Clustering open banking transactions

Juan Barragan

Abstract
In this note we explain the methodology which is used for clustering banking transactions. Our goal is that giving a list of bank statements, consisting of short text sentences, to group them into near classes and then to look for recurrent patterns.

Contents

1. Introduction

At Acme Corp, we finance digital companies and we do so by assessing the risk on a lending position using their data. The data we look at is mainly composed of their growth and advertising metrics as well as their banking transactions. One goal of this note is to look at the banking statement seeking for regular patterns. We make the hypothesis that a healthy business has necessarily a healthy and regular set of banking transactions.

There's a number of tools for studying text on the literature. Popular models use *Natural Language Processing* techniques. Yet our particular setting is not exactly the natural language. Instead, our texts are composed by a complicated combination of keywords used by banks combined with some explaining text, dates and a lot of times, numbers.

There is no semantic structure on all this, not at least as we are used to in natural languages like plain English. In plain English there is grammar, there is syntax, there is lexical structure. On banking transaction there is not such thing.

So we cannot exploit the nice and huge existing corpora. Yet we are going to proceed similarly.

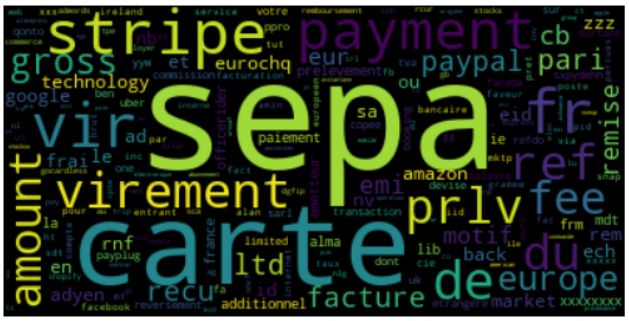


Figure 1. Word Cloud with all the bank transactions

On figure 1 we can have a look to the *Word Cloud* image generated with all the words on the bank descriptions.

2. Neural Networks

A *One Layer Neural Network* is a parametrized function between vector spaces, $f_\theta : \mathbb{R}^c \to \mathbb{R}^d$, where θ can be expressed as follows, let M be the matrix

$$M = \begin{pmatrix} \theta_{1,1} & \dots & \theta_{1,d} \\ \vdots & & \\ \theta_{c,1} & \dots & \theta_{c,d} \end{pmatrix} = (\theta_1, \dots \theta_d)$$

The θ_i are column vectors. Let σ a non-linear function (We often use sigmoids: $\sigma(t) = 1/(1+e^{-x})$), then f_θ is expressed as

$$f_\theta(x) = (\sigma(x^t \cdot \theta_1), \dots, \sigma(x^t \cdot \theta_d))$$

Here x^t is the transpose of the vector x and $\cdot$ the dot product. Neural networks are useful for regression as well as classification problems. Say we have sample observed data $(X_i, Y_i) \quad i = 1, \dots s$ we look at the *loss* function over all the sample:

$$L(\theta) = \sum_{j=1}^{s} \|f_\theta(X_j) - Y_j\|^2$$

We get a regressor when we solve for the set of parameters θ which minimizes the observed error:

$$\arg\min_\theta L(\theta)$$

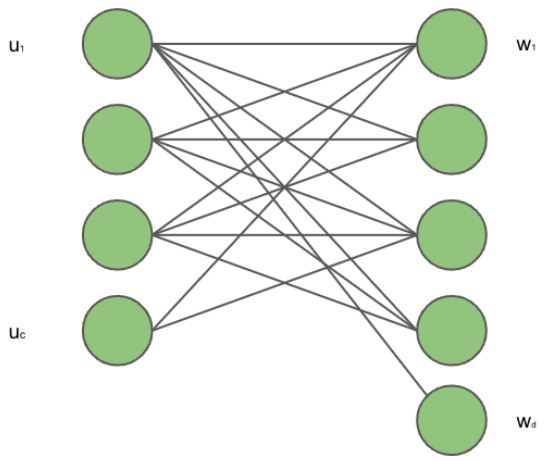


Figure 2. Neural Network, here $W_i = \sigma(x^t \cdot \theta_i)$

Is customary to represent Neural networks as on the graph show on figure 2

Layers can be added by function composition,

$$f(x) = f_{\theta_1}(f_{\theta_2}(\dots f_{\theta_p}(x)))$$

we talk of *Deep Neural Network* and the task of classification/regression as *Deep Learning*. You can check [3] for an introduction to neural networks.

