03-03
Systèmes de recommandation III

NOUS ÉCLAIRONS.
VOUS BRILLEZ.

FORMATION CONTINUE
ET SERVICES AUX ENTREPRISES

Cégep de Sainte-Foy

420-A58-SF — Algorithmes d'apprentissage non supervisé — Été 2021
Spécialisation technique en intelligence artificielle — M. Swawola, M.Sc.

Sommaire

Sommaire

Normalisation par la moyenne

$$\theta^{(4)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

- Utilisateur n'ayant noté aucune série

	Alice (1)	Bob (2)	Mike (3)	Alex (4)	x_1 (crime)	x_2 (horreur)
Money Heist	5	5	1	?	0.90	0
Mindhunter	4	5	?	?	0.95	0.2
The Walking Dead	1	?	5	?	0	0.80
The Haunting of Hill House	?	2	4	?	0.05	0.99
Ash vs Evil Dead	?	0	5	?	0	0.85

$$J(x^{(1)}, x^{(2)}, \cdots, x^{(n_s)}, \theta^{(1)}, \theta^{(2)}, \cdots, \theta^{(n_u)}) = \frac{1}{2} \sum_{i=1}^{n_s} \sum_{(i,j):r^{(i,j)}=1} ((\theta^{(j)})^T x^{(i)} - y^{(i,j)})^2 + \boxed{\frac{\lambda}{2} \sum_{i=1}^{n_s} \sum_{k=1}^{n} (x_k^{(i)})^2 + \frac{\lambda}{2} \sum_{j=1}^{n_u} \sum_{k=1}^{n} (\theta_k^{(j)})^2}$$

Normalisation par la moyenne

$$\theta^{(4)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

- Utilisateur n’ayant noté aucune série

	Alice (1)	Bob (2)	Mike (3)	Alex (4)	x_1 (crime)	x_2 (horreur)
Money Heist	5	5	1	**0**	0.90	0
Mindhunter	4	5	?	**0**	0.95	0.2
The Walking Dead	1	?	5	**0**	0	0.80
The Haunting of Hill House	?	2	4	**0**	0.05	0.99
Ash vs Evil Dead	?	0	5	**0**	0	0.85

$$J(x^{(1)}, x^{(2)}, \cdots, x^{(n_s)}, \theta^{(1)}, \theta^{(2)}, \cdots, \theta^{(n_u)}) = \frac{1}{2}\sum_{i=1}^{n_s} \sum_{(i,j):r^{(i,j)}=1} ((\theta^{(j)})^T x^{(i)} - y^{(i,j)})^2 + \boxed{\frac{\lambda}{2}\sum_{i=1}^{n_s}\sum_{k=1}^{n}(x_k^{(i)})^2 + \frac{\lambda}{2}\sum_{j=1}^{n_u}\sum_{k=1}^{n}(\theta_k^{(j)})^2}$$

Normalisation par la moyenne

$\theta^{(4)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

- Utilisateur n'ayant noté aucune série

	Alice (1)	Bob (2)	Mike (3)	Alex (4)	x_1 (crime)	x_2 (horreur)
Money Heist	[illegible]	[illegible]	[illegible]	0	[illegible]	0
Mindhunter	4	5	?	0	0.95	0.2
The Walking Dead	1	?	5	0	0	0.80
The Haunting of Hill House	?	2	4	0	0.05	0.99
Ash vs Evil Dead	?	0	5	0	0	0.85

$$J(x^{(1)}, x^{(2)}, \cdots, x^{(n_s)}, \theta^{(1)}, \theta^{(2)}, \cdots, \theta^{(n_u)}) = \frac{1}{2} \sum_{i=1}^{n_s} \sum_{(i,j):r^{(i,j)}=1} ((\theta^{(j)})^T x^{(i)} - y^{(i,j)})^2 + \boxed{\frac{\lambda}{2} \sum_{i=1}^{n_s} \sum_{k=1}^{n} (x_k^{(i)})^2 + \frac{\lambda}{2} \sum_{j=1}^{n_u} \sum_{k=1}^{n} (\theta_k^{(j)})^2}$$

Prédiction de mauvaises notes

⇒ Recommandation impossible

Normalisation par la moyenne

$$Y = \begin{bmatrix} 5 & 5 & 1 & ? & ? \\ 4 & 5 & ? & 0 & ? \\ 1 & ? & 5 & 4 & ? \\ ? & 2 & 4 & 5 & ? \\ ? & 0 & 5 & ? & ? \end{bmatrix} \qquad \mu = \begin{bmatrix} 3.67 \\ 3 \\ 3.33 \\ 3.67 \\ 2.5 \end{bmatrix}$$

$$Y = \begin{bmatrix} 1.33 & 1.33 & -2.67 & ? & ? \\ 1 & 2 & ? & -3 & ? \\ -2.33 & 1.67 & & 2.67 & ? \\ ? & -1.67 & 0.33 & 1.33 & ? \\ ? & -2.5 & 2.5 & ? & ? \end{bmatrix}$$

Apprendre $x^{(i)}, \theta^{(j)}$

Normalisation par la moyenne

	Alex (4)
Money Heist	?
Mindhunter	?
The Walking Dead	?
The Haunting of Hill House	?
Ash vs Evil Dead	?

