WORKING PAPERS

N° 1471

September 2023

“Parsimonious Wasserstein Text-mining”

Sébastien Gadat and Stéphane Villeneuve

Toulouse School of Economics

Parsimonious Wasserstein Text-mining*

Sébastien Gadat† and Stéphane Villeneuve‡

September 20, 2023

Abstract

This document introduces a parsimonious novel method of processing textual data based on the NMF factorization and on supervised clustering with Wasserstein barycenter's to reduce the dimension of the model. This dual treatment of textual data allows for a representation of a text as a probability distribution on the space of *profiles* which accounts for both uncertainty and semantic interpretability with the Wasserstein distance. The full textual information of a given period is represented as a random probability measure. This opens the door to a statistical inference method that seeks to predict a financial data using the information generated by the texts of a given period.

Keywords: Natural Language Processing, Textual Analysis, Wasserstein distance, clustering

1 Introduction

Word processing technologies have recently colonized the financial industry with the booming in natural language processing (NLP). NLP includes many advanced methods based on machine-learning techniques which aim to extract the information contained in a text (press article, report, email, tweet, etc.). The information thus identified is intended to enlighten decision making like, for instance, portfolio allocation or real-life investment decision.

The use of texts as data in finance and more generally in social sciences is not new, see (10) that offers a survey of text analysis in economics. Finance initially focused on sentiment analysis which classifies texts on a limited number of emotions (positive, negative

*The authors gratefully acknowledge the support of Causality Link. This research has benefited from the ANR-17-EURE-0010 (Investissements d'Avenir program).

†Toulouse School of Economics (CNRS UMR 5314) & Institut Universitaire de France, 1 esplanade de l'université, 31000 Toulouse, France. Email: sebastien.gadat@tse-fr.eu

‡Toulouse School of Economics, 1 esplanade de l'université, 31000 Toulouse, France. Email: stephane.villeneuve@tse-fr.eu

or neutral). Although very succinct, the concept of sentiment analysis is a cornerstone of behavioral finance because of its implications for understanding how financial markets work. The question is no longer whether investor sentiment influences stock prices, but rather how to quantify its effects. The analysis of the sentiment of a text predates the rise of NLP techniques. For instance, (16) built a dictionary by hand to measure the tone of a financial text and a very good survey of the textual sentiment literature can be found in (13). On the other hand, (14) explored various applications in the financial domain where text mining could play an important role. They concluded that it has many applications especially in the prediction of financial variables. Text analysis may thus contribute to stock market prediction and enable those involved to make decisions based on raw data rather than pure speculation.

However, the statistical analysis of text-mining in finance, while promising, is just dawning for two main reasons. One is technical. Machines simply cannot process textual data in its raw form. They need humans to break down the text into a digital format that they can read. Numerical representation of texts as data for statistical analysis is very high-dimensional and empirical research seeking to exploit the information contained in texts must first confront its dimensional challenge. All methods thus share one main idea- to map texts to tractable embeddings (vectors or measures) that lie in a low dimensional space.

The other reason is a deeper statistical reason. What is the real predictability of the information contained in texts? One of the most pressing issues behind this growing interest in NLP methods in finance is to provide an effective representation of textual data that is suitable for statistical processing and prediction. All the existing attempts share the common goal of representing texts in a small-dimensional vector space, with the constraint that the distance between two representations is small if the two original texts are semantically close. It has been already observed that machine learning methods predict but do not explain the underlying economic mechanisms. The situation is even worse in the case of text analysis where the preliminary step of transforming raw textual data into numerical representations exploitable by the statistician requires to be able to model the semantic proximity of two texts by a well chosen distance on the space of numerical representations. A strong motivation for our approach is to build text representations that are able to capture the semantic proximity of two texts. Based on recent research (12), we claim that distributional representations over point estimates - hereafter called profiles- allows to capture more the meaning of texts than historical point-wise representation alone.

The objective of this paper is twofold. First, it introduces a parsimonious method of processing textual data based on both the NMF factorization and on unsupervised clustering with Wasserstein barycenters to reduce the dimension of the model. This dual treatment of textual data allows to represent a text as a probability distribution on the space of profiles -that are a mixture of words that synthesize the information contained in

the texts of a given period- and the information contained in a given period as a random probability measure which allows for a good semantic interpretability within the representation space. Second, it opens the door to a statistical inference method that seeks to predict a financial data using the information generated by the texts of a given period. This results in a flexible method that frees itself from an a priori statistical model of the joint distribution of textual and financial variables. An important feature of our paper is that our method will extract a predictive content of texts.

2 Inference with textual dataset

Text mining for the prediction of financial quantity movements is an emerging topic in today's data mining. Previous research has already suggested the existence of a relationship between news articles and stock prices, making a better understanding of statistical inference on textual data a necessity. One of the main task is to develop a textual data representation with the following dual objective:

- a drastic reduction in the size of the data space,
- a good semantic interpretability within the representation space.

This preliminary but crucial task is the main goal of our work. We below go into details to describe the supervised machine learning task we want to address in our framework, and specify the structure of the data.

2.1 Inference with a corpus of documents

The dataset is structured as a sequential recording on a given period of time, say a day for the sake of simplicity, but this period could either be several days, a week or a month for example. The data contains N days and each day is indexed by an integer $i \in \{1, \dots, N\}$.

