

April 13, 2017

Abstract

1 Qu'est ce qu'un système de recommandation?

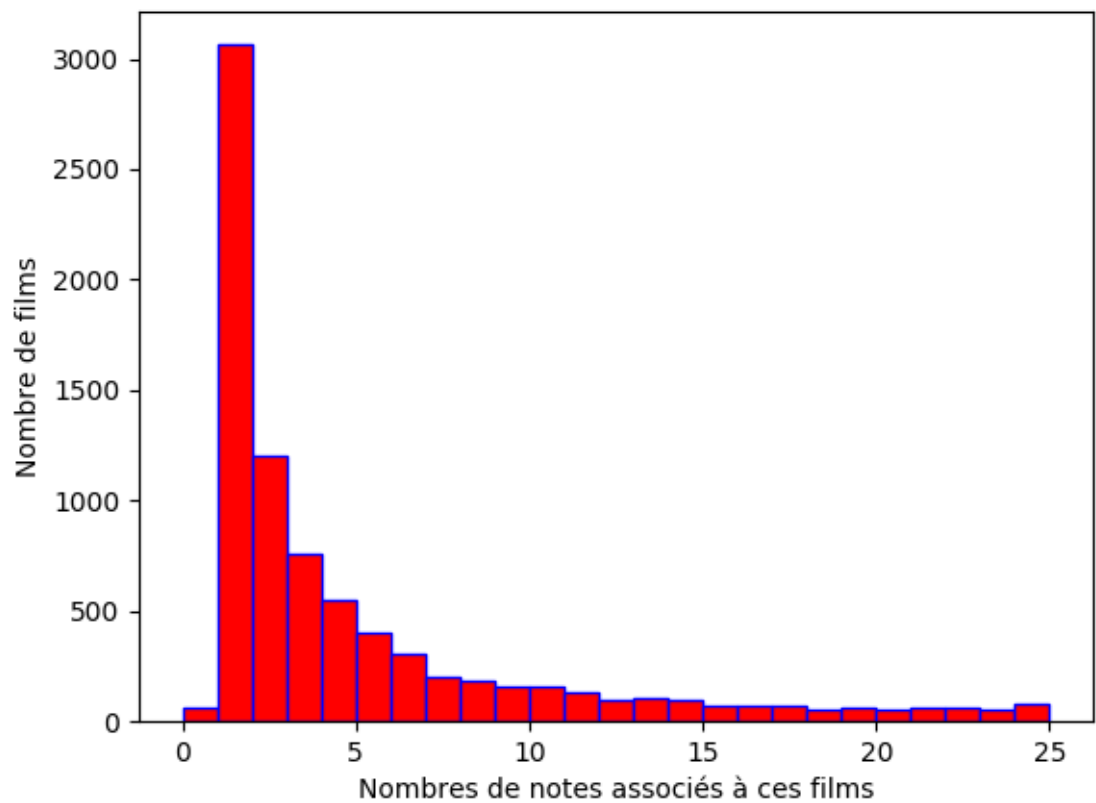
1.1 Notre approche

Le but de notre projet est de recommander un film à un utilisateur spécifique. Pour cela, nous devons savoir à quel point un utilisateur aime tel ou tel film pour lui recommander (ou pas). Il nous faudrait donc une sorte d'échelle d'affinité de l'utilisateur au film, ce qui paraît pertinent est donc d'estimer une note qu'un utilisateur mettrait à un film s'il avait à le noter par exemple. Si cette note est au dessus d'un certain seuil, on lui recommandera (ou on recommandera le film ayant la meilleure note prédite par exemple). Notre approche du problème est donc de trouver un modèle mathématique permettant de trouver la note que mettrait un utilisateur à un film qu'il n'a pas vu. Cependant, sans aucune connaissance des goûts l'utilisateur et des caractéristiques des films ceci est impossible. Il nous faut donc récupérer des données déjà existantes sur des utilisateurs et des films pour avoir une base et commencer à chercher une méthode. Nous avons donc récupéré des données.

1.2 Première difficulté : les données

1.2.1 Extraction des données

Le besoin d'avoir des données nous a guidé vers une grande base de données appelée "movielens" qui est un site communautaire de recommandation de films où les utilisateurs du site notent des films de 0 à 5. Plusieurs jeux de données étaient disponibles et différaient entre eux selon leur taille. Nous avons choisi de travailler avec une base de données de 670 utilisateurs (nu) et 9125 films (nf). On a donc extrait 2 fichiers : l'un contenant les 9125 films avec leurs titres et un numéro attribué, l'autre avec les notes des utilisateurs qui avaient chacun un identifiant, le fichier était du type : identifiant de l'utilisateur, id du film qu'il a noté, note. Ces 2 fichiers étaient donc peu pratiques pour commencer à faire quelque chose avec, nous avons donc créé une fonction `tableau_des_notes()` qui permet de ranger toutes ces notes dans un tableau numpy `nu*nf` avec en lignes les utilisateurs, et en colonnes les films. Quand un utilisateur n'a pas vu un film donc qu'il ne l'a pas noté, on insère un "Nan" (Not a Number) qui est un "symbole" facile à traiter. On appellera ce tableau `Y` tout au long du projet. (detail).png



1.2.2 Analyse des données

1.3

2 Formalisation du problème d'optimisation y compris modelisation

2.1 Factorisation

2.2 Fonction cout

2.2.1 Pourquoi on cherche a minimiser

3 Implémentation algo

3.1 descente du gradient en general optimisation

▽

3.1.1 écrire algo puis dire comment on a fait au début

4 Mise en application

4.1 Notre approche pour familiariser au problème

4.2 yassine

4.2.1 taux d'erreur

4.2.2 influence des paramètres

4.3 limites de notre approche

4.3.1 Est-ce qu'on arrive à recommander un film à un utilisateur

5 conclusion

5.1 perspectives