3. Word Embeddings

We cannot do Mathematics with words, at least not in a metric way. Using Neural Networks we are going to make *Word Embeddings* using an idea coming from *autoencoders*. The technique is called Word2Vec and it was introduced by T. Nikolov et al. at google, see [4]. First let $V = \{w_i | i = 1, \dots, v\}$ a vocabulary composed by v words. Each word is simply an arbitrary concatenation of chars $w = a_1 a_2 \dots a_k$. We embed each word using for example the *One Hot Encoding*, namely each word w_i is mapped to a vector $w = (\delta_i)$ with the entries all equal to zero except the i-th one.

A bank description is a *context*. Write $\mathscr{C}$ for all the contexts. We would like to measure how frequent adjacent words appear. Say how often a word w_i appears with word w_k on a given context. More precisely, we define (a two layers) Neural Network with input and output as follows:

$$(\delta_i) \to \sum_k (\delta_k) \times \frac{\text{count}_{C \in \mathscr{C}}(w_k, w_i \in C)}{\#\mathscr{C}}$$

As this is two layers Neural Network, there are inside the network, two matrices, M and N. Matrix M has dimensions $v \times d$ and N has $d \times v$, v is of course the size of the vocabulary and d is an arbitrary number known as the *dimension* of the embedding. The *Word Embedding* is defined as $w \to w^t M$. Which simply gives for the i-th word $w_i = (\delta_i)$ the i-th column of the matrix M. This model is known as the *Continuous Bag Of Words* or CBOW (With two word context or bigram)

In our exploratory work, we have found that *skipgrams* where more effective. The idea of a skipgram is that for a given word w, what is the probability of having that word in the middle of a given context of length $2\kappa + 1$, say $C_\kappa(w) = [w_{-\kappa}, \dots, w, \dots w_\kappa]$?

In that case the Neural Network has the form:

$$(\delta_i) \to (\delta_i) \sum_{\kappa=1}^{c} \frac{\text{count}_{C_\kappa \in \mathscr{C}}(w_i \in C_\kappa)}{\#\mathscr{C}}$$

In both cases, CBOW and skipgrams models, what we have to keep in mind is the following. The embedding, given the evidence, is looking for the optimal factor allowing for the greatest probability of finding words on a given context. And that's exactly what we are looking for.

Some final words concerning our particular setting. For our study case, we proceed as follows

Clean bank descriptions of accents and extraneous chars.

Tokenize the bank statement (split them by spaces)

Choose the longest context as the dimension of the embedding

Embed a full context C as the mean of its embedding:

$$e(C) = \frac{1}{\#C} \sum_{w \in C} e(w)$$

where $e(w)$ is the individual word embedding.

We hence have now a mapping $i : \mathbb{B} \to \mathbb{R}^d$ from the set of banking transactions to a real vector space such that words which are *frequently* present on similar bank statement contexts are deemed to be similar and are translated to vectors which are *metrically* closer. This former affirmation is called *The Distributional Hypothesis*, you can have more details about this methodology on [2].

4. Clustering

By far the most well known algorithm for clustering is the k-means. Basically given a number k we look for k centers such that the energy required by an arbitrary point for going to the closest center would be minimum. The caveat with k-means is that we need to know in advance the number of centers. There are a number of methods for determining it, empirical, graphical and force brute with a generalized Gini coefficient as a measure of how well classified the points are. None of them where satisfying for us.

Instead, what gave us results was an agglomerative clustering. A clustering is an algorithm for partitioning a set of point $S = \{x_i, i = 1, \dots n\}$ into subsets $S_i, i = 1, \dots m$ such that $S = \cup_{i=1}^m S_i$ and $S_i \cap S_j = \emptyset$ for all i, j $i \neq j$.

An *Agglomerative Hierarchical Clustering* start bottom up: initially the set of points S is partitioned trivially: each subset contains only one point, $S_i = \{x_i\}$, then we proceed to merge two subsets using a *linkage* function, we use *Ward*'s linkage function which for a given set S measures the deviation from its centroid:

$$ESS(S) = \sum_{i=1}^{n} \|x_i - \frac{1}{n}\sum_{j=1}^{n} x_j\|^2$$

Given two sets S and T, the Ward's linkage, the distance between two clusters is given By

$$D(S,T) = ESS(S \cup T) - [ESS(S) + ESS(T)]$$

We calculate the function D among all the subsets, and we choose to merge the two subsets whose distance is minimal.

