Méthodes Mathématiques pour la Grande Dimension

Laboratoire

Vue d'ensemble du projet :

Le but de ce projet est de vous faire explorer plusieurs techniques fondées sur la factorisation de matrice et leurs applications. Vous aurez une grande liberté dans le déroulement du TP et le rendu, afin d'encourager votre créativité et votre analyse. Les informations listées ici sont données à titre indicatif, dans le but de vous guider vers des pistes intéressantes. Vous êtes volontairement laissés libres de votre travail, sans objectif de performance. Vous n'avez pas d'obligation de résultat, mais une obligation de moyens.

Normes :

- Vous travaillerez impérativement par groupes de deux ou trois
- Vous rendrez un rapport au format PDF de 5 pages maximum
- Les étudiants me soumettant le corrigé du TD avant le 20/05/2020 avant 23h59 auront une bonification d'un point (sous réserve de travail sérieux)
- La date maximale de rendu est le 29/05/2020 à 23h59.
- Afin de garder la communication la plus fluide possible, je vous invite à rejoindre l'espace de travail : https://join.slack.com/t/tpmmgd/shared_invite/zt-e303v9kn-SsewlcKJ~S~c9PaMsLmKQg

Objectifs :

- Vous appliquerez au moins un algorithme de factorisation de matrice à un des jeux de données, vous expliquerez ses fondements mathématiques ainsi qu'algorithmiques et présenterez vos résultats.
- J'évaluerai principalement votre capacité d'analyse, de synthèse et d'esprit critique : il est essentiel d'expliquer clairement ce que vous faites, et pour quoi. Un seul algorithme testé, mais expliqué clairement vaut mieux que 10 méthodes testées et laissées sans explication.
- A la fin de ce document, vous trouverez des références. N'hésitez pas à vous en servir.
- N'hésitez pas non plus à chercher d'autres références (sans oublier bien sûr de les citer).
- Les exemples de codes sont donnés en Python. Si vous êtes plus à l'aise avec un autre langage, n'hésitez pas à l'utiliser.
- Gardez à l'esprit qu'il faut toujours être critique avec votre démarche : justifiez vos choix techniques tels que les algorithmes ou les normes.

Jeux de données :

Vous avez deux jeux de données à disposition :

- Un jeu de donnée composé d'une matrice M d'occurrences de mots dans un corpus de 8447 articles du New York Times et une liste de 3012 mots associés. Plus précisément,

la matrice M est composée de 3012 lignes et 8447 colonnes. Mi,j correspond au nombre de fois ou le mot i apparait dans le texte de la colonne j.

- Un jeu de données de 3141793 notes [1-10] données par 39189 utilisateurs à 4365 films (en l'occurrence, des animes). Chaque utilisateur a noté au moins 20 films. Le fichier correspond relie les id des anime à leurs noms.

Pistes de recherche :

Nous avons vu en cours que les algorithmes de factorisation de matrice peuvent être utilisés pour la classification ou pour la complétion de matrices. Comment pouvez vous appliquer ces techniques aux données fournies pour obtenir des résultats intéressants ?

Références :

Les fondamentaux :

