Intégration de données hétérogènes

Arthur Imbert

Octobre 2017

Le stage s'est déroulé au sein de l'équipe Parietal à l'INRIA. Mon superviseur était Gaël Varoquaux.

1 Introduction

De nos jours, le volume de données s'est considérablement accru. Chaque organisation exploite et communique des données en utilisant ses propres règles. La plupart du temps, ces fichiers sont créés avec une finalité bien précise, prenant en compte les spécificités et les contraintes internes au producteur. Cela rend accessible en ligne un volume important de données. Souvent comparables du fait de leur format, elles restent extrêmement hétérogènes par leur structure et leur contenu. Ainsi, cette hétérogénéité complique l'intégration des fichiers issus de sources différentes. Il y a autant de manières de stocker et de structure des données qu'il y a de sources de données. En outre, l'enthousiasme actuel autour de la *data science* et de ses modèles prédictifs révèle un potentiel important en faveur du croisement de données pertinentes.

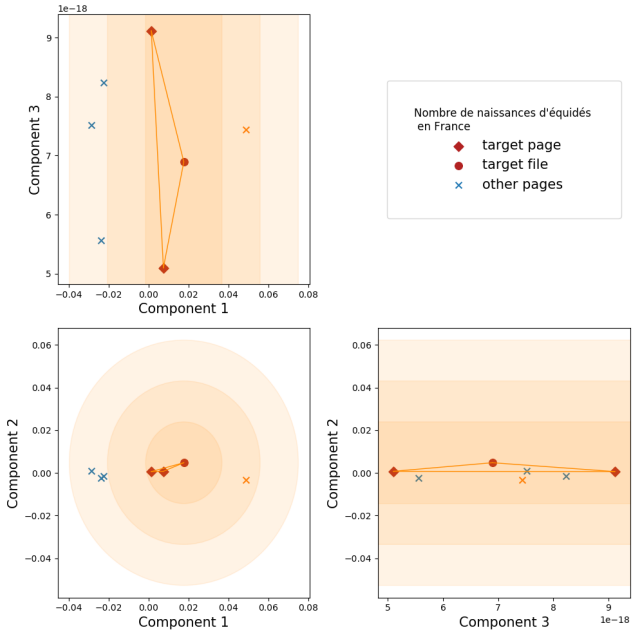
Ce travail vise à intégrer de telles données hétérogènes. Cela implique à la fois une capacité de reconstruire et de nettoyer des tables de données, mais également une méthode pour révéler de potentielles connexions entre différentes tables. Notre objectif est de construire un *file embedding* (un espace vectoriel) où chaque base de données est représentée par un vecteur. Cela implique, entre autres choses, l'utilisation de modèles d'extraction de *topics* [1] et de *metric learning* [2]. En calculant des distances au sein de notre espace vectoriel, nous pouvons automatiquement suggérer des connexions entre les bases de données. Finalement, à partir d'une requête sur un fichier, nous avons développé un programme qui permet de recommander des bases de données intéressantes à croiser.

Afin de collecter un volume suffisant de données hétérogènes, nous avons décider d'exploiter l'*open data*. Ce sont des données auxquelles nous avons librement accès. Nous pouvons les utiliser, les réutiliser et les redistribuer sans aucune restriction de licence. Elles peuvent prendre la forme de données textuelles ou non, de tables, de données géographiques, etc.

Un site web français répertoriant une grande partie de *open data* national existe. Il affiche toutes les informations pertinentes pour utiliser ces fichiers issus des administrations publiques et des collectivités territoriales : data.gouv.fr. Cette plateforme indexe notamment plusieurs bases de données ainsi que leur source. Néanmoins, ce n'est pas à proprement parlé du *big data*. La plupart des bases de données, bien qu'ayant un potentiel statistique important, restent de taille moyenne avec des structures hétérogènes (valeurs manquantes, fautes d'orthographe, commentaires, format non tabulaire). Par conséquent, chaque analyse utilisant ces données demanderait un temps non négligeable pour les préparer. Nous voulons réduire ce coût en développant des algorithmes généraux pour nettoyer ces données.

En outre, data.gouv.fr liste les réutilisations réalisées à partir d'une ou plusieurs bases de données ouvertes. Une réutilisation peut, par exemple, prendre la forme d'une carte représentant différents services publics, d'une analyse socio-économique ou d'une visualisation des résultats de élection présidentielle. Ainsi, nous entendons répondre à notre problématique d'intégration de données en nous concentrons sur la prédiction de réutilisations de fichiers. Cela nous permet de nous situer dans un cadre supervisé avec des données labelisées (deux fichiers ont été réutilisés ou non). La prediction de réutilisation devient un prétexte pour valider nos méthodes.