5. Optimal silhouette

Even if we have a way of linking subset on partition we still don't have a way to stop. Namely, what is the right number of clusters for a given problem? We explore here a measure of error. For a point on a given cluster, $x \in S_i$, let

$$m(x) = \frac{1}{\#S_i} \sum_{y \in S_i, y \neq x} \|x - y\|$$

Being the mean distance to the remaining members of the cluster, this measures how well x is on the cluster S_i, the smaller, the better. Now for all other clusters S_j, let

$$d(x) = \min_{j \neq i} \frac{1}{\#S_j} \sum_{y \in S_j} \|x - y\|$$

Here we are looking for the cluster S_j which is the closest one to the given point x.

The silhouette for the point x is then defined by:

$$s(x) = \frac{d(x) - m(x)}{\max[m(x), d(x)]}, \quad \#S_i > 1$$

In the initial case of $\#S_i = 1$, we set $s(x) = 0$. By definition, we have $-1 \leq s(x) \leq 1$. If a point x is well-placed on cluster S_i, the value $m(x)$ is small whereas the value $d(x)$ is big. So the best placed, the closer the silhouette is to 1. Hence, a measure of how appropriately the data have been clustered.

This is going to be our goal now: to look for the *maximal* silhouette. Say we have k clusters, let $\hat{s_k}$ the average of the individual silhouette

$$\hat{s_k} = \frac{1}{\#S} \sum_{x \in S} s_k(x)$$

As we move the number of clusters k, we are looking for the one making the bigger silhouette.

$$K_0 = \arg\max_k \hat{s_k}$$

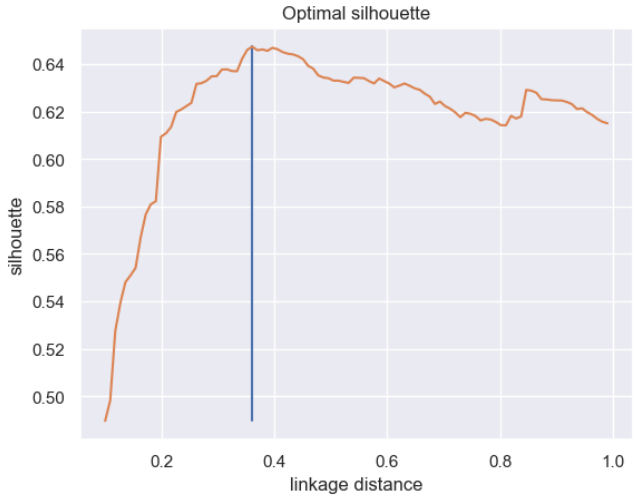


Figure 3. Silhouette for our case study

How to adjust? We will adjust with *The Linkage Distance* which is the threshold distance above which, clusters will not be merged. Moving the distance on the interval $[0,1]$ we get figure 3.

Using the distance found on the optimal silhouette, we found the optimal number of clusters.

6. Periodic transactions

All the transactions have the date where it was performed. For checking the periodicity of transactions, we consider the different periods: W, (Week), F (Fornight), M (Monthly), 2M, Q (Quaterly) and we consider also a tolerance t_P for each period (How many error days we allow for a given period). Let S, be a cluster and let $D(S) = \{d_i\}$ their set of dates. For each $d_0 \in D(S)$, and each period $P = W, F, M, 2M, Q$, we check the combinations $d_0 + kP$, looking for the largest series contained on $D(S)$ within the tolerance. This means that for each $d_i + kP$ there is a $d_{k_i} \in D(S)$ such that $|d_i + kP - d_{k_i}| \leq t_P$.

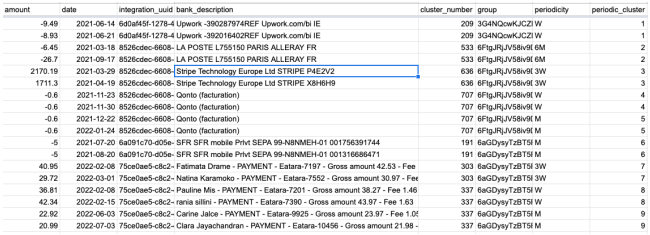
Still, once we have a regular set of dates on the cluster, we can have for a given date d have several transactions. Our choice is to select the ones which are closer:

- If there are only two dates, select the transactions being the closest on amount value.
- If there are more than two dates. Select the two first dates, proceed as before and generate a new transaction with amount the mean.
- Apply the former until all transactions are processed.