Each day i, we record a corpus of texts denoted by $\mathcal{X}_i$. This corpus of texts contains n_i documents and we shall write $\mathcal{X}_i = (T_{1,i}, \dots, T_{n_i,i})$ where $T_{j,i}$ is then the document j of day i. In addition to our corpus of texts recorded each day, we also assume that we observe an output Y_i. We assume that a statistical model exists that describes the joint distribution between $\mathcal{X} = (\mathcal{X}_\rangle)_{1 \leq i \mathbb{N}}$ and Y through a standard relationship in supervised learning tasks:

- $Y = \Psi(\mathcal{X}) + \varepsilon$ if Y is a real valued output (supervised regression)
- $\mathbb{P}[Y = 1|\mathcal{X}] = \Psi(\mathcal{X})$ if Y is a binary output (supervised classification).

Our objective is then to forecast a new output Y while we record a corpus of texts $\mathcal{X}$. Of course, a such forecasting task seems very hard, and the difficulty especially comes

from the complexity of the corpus dataset recorded during N days, since $(\mathcal{X}_i)_{1\leq i\leq N}$ do not present any clear structure. Hence, our first important task is to describe an efficient way to represent each corpus of documents of each day.

2.2 Corpus of text representation

Vector space models (VSM) seek to represent the meaning of words using real-valued vectors. These vector representations can be used to induce similarity measures by calculating distances between vectors. Before describing two examples below, we assume that we have at our disposal a dictionary of words that is preliminary determined before our learning procedure. A text can be a sentence or a paragraph of news articles described with the help of the pre-specified dictionary $\mathcal{D}$ of D words. Consider a collection of n_i texts for a given day, there are several ways to embed a text as a real-valued vector.

Bag of words embedding For instance, the most naive is the bag-of-words method where $\mathcal{X}_i(\ell, j)$ counts the number of times the word j appears in text $\ell \in \{1, \ldots, n_i\}$. In matrix form, the raw data of a given day take the form of a matrix $\mathcal{X}_i$ of size $n_i \times D$, whose row $\mathcal{X}_i(\ell, .)$ is the vector of $\mathbb{R}_+^D$ that represents the text i. There may be quite a few variations while preparing the above matrix $\mathcal{X}_i$, in the dictionary specification. With the bag-of-words, the calculation of similarity between textual objects is based on the frequency of the words composing the textual objects to be compared. Unfortunately, this approach does not always take into account the semantic dependence between these words.

Tokenization and TF-IDF Another popular embedding consists in a list of integers that encodes the number of occurences of the term after a standard tokenization preprocessing, which is a standard approach in NLP. We refer to (18) for a large overview of all computer science methods on information retrieval and in particular an introduction to tokenization.

Once we obtain the preliminary list of terms with a bag of words/tokens approach, we then parameterize each text into the token basis using a standard TF-IDF encoding, which is still one of the most popular nowadays term-weighting scheme introduced in (21). Each token of the dictionary in each text of the corpus is weighted according to the by-product of the term frequency and the inverse document frequency. More precisely, if j is a term of the dictionary and $\mathcal{X}_i(\ell, .)$ is a document of the corpus $\mathcal{X}_i$, then:

$$TF_{j,\ell} = \frac{f_{j,\ell}}{\sum_{j' \in \mathcal{D}} f_{j',\ell}}.$$

Hence, $TF_{j,\ell}$ stands for the relative frequency of the word j in the document $\mathcal{X}_i(\ell, .)$. In the meantime, the Inverse Document Frequency accounts for how much information is provided

by the word j among the global corpus $\mathcal{X}_i$. $IDF_{j,\mathcal{X}_i}$ is given by:

$$IDF_{j,\mathcal{X}_i} = \log \frac{n_i}{\sum_{\ell=1}^{n_i} \mathcal{X}_i(\ell, j)}.$$

Then, the TF-IDF score of a term j in a document ℓ of the corpus $\mathcal{X}_i$ is defined by:

$$\mathbb{X}_{\ell,j,\mathcal{X}_i} = TF_{j,\ell} \times IDF_{j,\mathcal{X}_i}.$$

This produces for each document of a corpus $\mathcal{X}_i$ a list of non-negative weights that account for the relative importance of a word j inside each document and inside the overall corpus $\mathcal{X}_i$.

In our experiments, we will use the two previous parameterization: the simplest one derived from a straightforward bag of words approach, which leads to the matrix $\mathcal{X}_i$, and the one with a supplementary TF-IDF weighting procedure, which leads to $\mathbb{X}$. That being said, the rest of our construction is theoretically independent from the way the matrix $\mathcal{X}_i/\mathbb{X}_i$ is obtained with our initial natural language processing.

As observed in (12), most VSMs have a common problem: each word is represented by a single vector, which doesn't always allow for semantics and polysemy. In addition, the dimension of the vector is given by the size of the dictionary which can be very high-dimensional and can cause the phenomenon called the curse of dimensionality. To tackle these two issues, we introduce a profile modelling method based on the NMF factorization and a clustering method using Wassersetin barycenters. This method will treat each document as a mixture of profiles, each profile as a mixture of the words from the dictionary. Therefore, each document is represented as a probability distribution over the set of profiles whose dimension is controlled by the statistician. This will allow us to cast the distance between documents via the Wasserstein distance.

3 Methodology

3.1 Global overview

Our methodology may be summarized, rouhgly speaking in a a global NLP+ML pipeline that contains essentially two main steps. The first one is described in Figure 1, and consists in a quantitative description of our corpus of texts with the help of a low-rank matrix factorization. This factorization appears to be a preliminary learning phase to be made before the daily information is considered and should be addressed carefully with more or less sophisticated NLP tools.

The second one is described in Figure 2, and brought the essential novelty of our paper. It leads to a mixture modelling of the daily information with the help of a K-means clustering in the Wasserstein space of weights over profiles of information. This second step may then feed any classification or regression tasks but highly depends on the results brought by the preliminary factorization step.

