

Introduction

Définition : collaborative filtering

Position du problème : Attentes vis-à-vis des algorithmes (précision/temps de réponse/ajout rapide d'un rating/cold start...)

Efficacité peut varier selon la forme du jeu de données (sparsity...)

Objectif : Implémenter (en Matlab) et comparer différents algorithmes en terme de précision et de temps d'exécution sur différents jeux de données, mettre en relief les compromis à faire.

1 Les algorithmes utilisés

1.1 Algorithmes témoins

Algorithmes naïfs servant de point de comparaison :

- "Witness" : Estimer un rating inconnu par la moyenne de tous les ratings connus
- "PerUserAverage" : Estimer la note donnée par un utilisateur à un objet par la moyenne des notes données par les autres utilisateurs à cet objet
- "UnbiasedWitness" : Retirer les biais comme dans le cours, estimer tous les ratings inconnus par 0, remettre les biais

1.2 SVD

Variantes sur la question bonus du DM (comment traiter les trous dans la matrice, avec ou sans traitement des biais...).

- "ShiftSVD" : traduire la matrice pour avoir 0 de moyenne, SVD, retraduire
- "UnbiasedSVD" : enlever les biais, SVD, remettre les biais

1.3 Algorithmes Slope One

Idée : régression linéaire, en imposant que la pente vaut 1.

Ajout d'un nouveau rating et traitement de requêtes très rapides, il faut voir à quel point c'est moins précis que d'autres algos plus sophistiqués.

Algorithmes tirés de [2].

1.4 Algorithmes par similarité cosinus

Implémentation des algorithmes vus en cours : "itemCosSimilarity", "userCosSimilarity".

1.5 Analyse en composantes principales : algorithme Eigentaste

Idée : s'appuyer sur un petit sous-ensemble d'objets notés par tous les utilisateurs (*gauge set*) pour projeter un utilisateur sur un espace de petite dimension puis estimer ses notes à partir de celles de ses voisins au sens d'un algorithme de clustering.

Défauts : on demande à un nouvel utilisateur de noter l'intégralité du *gauge set* pour l'ajouter, et le cold start pose problème.

La précision est-elle vraiment meilleure que celle d'autres algorithmes plus simples ?

Algorithme extrait de [1].

2 Observations expérimentales

2.1 Jeux de données

- Matrice de la question bonus du DM (pleine, on observe seulement une certaine fraction des ratings)
- Jeu de données Jester (ratings d’une centaine de blagues, dix blagues sont notées par toutes les utilisateurs) utilisé pour Eigentaste.
- Jeu de données plus grand (MovieLens) pour mesurer les difficultés liées aux temps d’exécutions dans des conditions plus réalistes ?

2.2 Mesures d’erreur

	p= 0.2	0.4	0.6	0.8
Witness	0.0148	0.0112	0.0074	0.0037
UnbiasedWitness	0.0085	0.0062	0.0041	0.0020
PerUserAverage	0.0149	0.0111	0.0075	0.0037
BiasFromMean	0.0086	0.0062	0.0041	0.0021
PlainSVD	0.0940	0.0490	0.0221	0.0084
ShiftSVD	0.0106	0.0065	0.0037	0.0016
UnbiasedSVD	0.0087	0.0060	0.0036	0.0016
UserCosSim	0.0094	0.0083	0.0076	0.0070
ItemCosSim	0.0091	0.0077	0.0062	0.0049
SlopeOne	0.0086	0.0062	0.0042	0.0020
WeightedSlopeOne	0.0085	0.0062	0.0041	0.0020

Sur la matrice du DM, on obtient ce tableau d’erreurs MSE pour différentes valeurs de p : Eigentaste et quelques autres ne sont pas encore implémentés. à venir aussi : une autre mesure d’erreur, et des mesures sur les autres jeux de données.

2.3 Temps d’exécution

Les algorithmes qui enlèvent et remettent les biais comme dans le cours s’exécutent beaucoup plus lentement : il faut inverser une matrice creuse, et on n’a pas réussi à vectoriser une partie du calcul de l’argmin pour les biais. Il faudra essayer l’expression plus naïve des biais et voir ce qu’on gagne en temps/perd en précision.

Il faut aussi tenir compte du fait que certains algos (Slope One oui, SVD non) calculent des choses ”une fois pour toutes”, et c’est rapide d’ajouter un utilisateur et/ou de faire une requête.

Références

- [1] Ken Goldberg, Theresa Roeder, Dhruv Gupta, and Chris Perkins. Eigentaste : A constant time collaborative filtering algorithm. *Inf. Retr.*, 4(2) :133–151, July 2001.

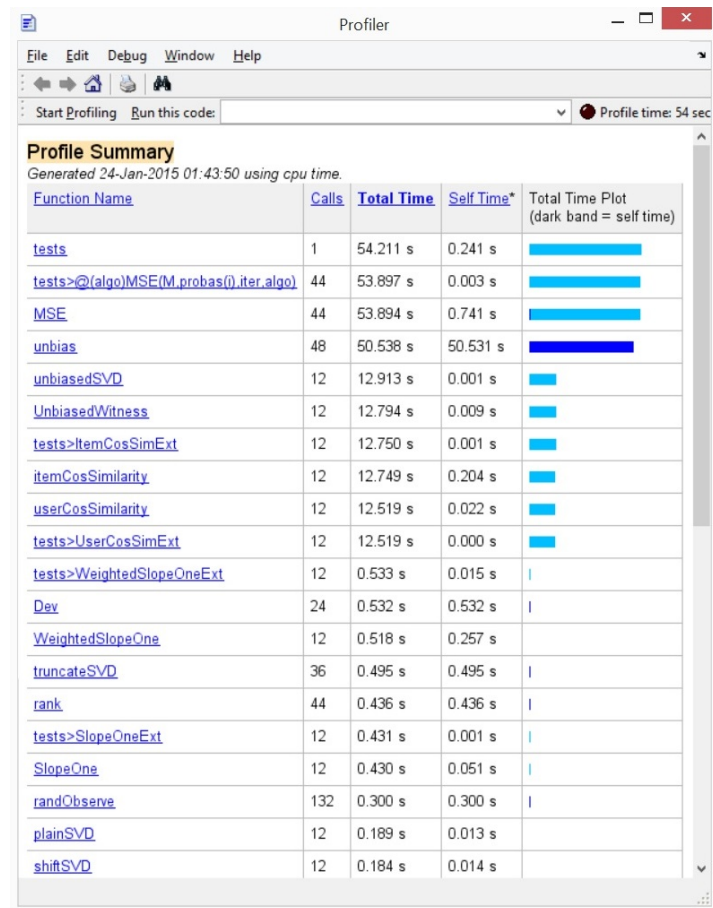


FIGURE 1 – Profiling fourni par Matlab

- [2] Daniel Lemire and Anna Maclachlan. Slope one predictors for online rating-based collaborative filtering. *CoRR*, abs/cs/0702144, 2007.