Text mining with constrained tensor decomposition

Elaheh Sobhani, Pierre Comon, Christian Jutten, Massoud Babaie-Zadeh

Text mining with constrained tensor decomposition [⋆]

Elaheh Sobhani[1,2][0000−0002−4532−7168], Pierre Comon[1][0000−0001−9436−9228],
Christian Jutten[1][0000−0002−4477−4847], and
Massoud Babaie-Zadeh[2][0000−0001−8864−4756]

[1] GIPSA-Lab, Univ. Grenoble Alpes, CNRS, F-38000 Grenoble, France
firstname.lastname@gipsa-lab.grenoble-inp.fr
[2] Sharif University of Technology, Tehran, Iran
mbzadeh@sharif.edu

Abstract. Text mining, as a special case of data mining, refers to the estimation of knowledge or parameters necessary for certain purposes, such as unsupervised clustering by observing various documents. In this context, the topic of a document can be seen as a hidden variable, and words are multi-view variables related to each other by a topic. The main goal in this paper is to estimate the probability of topics, and conditional probability of words given topics. To this end, we use non negative Canonical Polyadic (CP) decomposition of a third order moment tensor of observed words. Our computer simulations show that the proposed algorithm has better performance compared to a previously proposed algorithm, which utilizes the Robust tensor power method after whitening by second order moment. Moreover, as our cost function includes the non negativity constraint on estimated probabilities, we never obtain negative values in our estimated probabilities, whereas it is often the case with the power method combined with deflation. In addition, our algorithm is capable of handling over-complete cases, where the number of hidden variables is larger than that of multi-view variables, contrary to deflation-based techniques. Further, the method proposed therein supports a larger over-completeness compared to modified versions of the tensor power method, which has been customized to handle over-complete case.

Keywords: data mining · learning · latent variable · multi-view · non negative · tensor · CP decomposition · eigenvalue

1 Introduction

Modeling. Data mining covers a wide range of methods and tools utilized for discovering knowledge from data [17]. In the context of multi-modal data (*e.g.* text, audio and video for the same event), mining can be considered as the estimation of latent variables that are present in several modalities [6, 24]. Multi-view models are useful tools to represent the relationships in this framework [27]. The graphical model we assume is depicted in Fig. 1; a set of multi-view variables, $\{x_1, x_2, \ldots, x_L\}$, and their corresponding latent variable, h, have been represented. This general model does not impose any particular distribution (Gaussian, Dirichlet, etc), and is very common in several tasks

[⋆] Supported by Gipsa-Lab.

of data (e.g. text) mining. Text mining as a special case of data mining is the center of attention of this paper. With this in mind, words in a document are considered as the multi-view variables $\{x_1, x_2, \ldots, x_L\}$ and their topic as the hidden variable h, as in [5].

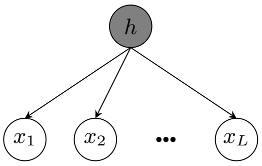

Fig. 1: multi-view variables and their corresponding hidden variable.

Related works. Daily increasing amount of text data available online have raised many challenging tasks such as unsupervised learning or clustering, which can be investigated by text mining [17]. Among plenty of algorithms customized for these purposes, we focus on some specific recent methods [2–4], which employ tensor factorization for learning latent variables and of course text mining as a special case. The key is that the latter tensor decomposition is unique, and hence is able to provide an estimation of latent parameters. There exist some other works, such as [9], employing tensors in a similar context. However, those aim at measuring word similarity rather than learning parameters of an underlying graphical model. Therefore, they may resort to the Tucker decomposition, which is not unique, but permits low *multilinear rank* approximation. Unlike [2–4] and the present contribution, which can be applied to other problems than text mining, [9] is hence dedicated to text mining, and does not aim at learning parameters of a model such as that of Fig. 1.

Resorting to moments for estimating parameters has a long history since 1894 till now in various fields such as mixture model [1,5,22], Blind Source Separation [10,11], and machine learning [1–5], just to name a few. For instance, in [5], the graphical model shown in Fig. 1 has been considered, and authors tried to estimate its parameters (probability of h and conditional probability of x_ℓ given h) by the means of diagonalization of two moment matrices.

In the same vein, several recent works have extended the idea of [5] to the tensor framework. To be more precise, [2] is the first contribution in which the Robust tensor power method is used to estimate the above mentioned parameters. One of the limitation of [2] is that it can handle only under-complete cases (*i.e.* when the number of hidden variables is smaller than that of multi-view). In order to overcome this limitation, authors of [3, 4] proposed to use other decompositions such as Tucker [12], or enhanced the tensor power method [15, 16].

Contribution. In this paper, we use the same formulation as in [2], relating moments and desired parameters. But one important difference lies in assuming a more appropriate cost function incorporating non negative constraints, which we minimize by

Alternating-Optimization Alternating Direction Method of Multipliers (AO-ADMM) [18]. This results in better performance than that of [2]. We shall show some figures in Section 5.2 demonstrating that our method, unlike the Robust tensor power method [2], does not return negative values for estimated probabilities. Moreover, our method can handle over-complete cases up to higher levels than [3].