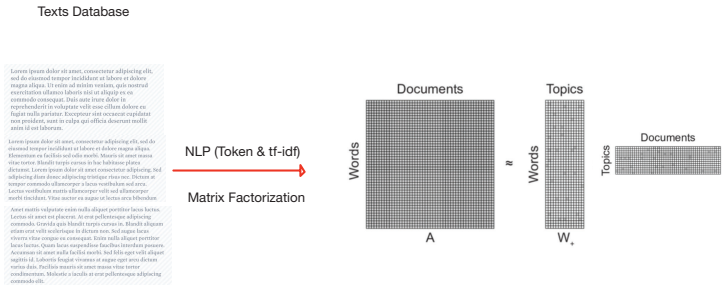

Figure 1: Learning step: From a corpus of texts to a quantitative weights/profile parametrization. The new parametrization is obtained with NMF.

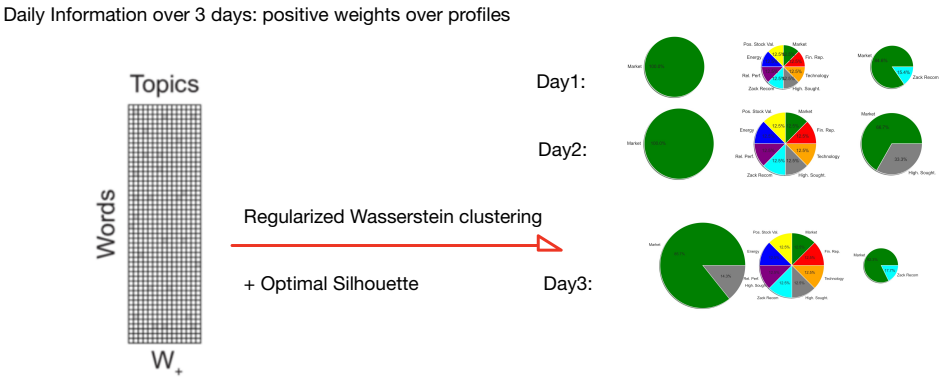

Figure 2: Daily analysis: From a daily corpus of texts to a weighted mixture of clusters. A cluster is a probability distribution over the profiles. The size of the balls are proportional to the weights of the clusters. For each of the 3 days we represent the three most important clusters of the daily information.

3.2 Profile modelling via Matrix factorization

The previous paragraph permits to embed the corpus of texts into a matrix representation (obtained with $\mathcal{X}_i$ or $\mathbb{X}_i$ following the previous paragraph). For the sake of simplicity, we will denote X_i as the matrix associated to day i, which is therefore of size $n_i \times D$.

It is well known in mathematical statistics and in machine learning that supervised learning highly suffers from the curse of dimensionality (see *e.g.* (11)). Faced with a large sample of data, our first step is to use non-negative matrix factorisation (NMF) to reduce the dimensionality of the problem by identifying the dominant profiles. Popularized by (15), NMF is a way to automatically extract sparse and meaningful features from a set of nonnegative data vectors.

Latent hidden structure of texts It is clear from the very nature of the data that indeed, a single document only brings a finite amount of information that may be decomposed on a finite small number of profile of information. Even though we could imagine some situations where a text may be somewhat unstructured, it is also a reasonable assumption to impose that the information contained in a text deals with a finite small number of themes (for example "crisis", "war", "epidemy", "ecology", "growth GDP", etc.). Of course, these typical profiles/themes are not available in the document and are hidden in the corpus of texts. Hence, they must be determined from the data and we assume that the hidden profiles are "stationary", *i.e.* we assume that the latent structure of the corpus of texts is decomposed inside the same basis of hidden profiles, all along the days and for all documents of every days during the period of study.

Statistical point of view Said differently, we assume the next important approximation, which may be translated into a statistical model (see below):

$$X_i \simeq W_i P, \tag{3.1}$$

where W_i corresponds to a set of weights for the n_i texts inside the profile matrix P. The matrix P and the weights W may be written as:

$$W_i = \begin{pmatrix} w_i(1,1) & w_i(1,2) & \dots & w_i(1,r) \\ \vdots & \dots & \dots & \vdots \\ w_i(n_i,1) & w_i(n_i,2) & \dots & w_i(n_i,r) \end{pmatrix} \quad \text{and} \quad P = \begin{pmatrix} P(1,1) & P(1,2) & \dots & P(1,D) \\ \vdots & \dots & \dots & \vdots \\ P(r,1) & P(r,2) & \dots & P(r,D) \end{pmatrix}.$$

Hence, each line of W_i, denoted by $W_i(\ell,.) = (W_i(\ell,u))_{1\leq u\leq r}$ corresponds to the set of weights on the latent profiles $P(1,.),\dots,P(r,.)$, meanwhile for any $u \in \{1,\dots,r\}$ each row $P(u,.)$ contains some non-negative coefficients that describe the coordinates of the profile $P(u,.)$ in the initial dictionary.

As a standard dichotomy in machine learning, we can think about Equation (3.1) in two different ways. Either from an optimization point of view or from a statistical modeling

point of view. We refer to (6) for a short overview on several statistical models (among many others) related to matrix factorization based on Gaussian, Poissonian or Gamma distributions.

It is in general somewhat difficult to assume a rigorous statistical model that describes the distribution of the difference between X_i and W_iP, which is the weighted decomposition over the latent profiles, because of the positivity constraint imposed to the entries of X_i.

