Model Based Collaborative Filtering : Matrix Factorization

Nassim Ait Ali Braham, Jean Dupin, Vincent Gouteux

Projet Science des Données

28 Octobre 2019

Résumé

Latent Factor Models

- Intuition:
  - Existe-t-il des attributs cachés (*latent features*) qui déterminent comment un utilisateur va évaluer un film donné ?
  - Comment les identifier ?
- Méthode : *Matrix Factorization*
  - m utilisateurs, n films, et $R \in \mathbb{R}^{m \times n}$ la matrice des évaluations connues.
  - Idée: Approcher la matrice R par

$$R \approx U.V^T$$

avec $U \in \mathbb{R}^{m \times k}$ et $V \in \mathbb{R}^{n \times k}$ où k est le nombre d'attributs cachés. Chaque vecteur ligne de U, (resp. V) représente le degré d'association entre un utilisateur donné (resp. un film donné) et chacun des attributs cachés.

Implémentation

- Implémenter la descente de gradient
- Implémenter les moindres carrés alternés
- Régularisation: contrôler l'ordre de grandeur des matrices U, V pour obtenir une bonne approximation de R en évitant des degrés d'association trop élevés.
- Optimisation du paramètre k et des paramètres de régularisation.
- Comparer performances

Description et préparation des données

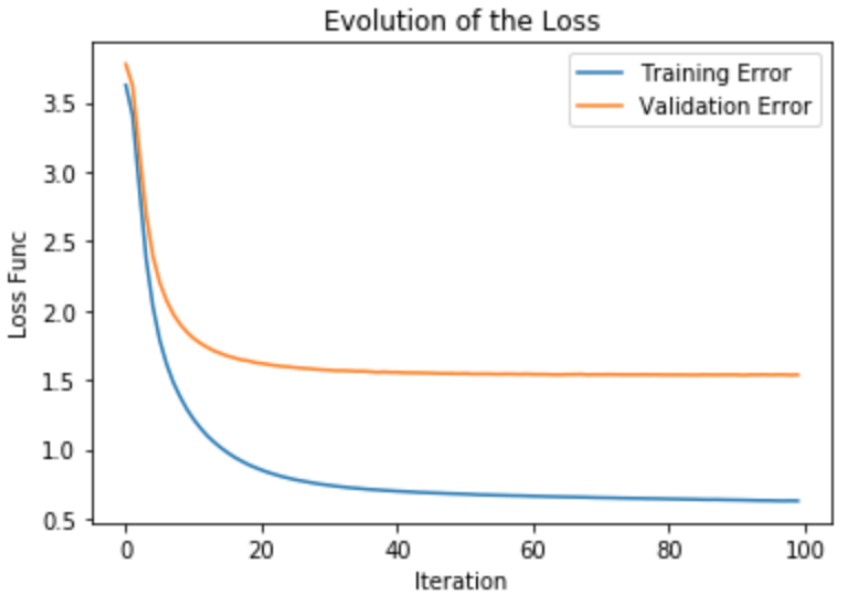
- Dataset: *Small MovieLens Dataset*
  - 100 836
  - 610 individus, 9 742 films.
  - 100 836 - 610*9 742 i.e 5 841 784 valeurs manquantes à prédire.
  - Split de données training/validation/test 60/20/20

Descente de gradient

$$U_{t+1} = U_t - \alpha_t \frac{\partial C}{\partial U}(U_t, V_t)$$

$$V_{t+1} = V_t - \alpha_t \frac{\partial C}{\partial U}(U_t, V_t)$$

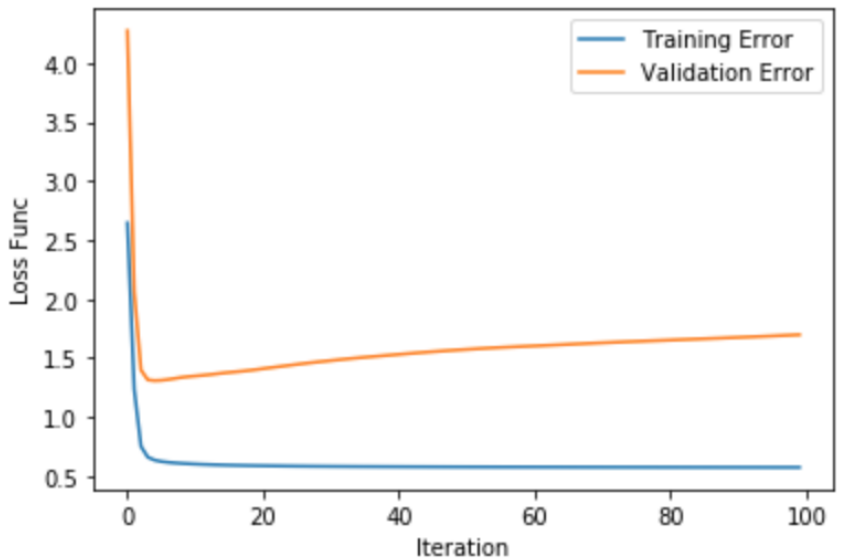
Avec $\alpha = 0.01$ et $(k, \lambda, \mu) = (4, 0.02, 0.02)$




```
Training error is: 0.628649404165025
Validation error is: 1.5375442449644279
```