Notation. Vectors, matrices, and tensors will be denoted with bold lowercases (*e.g.* $\boldsymbol{a}$), bold uppercases (*e.g.* $\boldsymbol{A}$) and bold calligraphic letters (*e.g.* $\boldsymbol{\mathcal{T}}$), respectively. The contraction symbol[3] $\bullet_k$ indicates a summation on the kth tensor index, and $\otimes$ the tensor (outer) product.

2 Problem formulation

Although the graphical model in Fig. 1 can be used for a wide range of learning problems [21], we customize the formulation of this section for the particular problem of text mining. Let L be the number of words in a given document, $\boldsymbol{x}_\ell$ the observed words, $\ell \in \mathcal{L} = \{1, 2, \dots, L\}$, and h a topic encoded into a discrete variable taking K possible integer values, say in $\mathcal{H} = \{1, 2, \dots, K\}$ with probability $\varphi(k) = Prob(h = k)$. All the words belong to a known encoded dictionary $\Omega = \{\boldsymbol{u}_1, \dots, \boldsymbol{u}_D\}$ of cardinality D. Put in other words, we can consider a mapping γ (generally not injective) from $\mathcal{L}$ to Ω such that $\boldsymbol{x}_\ell = \boldsymbol{u}_{\gamma(\ell)}$. In the context of text mining, D would be the number of words and K the number of topics. The number of documents, N, is generally larger than D.

Besides the probability of topics, the conditional probability of each word given the topic is also an important parameter in the text mining task. We denote the conditional probability of each word $\boldsymbol{u}_d$ of dictionary Ω, given a particular topic, $h = k$, by $f_k(d) = Prob(\boldsymbol{x} = \boldsymbol{u}_d | h = k)$. Therefore, the joint distribution of $\boldsymbol{X} = [\boldsymbol{x}_1, \dots, \boldsymbol{x}_L]$ can be written as:

$$p_{\boldsymbol{X}}(\boldsymbol{u}_{\gamma(1)}, \dots, \boldsymbol{u}_{\gamma(L)}) = \sum_{k=1}^{K} \varphi(k)\, f_k(\gamma(1)) \dots f_k(\gamma(L)), \tag{1}$$

which is referred to as the *naive Bayes* model. The main task is to estimate the quantities appearing in the right-hand side of (1) from realizations of $\boldsymbol{X}$. We assume the same assumptions as in [2,5]:

- conditional probabilities do not depend on the order of words (exchangablility),
- words are conditionally independent given the topic,
- words have the same conditional distribution given the topic.

In the sequel, we will need the second and third order moments:

$$\boldsymbol{P} \overset{\text{def}}{=} \mathbb{E}\{\boldsymbol{x}_p \otimes \boldsymbol{x}_q\} \tag{2}$$

$$\boldsymbol{\mathcal{T}} \overset{\text{def}}{=} \mathbb{E}\{\boldsymbol{x}_p \otimes \boldsymbol{x}_q \otimes \boldsymbol{x}_r\}, \tag{3}$$

[3] mode-k product between tensors.

where $\boldsymbol{P}$ is a $D \times D$ symmetric matrix and $\boldsymbol{\mathcal{T}}$ a $D \times D \times D$ symmetric tensor. These moments do not depend on $\{p, q, r\}$ provided these 3 integers are all different, which ensures the conditional independence assumed in (1). But note that $\{\gamma(p), \gamma(q), \gamma(r)\}$ may not be different because γ is not injective.

As in [2], we encode $\boldsymbol{u}_d$ into the columns of the $D \times D$ identity matrix. Because of this choice made for $\boldsymbol{u}_d$, these moments exhibit the following relations:

$$\boldsymbol{P} = \sum_{k=1}^{K} \varphi_k \; \boldsymbol{a}_k \otimes \boldsymbol{a}_k \tag{4}$$

$$\boldsymbol{\mathcal{T}} = \sum_{k=1}^{K} \varphi_k \; \boldsymbol{a}_k \otimes \boldsymbol{a}_k \otimes \boldsymbol{a}_k, \tag{5}$$

where $\boldsymbol{a}_k$ denotes the kth column of matrix $\boldsymbol{A}$. Note that $\boldsymbol{a}_k$ contains the values of $f_k(d)$ for all d and each k. Rewriting (1) reveals the nice property is that arrays $\boldsymbol{P}$ and $\boldsymbol{\mathcal{T}}$ are actually *joint probabilities* of observations, *i.e.*, $P_{ij} = Prob\{\boldsymbol{x}_p = \boldsymbol{u}_i, \boldsymbol{x}_q = \boldsymbol{u}_j\}$ and $\mathcal{T}_{ijk} = Prob\{\boldsymbol{x}_p = \boldsymbol{u}_i, \boldsymbol{x}_q = \boldsymbol{u}_j, \boldsymbol{x}_r = \boldsymbol{u}_k\}$.

3 The Robust tensor power method

We describe in this section the approach of [2]. We shall compare this method with ours in Subsection 5.2. In [2], the authors propose to use the two moments defined in (2) and (3) with exactly the same encoding explained in Section 2. Therefore, these moments enjoy relations (4) and (5).