Assumption (Latent structure)**.** *A finite number of hidden profiles $P_1, \ldots, P_r$ exists such that:*

$$\forall i \in \{1, \ldots, N\} \quad \forall \ell \in \{1, \ldots, n_i\} : \qquad X_i(\ell, .) = \sum_{k=1}^{r} w_{i,\ell}(k) P_k + \varepsilon_{i,\ell},$$

where $\varepsilon = X - WP$ follows a parametric distribution for which the ML estimation is recovered through a minimization procedure

$$(\hat{W}, \hat{P}) = \arg\min_{(W,P)\geq 0} - \log \mathbb{P}_{W,P}(X).$$

The difficulty involved in the previous formalism are twofolds: first, it is not obvious to define a statistical model that generates entries of X that are non-negative, and in particular the (untruncated) Gaussian model does not permit to achieves this property. Second, statistical identifiability is also a highly discussed subject of research, we refer to (7) for several geometrical insights on this identifiability issue, and for the recent monograph (9) for a global up to date overview.

Machine Learning point of view We are led to use instead the optimization point of view, that is essentially based on the definition of a loss function between the data X_i and the reconstruction W_iP obtained by the "learnt" algorithm. The loss may be measured in several ways, the Frobenius norm being the most popular one, each of them is based on a Bregman divergence between entries of X and the entries of the product W_iP. In this settings, to efficiently use NMF, we need to introduce a control parameter r which is an integer very small when compared to both n_i and D that will encode the dimensionality of the latent factors. We therefore define a NMF decomposition of X_i as a solution of the minimization problem:

$$\min_{(W_1,\ldots,W_N)\geq 0, P\geq 0} \sum_{i=1}^{N} ||X_i - W_iP||.$$

where the W_i are respectively $n_i \times r$ and P is $r \times D$ nonnegative matrices while $\|.\|$ is a data-fitting norm that measures the difference between the observations and the reconstructed data.

In the analysis, the matrix W_i will be called the *loading matrix* and the matrix P will be the *profile matrix*. Therefore, text i, represented by the vector X_i, can be approximatively

decomposed in $\mathbb{R}^D$ as the following linear combination of the r profiles. As introduced above, the r profiles $(P_k)_{1\leq k\leq r} \in \mathbb{R}^D$ can be viewed as the most prominent topics of the given period while $W_i(\ell,.)$ refers to the non-negative decomposition of text ℓ of day i on the profiles. This decomposition encodes an unnormalized positive measure whose total mass is generally different from 1. For the purpose to use an optimal transport tool, we have decided to normalize each text with the TV norm of $W_i(\ell,.)$, leading to a probability distribution that describes the text. We therefore define the normalize set of weights as $\tilde{W}_i$:

$$\tilde{W}_i(\ell,.) := \frac{W_i(\ell,.)}{\|W_i(\ell,.)\|_{TV}}$$

and the set of texts of day i is now described as: $(\tilde{W}_i(\ell,.), \|W_i(\ell,.)\|_{TV})_{1\leq\ell\leq n_i}$.

As a consequence, text $\ell \in \{1,\dots,n_i\}$ of day i can be now viewed as a probability measure over the finite set of profiles so that the dimension of the problem has been greatly reduced with the NMF factorization by representing a vector of $\mathbb{R}^D$ as an element of the $(r-1)$-simplex. Nevertheless, as indicated in the next paragraph, this dimensionality reduction will not be "enough" to handle a such large amount of information. Indeed, after the matrix factorization, each day is now represented as a collection of n_i texts decomposed on the r profiles, *i.e.* each day (say i) is parameterized with the help of a matrix (denoted by W_i) whose dimension is $n_i \times r$. We describe below a second important pre-processing step that leads to a more reasonable size of the dataset.

3.3 Wasserstein clustering

To even more reduce the dimension of the dataset, we then focus for a given day i on the population of the n_i probability distributions over profiles that describes this day. From a mathematical point of view, a daily information is summarized as a population of n_i elements of the simplex of dimension $r-1$. To produce a suitable summary of the n_i texts, we are then led to use a clustering (unsupervised classification) approach.

3.3.1 Wasserstein geometry of discrete probability measures over profiles

Wasserstein cost Statistical learning methods generally refer to data that are considered as points on a high-dimensional Euclidean space (e.g. $\mathbb{R}^D$ for large D in a word representation of texts), and deal with the problem of their classification using a distance on the Euclidean space, whose choice is not neutral in the classification process. These methods have led to the development of efficient algorithms, such as the K-means algorithm, widely used in statistical learning, (see Hastie et al). Unlike these traditional methods, our methodology offers a text representation in the space of probability measure on the finite set $\mathcal{P} = \{P_1,\dots,P_r\}$ that will be denoted by $\Delta(\mathcal{P})$ below. For the purpose of dimension reduction, we pursue the following objective: given collection of $\Delta(\mathcal{P})$-valued data, how to cluster the data into k groups. Because of the nonlinear structure of the space

of probability measures, the standard K-means algorithm in Euclidean space is no longer useful for this task. On the other hand, clustering is based on the concept of distance between the data representation. A very appropriate concept of distance in the space of probability measures is the Wasserstein distance. More precisely, we define on the product space $\mathcal{P} \times \mathcal{P}$ the applications $\pi_i : \mathcal{P} \times \mathcal{P} \to \mathcal{P}$ for $i = 1, 2$ as follows:

$$\pi_1(x_1, x_2) = x_1 \text{ and } \pi_2(x_1, x_2) = x_2.$$

A coupling between μ and ν is a probability measure γ on $\mathcal{P} \times \mathcal{P}$ such that $\pi_1 \# \gamma = \mu$ and $\pi_2 \# \gamma = \nu$. We denote by $\Pi(\mu, \nu)$ the set of couplings between μ and ν. The Wasserstein distance is thus defined as

$$W(\mu, \nu) = \inf_{\gamma \in \Pi(\mu,\nu)} \int \int_{\mathcal{P} \times \mathcal{P}} c(x, y) \gamma(dxdy),$$

where c is a unitary transport cost on $\mathcal{P}$ that is chosen to reflect the semantic similarity between profiles. In particular, given two profiles P_u and P_v, $c(P_u, P_v)$ encodes the proximity of the two profiles, *i.e.* this cost is large when P_u and P_v have nothing in common whereas it is small when they share some common information. Intuitively, two documents are similar in meaning if the prominent profiles of one document can be cheaply transported to the prominent profiles of the other document. The cost of transportation being measured by the function c.

