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 exprimés par des utilisateurs, l'on veut une méthode de prédire les futures préférences des utilisateurs, c'est le problème qu'on essayera de résoudre.

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, l'on veut apprendre un classificateur capable de prédire $\widehat{m_{i,j}}$ si celui-ci n'est pas connu, avec, 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 [7] qui comporte :

- 2289 utilisateurs ;
- 12479 œuvres issus de l'animation japonaise (animes, 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, calculer des voisins, qu'on notera $\mathcal{N}(i) \subset [[1, n]]^k$ au sens d'une métrique $d : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}_+$ opérant sur les distributions d'utilisateurs, prendre les k plus proches
- Pour classifier une nouvelle entrée pour un utilisateur i, on fait « voter » les voisins de i et on prend la majorité 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.

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

On abrégera KNN pour 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 effectuement préalablement et fournit que $k = 20$ donne une bonne performance relativement à la racine carrée de l'erreur moyenne au carrée (RMSE).

Remarque 1 : En pratique, on peut apprendre k par recherche d'hyper-paramètres durant l'entraînement du modèle, mais ceci ne serait pas fait en raison du coût en complexité.

Ensuite, pour la métrique, on utilise la similarité cosinus qui possède de bonnes performances empiriquement sur les tâches de recommandation d'après [5], 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 :

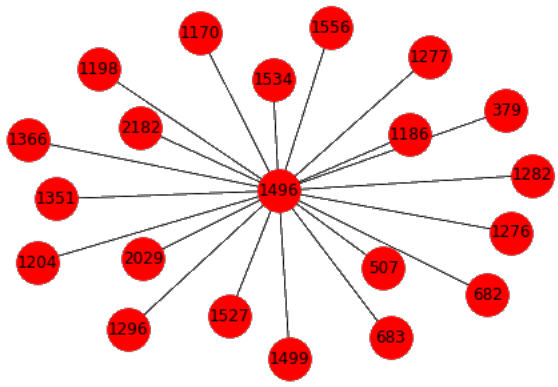

FIG. 1 : 20 plus proches voisins d'un utilisateur avec plus de 200 notes sur le jeu de données Mangaki

2.3 Défauts et limites du modèle

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

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

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

Or, la situation dans laquelle l'utilisateur u a lu les versions mangas d'une œuvre et v a vu les versions animes 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 t'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 t'on un 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 [8] et forme 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 chercher à intégrer des métadonnées visuelles dans un système de recommandation afin d'en améliorer sa qualité et son interprétabilité, on notera [8] où il s'agit d'un modèle qui combine un 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, [3]

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 [1] qui a permis le calcul effectif et approximatif des objets de façon tractable.

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

3.1 Propriétés de $\mathcal{W}$

– $\mathcal{W}$ est bien une distance sur les distributions de probabilités (discrètes), démontrée dans [9] ; —

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

Par l'algorithme de Sinkhorn-Knopp, présenté initialement dans [6], présenté en détails dans [1], il est possible de calculer une approximation de $\mathcal{W}$, pour $\varepsilon > 0$, un paramètre de régularisation entropique :

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

On prouve aussi dans [1] que $(r,c) \mapsto \mathbb{1}_{r\neq c}\mathcal{W}_\varepsilon(r,c)$ est une distance par la même approche employée dans [9].

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

En utilisant les travaux de [4] et [8], 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** :

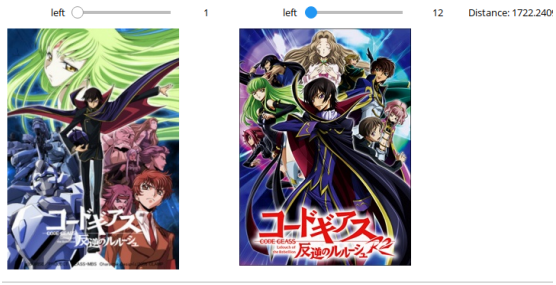

FIG. 2 : Distance entre les deux couvertures où le coût de déplacement est minimal sur toutes les œuvres du jeu de données

On appellera donc désormais 20-WKNN 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.

Ceci **remplit** le premier objectif du TIPE.

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, pour un utilisateur u donnée :

$$v = \text{argmin}_{v\in\Sigma_m} \sum_{u'\in\mathcal{N}(u)} \mathcal{W}(v,u')$$

Ainsi, on connaît la famille de probabilités :

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

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 prédiction.

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.

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 reproducibles.

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 réutilisés 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 [2] qui fournit des implémentations de l'algorithme de Sinkhorn.

4.1 AUC

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

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

4.2 Analyse qualitative

5 Prolongements envisagables

Réferences bibliographiques

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

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

[3] 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.

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

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

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

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

[8] 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.

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