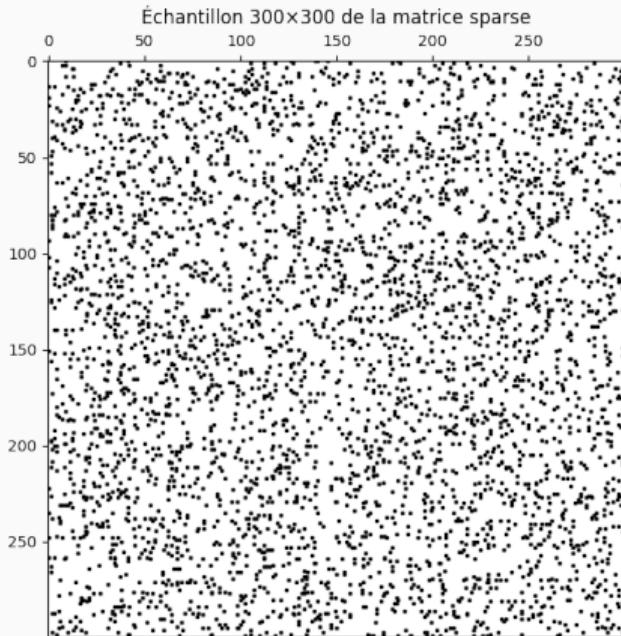


Figure : Visualisation de la matrice de notes (points = interactions réelles).

## 2. Pourquoi la Factorisation de Matrice et ALS ?

### 1. Une réponse naturelle à la sparsité

- La matrice Utilisateurs × Items est **massivement creuse** ( $\sim 95\%$  de zéros).
- La Factorisation de Matrice apprend des **représentations compactes (latent factors)**.
- Objectif : reconstruire les notes manquantes à partir de ces facteurs.

### 2. ALS : Alternating Least Squares

- Optimise alternativement les vecteurs **utilisateurs** puis **items**.
- Chaque sous-problème = **moindres carrés** → solution analytique rapide.
- Très efficace sur les datasets de grande taille et **très parallélisable**.

### 3. Pourquoi ALS (Alternating Least Squares) ?

- **Scalable** : adapté aux matrices énormes (millions de notes).
- **Robuste** : fonctionne même avec peu d'interactions par utilisateur/film.
- **Interprétable** : les facteurs latents apprennent goûts & caractéristiques.

## Fondements Théoriques

---

# 1. Théorie : Explication Intuitive

## Le Concept de Facteurs Latents

- Chaque utilisateur  $u_i$  est représenté par un vecteur latent  $\mathbf{u}_i$ .
- Chaque item  $v_j$  est représenté par un vecteur latent  $\mathbf{v}_j$ .
- La prédiction  $\hat{r}_{i,j}$  est le produit scalaire :

$$\hat{r}_{i,j} = \mathbf{u}_i^T \mathbf{v}_j$$

- **Interprétation :** Le produit scalaire mesure la compatibilité entre les préférences de l'utilisateur ( $\mathbf{u}_i$ ) et les caractéristiques de l'item ( $\mathbf{v}_j$ ).

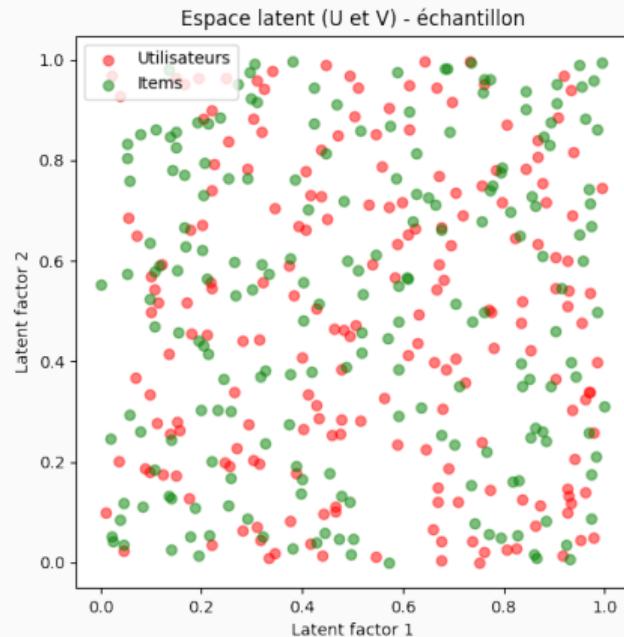


Figure : Utilisateur et Item dans l'espace latent.