- Pour l'utilisateur j et la série i

  → prédiction $(\theta^{(j)})^{\mathsf{T}} x^{(i)} + \mu^{(i)}$

- Pour l'utilisateur 4 (Alex)

  → prédiction $\mu^{(i)}$ **car** $(\theta^{(j)})^{\mathsf{T}} x^{(i)} = 0$

Normalisation par la moyenne

	Alex (4)
Money Heist	? $\longrightarrow$ 3.67
Mindhunter	? $\longrightarrow$ 3
The Walking Dead	? $\longrightarrow$ 3.33
The Haunting of Hill House	? $\longrightarrow$ 3.67
Ash vs Evil Dead	? $\longrightarrow$ 2.5

- Pour l'utilisateur j et la série i

  $\rightarrow$ prédiction $(\theta^{(j)})^{\mathsf{T}} \boldsymbol{x}^{(i)} + \boldsymbol{\mu}^{(i)}$

- Pour l'utilisateur 4 (Alex)

  $\rightarrow$ prédiction $\boldsymbol{\mu}^{(i)}$ **car** $(\theta^{(j)})^{\mathsf{T}} \boldsymbol{x}^{(i)} = 0$

- **La normalisation par la moyenne apporte la notion de note négative v. note positive**

Exercice

- Pourquoi ne divise t-on pas par l'écart-type ou l'étendue (range) comme en normalisation standard ou autres méthodes étudiées précédemment ?

- Réponse: les notes (par exemple les étoiles) sont sur la même échelle !

Sommaire

Factorisation matricielle de bas rang

$$Y = \begin{bmatrix} 5 & 5 & 1 & ? \\ 4 & 5 & ? & 0 \\ 1 & ? & 5 & 4 \\ ? & 2 & 4 & 5 \\ ? & 0 & 5 & ? \end{bmatrix} \qquad \text{prédictions} = \begin{bmatrix} (\theta^{(1)})^T x^{(1)} & (\theta^{(2)})^T x^{(1)} & \dots & (\theta^{(n_u)})^T x^{(1)} \\ (\theta^{(1)})^T x^{(2)} & (\theta^{(2)})^T x^{(2)} & \dots & (\theta^{(n_u)})^T x^{(2)} \\ \vdots & \vdots & \vdots & \vdots \\ (\theta^{(1)})^T x^{(n_s)} & (\theta^{(2)})^T x^{(n_s)} & \dots & (\theta^{(n_u)})^T x^{(n_s)} \end{bmatrix}$$

$$X = \begin{bmatrix} (x^{(1)})^T \\ (x^{(2)})^T \\ \vdots \\ (x^{(n_s)})^T \end{bmatrix} \qquad \Theta = \begin{bmatrix} (\theta^{(1)})^T \\ (\theta^{(2)})^T \\ \vdots \\ (\theta^{(n_u)})^T \end{bmatrix}$$

$$X\Theta^T$$

Matrice de bas rang

Sommaire

Identification des produits similaires

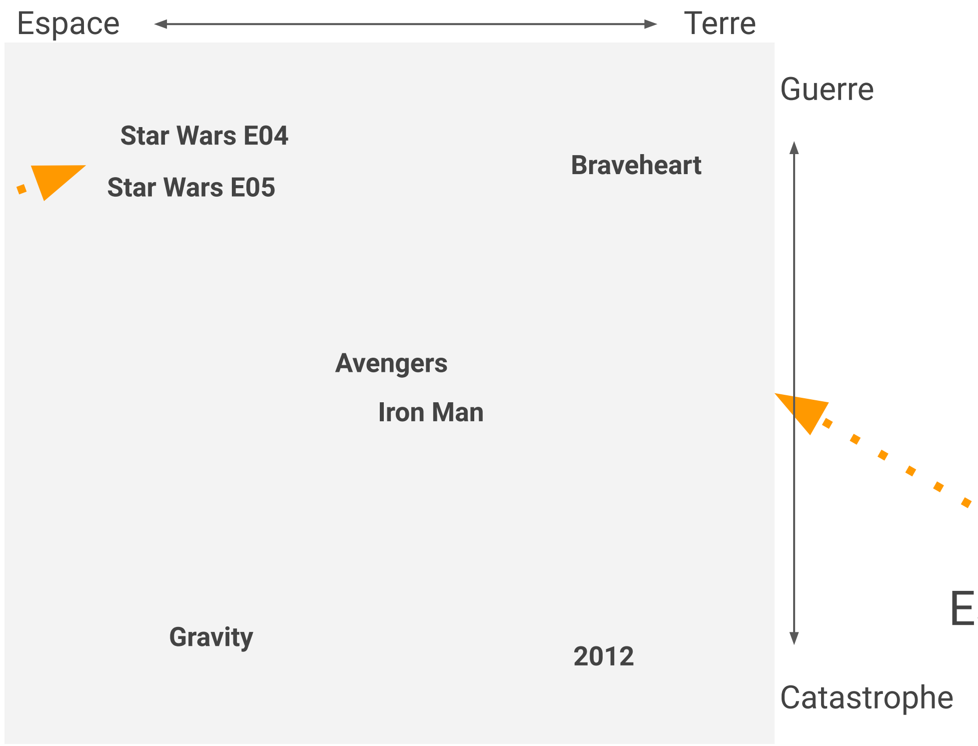
- Pour chaque produit i, nous apprenons un vecteur de feature $\boldsymbol{x}^{(i)} \in \mathbb{R}^n$
- Exemple avec les films:
  - $\boldsymbol{x}_1$ = action, $\boldsymbol{x}_2$ = comédie, $\boldsymbol{x}_3$ = drame, $\boldsymbol{x}_4$ = Steven Seagal dedans, etc ...