Matrix $\boldsymbol{P}$ is theoretically positive semi-definite, since φ_k are positive numbers in (4). Hence in a similar way as had been done for Blind Source Separation [10, 11], there exists a "whitening" matrix $\boldsymbol{W}$ such that $\boldsymbol{W}^{\mathsf{T}}\boldsymbol{P}\boldsymbol{W} = \boldsymbol{I}$, where $\boldsymbol{I}$ is the identity matrix; $\boldsymbol{W}$ can theoretically be easily obtained from the EigenValue Decomposition (EVD) $\boldsymbol{P} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^{\mathsf{T}}, \boldsymbol{U}^{\mathsf{T}}\boldsymbol{U} = \boldsymbol{I}$, by setting $\boldsymbol{W} = \boldsymbol{U}\boldsymbol{D}^{-1/2}$. Therefore,

$$\sum_{k=1}^{K} \tilde{\boldsymbol{a}}_k \tilde{\boldsymbol{a}}_k^{\mathsf{T}} = \boldsymbol{I}$$

if $\tilde{\boldsymbol{a}}_k \overset{\text{def}}{=} \sqrt{\varphi_k}\, \boldsymbol{W}^{\mathsf{T}}\boldsymbol{a}_k$. In other words, the matrix $\tilde{\boldsymbol{A}} \in \mathbb{R}^{K \times K}$ containing $\tilde{\boldsymbol{a}}_k$ as its columns is orthogonal, *i.e.* $\tilde{\boldsymbol{A}}\tilde{\boldsymbol{A}}^{\mathsf{T}} = \boldsymbol{I}$. As proposed in [2], in practice, matrix $\boldsymbol{P}$ can be estimated via the mere averaging of samples (7), and consequently, $\boldsymbol{P}$ may have negative eigenvalues because of estimation errors. This issue was not investigated in [2], but they used left singular vectors of $\boldsymbol{P}$ instead of eigenvectors to construct $\boldsymbol{W}$ (cf. Algorithm 1). Next, this whitening matrix is applied to tensor $\boldsymbol{\mathcal{T}}$ to yield:

$$\tilde{\boldsymbol{\mathcal{T}}} \overset{\text{def}}{=} \boldsymbol{\mathcal{T}} \underset{1}{\bullet} \boldsymbol{W} \underset{2}{\bullet} \boldsymbol{W} \underset{3}{\bullet} \boldsymbol{W}$$

which have the following element-wise definition [12]:

$$\tilde{\boldsymbol{\mathcal{T}}}(r, t, s) = \sum_{r', t', s'} \boldsymbol{\mathcal{T}}(r', t', s')\boldsymbol{W}(r, r')\boldsymbol{W}(t, t')\boldsymbol{W}(s, s').$$

Algorithm 1 The Robust tensor power method [2]

```
Input: $\mathcal{T}, \boldsymbol{P}$
Output: $\boldsymbol{A}, \varphi$
Whitening: $\boldsymbol{P} = \boldsymbol{U\Sigma V}^{\mathsf{T}}$; $\boldsymbol{W} = \boldsymbol{U\Sigma}^{-1/2}$; $\tilde{\mathcal{T}} \overset{\text{def}}{=} \mathcal{T} \bullet_1 \boldsymbol{W} \bullet_2 \boldsymbol{W} \bullet_3 \boldsymbol{W}$
for $k = 1, \dots, K$ do
    for $m = 1, \dots, M_1$ do
        1) draw an initial vector $\boldsymbol{\theta}_k$ of unit 2-norm for $\hat{\boldsymbol{a}}_k$
        2) compute $M_2$ power iteration updates in norm 2, i.e.:
        $\boldsymbol{t} = \tilde{\mathcal{T}} \bullet_2 \hat{\boldsymbol{a}}_k \bullet_3 \hat{\boldsymbol{a}}_k$; $\hat{\varphi}_k \leftarrow \|\boldsymbol{t}\|_2$; $\hat{\boldsymbol{a}}_k \leftarrow \dfrac{\boldsymbol{t}}{\hat{\varphi}_k}$
    end for
    Pick the trial $(m)$ having the largest $\hat{\varphi}_k$
    Refine $\hat{\boldsymbol{a}}_k, \hat{\varphi}_k$ by $M_2$ extraneous iterations
    Deflation:$\tilde{\mathcal{T}} \leftarrow \tilde{\mathcal{T}} - \hat{\varphi}\ \hat{\boldsymbol{a}}_k \otimes \hat{\boldsymbol{a}}_k \otimes \hat{\boldsymbol{a}}_k$
end for
De-whitening: $\boldsymbol{B} = (\boldsymbol{W}^T)^{\dagger}$; $\boldsymbol{a}_k = \hat{\varphi}_k \boldsymbol{B}\hat{\boldsymbol{a}}_k$; $\varphi = \dfrac{1}{\hat{\varphi}_k^2}$ for $k = 1, \dots, K$
Back to norm 1: $\alpha = \|\boldsymbol{a}_k\|_1$; $\boldsymbol{a}_k \leftarrow \dfrac{\boldsymbol{a}_k}{\alpha}$; $\varphi_k \leftarrow \dfrac{\varphi_k}{\alpha}$ for $k = 1, \dots, K$ (our suggestion)
```