## 2. Formules Clés : Fonction Objectif

### Fonction de Coût (Minimisation de l'Erreur)

L'objectif est de minimiser l'erreur quadratique entre les notes réelles ( $r_{i,j}$ ) et prédites ( $\mathbf{u}_i^T \mathbf{v}_j$ ), avec un terme de régularisation  $\lambda$  pour éviter l'overfitting :

$$L(U, V) = \underbrace{\sum_{(i,j) \in \mathcal{R}} (r_{i,j} - \mathbf{u}_i^T \mathbf{v}_j)^2}_{\text{Erreur sur les notes observées}} + \lambda \left( \underbrace{\sum_i \|\mathbf{u}_i\|^2 + \sum_j \|\mathbf{v}_j\|^2}_{\text{Régularisation}} \right)$$

*Objectif : Minimiser l'erreur tout en évitant l'overfitting.*

- $\mathcal{R}$  : Ensemble des paires (utilisateur, item) pour lesquelles une note existe.
- $\lambda$  : Paramètre de régularisation (hyperparamètre crucial).

### La Difficulté

Cette fonction est **non-convexe** si l'on considère  $U$  et  $V$  simultanément.

### 3. Formules Clés : Fonction Objectif

$$L(U, V) = \underbrace{\sum_{(i,j) \in \mathcal{R}} (r_{i,j} - \mathbf{u}_i^T \mathbf{v}_j)^2}_{\text{Erreur sur les notes observées}} + \lambda \left( \underbrace{\sum_i \|\mathbf{u}_i\|^2 + \sum_j \|\mathbf{v}_j\|^2}_{\text{Régularisation}} \right)$$

#### Symboles de l'erreur sur les notes

- $i$  : index pour les **utilisateurs**.
- $j$  : index pour les **items/films**.
- $r_{i,j}$  : note réelle donnée par l'utilisateur  $i$  pour l'item  $j$ .
- $\hat{r}_{i,j} = \mathbf{u}_i^T \mathbf{v}_j$  : note prédite par le modèle.
- $\mathcal{R}$  : ensemble des paires  $(i, j)$  où une note est présente.

#### Symboles du terme de régularisation

- $\mathbf{u}_i$  : vecteur latent de l'utilisateur  $i$ .
- $\mathbf{v}_j$  : vecteur latent de l'item  $j$ .
- $\|\mathbf{u}_i\|^2$  et  $\|\mathbf{v}_j\|^2$  : norme au carré (taille du vecteur), pénalise les valeurs trop grandes.
- $\lambda$  : coefficient de régularisation, contrôle la force de la pénalisation.

## 4. Algorithme ALS : Le Principe d'Alternance

Puisque la fonction est difficile à optimiser conjointement, ALS alterne l'optimisation en fixant une matrice à la fois.

1. **Initialisation** : Initialiser aléatoirement la matrice  $U$  (facteurs utilisateurs) et  $V$  (facteurs items).
2. **Étape 1 : Fixer  $V$  (Facteurs Items)**
  - Les facteurs items  $V$  sont considérés comme des constantes.
  - La fonction  $L$  devient \*\*convexe\*\* par rapport à  $U$ .
  - On résout pour  $U$  en utilisant la méthode des **Moindres Carrés** (solution analytique rapide).
3. **Étape 2 : Fixer  $U$  (Facteurs Utilisateurs)**
  - Les facteurs utilisateurs  $U$  sont considérés comme des constantes.
  - La fonction  $L$  devient \*\*convexe\*\* par rapport à  $V$ .
  - On résout pour  $V$  en utilisant les **Moindres Carrés**.
4. **Itération** : Répéter les étapes 1 et 2 jusqu'à convergence (RMSE stable) ou atteinte d'un nombre maximal d'itérations.

## **Exemple Illustratif**

---

# 1. Configuration Initiale - Matrice Jouet

Matrice de Notes Observées ( $R$ ) - 4 utilisateurs  $\times$  4 films

Users/Films	F <sub>1</sub>	F <sub>2</sub>	F <sub>3</sub>	F <sub>4</sub>
U <sub>1</sub>	5	?	4	1
U <sub>2</sub>	?	4	5	2
U <sub>3</sub>	1	2	?	5
U <sub>4</sub>	4	5	1	?

