Transport optimal appliqué à la recommandation d'œuvres

Ryan Lahfa

Table des matières

Table des figures

1 Position du problème

1.1 Recommandation d'entrées

À partir d'un ensemble de préférences exprimées par des utilisateurs, on voudrait une méthode de prédiction des futures préférences des utilisateurs. On s'attachera à résoudre ce problème grâce à l'apprentissage automatique supervisé.

Précisément, on se donne une base de données représentée par une matrice $M \in \mathcal{M}_{n,m}(\{0,1\})$ dont le terme général $(m_{i,j})$ indique si l'utilisateur i a aimé l'entrée j.

À partir de M, on veut apprendre un classifieur, i.e. $f : [[1,n]] \times [[1,m]] \to \{0,1\}$, capable de prédire $\widehat{m_{i,j}} = f(i,j)$ si celui-ci n'est pas connu, avec la probabilité, ou non, $p_{i,j} = \mathbb{P}(m_{i,j} = \widehat{m_{i,j}} \mid \theta) \in [0,1]$ où θ est une forme d'information partielle sur M.

On notera aussi $r_u \in \mathbb{R}^m$ la distribution de l'utilisateur $u \in [[1,n]]$ sur les entrées.

1.2 Jeu de données : Mangaki

Le modèle sera testé sur le jeu de données fournis par le site [14] qui comporte :

- 2289 utilisateurs ;
- 12479 œuvres issus de l'animation japonaise (anime, mangas) ;
- plus de 350000 notes

2 Modèle de comparaison : 20-KNN

2.1 Introduction

Pour $k \geq 1$ (ici $k = 20$ d'où 20-KNN), le modèle des k-plus proches voisins consiste en :

- Pour chaque utilisateur i, déterminer les k plus proches voisins, qu'on notera $\mathcal{N}(i) \subset [[1, n]]^k$. Par voisin proche, on entend au sens d'une métrique $d : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}_+$ opérant sur les distributions d'utilisateurs.
- Pour classifier une nouvelle entrée pour un utilisateur i, on demande l'avis de ses voisins qui connaissent la nouvelle entrée (en question) et on prend l'avis majoritaire sur cette nouvelle entrée comme prédiction, i.e.

En notant $\mathcal{N}'(i)_j = \{j \in \mathcal{N}(i) \mid m_{u,j} \text{ est connu}\}$ et m son cardinal.

$$\widehat{m_{i,j}} = \begin{cases} 1 \text{ si } \displaystyle\sum_{k \in \mathcal{N}'(i)_j} m_{k,j} \geq \frac{m}{2} \\ 0 \text{ sinon.} \end{cases}$$

On abrégera par KNN le modèle des k-plus proches voisins dans le reste du document et 20-KNN pour $k = 20$.

2.2 Choix du paramètre k, de la métrique et visualisation des voisins

Le choix du paramètre k peut s'effectuer par validation croisée sur le jeu de données. Cette validation croisée a été effectuée préalablement et fournit que $k = 20$ donne de bonnes performances relativement à la racine carrée de l'erreur quadratique moyenne (RMSE).

Remarque 1 : En pratique, on peut apprendre k par recherche d'hyper-paramètres durant l'entraînement du modèle, mais on ne le fera pas en raison d'un coût temporel trop élevé.

Ensuite, pour la métrique, on utilise la similarité cosinus qui possède empiriquement de bonnes performances sur les tâches de recommandation d'après [10], dont on rappelle la définition :

$$\text{sim}(u, u') = \frac{r_u^\intercal r_{u'}}{\|r_u\|_2 \|r_{u'}\|_2}$$

où $\|\cdot\|_2$ est la norme ℓ_2.

On peut aussi procéder à une visualisation des graphes de voisins dans les figures 1 et 2.

