Data mining with tensor decompositions

Elaheh Sobhani, Pierre Comon, Massoud Babaie-Zadeh

Data mining with tensor decompositions

Elaheh SOBHANI[1], Pierre COMON[1], Massoud BABAIE-ZADEH[2]

[1]GIPSA-Lab, Univ. Grenoble Alpes, CNRS, F-38000 Grenoble, France

[2]Sharif University of Technology, Tehran, Iran

prenom.nom@gipsa-lab.grenoble-inp.fr

Résumé – La fouille de textes est abordée en identifiant des modèles multi-vues. Suite aux travaux récents de A.Anandkumar, nous montrons que les algorithmes précédemment proposés présentent des inconvénients. L'algorithme que nous proposons permet de décomposer un tenseur non négatif construit avec des probabilités empiriques. Il se compare favorablement aux précédents sur des données synthétiques.

Abstract – Text mining is addressed by identifying multiview models. Following the recent works of A.Anandkumar, we show that the algorithms previously proposed exhibit important drawbacks. The algorithm subsequently proposed permits to decompose a nonnegative tensor built with sample joint probabilities. It compares favorably to the former on synthetically generated data sets.

1 Introduction

Multi-view models are useful in the context of multimodal recordings (such as text and audio), and offer a means to improve the estimation of latent variables that are present in several recordings. In this respect, they are related to data fusion. The model we assume does not impose any particular distribution (Gaussian, Dirichlet, etc). The notation is as follows.

Let L be the number of views, $\boldsymbol{x}_\ell$ the observed views, $\ell \in \mathcal{L} = \{1, 2, \ldots, L\}$, and h a discrete latent variable taking K possible integer values, say in $\mathcal{H} = \{1, 2, \ldots, K\}$ with probability $\varphi(k) = Prob(h = k)$. Every view belongs to a known dictionary $\Omega = \{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_D\}$ of cardinality D, so that there exists a mapping δ (not necessarily injective) from $\mathcal{L}$ to Ω such that $\boldsymbol{x}_\ell = \boldsymbol{u}_{\delta(\ell)}$. Then the key assumption is that views are statistically independent conditionally to h. More precisely, if we denote $f_k(d) = Prob(\boldsymbol{x} = \boldsymbol{u}_d | h = k)$, this means that the joint distribution of $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_L]$ can be written as:

$$p_{\boldsymbol{X}}(\boldsymbol{u}_{\delta(1)}, \ldots, \boldsymbol{u}_{\delta(L)}) = \sum_{k=1}^{K} \varphi(k)\, f_k(\delta(1)) \ldots f_k(\delta(L)) \quad (1)$$

The above equation is sometimes referred to as the *naive Bayes* model. The goal is to estimate the quantities appearing in the right-hand side of (1) from realizations of $\boldsymbol{X}$.

In the context of text mining, $\boldsymbol{x}_\ell$ may correspond to words and h to a topic, for instance. As a function of the words appearing in a document containing L words, each belonging to a known dictionary of D possible words, it is thus possible to infer what is the probability that a topic is addressed therein.

The problem presentation we make in Sections 1-3 is the same as that described in [1,2], but adapted to the vocabulary of the Signal & Data Processing readership for the sake of clarity.

2 Generative Model

To generate a data set, we need first to define the distributions $\varphi(k)$ and $f_k(d)$ for all $(k, \boldsymbol{u}_d) \in \mathcal{H} \times \Omega$. The values of $\varphi(k)$ are stored in a K-dimensional vector $\boldsymbol{\varphi}$. Similarly, the values of $f_k(d)$ are stored in a $D \times K$ matrix $\boldsymbol{A}$.

But it is actually useful to use their cumulative distributions, $\Phi(k) = Prob(h \leq k)$ and $F_k(d)$ with an appropriate encoding, as we shall see. The values of $\Phi(k)$ are stored in a $K \times 1$ vector $\boldsymbol{\Phi}$ obtained from $\boldsymbol{\varphi}$ by summing[1] its entries. Similarly, the values of $F_k(d)$ are obtained by summing the entries of matrix $\boldsymbol{A}$ and stored in a $D \times K$ matrix $\boldsymbol{F}$. Our generative algorithm goes along the following lines:

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

3 Moments

As in [1], to ease the presentation without restricting the generality, we may assume that $\boldsymbol{u}_d$ are the columns of the $D \times D$ identity matrix. Because of this choice made for $\boldsymbol{u}_\ell$, the joint probability (1) can be obtained by expectation as explained now. By definition of the encoding of $\boldsymbol{x}_\ell$, we have:

$$\boldsymbol{x}_p \otimes \boldsymbol{x}_q \otimes \boldsymbol{x}_r = \boldsymbol{u}_{\delta(p)} \otimes \boldsymbol{u}_{\delta(q)} \otimes \boldsymbol{u}_{\delta(r)} \quad (2)$$