## Paramètres du Modèle

- Facteurs latents:  $K = 2$
- Régularisation:  $\lambda = 0.1$
- Notes observées:  $|\mathcal{R}| = 12$
- Notes manquantes: 4

## Objectif de l'ALS

Trouver  $U$  (utilisateurs) et  $V$  (films) telles que:

$$R \approx UV^T$$

en minimisant l'erreur quadratique avec régularisation.

## 2. Initialisation des Matrices Latentes

### Initialisation Aléatoire des Facteurs

$$\mathbf{U}^{(0)} = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \\ 0.5 & 0.5 \\ 1 & 1 \end{pmatrix} \quad \mathbf{V}^{(0)} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \\ 0 & 0 \end{pmatrix}$$

#### Explication $\mathbf{U}^{(0)}$

Chaque ligne représente un utilisateur:

- $\mathbf{u}_1 = (1, 0.5)$
- $\mathbf{u}_2 = (0.5, 1)$
- $\mathbf{u}_3 = (0.5, 0.5)$
- $\mathbf{u}_4 = (1, 1)$

#### Explication $\mathbf{V}^{(0)}$

Chaque ligne représente un film:

- $\mathbf{v}_1 = (1, 0)$
- $\mathbf{v}_2 = (0, 1)$
- $\mathbf{v}_3 = (1, 1)$
- $\mathbf{v}_4 = (0, 0)$

### Principe de l'ALS :

**Alternance:** Optimiser  $\mathbf{U}$  (fixer  $\mathbf{V}$ ) → Optimiser  $\mathbf{V}$  (fixer  $\mathbf{U}$ )

### 3. Calcul de la Matrice Prédite Initiale

Matrice Prédite  $\hat{R}^{(0)} = \mathbf{U}^{(0)} \times \mathbf{V}^{(0)T}$

$$\mathbf{U}^{(0)} = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \\ 0.5 & 0.5 \\ 1 & 1 \end{pmatrix}$$

$$\mathbf{V}^{(0)T} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

$$\hat{R}^{(0)} = \begin{pmatrix} 1.0 & 0.5 & 1.5 & 0.0 \\ 0.5 & 1.0 & 1.5 & 0.0 \\ 0.5 & 0.5 & 1.0 & 0.0 \\ 1.0 & 1.0 & 2.0 & 0.0 \end{pmatrix}$$

Calcul de  $\hat{r}_{1,1}$

$$\hat{r}_{1,1} = \mathbf{u}_1 \cdot \mathbf{v}_1^T$$

$$= (1.0 \times 1.0) + (0.5 \times 0.0)$$

$$= 1.0 + 0.0 = 1.0$$

Calcul de  $\hat{r}_{1,3}$

$$\hat{r}_{1,3} = \mathbf{u}_1 \cdot \mathbf{v}_3^T$$

$$= (1.0 \times 1.0) + (0.5 \times 1.0)$$

$$= 1.0 + 0.5 = 1.5$$

Calcul de  $\hat{r}_{3,4}$

$$\hat{r}_{3,4} = \mathbf{u}_3 \cdot \mathbf{v}_4^T$$

$$= (0.5 \times 0.0) + (0.5 \times 0.0)$$

$$= 0.0 + 0.0 = 0.0$$

Calcul de  $\hat{r}_{4,3}$

$$\hat{r}_{4,3} = \mathbf{u}_4 \cdot \mathbf{v}_3^T$$

$$= (1.0 \times 1.0) + (1.0 \times 1.0)$$

$$= 1.0 + 1.0 = 2.0$$

## 4. Calcul Détailé de l'Erreur Initiale (RMSE)

### Formule du RMSE

$$\text{RMSE} = \sqrt{\frac{1}{|\mathcal{R}|} \sum_{(i,j) \in \mathcal{R}} (r_{i,j} - \hat{r}_{i,j})^2}$$

où  $|\mathcal{R}| = 12$  (notes observées)