Computational and robustness point of views Observe that in our case, the set $\mathcal{P}$ is finite and thus the computation of the Wasserstein distance between $\mu = \sum_{i=1}^{r} \alpha_i \delta_{P_i}$ and $\nu = \sum_{i=1}^{r} \beta_i \delta_{P_i}$ is reduced to a linear programming problem. More precisely, we define $\Pi(\mu, \nu)$ as the set of probability measures on $\mathcal{P} \times \mathcal{P}$ with marginals μ and ν.

$$W(\mu, \nu) = \inf_{\pi \in \Pi(\mu,\nu)} \left\{ \sum_{i,j} \pi_{ij} c(P_i, P_j) \right\}. \tag{3.2}$$

In that particular situation where the measures are discrete, the optimal transport problem and the assignment problem, described in (1) coincide. This latter assignment problem can be solved using various techniques, for example with the help of the Bertsekas' auction algorithm, which is a dual ascent method see (1).

Nevertheless, it has been observed both from a computational and statistical point of views that a direct use of the Wasserstein distance yields some numerical difficulties. First, the optimal transport plan is an irregular function of the measures μ and ν, which induces a highly lack of robustness from a statistical point of view. We refer, as an example, to the recent work (23) (among others). Second, the computational cost of solving the linear program (3.2) with the Bertsekas' auction algorithm can be greatly improved, with the help of a penalty term as indicated below.

A recent popular alternative (see (8; 3)), that may be solved more efficiently with the dual Sinkhorn alternate projection algorithm, focuses on a penalized entropic criterion:

$$W_\varepsilon(\mu,\nu) = \inf_{\pi\in\Pi(\mu,\nu)} \left\{ \sum_{i,j} \pi_{ij} c(P_i,P_j) + \varepsilon KL(\pi, \mu\otimes\nu) \right\}$$

where ε is a penalization parameter and KL refers to the Kullback-Leibker divergence, defined by:

$$KL(\pi,\mu\otimes\nu) = \sum_{i,j} \log\left(\frac{\pi_{i,j}}{\mu_i\nu_j}\right)\pi_{i,j}$$

The essential interest of this penalty is to brought strong convexity induced by KL, which is a 1-strongly convex function so that minimizing W_ε may be achieved using the Bregman iterative mirror descent that converges exponentially fast (with a rate that is degraded with ε). Therefore, instead of the algorithm of (1), we will use the alternate Sinkhorn and Knopp projection introduced in (3) to compute efficiently $W_\varepsilon(\mu,\nu)$ the Wasserstein distance between the probability distribution over profiles, *i.e* the distance between the lines of the matrix W_i that aggregated the information gathered at day i. This algorithm, which is based on a gradient ascent strategy in the dual space iteratively computes a sequence of two vectors of size r, denoted by $(\mathbf{u}^{(k)},\mathbf{v}^{(k)})_{k\geq 1}$, and is described precisely in Algorithm 1.

Data: Measures (μ,ν), Matricial cost $C = (c(P_i,P_j))_{i,j}$,
Penalty parameter ε, Iterations N_{iter}
$u^{(0)} = \mathbf{1}$ and $K = \exp(-C/\varepsilon)$;
for $k = 1 \dots N_{iter}$ **do**
  $\mathbf{v}^{(k)} = \nu \oslash K^T\mathbf{u}^{(k-1)}$;
  $\mathbf{u}^{(k)} = \mu \oslash K\mathbf{v}^{(k)}$;
end
Compute $\hat{\pi} = \mathrm{diag}(\mathbf{u}^{(N_{iter})})K\,\mathrm{diag}(\mathbf{v}^{(N_{iter})})$;
Result: $W_\varepsilon(\mu,\nu) \simeq \langle\hat{\pi}, C\rangle$

Algorithm 1: Alternate Sinkhorn-Klopp algorithm.

Data-dependent cost between profiles In the previous paragraph, we propose to measure the distance between texts using a transport distance, while considering a text as a weighted combinations of profiles. Hence, the Wasserstein loss W defined in (3.2) crucially depends on the cost induced by a displacement of one profile to another, *i.e.* on the definition of the matrix $C = (c(P_i,P_j))_{1\leq i,j\leq r}$. We can imagine simply reducing the cost C to a very simple matrix $c(P_i,P_j) = 1-\delta_{i=j}$, which translates the fact that all the profiles are at the same distance. This may be a reasonable assumption when r is chosen not so large when compared to the corpus of texts.