Moindres carrés alternés

Avec $\alpha = 0.01$ et $(k, \lambda, \mu) = (4, 0.02, 0.02)$



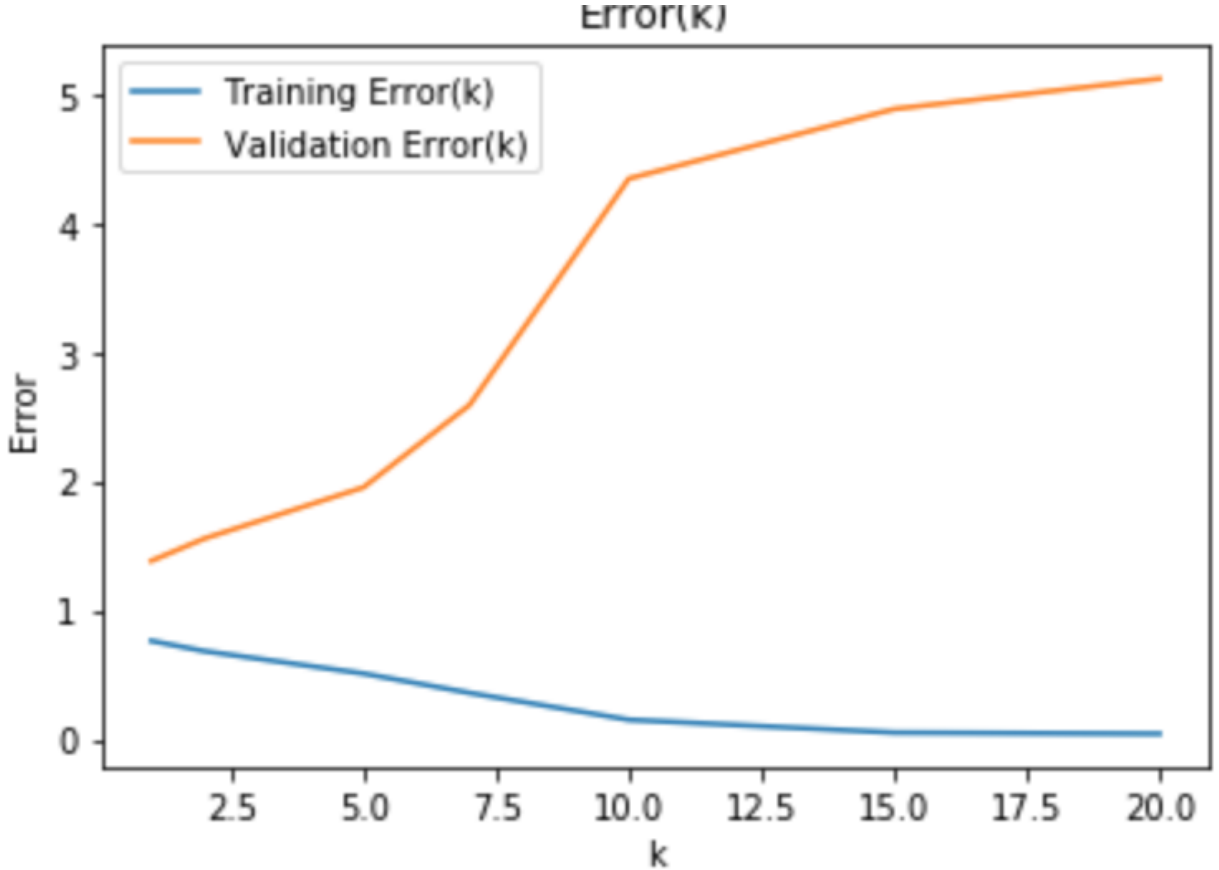

```
Training error is: 0.5760869652455342
Validation error is: 1.6982579720054556
```