This new tensor enjoys the relationship

$$\tilde{\mathcal{T}} = \sum_{k=1}^{K} \varphi_k^{-1/2}\ \tilde{\boldsymbol{a}}_k \otimes \tilde{\boldsymbol{a}}_k \otimes \tilde{\boldsymbol{a}}_k.$$

The conclusion is that $\tilde{\mathcal{T}}$ ideally admits an *orthogonal* Canonical Polyadic (CP) decomposition; see *e.g.* [11, 12] for an introduction. Many algorithms have been devised for this kind of decomposition, including the pair-sweeping CoM algorithm, or Joint Approximate Diagonalization (JAD) algorithms [10]. But authors in [2] utilized the tensor power iteration [15] to extract the dominant "eigenvector[4]", $\hat{\boldsymbol{a}}$, and dominant "eigenvalue", $\hat{\varphi}$, of $\tilde{\mathcal{T}}$ and then proceeded by deflation, *i.e.* $\tilde{\mathcal{T}} \leftarrow \tilde{\mathcal{T}} - \hat{\varphi}\ \hat{\boldsymbol{a}} \otimes \hat{\boldsymbol{a}} \otimes \hat{\boldsymbol{a}}$, to get the remaining ones.

Unfortunately, except the tensor power iteration [15], there is no pseudo code in [2] describing the entire algorithm. Our description in Algorithm 1 hopefully fills this lack; we used this algorithm in our subsequent computer experiments in Section 5.2.

4 Proposed method

In this section, we detail our contributions, namely a non negative Canonical Polyadic (CP) decomposition executed on tensor (3). Therefore, it should not be considered as a modified version of the Robust Tensor Power Method, since it is a completely different

[4] Recall that there exist several definitions of tensor eigenvectors [19, 23]. The definition used in [2] – and hence here – is $\mathcal{T} \bullet \boldsymbol{v} \bullet \boldsymbol{v} = \lambda \boldsymbol{v}$, which has the undesirable property that λ depends on the norm of $\boldsymbol{v}$. In fact, if $(\lambda, \boldsymbol{v})$ is an eigenpair, then so is $(\alpha\lambda, \alpha\boldsymbol{v})$ for any nonzero α. This is analyzed in *e.g.* [23].

Algorithm 2 AO-ADMM for Problem (6)

```
Input: $\boldsymbol{\mathcal{T}}, K, \rho, Q$
Output: $\boldsymbol{A}, \boldsymbol{\varphi}$
for $q = 1 : Q$ (loop on starting points) do
    Initialize $\boldsymbol{H}_1, \boldsymbol{H}_2, \boldsymbol{H}_3$
    Initialize $\boldsymbol{U}_1, \boldsymbol{U}_2, \boldsymbol{U}_3$
    repeat
        for $n = 1, 2, 3$ do
            $\boldsymbol{T} = \boldsymbol{\mathcal{T}}_{(n)}$ (Unfold mode $n$)
            $\boldsymbol{Z} = \odot_{j \neq n} \boldsymbol{H}_j$
            Compute the Cholesky factor $\boldsymbol{L}$ of $(\boldsymbol{Z}^{\mathsf{T}} \boldsymbol{Z} + \rho \boldsymbol{I})$
            Update $\boldsymbol{H}_n$ as in Alg.1 of [18]:
            (a) $\boldsymbol{G}_n \leftarrow \boldsymbol{L}^{-\mathsf{T}} \boldsymbol{L}^{-1} [\boldsymbol{Z}^{\mathsf{T}} \boldsymbol{T} + \rho (\boldsymbol{U}_n + \boldsymbol{H}_n)^{\mathsf{T}}]$      # Auxiliary variables
            (b) $\boldsymbol{H}_n \leftarrow \max[\boldsymbol{0}, \boldsymbol{G}_n^{\mathsf{T}} - \boldsymbol{U}_n]$      # Primal variables
            (c) $\boldsymbol{U}_n \leftarrow \boldsymbol{U}_n + \boldsymbol{H}_n - \boldsymbol{G}_n^{\mathsf{T}}$      # Dual variables
        end for
    until some termination criterion
    for $k = 1 : K$ do $\boldsymbol{a}_k = H_1(:, k) / \|H_1(:, k)\|_1$; $\varphi_k = \prod_n \|H_n(:, k)\|_1$ end for
end for
Pick the best among the $Q$ estimates.
```

way of computing the unknown parameters. It is clear that $\boldsymbol{\mathcal{T}}$ in (5) corresponds to a CP decomposition structure. As we seek $\boldsymbol{A}$ and $\boldsymbol{\varphi}$, which represent probability values, it would be inevitable to consider non negativity constraint during the decomposition process. However, at least theoretically, we do not need to impose symmetry during the course of the CP decomposition, even if (5) reveals that $\boldsymbol{\mathcal{T}}$ is a symmetric tensor, since it will be expected to be obtained automatically.