Another solution would be to use the cosine similarity index to measure the distance

over profiles associated to their tf-idf coordinate in the initial dictionary:

$$\text{Cosine}(P_u, P_v) = \frac{\langle P_u, P_v \rangle}{\|P_u\| \|P_v\|},$$

since the cosine index traditionnally measures the proximity between texts (see *e.g.* (18)). Indeed, as indicated by our numerical experiments, and thanks to the good compression property induced by the preliminary NMF algorithm, it appears that the cosine index translates an almost null cosine between all different profiles, which shows that the simple cost matrix:

$$C(P_u, P_v) = 1 - \delta_{u=v}$$

seems a reasonable and simple assumption. Said differently, we then consider the parametrisation of texts *via* W as a set of coordinates in the directions of profiles P that is considered as an orthonormal basis. This latter fact even more simplifies the computation of the pairs of penalized Wasserstein distances $W_\varepsilon(\mu, \nu)$ thanks to the very simple form of the matrix K involved in Algorithm 1.

3.3.2 Clustering the information in the Wasserstein space

The essential interest of the previous cost matrix C and of the Wasserstein distance W is the use of standard unsupervised classification algorithms based on metric considerations. Among them, the K-means algorithm is one of the standard method that relies on an iterative procedure of assignment to the nearest class and barycenter update.

Each day, we collect n_i measures $\{\mu_1, \ldots, \mu_{n_i}\}$ on $\mathcal{P}$ and a natural approach to simplify and aggregate the daily information is to produce a clustering with the help of an unsupervised classification method, *i.e.* we will organize this collection into K clusters in the space $\Delta(\mathcal{P})$.

Entropic regularized Wasserstein barycenter To obtain a tractable clustering, an almost straightforward application of the K-means algorithm of (17) can be implemented, once the technical details of the (penalized) Wasserstein distance and Barycenter computations are fixed. For this purpose, the keystone barycenter computation in terms of penalized Wasserstein distance is inspired from the contributions of (4) (see also (5) and (2) for recent generalizations). We detail below the important steps involved in the iterations of the K-means method. Among the increasing literature on Wasserstein barycenters, we have choosen to use the smoothed dual approach of (5) to obtain the entropic regularized Wasserstein barycenter to stay consitent with our use of the Sinkhorn-Klopp algorithm, and to obtain a rapid implementation.

Entropic regularized Wasserstein-Silhouette K-means We finally describe the overall procedure to represent the daily information as a mixture of probability distributions

over profiles, with the help of (entropic regularized) Wasserstein distances, barycenters and K-means. The method is given in Algorithm 2, and the number of clusters K used for K-means clustering is obtained with the help of the Silhouette score medoid.

A standard bottleneck when using K-means clustering relies on the intricate choice of K, that determines the number of clusters that are used for clustering. To properly address this final issue, it would be tempting to use the silhouette coefficient introduced in (20), which is defined for any measure μ_j of our dataset and for any clustering of the family of measures, as:

$$S_{sil}(\mu_j) = \frac{b(\mu_j) - a(\mu_j)}{\max\{a(\mu_j), b(\mu_j)\}}, \tag{3.3}$$

where $b(\mu_j)$ stands for the average Wasserstein distance between μ_j and its closest cluster while $a(\mu_j)$ is the average Wasserstein distance between μ_j and all other measures in the cluster of μ_j. Then, the overall Silhouette coefficient is simply the average number of individual coefficients $S_{sil}(\mu_j)$.

However, the computation of the Silhouette coefficient appears to be costly in our framework, and we simply use instead the medoid Silhouette, presented in (22), where average intra and inter Wasserstein distances in a and b coefficients involved in (3.4) are simply replaced by direct distances to barycenters of clusters. Said differently,

$$S_{sil,medoid}(\mu_j) = \frac{b'(\mu_j) - a'(\mu_j)}{\max\{a'(\mu_j), b'(\mu_j)\}}. \tag{3.4}$$

with

$$b'(\mu_j) = W_\varepsilon(\mu_j, \nu_{\overline{k_j}}) \qquad \text{and} \qquad a'(\mu_j) = W_\varepsilon(\mu_j, \nu_{k_j}),$$

where ν_{k_j} is the barycenter of the cluster where μ_j is and $\nu_{\overline{k_j}}$ is the second closest barycenter of μ_j. The value of $S_{sil,medoid}$ varies between -1 and 1 and it is commonly admitted in descriptive statistics that the closest to 1 $S_{sil,medoid}$, the better the clustering result. Accordingly, we have implemented a loop over K to maximise the value of $S_{sil,medoid}$.

Data: Day i, measures $(\mu_1, \dots, \mu_{n_i})$, Matricial cost $C = (c(P_\ell, P_{\ell'}))_{\ell,\ell'}$,
Penalty parameter ε
K probability measures $\{\nu_1, \dots, \nu_K\}$ on $\Delta(\mathcal{P})$ sampled uniformly;
Initial cluster assignment:

$$\forall j \in \{1, \dots, n_i\} : \qquad k(j) = \text{Argmin}_{1 \le k \le K} W_\varepsilon(\mu_j, \nu_k).$$

while *Clusters are not stable* **do**
  Compute the smoothed barycenters with (5):

$$\nu_k^i = \text{Argmin}_{\nu \in \Delta(\mathcal{P})} \sum_{j : k(j) = k} W_\varepsilon(\nu, \mu_j);$$

  Update the clusters assignments:

$$\forall j \in \{1, \dots, n_i\} : \qquad k(j) = \text{Argmin}_{1 \le k \le K} W_\varepsilon(\mu_j, \nu_k).$$

end
Compute the weights of each clusters:

$$\omega_k^i = \frac{|j : k(j) = k|}{n_i}$$

Result: Weights $(\omega_1^i, \dots, \omega_K^i)$, Final barycenter measures $(\nu_1^i, \dots, \nu_K^i)$
Algorithm 2: Smoothed entropic Wasserstein K-means with known K.