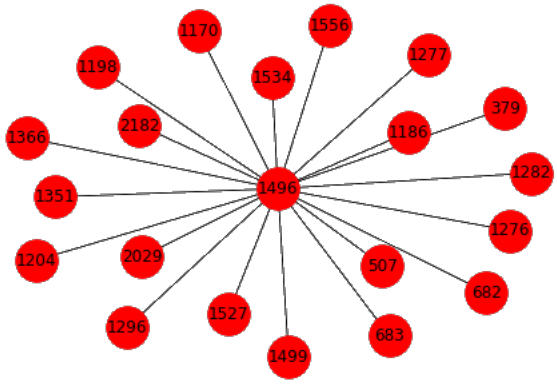

FIG. 1 : 20 plus proches voisins d'un utilisateur avec plus de 200 notes sur le jeu de données Mangaki[1]

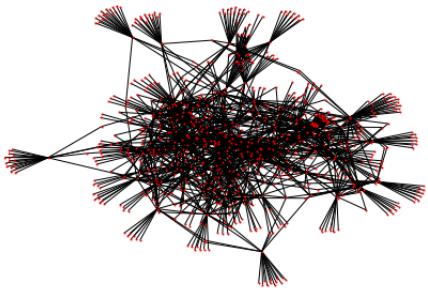

FIG. 2 : Sous-graphe des 20 plus proches voisins sur le jeu de données Mangaki entier

2.3 Défauts et limites du modèle

Comme défini en introduction, l'entraînement de 20-KNN dépend de la métrique d employée. Si on note $\text{Supp}(u)$ pour $u \in [[1, n]]$ le support des utilisateurs, défini par :

$$\text{Supp}(u) = \{j \in [[1, m]] \mid m_{u,j} \text{ est connu}\}$$

Ainsi, pour u, v deux utilisateurs tels que $\text{Supp}(u) \cap \text{Supp}(v) = \emptyset$, alors : $\text{sim}(u, v) = 0$.

Or, la situation dans laquelle l'utilisateur u a lu les versions mangas[2] d'une œuvre et v a vu les versions anime[3] de celle-ci peut se présenter, cependant la métrique n'en tient pas compte et ne peut le calculer puisqu'il s'agit d'une information propre à l'œuvre.

2.4 Objectifs du TIPE

Nous répondrons aux questions suivantes :

- Sachant qu'on dispose de l'ensemble des couvertures des œuvres, peut-on calculer une métrique qui tient compte de l'information visuelle de ces couvertures et des similarités entre les distributions d'utilisateurs ?
- En la remplaçant par la similarité cosinus, obtient-on une meilleure performance au sens d'une métrique d'erreur ?
- Est-ce qu'on constate des transferts d'information pertinents et intéressants tels que : la saison i d'une œuvre vers la saison $i + j$ de la même œuvre, du format manga vers le format anime ou vice versa ?

Ces travaux sont motivés notamment par [15] et forment un prolongement possible de cet article.

2.5 État actuel de la recherche

À notre connaissance, la littérature ne mentionne pas beaucoup de travaux qui cherche à intégrer des métadonnées visuelles dans un système de recommandation afin d'en améliorer sa qualité et son interprétabilité.

On notera : [2], [12] et

- [15] où il s'agit d'un modèle qui combine un filtrage collaboratif et ensemble de régresseurs linéaires par utilisateur afin d'apprendre des préférences visuelles dans un cadre de démarrage à froid, ce travail permet l'interprétabilité des goûts d'un utilisateur en inspectant la matrice du régresseur linéaire, travaux qui ont inspiré celui-ci ;
- [6] où il s'agit d'un modèle purement basé sur le contenu qui extrait automatiquement des représentations visuelles profondes et des métadonnées telles que le contraste afin de recommander des œuvres artistiques digitales– « artworks ».

1. Les étiquettes sont les identifiants d'utilisateurs.
2. Le format livre de l'œuvre.
3. L'adaptation animée de l'œuvre.

3 Raffinement par le transport optimal : impact de la distance de Wasserstein

Le transport optimal est un domaine qui est de plus en plus appliqué notamment grâce à [3] qui a permis le calcul effectif et approximatif des objets de façon traitable.

Au préalable, notons $\Sigma_d = \{x \in \mathbb{R}_+^d \mid \sum_{i=1}^{d} x_i = 1\}$ le simplexe de dimension d, qui peut s'interpréter de façon probabiliste comme une distribution de probabilité discrète à valeurs dans $[[1, d]]$.

Si l'on dispose de $r, c \in \Sigma_d$ deux distributions de probabilités discrètes.

On pose $U(r, c) = \{M \in \mathcal{M}_{d,d}(\mathbb{R}_+) \mid M\mathbb{1}_d = r \text{ et } M^\intercal \mathbb{1}_d = c\}$, l'ensemble des probabilités jointes sur r et c à valeurs dans $[[1, d]]^2$.