Generally, computing a tensor low-rank approximation is ill-posed, except if some constraints, such as *non negativity* or *orthogonality*, are imposed [12, 20]. Taking this into account, in order to obtain reliable estimated quantities, $\boldsymbol{A}$ and $\boldsymbol{\varphi}$, we consider the following minimization in $\mathbb{R}_+$:

$$\min_{\boldsymbol{A}, \boldsymbol{\varphi}} \|\boldsymbol{\mathcal{T}} - \sum_{k=1}^{K} \boldsymbol{\varphi}(k)\, \boldsymbol{a}_k \otimes \boldsymbol{a}_k \otimes \boldsymbol{a}_k\|_2 \qquad \text{s.t.} \quad \boldsymbol{A} \in \mathbb{R}_+^{D \times K},\ \boldsymbol{\varphi} \in \mathbb{R}_+^{K}. \tag{6}$$

Since (6) is a non convex problem, Alternating Optimization (AO) is usually executed for such a problem [13, 18, 26]. AO is also known as Block Coordinate Descent (BCD) whose convergence has been discussed in [7]. When the cost function is simply least squares with no regularization, AO converted into the well-known method Alternating Least Squares (ALS) [18]. A recent modification named AO-ADMM [18] has been done to improve AO by the means of Alternating Direction Method of Multipliers (ADMM) [8], so that AO can handle constraints as well. Algorithm 2 shows the details of applying AO-ADMM to our problem (6).

It should be noted that in order to ensure the uniqueness of the CP decomposition in Algorithm 2, the rank of $\boldsymbol{\mathcal{T}}$, K, should be less than the *expected rank* R^o, or than R_s^o

in the symmetric case [12], with:

$$R^o = \left\lceil \frac{D^3}{3D-2} \right\rceil, \; R^o_s = \left\lceil \frac{(D+1)(D+2)}{6} \right\rceil,$$

Therefore, because the proposed algorithm only uses the third order moment tensor $\boldsymbol{\mathcal{T}}$, it is capable of handling over-complete cases ($K > D$) for estimating probabilities of hidden multi-view variables, up to $K = O(D^2)$.

5 Simulations

In this section we describe computer simulations aiming at comparing the proposed method with that of [2] in terms of performance in estimating the probability of topics and the conditional probabilities of words given the topics. In Subsection 5.1, we explain how one can generate synthetic data according to the graphical model in Fig. 1. The relative error $e_{\boldsymbol{\varphi}}$ (resp. $e_{\boldsymbol{A}}$) in estimating $\boldsymbol{\varphi}$ (resp. $\boldsymbol{A}$) made by both methods are reported in Subsection 5.2. If $\hat{\boldsymbol{\varphi}}$ is an estimate of $\boldsymbol{\varphi}$, it is defined as:

$$e_{\boldsymbol{\varphi}} = \frac{\|\boldsymbol{\varphi} - \hat{\boldsymbol{\varphi}}\|_2}{\|\boldsymbol{\varphi}\|_2}$$

5.1 Data generation

In this subsection we explain how one can generate a set of synthetic encoded words that are related to the same topic (cf. Fig. 1), with these two properties simultaneously:

- conditionally independent given the topic
- having the same conditional distribution

The first step to generate a synthetic data set is to define the arbitrary distributions $\varphi(k)$ and $f_k(d)$ for all $(k, \boldsymbol{u}_d) \in \mathcal{H} \times \Omega$. As mentioned before, the values of $\varphi(k)$ and $f_k(d)$ are stored in a K-dimensional vector $\boldsymbol{\varphi}$ and in a $D \times K$ matrix $\boldsymbol{A}$ respectively.

In order to generate synthetic data according to graphical model in Fig. 1, it is actually useful to use their cumulative distributions, $\Phi(k) = Prob(h \leq k)$ and $F_k(d) = Prob(\gamma \leq d \,|\, h = k)$. $\boldsymbol{\Phi}$ is a $K \times 1$ vector constructed by summing[5] the entries of $\boldsymbol{\varphi}$. Similarly, matrix $\boldsymbol{F}$ is a $D \times K$ matrix obtained by summing the entries of matrix $\boldsymbol{A}$. Our generative algorithm is as follows:

- draw $z \in [0,\ 1]$, and pick $k = \Phi^{-1}(z)$, by selecting the first entry in $\boldsymbol{\Phi}$ that is larger than z.
- for each $\ell \in \{1, 2, \ldots, L\}$,
  - draw $z_\ell \in [0,\ 1]$, and pick $\gamma(\ell) = F_k^{-1}(z_\ell)$, by selecting the first entry in the kth column of $\boldsymbol{F}$ that is larger than z_ℓ.
  - set $\boldsymbol{x}_\ell = \boldsymbol{u}_{\gamma(\ell)}$.

[5] With this goal, the matlab function `cumsum()` can be used.

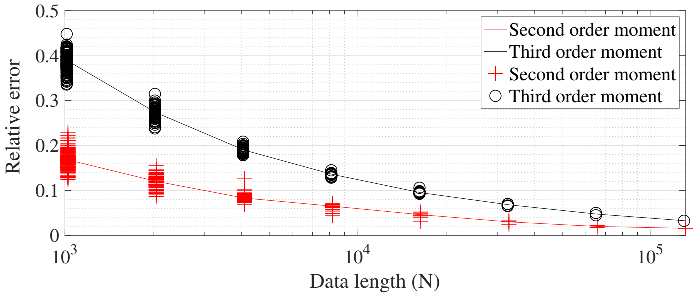

Fig. 2: Consistency of moment estimates. In this figure, the larger the data length, the fewer the number of realizations, to keep the total block length (and hence the computational load) constant.