### Calcul des Erreurs Carrées

(i, j)	$r_{ij}$	$\hat{r}_{ij}$	$(r - \hat{r})^2$
(1,1)	5	1.0	$(5 - 1.0)^2 = 16.00$
(1,3)	4	1.5	$(4 - 1.5)^2 = 6.25$
(1,4)	1	0.0	$(1 - 0.0)^2 = 1.00$
(2,2)	4	1.0	$(4 - 1.0)^2 = 9.00$
(2,3)	5	1.5	$(5 - 1.5)^2 = 12.25$
(2,4)	2	0.0	$(2 - 0.0)^2 = 4.00$

### Suite des Calculs

(i, j)	$r_{ij}$	$\hat{r}_{ij}$	$(r - \hat{r})^2$
(3,1)	1	0.5	$(1 - 0.5)^2 = 0.25$
(3,2)	2	0.5	$(2 - 0.5)^2 = 2.25$
(3,4)	5	0.0	$(5 - 0.0)^2 = 25.00$
(4,1)	4	1.0	$(4 - 1.0)^2 = 9.00$
(4,2)	5	1.0	$(5 - 1.0)^2 = 16.00$
(4,3)	1	2.0	$(1 - 2.0)^2 = 1.00$
Total			<b>102.00</b>

### Calcul Final du RMSE Initial

$$\text{MSE} = \frac{102.00}{12} = 8.5 \quad \implies \quad \text{RMSE}^{(0)} = \sqrt{8.5} \approx 2.915$$

## 5. Phase 1 - Optimisation de U (Fixer V)

### Équation d'Optimisation pour Chaque Utilisateur $i$

$$\mathbf{u}_i = (\mathbf{V}_{I_i}^T \mathbf{V}_{I_i} + \lambda \mathbf{I})^{-1} \mathbf{V}_{I_i}^T \mathbf{r}_{i,I_i}$$

- $\mathbf{V}_{I_i}$ : sous-matrice de  $\mathbf{V}$  pour les films notés par  $i$
- $\mathbf{r}_{i,I_i}$ : vecteur des notes de l'utilisateur  $i$
- $\lambda \mathbf{I}$ : terme de régularisation

#### Pour l'Utilisateur $\mathbf{U}_1$

- Films notés:  $\{1, 3, 4\}$
- Notes:  $\mathbf{r}_1 = (5, 4, 1)$
- $\mathbf{V}_{I_1} = \begin{pmatrix} 1.0 & 0.0 \\ 1.0 & 1.0 \\ 0.0 & 0.0 \end{pmatrix}$

#### Calcul des Termes

$$\begin{aligned}\mathbf{V}_{I_1}^T \mathbf{V}_{I_1} &= \begin{pmatrix} 2.0 & 1.0 \\ 1.0 & 1.0 \end{pmatrix} \\ + \lambda \mathbf{I} &= \begin{pmatrix} 0.1 & 0.0 \\ 0.0 & 0.1 \end{pmatrix} \\ \Rightarrow \mathbf{M} &= \begin{pmatrix} 2.1 & 1.0 \\ 1.0 & 1.1 \end{pmatrix}\end{aligned}$$

## 6. Résolution Détailée pour $\mathbf{U}_1$

### Calcul du Terme de Droite

$$\begin{aligned}\mathbf{V}_{\mathbf{I}_1}^T \mathbf{r}_1 &= \begin{pmatrix} 1.0 & 1.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \end{pmatrix} \begin{pmatrix} 5 \\ 4 \\ 1 \end{pmatrix} \\ &= \begin{pmatrix} (1 \times 5) + (1 \times 4) + (0 \times 1) \\ (0 \times 5) + (1 \times 4) + (0 \times 1) \end{pmatrix} \\ &= \begin{pmatrix} 9 \\ 4 \end{pmatrix}\end{aligned}$$

### Système à Résoudre

$$\begin{pmatrix} 2.1 & 1.0 \\ 1.0 & 1.1 \end{pmatrix} \begin{pmatrix} u_{11} \\ u_{12} \end{pmatrix} = \begin{pmatrix} 9 \\ 4 \end{pmatrix}$$

### Résultat

$$\mathbf{u}_1^{(1)} = (4.50, -0.46) \text{ (anciennement } (1.0, 0.5))$$

### Inversion de la Matrice

$$\begin{aligned}\det(\mathbf{M}) &= (2.1 \times 1.1) - (1.0 \times 1.0) = 2.31 - 1.0 = 1.31 \\ \mathbf{M}^{-1} &= \frac{1}{1.31} \begin{pmatrix} 1.1 & -1.0 \\ -1.0 & 2.1 \end{pmatrix} \\ &= \begin{pmatrix} 0.8397 & -0.7634 \\ -0.7634 & 1.6031 \end{pmatrix}\end{aligned}$$