Le graphique 1 illustre le type de résultat que l'on cherche à généraliser. A partir d'un *file embedding* et d'une requête sur un fichier, nous calculons des distances pour retrouver les plus proches voisins. Plus proches sont les fichiers, plus pertinente devrait être la réutilisation entre ces fichiers. Les trois fichiers avec une réutilisation existante sont des tables de données sur la naissance des chevaux. Ici, il est suggérer de les croiser avec trois autres bases traitant de la saillie de juments.



1

Figure 1 – Suggestion de connexions potentielles

Note : Une vue partielle en 3 dimension du *file embedding* est représentée (ACP). Le fichier requêté est représenté par le cercle rouge. Tous les autres fichiers voisins sont des connexions potentielles. Les bases de données concernées par une réutilisation déjà existante sont reliées par un trait orange.

2 Résultats et validation

Depuis la plateforme data.gouv.fr nous récupérons les métadonnées de 26856 pages différentes. Chaque page vient d'un unique producteur et traite d'un sujet en particulier. Généralement, une page présente plusieurs bases de données, chacune ayant leur propre URL pour lancer leur téléchargement. Sur les 96629 URL disponibles, seuls 71368 sont directement utilisables. Nous avons téléchargé 55138 fichiers (soit 348Go) et nous avons nettoyé 23112 d'entres eux (soit 72Go) issus de 9092 pages uniques. Parmi eux, 74% n'ont pas été réutilisés, 4% l'ont été une seule fois et 20% à deux reprises.

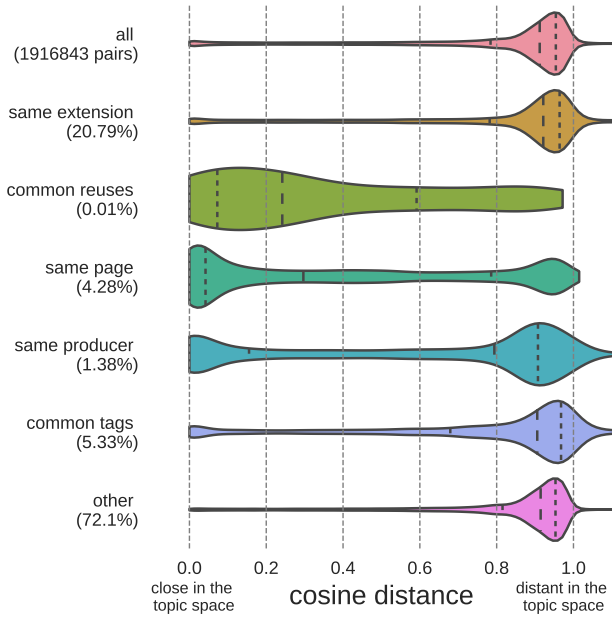
Le reste du programme nécessite l'optimisation de différents paramètres. Le processus de *metric learning* a également besoin d'être validé. Par conséquent, nous avons défini une méthode de validation croisée. Nous séparons nos données et nous validons sur l'*embedding* de test ce que nous avons appris avec l'*embedding* d'entraînement. Une fois paramétré, nous construisons notre *file embedding* avec 25 dimensions (ou *topics*) à l'aide d'un algorithme de Non-negative Matrix Factorization (NMF) et de Least Square-residual Metric Learning (LSML).

Dans le graphique 2, nous représentons la distribution de la distance cosinus pour différentes paires de fichiers. Premièrement, nous observons que les fichiers avec la même extension d'origine ne semblent pas être plus rapprochés dans l'espace. L'extension d'origine n'aurait pas d'impact significatif dans la construction de notre *embedding*. Cela valide en partie notre étape de nettoyage des données où l'on récupère des tables à partir de fichiers hétérogènes. Enfin, le graphique nous permet d'observer une plus grande variance dans la distance entre fichiers réutilisés. Ces paires semblent également relativement plus proches au sein de notre *file embedding*.

Un autre moyen d'analyser notre *embedding* est de représenter des nuages de mots pour chaque *topic*. Le poids de chaque mot dans un *topic* détermine sa taille dans le graphique.