In the second step, we utilize the averages of large number of realizations, say $N = 2^{14}$, to obtain an acceptable approximation of $\boldsymbol{P}$ and $\boldsymbol{\mathcal{T}}$. In each realization, we draw a random encoded topic, k, in the way described above. Then, according to the chosen topic, we draw three random encoded words, $\{\boldsymbol{x}_p = \boldsymbol{u}_{\gamma(p)}, \boldsymbol{x}_q = \boldsymbol{u}_{\gamma(q)}, \boldsymbol{x}_r = \boldsymbol{u}_{\gamma(r)}\}$, such that they satisfy the two properties mentioned above. At the end, by using the following averages, we have an *empirical* approximation of second and third order moments with sample moments below:

$$\boldsymbol{P}_e = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{u}_{\gamma(p)} \otimes \boldsymbol{u}_{\gamma(q)} \tag{7}$$

$$\boldsymbol{\mathcal{T}}_e = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{u}_{\gamma(p)} \otimes \boldsymbol{u}_{\gamma(q)} \otimes \boldsymbol{u}_{\gamma(r)} \tag{8}$$

Moment consistency As mentioned in Section 2, it can be easily shown that (2) and (3) are respectively equal to (4) and (5). Therefore, the above sample estimates of (2) and (3), *i.e.* (7) and (8), should converge to the true moments $\boldsymbol{P}$ and $\boldsymbol{\mathcal{T}}$ defined in (4) and (5). Hence, it seems essential to check the consistency of generated data in order to make sure that sample moments indeed converge to the true joint probabilities. To this end, we generated N realizations of our synthetic data set, $\boldsymbol{x}_\ell = \boldsymbol{u}_{\gamma(\ell)}$, (in a way explained above) with the following parameter values: $K = 4, D = 6, N = 2^{17}$. We chose a uniform distribution for φ, and K arbitrary distributions (cf. Section 5.2) to build a synthetic matrix $\boldsymbol{A}$. Lastly, we calculated sample moments in (7) and (8), and compared them to the true ones (4) and (5). Fig. 2 reports the discrepancy between both in terms of Euclidean norm. As can be seen in Fig. 2, the generated data in the way described above are consistent for $N > 10000$.

5.2 Performance

Results as a function of N. As mentioned before, the main goal is the estimation of $\boldsymbol{A}$ and φ. We report in this section the comparison results obtained with only one dictionary size, $D = 8$, with a fixed number of topics, $K = 4$, and for various data lengths up to $N = 2^{17}$. The power method is run with with $M_1 = 10$ and $M_2 = 5$, and the proposed algorithm with $Q = 100$. Moreover, the results are obtained for one particular realization[6] of matrix $\boldsymbol{A}$ and vector $\boldsymbol{\varphi}$:

$$\boldsymbol{A} = \begin{bmatrix} 0.162 & 0.211 & 0.000 & 0.000 \\ 0.082 & 0.130 & 0.000 & 0.000 \\ 0.022 & 0.235 & 0.000 & 0.403 \\ 0.196 & 0.423 & 0.000 & 0.038 \\ 0.174 & 0.000 & 0.276 & 0.119 \\ 0.104 & 0.000 & 0.133 & 0.439 \\ 0.113 & 0.000 & 0.119 & 0.000 \\ 0.147 & 0.000 & 0.473 & 0.000 \end{bmatrix}, \; \boldsymbol{\varphi} = \begin{bmatrix} 0.256 \\ 0.163 \\ 0.201 \\ 0.380 \end{bmatrix}.$$

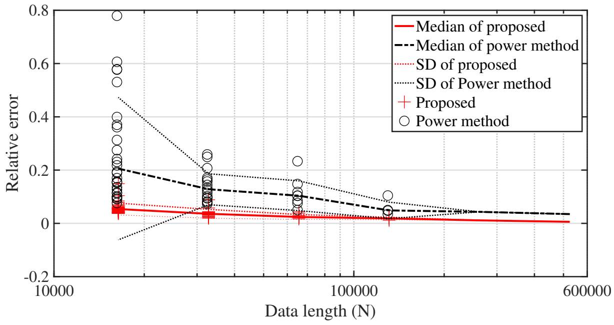

Fig. 3: Performance (relative error) in estimating $\boldsymbol{A}$

In order to compare the performance of both methods, we run the above simulation for increasing data lengths (note that the larger the data length, the fewer the number of realizations, to keep a computational load constant). To be more precise, by keeping the values of parameters $D = 8$, $K = 4$, we increase exponentially the value of N from 2^{14} to 2^{17}. For each value of N, we generated several synthetic data sets of size N in the way that has been explained in Subsection 5.1, and then the methods have been applied to the obtained data to compare relative errors as a function of of data length. The results are reported in Fig. 3-4; each black circle (resp. red cross) corresponds to the error of power method (resp. proposed method) in estimating $\boldsymbol{\varphi}$ or $\boldsymbol{A}$ for a specific realization of a particular data length, N. To ease comparison, the median (resp. standard deviation) among realizations is also plotted in solid (resp. dotted) line for every data length. As Figures 3 and 4 show, the error of the proposed method converges faster to zero as the data length increases, and unlike the Robust tensor power method [2], it appears to be much more robust, *i.e.* smaller standard deviation of the relative error.