Optimisation des paramètres

- Ensemble de valeurs candidates pour k, λ et μ
- Trouver le *meilleur triplet* : plus faible erreur sur les données de validation
- Temps d'exécution pour SGD avec 100 itérations: 1 à 5 mn. Sur un grid de taille 5x5x5, temps d'exécution $\sim$ 2h
- Temps d'exécution pour ALS pour 100 itérations: 20 à 30min. Sur un grid de taille 5x5x5, temps d'exécution $\sim$ 50h...
- Meilleure méthode ?
  - Déterminer un k optimal pour un couple λ, μ fixé puis parcourir (λ, μ)
  - Un unique paramètre de régularisation λ ?

Recherche du k optimal

Avec $\alpha = 0.01$ et $(\lambda, \mu) = (0.02, 0.02)$

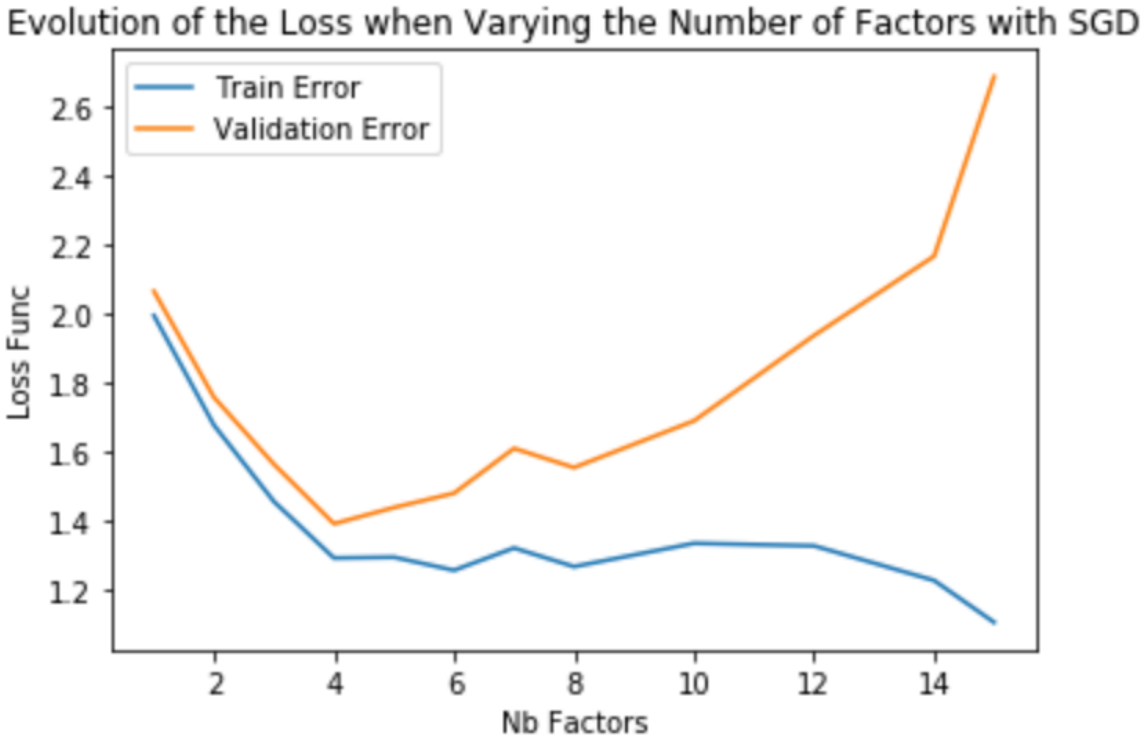


Difficultés rencontrées et pistes à explorer

Problème du *cold start user* : certains individus n'avaient plus de notes après la séparation *Training, Validation et Testing Sets*.
Comment traiter les valeurs manquantes ?

Résultats étranges : les courbes obtenues par variation de k n'ont pas l'allure attendue.

Difficultés rencontrées et pistes à explorer



Graphique des valeurs des train error et validation error en fonction de k, obtenu avec une matrice artificielle pleine de rang 5

Difficultés rencontrées et pistes à explorer

- Prédictions ≥ 5 : instaurer une contrainte sur les notes ?
- Explorer les techniques pour la gestion des valeurs manquantes
- Optimisation des hyper-paramètres : k, μ et *lambda*
- Essayer de nouveau algorithmes : *Mini Batch Gradient Descent*
- Normalisation des données ? Termes de biais ?
- Optimisation du code