This tensor has only one nonzero entry, which means that by summing N realizations, we obtain an estimate of the number

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

of occurrences. Hence averaging N realizations yields the joint probability distribution. In other words,

$$\hat{p}_{x_p,x_q,x_r}(\boldsymbol{u}_{\delta(p)},\boldsymbol{u}_{\delta(q)},\boldsymbol{u}_{\delta(r)}) = \widehat{\mathbb{E}}\{\boldsymbol{x_p}\otimes\boldsymbol{x}_q\otimes\boldsymbol{x}_r\}\Big|_{\delta(p),\delta(q),\delta(r)} \quad (3)$$

where $\hat{}$ denotes sample estimate. In the sequel, we shall need the moments of order two and three, which correspond to double and triple joint probabilities, denoted as follows:

$$\boldsymbol{P} = \mathbb{E}\{\boldsymbol{x}_p\otimes\boldsymbol{x}_q\} \quad (4)$$

$$\boldsymbol{\mathcal{T}} = \mathbb{E}\{\boldsymbol{x}_p\otimes\boldsymbol{x}_q\otimes\boldsymbol{x}_r\} \quad (5)$$

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). Note that $\{\delta(p),\ \delta(q),\ \delta(r)\}$ may not be different because δ is not injective.

4 Previous approaches

We describe in this section the approaches proposed by A.Anandkumar [1, 2]. These approaches may seem attractive at first glance, but it turns out that they are not usable in practice for the reasons we subsequently point out. In fact, there is no computer experiment in the latter papers, which may explain why the serious problems encountered in these algorithms have not been detected.

4.1 Power method with deflation

In [1], the authors propose to use the two moments defined in (4) and (5). Because of the Bayesian rule (1) and (3), these moments enjoy the following relations:

$$\boldsymbol{P} = \sum_{k=1}^{K}\varphi_k\ \boldsymbol{a}_k\otimes\boldsymbol{a}_k \quad (6)$$

$$\boldsymbol{\mathcal{T}} = \sum_{k=1}^{K}\varphi_k\ \boldsymbol{a}_k\otimes\boldsymbol{a}_k\otimes\boldsymbol{a}_k \quad (7)$$

where $\boldsymbol{a}_k$ denotes the kth column of matrix $\boldsymbol{A}$ defined in Section 2. Because of (6), matrix $\boldsymbol{P}$ is theoretically positive semidefinite, since φ_k are positive numbers. Hence, in a similar way as had been done for Blind Source Separation [3], 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 $\boldsymbol{W}=\boldsymbol{U}\boldsymbol{D}^{-1/2}$. Then we have

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

if $\tilde{\boldsymbol{a}}_k \stackrel{\text{def}}{=} \sqrt{\varphi_k}\,\boldsymbol{W}^{\mathsf{T}}\boldsymbol{a}_k$. In other words, the matrix $\tilde{\boldsymbol{A}}$ containing $\tilde{\boldsymbol{a}}_k$ as columns is now $K\times K$ orthogonal: $\tilde{\boldsymbol{A}}\tilde{\boldsymbol{A}}^{\mathsf{T}}=\boldsymbol{I}$. However, if matrix $\boldsymbol{P}$ is estimated via the mere averaging (4) as proposed in [1], $\boldsymbol{P}$ may have negative eigenvalues because of estimation errors; this has been overlooked in [1].

Next, this whitening matrix is applied to tensor $\boldsymbol{\mathcal{T}}$ to yield $\tilde{\boldsymbol{\mathcal{T}}} \stackrel{\text{def}}{=} \boldsymbol{\mathcal{T}}\bullet\boldsymbol{W}\bullet\boldsymbol{W}\bullet\boldsymbol{W}$. This new tensor enjoys the relationship

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

The conclusion is that $\tilde{\boldsymbol{\mathcal{T}}}$ ideally admits an *orthogonal* Canonical Polyadic (CP) decomposition; see *e.g.* [4] for an introduction. Many algorithms have been devised for this task, including the pair-sweeping CoM algorithm, or Joint Approximate Diagonalization (JAD) algorithms [3]. But it is proposed in [1] to execute the tensor power iteration [5] to extract the dominant "eigenvector[2]" of $\tilde{\boldsymbol{\mathcal{T}}}$ and then proceed by deflation to get the remaining ones.

One known problem with deflation, which has been observed even for matrices, is that the eigenvectors extracted may loose orthogonality because of rounding errors. The consequence is that the same (dominant) eigenvector may show up several times. This has been also overlooked in [1]. We propose in Section 5 to fix the problem by a re-orthogonalization with previously found vectors. For small size problem, drawing an initial value in the orthogonal subspace is sufficient.