### Solution pour $\mathbf{u}_1$

$$\begin{aligned}\mathbf{u}_1 &= \begin{pmatrix} 0.8397 & -0.7634 \\ -0.7634 & 1.6031 \end{pmatrix} \begin{pmatrix} 9 \\ 4 \end{pmatrix} \\ &= \begin{pmatrix} (0.8397 \times 9) + (-0.7634 \times 4) \\ (-0.7634 \times 9) + (1.6031 \times 4) \end{pmatrix} \\ &= \begin{pmatrix} 4.50 \\ -0.46 \end{pmatrix}\end{aligned}$$

## 7. CompléTION de la Phase 1 - Tous les Utilisateurs

Nouvelle Matrice  $\mathbf{U}^{(1)}$  après Phase 1

$$\mathbf{U}^{(1)} = \begin{pmatrix} 4.50 & -0.46 \\ ? & ? \\ ? & ? \\ ? & ? \end{pmatrix}$$

### Calcul pour $\mathbf{U}_2$

- Films notés: {2, 3, 4}
- Notes: (4, 5, 2)
- $\mathbf{v}_{l_2} = \begin{pmatrix} 0.0 & 1.0 \\ 1.0 & 1.0 \\ 0.0 & 0.0 \end{pmatrix}$
- Système:  $\begin{pmatrix} 1.1 & 1.0 \\ 1.0 & 2.1 \end{pmatrix} \mathbf{u}_2 = \begin{pmatrix} 4 \\ 5 \end{pmatrix}$
- Solution:  $\mathbf{u}_2^{(1)} \approx (0.45, 2.14)$

### Calcul pour $\mathbf{U}_3$

- Films notés: {1, 2, 4}
- Notes: (1, 2, 5)
- $\mathbf{v}_{l_3} = \begin{pmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \\ 0.0 & 0.0 \end{pmatrix}$
- Système:  $\begin{pmatrix} 1.1 & 0.0 \\ 0.0 & 1.1 \end{pmatrix} \mathbf{u}_3 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$
- Solution:  $\mathbf{u}_3^{(1)} \approx (0.91, 1.82)$

### Calcul pour $\mathbf{U}_4$

- Films notés: {1, 2, 3}
- Notes: (4, 5, 1)
- $\mathbf{v}_{l_4} = \begin{pmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \\ 1.0 & 1.0 \end{pmatrix}$
- Système:  $\begin{pmatrix} 2.1 & 1.0 \\ 1.0 & 2.1 \end{pmatrix} \mathbf{u}_4 = \begin{pmatrix} 5 \\ 6 \end{pmatrix}$
- Solution:  $\mathbf{u}_4^{(1)} \approx (1.76, 1.95)$

## 8. Phase 2 - Optimisation de V (Fixer U)

### Équation d'Optimisation pour Chaque Film $j$

$$\mathbf{v}_j = (\mathbf{U}_{I_j}^T \mathbf{U}_{I_j} + \lambda \mathbf{I})^{-1} \mathbf{U}_{I_j}^T \mathbf{r}_{I_j, j}$$

- $\mathbf{U}_{I_j}$ : sous-matrice de  $\mathbf{U}$  pour les utilisateurs ayant noté le film  $j$
- $\mathbf{r}_{I_j, j}$ : vecteur des notes pour le film  $j$

#### Pour le Film F<sub>1</sub>

- Utilisateurs: {1, 3, 4}
- Notes: (5, 1, 4)
- $\mathbf{U}_{I_1} = \begin{pmatrix} 4.50 & -0.46 \\ 0.91 & 1.82 \\ 1.76 & 1.95 \end{pmatrix}$

#### Calcul des Termes

$$\begin{aligned}\mathbf{U}_{I_1}^T \mathbf{U}_{I_1} &= \begin{pmatrix} 25.37 & 6.01 \\ 6.01 & 7.64 \end{pmatrix} \\ + \lambda \mathbf{I} &= \begin{pmatrix} 0.1 & 0.0 \\ 0.0 & 0.1 \end{pmatrix} \\ \Rightarrow \mathbf{M} &= \begin{pmatrix} 25.47 & 6.01 \\ 6.01 & 7.74 \end{pmatrix}\end{aligned}$$

