Alexia BOURMAUD

Louise MARCHAL

TP4 - Filtrage collaboratif

Dans ce TP nous allons tester plusieurs algorithmes permettant de faire du filtrage collaboratif. Le but est de recommander des films susceptibles de plaire à un utilisateur grâce aux notes que d'autres utilisateurs ont mis sur des films qu'il a déjà notés.

Pour réaliser nos recommandations, nous avons utilisé trois algorithmes de factorisation de matrice : SVD et NMF. Dans notre cas ces derniers sont intéressants car il manque des données dans notre matrice de départ et ces algorithmes permettent de combler les valeurs manquantes.

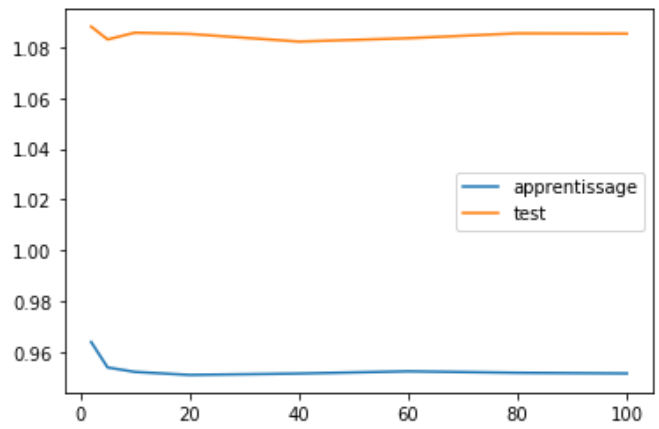
Nous avons séparé la base de données initiale en deux matrices sparses pour former un ensemble d'apprentissage contenant 90% des données et un ensemble de test contenant 10% des données

1. SVD

SVD (Singular Value Decomposition) est une méthode permettant de décomposer une matrice (de taille N*F) en trois sous matrices qui multipliés entre elles permettent de reconstruire cette matrice. Ces trois sous-matrices ont pour taille respectives N*C, C*C, C*F.

Pour améliorer notre recommandation, nous avons pris en compte le biais lié à la façon de noter de chaque utilisateur. En effet, certains utilisateurs auront tendances à donner de meilleures notes que d'autres et inversement. Pour cela, nous avons calculé pour chaque utilisateur la moyenne des notes qu'il avait mis puis nous avons retiré cette valeur à chaque note avant de faire un SVD.

Nous évaluons nos résultats à l'aide d'un coût quadratique en faisant varier la taille de la dimension C dans les matrices résultantes du SVD. Il est possible de calculer le coût MSE en comparant les notes prédites et les notes que l'on avait déjà.



	MSE en apprentissage	MSE en test
2	0.963915	1.088177
5	0.953888	1.083100
10	0.952139	1.085789
20	0.950948	1.085271
40	0.951494	1.082277
60	0.952377	1.083588
80	0.951793	1.085511
100	0.951532	1.085381

Figure 1 : Cout quadratique en apprentissage et en test en fonction de la dimension C

On observe que la valeur de dimension C n'a pas beaucoup d'influence sur l'efficacité de la recomposition de la matrice. On gardera donc une dimensionalité de 5. Globalement, on obtient une erreur quadratique proche de 0,95 en apprentissage et environ 1,1 en test.

On peut maintenant recommender des films à un utilisateur en prenant les meilleures notes prédites lorsqu'on recompose la matrice de notes. Il faut évidemment vérifier que l'utilisateur n'a pas déjà vu les films.

	film	note estimée
0	Little Women (1994)	4.749726
1	Brazil (1985)	4.696947
2	Princess Bride, The (1987)	4.662173
3	Taxi Driver (1976)	4.334350
4	Wings of Desire (Der Himmel über Berlin) (1987)	4.289879

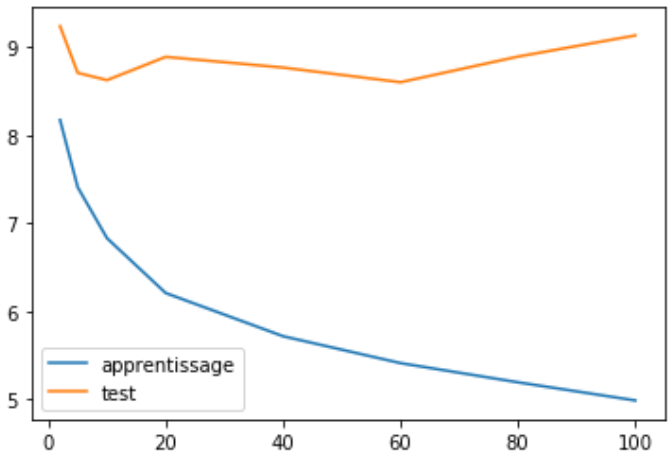
Figure 2 : Recommandation de films pour l'utilisateur 5

On peut en déduire que les cinq films ayant le plus de chance de plaire à l'utilisateur cinq sont Little Women, Brazil, The princess bride, Taxi Driver et Wings of Desire. De plus, on estime la note que pourrait mettre l'utilisateur après avoir vu le film.

2. NMF

NMF (Non-negative Matrix Approximation) est un autre algorithme permettant de décomposer une matrice (de taille N*F) à valeurs positives en deux sous-matrices (de taille N*C et C*F) contenant également des valeurs positives.

Comme pour SVD, nous avons fait varier la dimension C afin d'observer son impact sur la recomposition de la matrice initiale.



	cout MSE en apprentissage	Cout MSE en test
2	8.169461	9.233693
5	7.407550	8.704080
10	6.828230	8.621561
20	6.205860	8.885478
40	5.715776	8.764013
60	5.410702	8.597508
80	5.193152	8.888490
100	4.987090	9.128422

Figure 3 : Cout quadratique en apprentissage et en test en fonction de la dimension C

On remarque qu'augmenter la dimension de C n'est pas une bonne idée car cela mène qu'à augmenter le sur-apprentissage.

De plus, on remarque que l'erreur est nettement plus élevée que l'erreur obtenue avec l'algorithme SVD.

Voici ci-dessous, les recommandations faites par NMF pour l'utilisateur 5.

	film	note estimée
0	Scout, The (1994)	3.069851
1	Mass Appeal (1984)	2.843230
2	Don Juan DeMarco (1995)	2.337794
3	Onegin (1999)	2.299196
4	Price of Glory (2000)	2.223488

Nous avons cherché si sklearn ne possédait pas une fonction NMF qui ne considérait pas les valeurs manquantes de la matrice sparse comme des zéros. Cependant nos recherches sont restées vaines. La seule solution serait de réaliser nous même une descente de gradients afin de trouver les sous matrices qui recomposent au mieux la matrice initiale.

3. Conclusion

Nous avons testé plusieurs méthodes permettant de faire de la recommandation. La méthode qui nous fournit les meilleurs résultats est SVD. En effet, l'erreur quadratique est suffisamment faible pour que les recommandations soient considérées comme fiables.