Selon vous, comment peut-on identifier les films similaires ?

Identification des produits similaires

- Trouver les films f similaires au film i
  - Si || $\boldsymbol{x}^{(i)}$ - $\boldsymbol{x}^{(f)}$ || est petit, alors les films f et i sont similaires
  - Pour trouver le 3 films j les plus similaires à i, trouver les 3 films avec les plus petites valeurs de || $\boldsymbol{x}^{(i)}$ - $\boldsymbol{x}^{(f)}$ ||

Espace latent



Films similaires se retrouvent proches dans l'espace latent

Espace latent

Sommaire

Évaluation

- Les techniques de **validation croisée** ou **de jeu de test** s’appliquent de manière analogue

1	3	4			
	0	3			5
	4	5			5
	2			3	2
				2	
1	0	5	1		1

1	3	4			
	0	3			5
	4	5			5
	2			?	?
				?	
1	0	5	?		?

Jeu de test

Évaluation

- Autres méthode plus robuste par rapport à l’ordre les lignes / colonnes

1	3	4			
	0	3			5
	4	5			5
	2			3	2
				2	
1	0	5	1		1

1	3	?			
	?	3			?
	4	5			5
	2			3	2
				?	
1	0	?	1		1

Jeu de test “disséminé”

Évaluation

- **Comparaison** des prédictions avec les notes connues
  - RMSE
  - Précision du “top 10”
  - Corrélation du classement (Correlation de Spearman)

    → Comparaison des classements entre le système de recommandation et ceux des utilisateurs

Évaluation

- **Modèle 0/1**
  - Coverage: nombre de produits par utilisateur pour lequels le système peut réaliser une prédiction
  - Précision des prédictions
  - Courbe ROC

Évaluation

- **Désavantages des mesures d’erreurs**
    - Améliorer la précision peut impacter négativement la diversité ou le contexte des prédictions
    - En pratique, les prédictions de notes élevées sont plus intéressantes que la prédiction de notes basses

        → RMSE peut effectivement pénaliser une méthode fonctionnant bien sur les notes élevées et moins bien sur les notes faibles

Le plus de données, le mieux !

- **Quelques conseils**
  - Utiliser le maximum de données possibles !

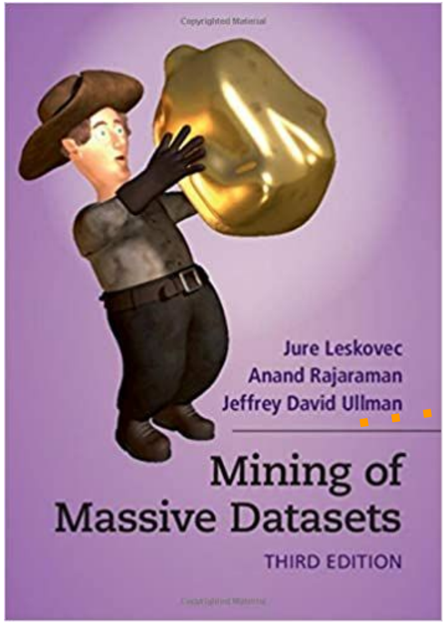
    → Mieux vaut un algorithme simple avec beaucoup de données qu'un algorithme complexe avec moins de données
  - Ajouter des données lorsque possible
- More data usually beats better algorithms

Sommaire

Pull de https://github.com/mswawola-cegep/420-a58-sf.git

03-03

Sommaire



9 Recommendation Systems
p. 319-353

Jure Leskovec, Anand Rajaraman, Jeffrey D. Ullman, **Mining of Massive Datasets, 3rd edition**

Références

[1] CS229: Machine Learning - Stanford University

[2] Mining of Massive Datasets, 3rd edition

[3] More data usually beats better algorithms