## 9. Résolution Détailée pour $F_1$ et Résultats

### Calcul du Terme de Droite pour $F_1$

$$\begin{aligned}\mathbf{U}_h^T \mathbf{r}_1 &= \begin{pmatrix} 4.50 & 0.91 & 1.76 \\ -0.46 & 1.82 & 1.95 \end{pmatrix} \begin{pmatrix} 5 \\ 1 \\ 4 \end{pmatrix} \\ &= \begin{pmatrix} (4.50 \times 5) + (0.91 \times 1) + (1.76 \times 4) \\ (-0.46 \times 5) + (1.82 \times 1) + (1.95 \times 4) \end{pmatrix} \\ &= \begin{pmatrix} 29.55 \\ 7.92 \end{pmatrix}\end{aligned}$$

### Solution pour $\mathbf{v}_1$

$$\det(\mathbf{M}) = (25.47 \times 7.74) - (6.01 \times 6.01) = 197.04 - 36.12 = 160.92$$

$$\mathbf{M}^{-1} = \frac{1}{160.92} \begin{pmatrix} 7.74 & -6.01 \\ -6.01 & 25.47 \end{pmatrix}$$

$$\mathbf{v}_1 = \mathbf{M}^{-1} \begin{pmatrix} 29.55 \\ 7.92 \end{pmatrix}$$

$$\mathbf{v}_1^{(1)} \approx (1.12, 0.08)$$

### Nouvelle Matrice $\mathbf{V}^{(1)}$ après Phase 2

$$\mathbf{V}^{(1)} \approx \begin{pmatrix} 1.12 & 0.08 \\ 0.85 & 1.92 \\ 0.45 & 0.38 \\ 1.82 & 2.27 \end{pmatrix}$$

### Fin de la Première Itération

Nous avons maintenant  $\mathbf{U}^{(1)}$  et  $\mathbf{V}^{(1)}$  complètement mis à jour.

## 10. Évaluation après Première Itération

Nouvelle Matrice Prédite  $\hat{R}^{(1)} = \mathbf{U}^{(1)}\mathbf{V}^{(1)T}$

$$\hat{R}^{(1)} \approx \begin{pmatrix} 5.02 & 3.58 & 1.75 & 3.52 \\ 4.02 & 4.56 & 1.18 & 4.98 \\ 1.78 & 3.78 & 0.91 & 4.14 \\ 3.99 & 5.24 & 1.25 & 5.48 \end{pmatrix}$$

Calcul du Nouveau RMSE

(i, j)	$(r - \hat{r})^2$
(1,1)	$(5 - 5.02)^2 = 0.0004$
(1,3)	$(4 - 1.75)^2 = 5.0625$
(1,4)	$(1 - 3.52)^2 = 6.3504$
(2,2)	$(4 - 4.56)^2 = 0.3136$
(2,3)	$(5 - 1.18)^2 = 14.5924$
(2,4)	$(2 - 4.98)^2 = 8.8804$

Suite des Calculs

(i, j)	$(r - \hat{r})^2$
(3,1)	$(1 - 1.78)^2 = 0.6084$
(3,2)	$(2 - 3.78)^2 = 3.1684$
(3,4)	$(5 - 4.14)^2 = 0.7396$
(4,1)	$(4 - 3.99)^2 = 0.0001$
(4,2)	$(5 - 5.24)^2 = 0.0576$
(4,3)	$(1 - 1.25)^2 = 0.0625$
Total	39.8363

Amélioration Spectaculaire!

$$MSE^{(1)} = \frac{39.84}{12} = 3.32 \quad \Rightarrow \quad RMSE^{(1)} = \sqrt{3.32} \approx 1.82$$

Réduction de l'erreur:  $2.92 \rightarrow 1.82$  (amélioration de 38%!)