- https://ocw.mit.edu/resources/res-18-010-a-2020-vision-of-linear-algebra-spring-2020/ Un cours d'algèbre linéaire de référence, allant de « pourquoi l'algèbre linéaire ? » à la notions de vecteurs singuliers sont très bien expliqués. Ce cours est très clair et didactique, les chapitres sont relativement courts et permettent de construire une vraie intuition des notions d'algèbre linéaire nécessaires pour les Mathématiques pour la Grande Dimension.
- https://www.youtube.com/playlist?list=PLtmWHNX-gukIc92m1K0P6bIOnZb-mg0hY Pour ceux qui veulent vraiment maitriser le sujet de fond en comble, ce cours (10 séances d'1h30, et les notebooks de TD/TP associés https://github.com/fastai/numerical-linear-algebra vous donneront une connaissance solide du sujet. Ce sont les cours de master donnés par l'élite mondiale des sciences des données (une des universités de San Francisco) aux futurs chercheurs et ingénieurs de la Silicon Valley. La deuxième vidéo recouvre particulièrement ce que nous avons vu ici.
- https://setosa.io/ev/eigenvectors-and-eigenvalues/ Un très joli site pour jouer avec les notions de valeurs propres, vecteurs propres etc. En plus de permettre la construction d'une intuition visuelle, il présente clairement des problèmes d'algèbre linéaires pour lesquels ces outils sont utiles.
- https://hadrienj.github.io/ Ce site est un peu mal organisé, mais les concepts de Mathématiques pour la science des données sont très bien expliqués, avec beaucoup d'illustrations, et le code python pour manipuler ces illustrations. Le livre d'Hadrien Jean « Essential Math for Data Science » sortira en Octobre (il est disponible en pré-accès payant), et il pourrait bien devenir votre lecture de chevet.
- http://wikistat.fr/pdf/st-m-explo-nmf.pdf , http://wikistat.fr/pdf/st-m-datSc3-colFil.pdf et http://wikistat.fr/pdf/st-m-explo-acp.pdf Des synthèses efficaces des techniques abordées. En règle générale, https://www.math.univ-toulouse.fr/~besse/Wikistat/ a beaucoup de bonnes fiches de synthèse.

Pour le TP, code, tutoriel et algorithmique :

- https://medium.com/@nixalo/comp-linalg-l2-topic-modeling-with-nmf-svd-78c94330d45f
  Cet article présente la « modélisation de sujet », soit la classification non supervisée à l'aide de techniques de réduction de matrice. Ici sont utilisées la décomposition en valeurs singulières et la factorisation de matrices non négatives. Vous trouverez quelques exemples intéressants de code en Python, dans l'article même, et dans le notebook Jupyter https://nbviewer.jupyter.org/github/fastai/numerical-linear-algebra/blob/master/nbs/2.%20Topic%20Modeling%20with%20NMF%20and%20SVD.ipynb associé. Le notebook présente aussi des exemples intéressants avec des images.
- https://www.kaggle.com/CooperUnion/anime-recommendations-database/kernels
  Beaucoup d'explications et de tutoriels pour un système de recommandation d'anime.
- https://campus.datacamp.com/courses/unsupervised-learning-in-python/discovering-interpretable-features?ex=1 Quelques exemples succincts de code pour la NNMF avec Python
- https://www.programcreek.com/python/example/101468/sklearn.decomposition.NMF
  Des exemples de projets utilisant la NNMF avec Python.
- https://towardsdatascience.com/prototyping-a-recommender-system-step-by-step-part-2-alternating-least-square-als-matrix-4a76c58714a1
  Un tutoriel de construction de moteur de recommandation avec une méthode d'approximation de matrice (ALS)

Niveau avancé :

- https://research.fb.com/fast-randomized-svd/
  Un article publié par la section de recherche fondamentale de Facebook (de la recherche de très haut niveau, donc). Cet article présente rapidement une nouvelle approche pour calculer une SVD approchée d'une grande matrice fondée sur l'aléatoire. Les algorithmes aléatoires sont un domaine de recherche très actif aujourd'hui car ils ont de bonnes garanties théoriques de fonctionnement. Pour ceux qui seraient plus intéressés par ce type d'algorithmes, je recommande : http://theory.stanford.edu/~valiant/teaching/CS265/index.html
- http://www.columbia.edu/~jwp2128/Teaching/E4903/papers/nmf_nature.pdf Le papier qui a lancé la NNMF : c'est court, efficace et élégant.
- https://arxiv.org/pdf/0909.4061.pdf
  Un papier de revue sur des techniques de construction de matrices approches fondées sur les algorithmes aléatoires. Beaucoup de mathématiques et de notions complexes, mais le feuilleter peut vous donner une idée des réflexions actuelles sur la question.
- https://deepai.org/publication/robust-non-linear-matrix-factorization-for-dictionary-learning-denoising-and-clustering Un papier de recherche su la factorisation non linéaire de matrice. Là, on entre dans le domaine des mathématiques vraiment costaudes.