[6] Results on other particular realizations may be found in [25].

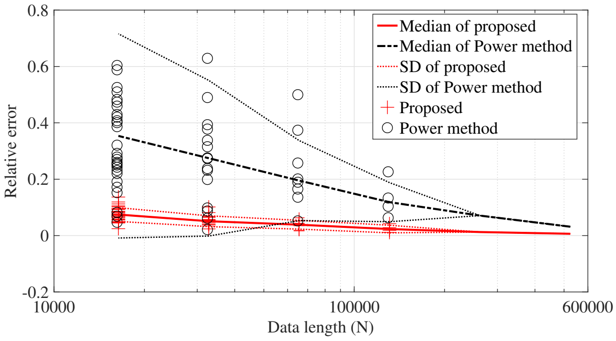

Fig. 4: Performance (relative error) in estimating φ

Results for random $\boldsymbol{A}$ and φ. When running extensive computer simulations, one can assume a uniform distribution for the topics, and K uniform distributions for the conditional probability of words given the topic, which are actually the columns of $\boldsymbol{A}$. Once $\boldsymbol{A}$ and φ are obtained, one can proceed exactly the same way as explained in Subsection 5.1 to generate synthetic datasets and sample moments. For space reasons, we only report in Table 1 the results obtained for a few values of (D, K), averaged over 20 independent trials. These extensive computer experiments confirm the robustness and superiority of the proposed algorithm compared to the power method.

Table 1: Median (left) and standard deviation (right) of error in estimating $\boldsymbol{A}$ and φ, with data length $N = 2^{17} \approx 1.2\,10^6$, and $K = 4$. In each cell, top: proposed method, bottom: power method.

Relative error in estimating $\boldsymbol{A}$

D	6	7	8	9	10	6	7	8	9	10
Proposed	0.087	0.077	0.070	0.104	0.073	0.100	0.123	0.083	0.104	0.120
Robust power method	0.382	0.178	0.243	0.181	0.118	0.313	0.314	0.386	0.327	0.235

Relative error in estimating φ

D	6	7	8	9	10	6	7	8	9	10
Proposed	0.145	0.105	0.115	0.096	0.128	0.223	0.118	0.213	0.285	0.315
Robust power method	0.657	0.283	0.310	0.287	0.228	0.545	0.524	0.591	0.407	0.399

6 Conclusion

In this paper, we focus on text mining as a special case of data mining: given the observation of words of several documents, we aim at estimating the probabilities of topics, φ, (as hidden variable cf. Fig. 1) as well as the conditional probabilities of words given topics, $\boldsymbol{A}$. For this purpose, we use (5), which is the same equation as in [2] relating the third order moment, $\mathcal{T}$, of observed words to desired probabilities. However, our cost function is more appropriate since non negativity is imposed. In fact, the Robust tensor power method [2] sometimes delivers negative values for estimated probabilities [25]. This difference appears in our computer results, which show that the performance of the proposed method is better compared to [2]. In addition, our algorithm only requires $\mathcal{T}$ as input, unlike the algorithm in [2], which permits to address the over-complete regime where number of hidden variables is larger than that of multi-view variables.

Further improvements of the power method. As mentioned in Section 3, the Robust tensor power method [2], combined with deflation, attempts to estimate eigenvectors and eigenvalues of the whitened tensor. One known problem with deflation, which has been observed even for matrices [14], is that the extracted eigenvectors may loose orthogonality because of rounding errors. The consequence is that the same (dominant) eigenvector may show up several times, especially if the array dimensions become large. This problem can be fixed by a re-orthogonalization of initial vectors with previously found vectors. Suppose that $\boldsymbol{V}(i) \overset{\text{def}}{=} [\tilde{\boldsymbol{a}}_1, \dots \tilde{\boldsymbol{a}}_i]$ is the matrix of the eigenvectors already obtained at iteration i. At the next iteration, instead of choosing simply a random vector $\boldsymbol{\theta}$ as in [2], we propose to pick it in the subspace orthogonal to $\boldsymbol{V}(i)$: $\tilde{\boldsymbol{\theta}} = \boldsymbol{\theta} - \boldsymbol{V}\boldsymbol{V}^{\mathsf{T}}\boldsymbol{\theta}$ is the next initial vector. In even larger dimensions, it may be necessary to also re-orthogonalize the intermediate iterates generated by the tensor power method, because of rounding errors. Another improvement consists in symmetrizing the sample estimates $\mathcal{T}_e$ and $\boldsymbol{P}_e$.

Over-complete regime. We previously mentioned the ability of our algorithm to address the so-called over-complete regimes, where $K > D$, up to $K = O(D^2)$. In this context, deflation is not possible, and if K eigenvectors are desired, the tensor power method must be run from K initial values, located close enough to the true eigenvectors. But the latter are unknown, which imposes to draw a very large number (growing as a Dth power) of initial vectors. We believe this solution is not workable. The sufficient condition requested in [3] is not only inaccurately defined, but is also very unlikely to be satisfied in practice.

