Integration of heterogeneous datasets

Arthur Imbert

October 2017

The internship took place in the Parietal team at INRIA. My supervisor was Gaël Varoquaux.

1 Introduction

Nowadays, the production of data has considerably grown. Each organization exploits and issues data, using its own schema. Most of the time, these files are produced for specific purpose, taking into account some internal rules or constraints. That leads to a vast amount of data available online, often sharing the same file formats, but still deeply heterogeneous by their structure. Indeed, this heterogeneity complicates the integration of files issued from different sources. There are as many different ways to store and structure data as there are producers. By the same time, the current enthusiasm around data science and its predictive models shows a high potential for crossing relevant data.

This paper aims to integrate such heterogeneous datasets. This involves both an ability to reshape and clean files and a method to reveal potential connections between them. Our goal is to build a file embedding (a vectorial space) where each dataset is represented with a vector. It involves, among other things, the use of topics extraction [1] and metric learning models [2]. By computing distances within this vectorial space, we want to automatically suggest connections between relevant datasets. Finally, from a query file, we design a pipeline to recommend some potential datasets to cross.

In order to collect a vast amount of heterogeneous data we exploit open data. It is freely available to everyone. We can use, reuse or redistribute it without any patent or copyright restrictions. Open data may includes textual and non-textual material, tables, geographical data, etc.

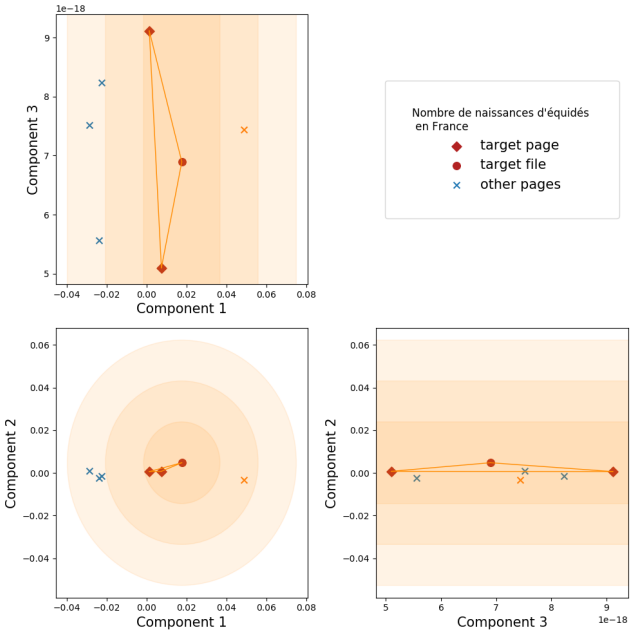
A French open data platform exists. It lists all the information needed to use files issued by the French public organizations: data.gouv.fr. This website hosts and indexes multiple datasets and give information about their producer. However, it is not Big data. Most of the datasets have a huge statistical potential, but a small or medium size and an extremely heterogeneous structures (missing values, misspelling, commentary, non structured format). Therefore, each analysis using these datasets would need laborious data manipulation and normalization. We want to reduce this cost and apply a general algorithm to clean datasets.

Additionally, data.gouv.fr lists reuses that have been done exploiting one or several open datasets. For example, a reuse can be a map of public services, a visualization of presidential election results, or any statistical analysis crossing socio-economics data. Thus, we expect to answer our integration problem by predicting reuses between files. It allows us to transform our problem as a supervised one and use a statistical learning framework. Existing reuses gives us labeled data and help us to learn new connections between files. Prediction of reuses becomes a pretext task to validate our methods and settings.

Figure 1 illustrates the typical result we are looking for in this paper. From a file embedding, we compute distances to reveal the closest neighbors from a query file. Closer they are, more relevant a reuse should be. The three reused files (connected with orange lines) include data about births of horses. Our embedding suggests to cross them again with three other datasets dealing with mare mating.

2 Results and validation

From the platform data.gouv.fr we collect the metadata of 26856 different pages. Each page comes from an unique producer and deals with a specific topic. Usually, one page hosts several files with their respective URL to download them. We manage to load 55138 files and store 348GB. We also manage to clean 23112 files (for 72GB) coming from 9092 different pages. Among them, 17147 (74%) have not been reused, 898 (4%) have been reused once and 4511 (20%) twice.



1

Figure 1: Suggestion of potential connections

Note: This is a partial 3-dimensional view (PCA) of our final file embedding. The query file is represented with the red circle. All the other points are files with a potential connection. Files with an actual connection are linked by an orange line.

The rest of the pipeline includes different parameters we need to set up. The metric learning process has to be evaluated too. Hence, we define a cross-validation framework (splitting our dataset to validate on test embedding what we learned from the train embedding). Once we find our optimal parametrization, we compute our file embedding with 25 dimensions (or topics) using Non-negative Matrix Factorization (NMF) and Least Square-residual Metric Learning (LSML) algorithms.

In the figure 2, we plot the distribution of cosine distances within our file embedding. It first shows that files from same extension do not appear to be closer in the embedding. The original extension doesn't have an impact on the embedding. It validates a part of our cleaning process where we return homogeneous dataframes from heterogeneous files with various format. Finally, figure 2 shows a higher variance for distances between files reused together. Yet, most of these pairs still present a relative closer distance than what we observe in the rest of the topic space.