Information representation Once our NLP method with matrix factorization and Wasserstein clustering is performed, we summarize the information of day i with a mixture of probability distributions over profiles $\mathcal{P}$, denoted by $\mathcal{I}_i$:

$$\mathcal{I}_i := \sum_{k=1}^{K} \omega_k^i \delta_{\nu_k^i}.$$

Each probability $(\nu_k^i)_{k=1,\dots,K}$ is a mixture of the profiles. They describe intrinsic structure of the textual information. As indicated above, the description of the information contained in the n_i documents of day i is given by the probability measure $\mathcal{I}_i$ which greatly simplifies the representation of the data while keeping a nice level of interpretability.

4 Numerical experiments

4.1 Matricial factorization and profiles

We close this report with some brief numerical illustrations. This numerical section is purely a proof of concept as it would require a more effective investigation on a richer database. Besides this prospective approach, we obtain encouraging result and are strongly convinced to be able to obtain very good results in numerous fields, including finance. We tested our method on a database of texts from the Yahoo! news finance site over the period April 2023, which generates a thousands of texts, and our dictionary of 1000 words which is rather small. Even with a small database, meaningful profiles emerge. Figure (3) shows the eight profiles generated by the NMF procedure and displays them as a mixture of words from the dictionary. For example, Profile 2 is a mixture of words clearly related to banking and market activities. The most frequent words in the bank and market profile are banks, market, investors. Profile 4 is clearly dominated by the word energy, but although of marginal contribution, the other common words of this profile are fuel, oil, renewable and investors. Importantly, words can be shared between profile, a word like investors appears twice.

Roughly speaking, we can summarise these profiles in the next table.

Profile 1	Financial reports
Profile 2	Bank and Market
Profile 3	Performing Shares
Profile 4	Energy
Profile 5	Relative performance from past year
Profile 6	Zack's market recommendation
Profile 7	Products in high demand
Profile 8	Technology

4.2 Representation of the daily information

According to the previous matrix factorization, all our modeling and data analysis then may lead to a representation of the daily information that is exemplifies in Figure 4. It displays the daily information as a mixture of clusters, which are probability measures over profiles. The number of clusters optimized by the silhouette method changes from day to day. We have 6 clusters for the 21th but 8 for the 30th. The smaller the number of clusters, the more concentrated the information, whereas the larger the number of clusters, the more dispersed the information. Additionally, the relative weight of each cluster is proportional to the radius of the circular diagram. For example, the largest number of texts of the 21th of April 2023 in Yahoo news were talking about the Market (in green) and the relative performance of a company when compared to previous reports (in grey). This representation is also quantitative as the weight of each cluster and the coordinate

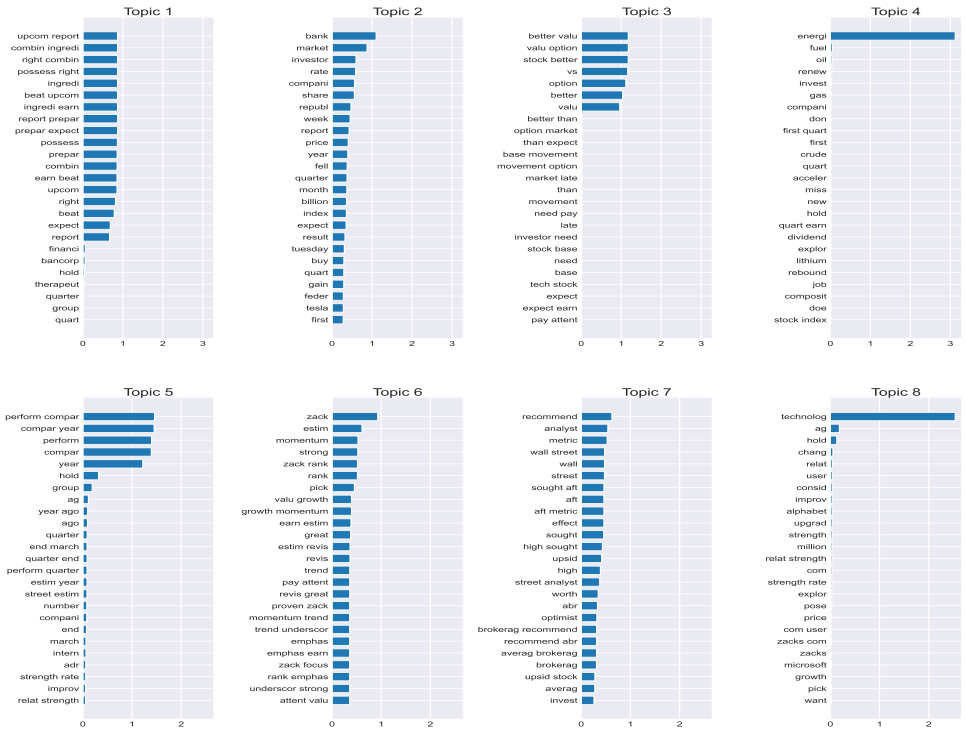

Figure 3: Description of the profiles

of the mixing profiles of each barycenters are explicitly known. This opens the door to initial statistical processing of the data. A financial variable such as the variation of a stock market index can be regressed on the number and the size of clusters to study the impact of information dissemination on index variations.