Ainsi, on définit la distance de Wasserstein comme étant :

$$\mathcal{W}(r, c) = \min_{\gamma \in U(r,c)} (\gamma \mid C)_F$$

où C est une matrice exprimant le coût de transporter de la masse de r_i vers c_j et $(\cdot \mid \cdot)_F$ est le produit scalaire de Frobenius défini par $(A \mid B)_F = \mathrm{Tr}(A^\intercal B)$ où $A, B \in \mathcal{M}_p(\mathbb{R}), p \in \mathbb{N}$.

Remarque : On rencontre aussi le nom de « Earth's Mover Distance » ou EMD pour la distance de Wasserstein, cela peut s'expliquer par l'interprétation intuitive suivante : si les distributions r, c sont des masses, au sens physique, renormalisées à 1, et que la matrice C représente des coûts par unité de masse de transporter les masses de r vers les masses de c, alors la distance de Wasserstein est le coût minimal de transport afin de transformer une masse en l'autre par déplacements successifs.

3.1 Autour de $\mathcal{W}$

$\mathcal{W}$ est bien une distance sur les distributions de probabilités (discrètes), démontrée dans [16], ce qui motive son usage en tant que métrique pour KNN.

En revanche, son temps de calcul est prohibitif, en effet, il est en $O(d^3 \log d)$ au mieux, par des variations de l'algorithme du simplexe sur un graphe, que l'on ne détaillera pas puisque cela dépasse largement le cadre de ce TIPE, d'après [7].

3.2 Intérêt : calcul efficace et rapide $\mathcal{W}$, propagation de l'information visuelle dans le modèle

On introduit $\varepsilon > 0$, un paramètre de régularisation entropique, qui se justifie pour deux raisons :

- Rendre le calcul plus rapide ;
- Rendre le plan de transport optimal le plus simple[4] possible en forçant son entropie à être faible

$$\mathcal{W}_\varepsilon(r, c) = \min_{\gamma \in U(r,c)} (\gamma \mid C)_F + \varepsilon\Omega(\gamma).$$

où $\Omega(A) = \sum_{i,j} A_{ij} \log A_{ij}$ pour $A \in \mathcal{M}_d(\mathbb{R}_+^*)$, qui représente une forme d'entropie.

De plus, le minimum est atteint pour un γ unique [8] et la solution est de la forme :

$$\forall (i, j) \in [[1, d]]^2, \gamma_{i,j} = u_i \exp(-C/\varepsilon) v_j$$

Pour $(\mathbf{u}, \mathbf{v}) \in (\mathbb{R}^d)^2$ des facteurs de dilations qu'on calcule itérativement.

C'est l'algorithme de Sinkhorn-Knopp, présenté initialement dans [13], réutilisé de façon fondamentale dans [3]. Il permet donc de calculer une approximation de $\mathcal{W}$ en temps quadratique en la dimension d.

On prouve aussi dans [3] que $(r, c) \mapsto \mathbb{1}_{r \neq c} \mathcal{W}_\varepsilon(r, c)$ est une distance par la même approche employée dans [16].

4. Lire : parcimonieux, avec le plus de zéros.

Remarque 1 : En raison de la nature itérative de l'algorithme de Sinkhorn-Knopp, on peut très facilement paralléliser le calcul des facteurs de dilatations (qui sont des vecteurs dans l'algorithme original) en des matrices lorsqu'on calcule des distances d'une distribution de probabilité vers M distributions de probabilités.

3.3 Calcul des représentations visuelles par le réseau de neurones convolutifs Illustration2Vec

En utilisant les travaux de [9] et [15], on peut calculer des représentations parcimonieuses de couvertures $(p_i)_i \in \left(\mathbb{R}^{512}\right)^m$ qui permettent de poser la matrice de coût comme étant :

$$C = \left(\|p_i - p_j\|_2^2\right)_{(i,j)\in[[1,m]]^2}$$

Donc, la matrice de coût représente la similarité visuelle entre deux couvertures, que l'on illustre entre les deux saisons de l'anime **Code Geass : Lelouch of the Rebellion** :

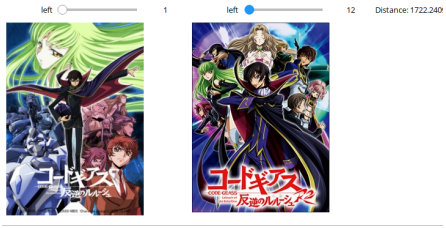

FIG. 3 : Distance entre les deux couvertures où le coût de déplacement est minimal sur toutes les œuvres du jeu de données

En vertu de cette propriété de coût minimal, la distance de Wasserstein propagera correctement l'information visuelle entre les deux couvertures, qui s'avère être l'information : « ces deux œuvres font partie du même univers et sont directement la suite ou l'antépisode l'un de l'autre ».