Perspectives. This work must be continued to allow the application of the algorithm we have proposed to real databases. The implementation of certain operations must be revised in large dimensions to allow firstly to save memory space, secondly to limit rounding errors, and thirdly to reduce biases specific to large dimensions.

References

1. Anandkumar, A., Foster, D.P., Hsu, D., Kakade, S.M., Liu, Y.: A spectral algorithm for latent Dirichlet allocation. In: NIPS. vol. 25, pp. 1–9 (2012)
2. Anandkumar, A., Ge, R., Hsu, D., Kakade, S.M., Telgarsky, M.: Tensor decompositions for learning latent variable models. J. Machine Learning Research **15**, 2773–2832 (Aug 2014)
3. Anandkumar, A., Ge, R., Janzamin, M.: Analyzing tensor power method dynamics in overcomplete regime. Jour. Machine Learning Research **18**, 1–40 (Jan 2017)
4. Anandkumar, A., Hsu, D., Janzamin, M., Kakade, S.: When are overcomplete topic models identifiable? uniqueness of tensor Tucker decompositions with structured sparsity. Jour. Machine Learning Research **16**, 2643–2694 (Dec 2015)

5. Anandkumar, A., Hsu, D., Kakade, S.M.: A method of moments for mixture models and hidden Markov models. In: 25th Conf. Learning Theory. vol. 23, pp. 33.1–33.34 (2012)
6. Atrey, P.K., Hossain, M.A., El Saddik, A., Kankanhalli, M.S.: Multimodal fusion for multimedia analysis: a survey. Multimedia systems **16**(6), 345–379 (2010)
7. Beck, A., Tetruashvili, L.: On the convergence of block coordinate descent type methods. SIAM journal on Optimization **23**(4), 2037–2060 (2013)
8. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends in Machine learning **3**(1), 1–122 (2011)
9. Chang, K.W., Yih, W.T., Meek, C.: Multi-relational latent semantic analysis. In: Empirical Methods in Natural Language Proc. pp. 1602–1612. Seattle (Oct 2013)
10. Comon, P., Jutten, C. (eds.): Handbook of Blind Source Separation. Academic Press, Oxford UK, Burlington USA (2010)
11. Comon, P.: Independent component analysis, a new concept? Signal Proc., Elsevier **36**(3), 287–314 (1994)
12. Comon, P.: Tensors: a brief introduction. IEEE Sig. Proc. Magazine **31**(3), 44–53 (2014)
13. Comon, P., Luciani, X., De Almeida, A.L.: Tensor decompositions, alternating least squares and other tales. Jour. Chemometrics **23**(7-8), 393–405 (2009)
14. Cullum, J., Willoughby, R.: Lanczos Algorithms, vol. I. Birkhauser (1985)
15. De Lathauwer, L., Comon, P., et al.: Higher-order power method, application in Independent Component Analysis. In: NOLTA. pp. 91–96. Las Vegas (Dec 1995)
16. De Lathauwer, L., De Moor, B., Vandewalle, J.: On the best rank-1 and rank-(r 1, r 2,..., rn) approximation of higher-order tensors. SIAM journal on Matrix Analysis and Applications **21**(4), 1324–1342 (2000)
17. Han, J., Pei, J., Kamber, M.: Data mining: concepts and techniques. Elsevier (2011)
18. K.Huang, Sidiropoulos, N., et al.: A flexible and efficient algorithmic framework for constrained matrix and tensor factorization. IEEE Trans. Sig. Proc. **64**(19), 5052–5065 (2016)
19. Lim, L.H.: Singular values and eigenvalues of tensors: a variational approach. In: IEEE SAM Workshop. Puerto Vallarta, Mexico (13-15 Dec 2005)
20. Lim, L.H., Comon, P.: Nonnegative approximations of nonnegative tensors. Jour. Chemometrics **23**, 432–441 (Aug 2009)
21. Murphy, K.P.: Machine Learning, A Probabilistic Perspective. MIT press, London (2012)
22. Pearson, K.: Contributions to the mathematical theory of evolution. Philosophical Transactions of the Royal Society of London. A **185**, 71–110 (1894)
23. Qi, L., Luo, Z.: Tensor Analysis, Spectral Theory and Special tensors. SIAM (2017)
24. Ramage, D., Manning, C.D., Dumais, S.: Partially labeled topic models for interpretable text mining. In: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 457–465. ACM (2011)
25. Sobhani, E., Comon, P., Babaie-Zadeh, M.: Data mining with tensor decompositions. In: Gretsi. Lille (Aug 26-29 2019)
26. Sobhani, E., Sadeghi, M., Babaie-Zadeh, M., Jutten, C.: A robust ellipse fitting algorithm based on sparsity of outliers. In: 2017 25th European Sig. Proc. Conference (EUSIPCO). pp. 1195–1199. IEEE (2017)
27. Whittaker, J.: Graphical models in applied multivariate statistics. Wiley Publishing (2009)