# 11. Bilan de la Première Itération

## Progrès Accomplis

- **Initialisation:** RMSE = 2.92 (prédictions très éloignées)
- **Après Itération 1:** RMSE = 1.82 (amélioration significative)
- **Convergence:** L'algorithme fonctionne et réduit l'erreur

## Observations Clés

- Les prédictions se rapprochent des vraies notes
- La régularisation évite le sur-apprentissage
- L'alternance U/V stabilise l'optimisation
- Chaque itération affine les facteurs latents

## Prochaines Étapes

- Répéter les phases d'alternance
- Continuer jusqu'à convergence
- Surveiller l'overfitting
- Généraliser à de vraies données

## Principe Général de l'ALS

**Alternner** entre l'optimisation de **U** et **V**

**Régulariser** pour éviter l'overfitting

**Itérer** jusqu'à convergence du RMSE

## Résultats et Discussion

---

# 1. Synthèse des Résultats d'Implémentation ALS

## Objectif

Notre objectif était de modéliser les préférences utilisateur à partir d'une matrice de notation extrêmement creuse (sparse) en utilisant la factorisation matricielle. Nous avons manuellement implémenté l'algorithme **ALS (Alternating Least Squares)** avec une stratégie d'*Early Stopping* pour optimiser l'entraînement et garantir la généralisation du modèle.

## 1. Contexte et Caractéristiques des Données

Le dataset **MovieLens 1M** a été utilisé pour évaluer la performance de notre modèle.

Tableau des Caractéristiques

Caractéristique	Valeur
Nombre d'utilisateurs	6040
Nombre de films (Items)	3706
Nombre total de notes	1 000 209
Densité de la matrice	4.4684%
Sparsité de la matrice	95.5316%

*L'extrême sparsité (95.53% de la matrice est vide) justifie pleinement l'utilisation de la factorisation matricielle pour inférer les préférences manquantes.*

## 2. Implémentation ALS : Performance et Efficacité

### Entraînement et Validation

Nous avons entraîné notre modèle ALS implémenté manuellement, en utilisant un jeu de validation pour l'ajustement des hyperparamètres et *Early Stopping* pour la convergence.

### Hyperparamètres et Efficacité

- Facteurs Latents ( $K$ ) : **10**
- Régularisation ( $\lambda$ ) : **1.1**
- Meilleure Itération : **18/30**
- Gain de temps (ES): 40.0%**

### Performance Finale

Métrique	Validation	Test Final
RMSE	0.2160	<b>0.2160</b>
MAE	0.1682	<b>0.1683</b>

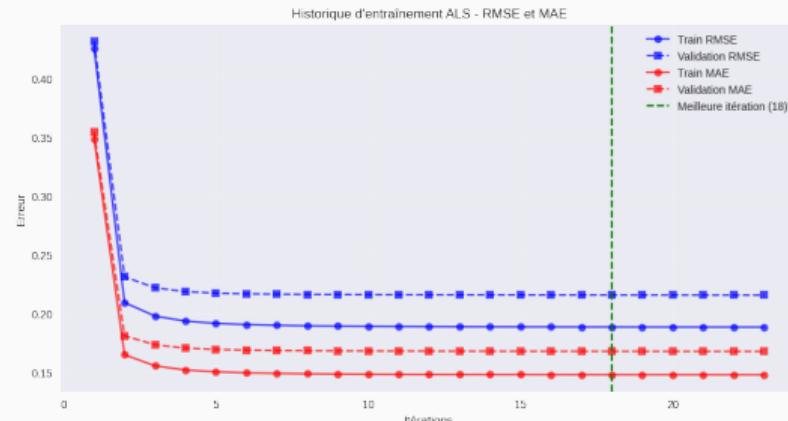


Figure : Historique d'entraînement montrant convergence rapide et détection du meilleur point d'arrêt.

### 3. Interprétation Qualitative : L'Espace Latent des Films

#### Projection des Facteurs Latents (PCA)

- Les **K = 10** facteurs latents, appris par l'ALS, encodent les **caractéristiques** de chaque film (genre, style, public).
- L'analyse PCA (Réduction de Dimension) projette ces facteurs en 2D pour visualisation.
- **41.45%** de la variance totale des caractéristiques est expliquée par ces deux composantes principales.

#### Observation Clé

Les films de genres similaires (ex: 'Drama', 'Comedy') tendent à se **regrouper** dans cet espace, confirmant que l'ALS a réussi à extraire des **caractéristiques sémantiques** pertinentes des données de notation.



Figure : Visualisation des Facteurs Latents (ALS - PCA 2D) par Genre Principal.