Le troisième objectif du TIPE **a été donc atteint**.

On appellera donc désormais 20-W-KNN le modèle 20-KNN dans lequel on remplace la similarité cosinus par une approximation de la distance de Wasserstein $\mathcal{W}$ avec la matrice C définie comme précédemment.

Le premier objectif du TIPE **a été atteint**.

3.4 Calcul des prédictions de 20-WKNN

Dans le modèle à similarité cosinus, on fait voter les voisins afin de calculer une prédiction. En revanche, puisqu'on manipule des distributions de probabilités, on ne peut plus procéder à la détermination de la classe majoritaire.

Au lieu de cela, on calcule le barycentre des distances de Wasserstein, pour un utilisateur u donné :

$$v = \frac{1}{\mathbf{card}\,\mathcal{N}(u)} \operatorname{argmin}_{v\in\Sigma_m} \sum_{u'\in\mathcal{N}(u)} \mathcal{W}(v, u')$$

Remarque 1 : Ce calcul peut être encore effectué rapidement par l'algorithme de Sinkhorn-Knopp, d'après [4].

Ainsi, on connaît le germe de probabilité :

$$(\mathbb{P}(m_{u,j} = 1))_{j\in[[1,m]]} = (v_j)_{j\in[[1,m]]}$$

À partir de cela, on peut opter pour une méthode à base de seuil, on fixe un paramètre $\alpha \in [0, 1]$ et on pose :

$$\widehat{m_{u,j}} = \begin{cases} 1 \text{ si } \mathbb{P}(m_{u,j} = 1) \geq \alpha \\ 0 \text{ sinon.} \end{cases}$$

On appelle α seuil de discrimination.

Remarque 1 : Ce paramètre peut être appris par recherche d'hyper-paramètres pendant l'entraînement du modèle.

Remarque 2 : Il peut être rendu dépendant de l'utilisateur, lors de l'entraînement en déterminant le profil d'un utilisateur. Par exemple, une approche naïve consisterait à dire que plus un utilisateur a de préférences négatives, plus le seuil de discrimination sera élevé, et vice versa.

Remarque 3 : Si les α sont dépendants des utilisateurs, on peut calculer un $\overline{\alpha}$ moyen que l'on peut employer pour une prédiction sur un nouvel utilisateur qui n'était pas présent dans la phase d'entraînement.

4 Résultats

Les code des expériences sont fournies sur le référentiel GitHub : https://github.com/mangaki/hiyajo-ot et reproductibles.

Le matériel employé pour l'expérience est un serveur muni d'un Intel(R) Atom(TM) CPU C2750 @ 2.40GHz à 8 cœurs et 16 Gio de RAM.

Plusieurs implémentations de référence seront réutilisées directement plutôt que de les réécrire car leur (ré)-implémentation ne concerne pas le fond de ce TIPE, on utilisera notamment NumPy, SciPy, NetworkX, IPyParallel et enfin POT [5] qui fournit des implémentations de l'algorithme de Sinkhorn.

4.1 Évaluation de l'erreur de recommandation : courbe ROC

Une courbe ROC est un graphique exprimant le taux de vrai positifs (TPR) par rapport au taux de faux positifs (FPR). On calcule l'ensemble des seuils de discriminations possibles et on trace une courbe ROC montrant ces seuils et on peut calculer l'aire sous la courbe, ce résultat est indépendant des seuils de discrimination. Ce qui rend le choix de cet outil intéressant puisque 20-WKNN dépend d'un seuil de discrimination pour effectuer des prédictions. Ce résultat est pris comme l'erreur de recommandation que l'on note en tant que « AUROC » pour « Area Under Receiving Operating Characteristics ».

AUROC	Ensemble de test
KNN	0.514
W-KNN	**0.625**

Fig. 4 : 5-fold où l'on s'assure que les classes sont équilibrées, répété 3 fois avec un seed de 42

Le second objectif du TIPE **a été atteint**, on constate que W-KNN a une meilleure performance que KNN de façon stable et reproductible.