On the figure 4 you can have a look to a sample of obtained results.

References

[1] C. Bishop Pattern Recognition and Machine Learning *Springer*



amount	date	integration_uuid	bank_description	cluster_number	group	periodicity	periodic_cluster
-9.49	2021-06-14	6d0af45f-1278-4	Upwork -390287974REF Upwork.com/bi IE	209	3G4NQcwKJCZI	W	1
-8.93	2021-06-21	6d0af45f-1278-4	Upwork -392016402REF Upwork.com/bi IE	209	3G4NQcwKJCZI	W	1
-6.45	2021-03-18	8526cdec-6608-	LA POSTE L755150 PARIS ALLERAY FR	533	6FtgJRjJV58iv9l	6M	2
-26.7	2021-09-17	8526cdec-6608-	LA POSTE L755150 PARIS ALLERAY FR	533	6FtgJRjJV58iv9l	6M	2
2170.19	2021-03-29	8526cdec-6608-	Stripe Technology Europe Ltd STRIPE P4E2V2	636	6FtgJRjJV58iv9l	3W	3
1711.3	2021-04-19	8526cdec-6608-	Stripe Technology Europe Ltd STRIPE X8H8H9	636	6FtgJRjJV58iv9l	3W	3
-0.6	2021-11-23	8526cdec-6608-	Qonto (facturation)	707	6FtgJRjJV58iv9l	W	4
-0.6	2021-11-30	8526cdec-6608-	Qonto (facturation)	707	6FtgJRjJV58iv9l	W	4
-0.6	2021-12-22	8526cdec-6608-	Qonto (facturation)	707	6FtgJRjJV58iv9l	M	5
-0.6	2022-01-24	8526cdec-6608-	Qonto (facturation)	707	6FtgJRjJV58iv9l	M	5
-5	2021-07-20	6a091c70-d05e-	SFR SFR mobile Prlvt SEPA 99-N8NMEH-01 001756391744	191	6aGDysyTzBTSI	M	6
-5	2021-08-20	6a091c70-d05e-	SFR SFR mobile Prlvt SEPA 99-N8NMEH-01 001316686471	191	6aGDysyTzBTSI	M	6
40.95	2022-02-08	75ce0ae5-c8c2-	Fatimata Drame - PAYMENT - Eatara-7197 - Gross amount 42.53 - Fee	303	6aGDysyTzBTSI	3W	7
29.72	2022-03-01	75ce0ae5-c8c2-	Natina Karamoko - PAYMENT - Eatara-7552 - Gross amount 30.97 - Fee	303	6aGDysyTzBTSI	3W	7
36.81	2022-02-08	75ce0ae5-c8c2-	Pauline Mis - PAYMENT - Eatara-7201 - Gross amount 38.27 - Fee 1.46	337	6aGDysyTzBTSI	W	8
42.34	2022-02-15	75ce0ae5-c8c2-	rania sillini - PAYMENT - Eatara-7390 - Gross amount 43.97 - Fee 1.63	337	6aGDysyTzBTSI	W	8
22.92	2022-06-03	75ce0ae5-c8c2-	Carine Jalce - PAYMENT - Eatara-9925 - Gross amount 23.97 - Fee 1.05	337	6aGDysyTzBTSI	M	9
20.99	2022-07-03	75ce0ae5-c8c2-	Clara Jayachandran - PAYMENT - Eatara-10456 - Gross amount 21.98 -	337	6aGDysyTzBTSI	M	9

Figure 4. Some results on classification

[2] J. Eisenstein Introduction to Natural Language Processing *MIT press*

[3] T. Hastie, R. Tibshirani and J. Friedman The Elements of Statistical Learning *Springer*

[4] T. Mikolov, K. Chen, G. Corrado, J. Dean Efficient Estimation of Word Representations in Vector Space *arXiv:1301.3781*

[5] X. Rong word2vec Parameter Learning Explained *arXiv:1411.2738*

[6] Peter J. Rousseuw Silhouettes: a graphical aid to the interpretation and validation of cluster analysis *Journal of Computational and Applied Mathematics 20 (1987) 53-65*