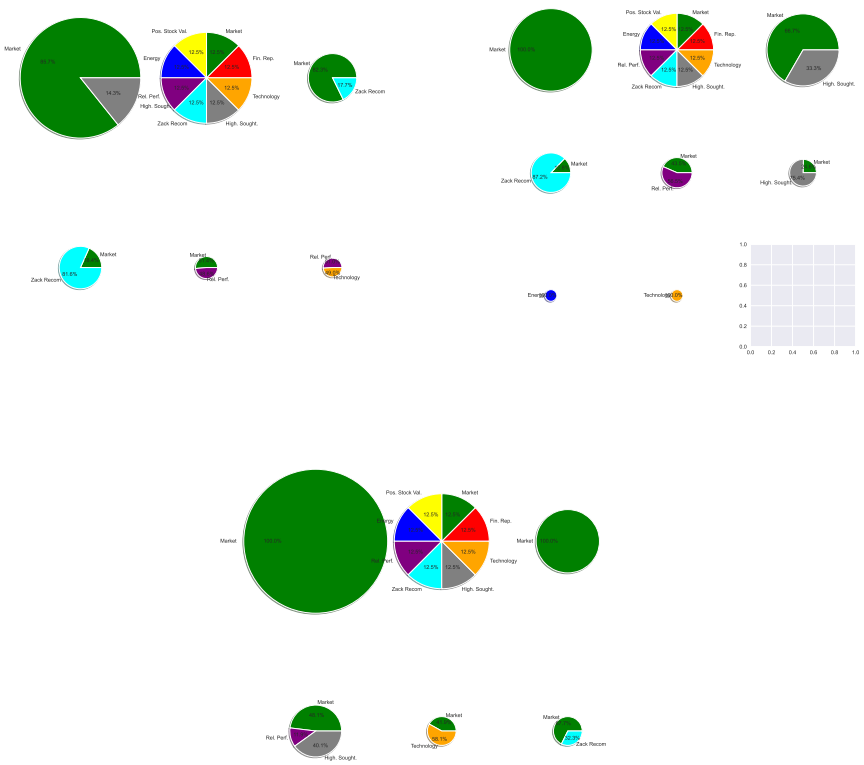

Figure 4: Information summary of Day 21th, 24th and 30th of April 2023

References

[1] Bertsekas, D.P. (1981) *A new algorithm for the assignment problem*, Mathematical Programming, **21**, 1, 152–171

[2] Chizat, L. and Peyré, G. and Schmitzer, B. Vialard, F.-X. (2018) *Scaling algorithms for unbalanced optimal transport problems*, Mathematics of Computation, 87(314):2563–2609

[3] Cuturi, M. (2013) *Sinkhorn distances: Lightspeed computation of optimal transport*, Advances in Neural Information Processing Systems, 2292–2300

[4] Cuturi, M. and Doucet, A. (2014) *Fast Computation of Wasserstein Barycenters*, Proceedings of the 31st International Conference on Machine Learning, **32**(2), 685–693

[5] Cuturi, M. and Peyré, G. (2016). *A smoothed dual approach for variational Wasserstein problems*, SIAM Journal on Imaging Sciences, **9**(1), 320–343

[6] Fevotte, C. and Cemgil, T. (2009) *Nonnegative matrix factorizations as probabilistic inference in composite models*, 17th European Signal Processing Conference (EUSIPCO 2009) Glasgow, Scotland, August 24-28

[7] Fu, X. and Huang, K. and Sidiropoulos, N. and Ma, W.-K. (2019) *Nonnegative Matrix Factorization for Signal and Data Analytics: Identifiability, Algorithms, and Applications*, IEEE, Signal Processing, **36**, 2, 59–80

[8] Galichon, A. and Salanié, B. (2010) *Matching with trade-offs: Revealed preferences over competing characteristics*, Tech. report, CEPR Discussion Papers.

[9] Giglis, N. (2020) *Nonnegative Matrix Factorization*, Siam, Data Science Book Series.

[10] Grimmer, J. and B. M. Stewart (2013) *Text as Data: The Promise and Pitfalls of Automatic Content Analysis Methods for Political Texts*, Political Analysis, vol. 21, issue 3, 267-297

[11] Hastie, T. and Tibshirani, R. and Friedman, J. (2009) *The Elements of Statistical Learning, Data Mining, Inference, and Prediction, Second Edition*, Springer, Series in Statistics.

[12] Huang, E.H., Socher R., Manning C. and Ng A. (2012) *Improving word representations via global context and multiple word prototypes.* Proceedings of the 50th Annual Meeting of the Associatin for Computatinal linguistics, **1, 873-882**

[13] Kearney C. and S. Liu(2014) *Textual Sentiment in Finance: A Survey of Methods and Models*, International Review of Financial Analysis, Vol. 33, 171-185

[14] Kumar, B.S. and Ravi, V. (2016) A Survey of the Applications of Text Mining in Financial Domain. Knowledge-Based Systems, 114, 28-147

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

[16] T. Loughran and B. McDonald. (2011) *When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks.* The Journal of Finance, 66(1):35–65.

[17] Lloyd, S. (1982) *Least squares quantization in PCM*, IEEE Transactions on Information Theory, **28**, (2): 129–137.

[18] Manning, C. and Raghavan, P. and Schütze, H. (2008) *Introduction to Information Retrieval*, Cambridge University Press.

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

[20] Rousseeuw, P. (1987) *Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis*, Computational and Applied Mathematics, **20**, 53-65.

[21] Spärck Jones, K. (1972) *A Statistical Interpretation of Term Specificity and Its Application in Retrieval*, Journal of Documentation, **28**, 1, 11–21.

[22] Van der Laan, M. and Pollard, K. and Bryan, J. (2003) *A new partitioning around medoids algorithm*, Journal of Statistical Computation and Simulation, **73** (8), 575-584.

[23] Zhuang, Y. and Chen, X. and Yang, Y. (2022) *Wasserstein K-means for clustering probability distributions*, Advances in Neural Information Processing Systems, 2292–2300.