5 Prolongements envisagables

Après avoir atteint tous les objectifs fixés par le TIPE, nous n'avons pas discuté des différences entre les temps d'entraînement et de prédiction de KNN et W-KNN. En l'état, W-KNN est 100 fois plus lent à entraîner que KNN malgré l'algorithme de Sinkhorn, cela s'explique que les expériences n'ont pas exploité le caractère hautement parallélisable de cet algorithme afin de faire diminuer les temps de calcul.

Cependant, une alternative est envisageable par [1], un algorithme quasi linéaire de calcul des distances approximatifs de Wasserstein est possible, et mis en place sous le nom de **Greenkhorn** dans la librairie POT, cependant il est très sensible aux erreurs numériques et n'est pas conçu pour la parallélisation automatique. Durant les essais préliminaires, aucune renormalisation n'a été fructueuse.

De plus, l'algorithme de Sinkhorn possède une version stabilisée qui fonctionne selon le principe décrit dans [11] mais Greenkhorn n'en possède aucune, il serait intéressant de contribuer à POT afin d'ajouter le support pour un tel schéma de calcul numérique et de résoudre le problème décrit ici : https://github.com/rflamary/POT/issues/54 par la même occasion.

Ces travaux feront l'objet d'une publication scientifique plus tard.

Réferences bibliographiques

[1] Altschuler, J., Weed, J. et Rigollet, P. 2017. Near-linear time approximation algorithms for optimal transport via Sinkhorn iteration. *Advances in Neural Information Processing Systems* (2017), 1964-1974.

[2] Chu, W.-T. et Tsai, Y.-L. 2017. A hybrid recommendation system considering visual information for predicting favorite restaurants. *World Wide Web*. 20, 6 (2017), 1313-1331.

[3] Cuturi, M. 2013. Sinkhorn distances : Lightspeed computation of optimal transport. *Advances in neural information processing systems* (2013), 2292-2300.

[4] Cuturi, M. et Doucet, A. 2014. Fast computation of Wasserstein barycenters. *International Conference on Machine Learning* (2014), 685-693.

[5] Flamary, R. et Courty, N. 2017. POT Python Optimal Transport library.

[6] Messina, P., Dominguez, V., Parra, D., Trattner, C. et Soto, A. 2019. Content-based artwork recommendation : integrating painting metadata with neural and manually-engineered visual features. *User Modeling and User-Adapted Interaction*. 29, 2 (2019), 251-290.

[7] Pele, O. et Werman, M. 2009. Fast and robust earth mover's distances. *2009 IEEE 12th International Conference on Computer Vision* (2009), 460-467.

[8] Peyré, G., Cuturi, M. et others 2019. Computational optimal transport. *Foundations and Trends in Machine Learning*. 11, 5-6 (2019), 355-607.

[9] Saito, M. et Matsui, Y. 2015. Illustration2vec : a semantic vector representation of illustrations. *SIGGRAPH Asia 2015 Technical Briefs* (2015), 5.

[10] Sarwar, B.M., Karypis, G., Konstan, J.A., Riedl, J. et others 2001. Item-based collaborative filtering recommendation algorithms. *Www*. 1, (2001), 285-295.

[11] Schmitzer, B. 2019. Stabilized sparse scaling algorithms for entropy regularized transport problems. *SIAM Journal on Scientific Computing*. 41, 3 (2019), A1443-A1481.

[12] Shah, R.R., Samanta, A., Gupta, D., Yu, Y., Tang, S. et Zimmermann, R. 2016. PROMPT : Personalized User Tag Recommendation for Social Media Photos Leveraging Personal and Social Contexts. *2016 IEEE International Symposium on Multimedia (ISM)* (déc. 2016), 486-492.

[13] Sinkhorn, R. 1967. Diagonal equivalence to matrices with prescribed row and column sums. *The American Mathematical Monthly*. 74, 4 (1967), 402-405.

[14] Vie, J.-J., Laıly, C. et Pichereau, S. 2015. Mangaki : an anime/manga recommender system with fast preference elicitation. *Tech. Rep.* (2015).

[15] Vie, J.-J., Yger, F., Lahfa, R., Clement, B., Cocchi, K., Chalumeau, T. et Kashima, H. 2017. Using posters to recommend anime and mangas in a cold-start scenario. *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)* (2017), 21-26.

[16] Villani, C. 2008. *Optimal transport : old and new*. Springer Science & Business Media.