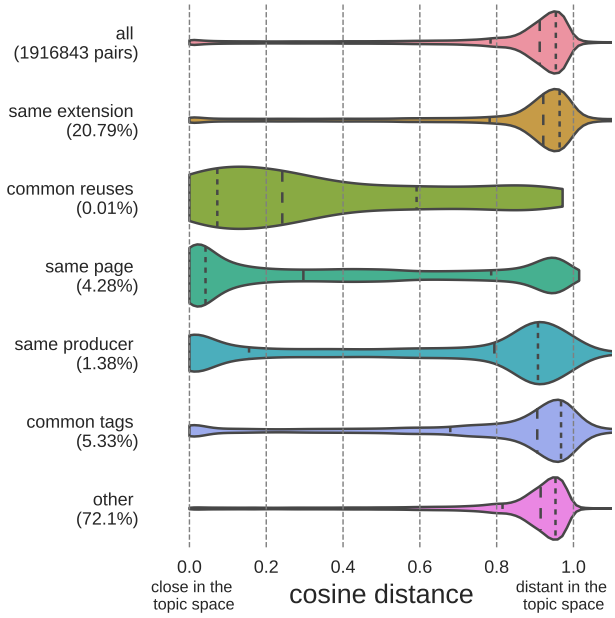
An other way to analyze our topic space is to plot wordclouds for specific dimensions (or topics). The weight of every word in a topic determines its size in the plot.

We present three examples of homogeneous topics in the figure 3. The left one gathers quasi exclusively geographical words indicating well known locations in France. The second one probably comes from a group of police reports. It mostly includes vocabulary about crimes, thefts or burglaries. The third example, on the right, shows a mixed vocabulary related to budget and bureaucracy concepts.

For each file, we return its neighbors within a specific radius in the embedding. We notice than 7663 of the 23112 files have at least one neighbor. If we plot their distribution (see figure 4), we observe a unbalanced shape. Most of the files with a neighborhood have less than 250 neighbors. On the top of the distribution, a group of files seems to share more than 2000 neighbors. Each neighborhood contains, in average, 143 neighbors and 69 reused pairs.

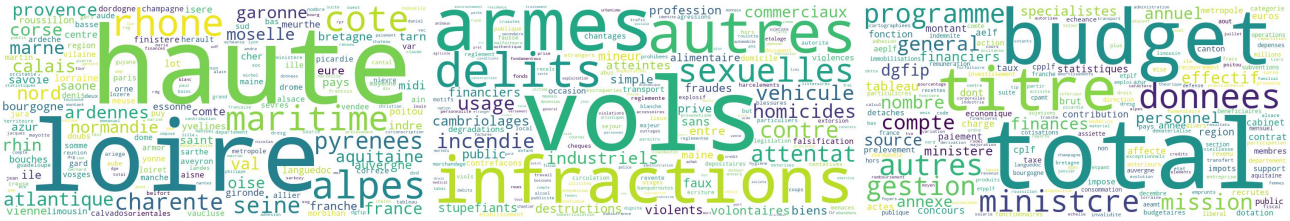
3 Discussion

This study involves complex data, unusual methods and ambitious problematic. We do not manage to download and clean all the data listed in data.gouv.fr. Our goal is simply to gather enough data to infer the most relevant statistical results. But additional datasets could be collected from diversified sources. Moreover, there are 1711



2

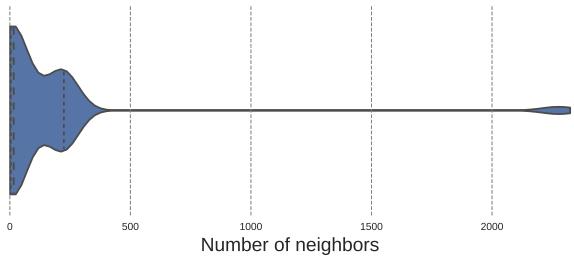
Figure 2: Cosine distance distribution in the best file embedding



(a) Region topic (b) Police and criminality topic (c) Budget topic

3

Figure 3: Three wordcloud topics



4

Figure 4: Neighbors distribution

Note: Only the file with at least one neighbor are considered for this distribution.

reuses listed in the platform, but 1390 of them (81%) concern only one page of data. During our learning process, we even had to undersample the non reused pairs of files. This makes reuses less relevant to learn how to cross files from different origins. With a majority of reuses concerning only one page fo data, the correlation between reuses and similarity increases. Our embedding learns that a reuse associates most of the time quasi identical files. It's relevant but it denies an interesting dimension of the reuse: crossing two files from completely different, but nonetheless complementary topics. Therefore, we certainly miss some reuse opportunities.

This study gives us several prospects. Firstly, we can focus on the geographical data and especially on the GEOJSON extension. As numerous geographical files are close from each other, being able to infer the geographical area concerned by their data would help us to discriminate them. Secondly, it would be possible to automatically merge several files once we recognized a common id columns between them. Lastly, our pipeline could be integrated to an user interface as data.gouv.fr

4 Conclusion

Integrate heterogeneous datasets can bring us to various directions. We choose to approach the problem with a statistical point of view and machine learning tools.

The first task is the building of a relevant dataset, big enough to ensure statistical results. It gives us the opportunity to develop algorithms in order to parse, clean and reshape numerous files on a large scale. The second task is the building of a file embedding, in order to easily compute distances between files and infer relevant connections. To fit with a learning framework, we define a pretext task: the suggestion of reuses between files.

References

[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.