Nous représentons trois nuages de mots dans le graphique 3. Celui de gauche rassemble des mots issus du champ lexical géographique des régions françaises. Celui du milieu est probablement issus des rapports de police. Il comprend pour l'essentiel des termes liés à des délits ou à des crimes. Enfin le troisième nuage de mots, à droite, mélange le champ lexical du budget avec celui de l'administration.

Pour chaque base de données, nous récupérons ses voisins situés dans un rayon fixé. Nous observons que 7663 des 23112 base de données ont au moins une base de données voisine dans le *file embedding*. Si nous représentons graphiquement leur distribution (graphique 4), nous observons une forme déséquilibrée. Mis à part un nombre



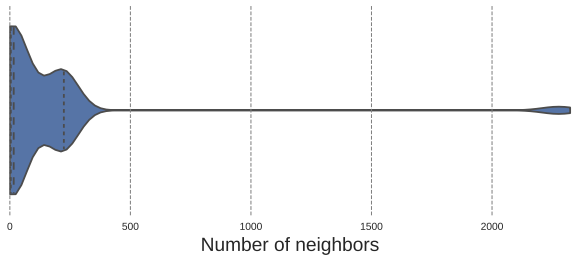
2

Figure 2 – Distribution de la distance cosinus au sein du *file embedding*

(a) Champ lexical de la région **(b) Champ lexical de la criminalité et de la police** **(c) Champ lexical du budget et de l'administration**

3

Figure 3 – Nuages de mots portant sur trois *topics*



4

Figure 4 – Distribution du voisinage

Note : Seuls les fichiers avec au minimum un voisin sont pris en compte dans cette distribution.

limité de voisinages avec plus de 2000 fichiers, les autres contiennent généralement moins de 250 fichiers. En moyenne, chaque voisinage comprend 143 fichiers et 69 pairs réutilisées.

3 Discussion

Cette étude utilise des données complexes, des méthodes inhabituelles et entend répondre à une problématique ambitieuse. Nous n'avons pas cherché à télécharger et à nettoyer l'ensemble des données répertoriées par data.gouv.fr. Notre but est surtout de rassembler assez de base de données pour obtenir des résultats statistiques significatifs. Pour autant, des données supplémentaires pourraient être collectées depuis des sources variées. De plus, 1711 réutilisations sont listées sur la plateforme, mais 1390 d'entres elles (81%) n'impliquent qu'une seule base de données. Durant notre processus de *metric learning*, nous avons du sous échantillonner les paires de fichiers non réutilisées. Dès lors, la Confusion entre réutilisation et similarité s'accroit. Les fichiers suggérés sont alors quasi identiques. Cela est pertinent mais une dimension importante de la réutilisation est alors écartée : croiser des fichiers strictement différents, mais néanmoins complémentaires. Par conséquent, nous passons certainement à côté de réutilisations intéressantes.

Cette étude nous ouvre plusieurs perspectives. Premièrement, nous pouvons attacher plus d'importance à l'analyse des fichiers géographiques issus notamment de l'extension GEOJSON. Comme nombre d'entres eux sont proches dans notre *embedding*, être capable d'estimer les zones géographiques dont il est question nous aiderait à les discriminer. Deuxièmement, il serait possible d'agréger automatiquement plusieurs fichiers une fois détecté un index commun entre eux. Enfin, l'ensemble de notre processus pourrait être intégré dans une interface utilisateur comme data.gouv.fr.

4 Conclusion

Intégrer des bases de données hétérogènes peut mener dans des directions très variées. Nous avons choisi d'adopter une approche statistique à ce problème.

La première tâche est de construire un corpus de fichiers pertinent et assez volumineux pour s'assurer des résultat statistiques significatifs. Cela nous donne l'opportunité de développer des algorithmes pour lire, nettoyer et reformater ces nombreux fichiers, à grande échelle. La seconde tâche est de construire un *file embedding* afin de pourvoir facilement calculer des distances entre les bases de données et suggérer des connexions, ou des similarités pertinentes. Pour s'intégrer dans un cadre d'apprentissage, nous avons défini une tâche pretexte : suggérer des réutilisations entre les fichiers de notre corpus en fonction de leur proximité dans l'*embedding*.

Références

[1] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature. 401*, page 788–791, 1999.

[2] E. Y. Liu, Z. Guo, X. Zhang, V. Jojic, and W. Wang. Machine learning from relative comparisons by minimizing squared residual. 2012.