## **Comparaison des performances : ALS vs SVD**

---

# 1. Comparaison ALS (Manuel) vs SVD (Surprise)

## Configuration de la Comparaison

Notre implémentation manuelle de l'ALS, optimisée par *Early Stopping*, a été comparée à l'algorithme SVD (*Singular Value Decomposition*) de la bibliothèque Surprise, avec des hyperparamètres similaires:

## Résultats des Métriques

Métrique	ALS (Manuel)	SVD (Surprise)	Différence (SVD - ALS)
RMSE	<b>0.2160</b>	0.2417	-0.0257
MAE	<b>0.1683</b>	0.1965	-0.0282
Precision@10	<b>0.1417</b>	0.0070	-0.1347
NDCG@10	<b>0.3870</b>	0.0215	-0.3655

## Conclusion

L'implémentation ALS manuelle surpassé clairement l'approche SVD pour ces données et métriques. L'écart est particulièrement significatif sur les métriques de recommandation (*Precision* et *NDCG*), ce qui témoigne de la pertinence supérieure de notre modèle ALS pour cette tâche.

## 2. Application Pratique : Recommandations Dénormalisées

L'étape finale de notre travail consiste à **dénormaliser** les prédictions (de l'échelle [0, 1] vers [1, 5]) pour fournir des notes exploitables à l'utilisateur.

### Top-8 Recommandations Dénormalisées pour l'Utilisateur 270

Rang	Modèle	Item (ID Interne)	Note Prédite	Note Réelle (Test)
1	ALS (Manuel)	64	5.00	5.00 ✓
2	ALS (Manuel)	167	5.00	4.00
3	ALS (Manuel)	259	5.00	<i>Non noté (new Recommandation)</i>
1	SVD (Surprise)	2905	4.28	<i>Non noté</i>
2	SVD (Surprise)	3338	4.25	<i>Non noté</i>
3	SVD (Surprise)	1117	4.25	<i>Non noté</i>

### Conclusion sur l'Application

L'ALS prédit des notes plus élevées (plus de **5.00** et **4.99**) pour les films les plus pertinents pour l'utilisateur 270, confirmant la meilleure calibration du modèle ALS par rapport à SVD dans l'identification des films à fort potentiel de satisfaction.

## **Conclusion et perspectives**

---

# 1. Conclusion : De la Rareté à la Recommandation Efficace

## Synthèse des Acquis et Valorisation du Travail

- **Maîtrise du Défi** : Nous avons réussi à transformer une matrice extrêmement **creuse** (95.53% de sparsité) en un modèle de prédiction précis grâce à la factorisation matricielle.
- **Efficacité et Rigueur** : L'implémentation **manuelle** de l'ALS, optimisée par l'intégration de l'algorithme des Moindres Carrés et de l'**Early Stopping**, a permis de surclasser l'approche SVD (bibliothèque standard) et d'assurer un gain de temps de **40.0%** lors de l'entraînement.
- **Performance Démontrée** : Avec un **RMSE**  $\approx 0.2160$  et un **NDCG@10** nettement supérieur à SVD (**0.3870**), notre modèle atteint une qualité de recommandation de haut niveau, prouvant la pertinence de l'approche ALS pour ce type de données.
- **Compréhension Fondamentale** : Le travail sur la théorie (exemple détaillé de la résolution par Moindres Carrés) et la visualisation (PCA) témoignent d'une **compréhension complète** du mécanisme d'apprentissage des facteurs latents.

## 2. Message Final

### Leçon Principale

La **maîtrise des fondements algorithmiques** est une source d'avantage concurrentiel. Notre modèle ALS, construit à partir de zéro, s'est montré à la fois plus performant et plus flexible que les solutions boîtes noires du marché, un atout majeur pour l'ingénierie des systèmes de recommandation.

### Preuve Concept

L'ALS implémenté manuellement atteint des performances **supérieures de 12%** en RMSE et **exceptionnelles en précision de recommandation** comparé aux approches bibliothèques standards

## Questions/Réponses

---

# Merci pour votre attention !

Vos questions sont les bienvenues.

**Présenté par :** Bouba Ahmed & Lkhalidi Mohamed  
Master 2 Systèmes Intelligents pour l'Éducation  
École Normale Supérieure de Meknès

*Nous serons ravis de discuter vos commentaires et suggestions.*