4.2 Diagonalization of two moment matrices

In [2], a joint diagonalization algorithm is promoted and uses two moment marices, namely $\boldsymbol{P}$ and a matrix contraction $\boldsymbol{T}(\boldsymbol{\eta})=\boldsymbol{\mathcal{T}}\bullet\boldsymbol{\eta}$, where $\boldsymbol{\eta}$ is a randomly drawn vector. The idea is very interesting, but the algorithm unfortunately has never been implemented and tested, according to the available literature. We describe it shortly in this section, and shall not report its poor performances in Section 6 for reasons of space.

The Algorithm A of [2] starts with an SVD of $\boldsymbol{P}$ as $\boldsymbol{P}=\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathsf{T}}$. Then the matrix $\boldsymbol{B}(\boldsymbol{\eta})=\boldsymbol{U}^{\mathsf{T}}\boldsymbol{T}(\boldsymbol{\eta})\boldsymbol{V}\,(\boldsymbol{U}^{\mathsf{T}}\boldsymbol{P}\boldsymbol{V})^{-1}$ is computed, as well as its K dominant eigenvectors, $\boldsymbol{\xi}_k$. Then an estimate of the columns of matrix $\boldsymbol{A}$ is given by $\hat{\boldsymbol{a}}_k=\boldsymbol{U}\boldsymbol{\xi}_k$, up to a scaling factor depending on how $\hat{\boldsymbol{a}}_k$ are normalized. The 'eigenvalues" of $\boldsymbol{\mathcal{T}}$ can be obtained in a second stage by contraction as $\hat{\varphi}_k=\boldsymbol{\mathcal{T}}\bullet\hat{\boldsymbol{a}}_k\bullet\hat{\boldsymbol{a}}_k\bullet\hat{\boldsymbol{a}}_k$.

5 The algorithms proposed

5.1 Re-orthogonalization of the Power Method

Let $\boldsymbol{V}(i) \stackrel{\text{def}}{=} [\tilde{\boldsymbol{a}}_1,\dots\tilde{\boldsymbol{a}}_i]$ be the matrix containing the eigenvectors already output at iteration i. To compute the next iteration, we draw a random initial vector $\boldsymbol{\theta}$ and orthogonalize it with respect to $\boldsymbol{V}$ by assuming $\tilde{\boldsymbol{\theta}}=\boldsymbol{\theta}-\boldsymbol{V}\boldsymbol{V}^{\mathsf{T}}\boldsymbol{\theta}$ as initial vector, instead of $\boldsymbol{\theta}$ as in [1]. This modification is sufficient in small dimensions. However, in larger dimensions, it will be necessary

[2]Recall that there exist several definitions of tensor eigenvectors. The definition used in [1] – and hence here – is $\boldsymbol{\mathcal{T}}\bullet\boldsymbol{v}\bullet\boldsymbol{v}=\lambda\boldsymbol{v}$, 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.

to also re-orthogonalize the intermediate iterates generated by the Power Method because of rounding errors.

5.2 Algorithm under nonnegative constraint

As can be seen in (7), $\boldsymbol{\mathcal{T}}$ has a CP decomposition stucture. Since we expect to obtain $\boldsymbol{A}$ and $\boldsymbol{\varphi}$, which contain probability values, nonnegativity is an essential constraint that should be taken into account during the decomposition. Although (7) reveals that $\boldsymbol{\mathcal{T}}$ is a symmetric tensor, we do not need to impose symmetry during the course of the CP decomposition.

CP decomposition viewed as a sort of low-rank approximation is ill-posed in general, unless some constraints such as nonnegativity are imposed [4]. Therefore, we consider the following minimization in order to obtain reliable estimated quantities, $\boldsymbol{A}$ and $\boldsymbol{\varphi}$, in $\mathbb{R}_+$:

$$\min_{\boldsymbol{A},\boldsymbol{\varphi}} \|\boldsymbol{\mathcal{T}} - \sum_{k=1}^{K} \boldsymbol{\varphi}(k)\boldsymbol{a}_k \otimes \boldsymbol{a}_k \otimes \boldsymbol{a}_k\|_2 \quad \text{s.t.} \quad \boldsymbol{A} \in \mathbb{R}_+^{D\times K},\ \boldsymbol{\varphi} \in \mathbb{R}_+^{K}. \tag{8}$$

Obviously, (8) is a nonconvex problem, and Alternating Optimization (AO) is usually utilized for such a problem [6–8]. AO is also known as Block Coordinate Descent (BCD) and its convergence has been discussed in [9]. Alternating Least Squares (ALS) is a special case of AO when the cost function is simply least squares with no regularization [6].

Algorithm 1 AO-ADMM for Problem (8)

Input: $\boldsymbol{\mathcal{T}}, K, \rho$
Output: $\boldsymbol{A}, \boldsymbol{\varphi}$

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 [6]:
    (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
$\boldsymbol{A} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_K]$, $\boldsymbol{\varphi} = [\varphi_1, \varphi_2, \ldots, \varphi_K]^T$

Recently, a new method has been introduced in [6], which improves AO by the means of Alternating Direction Method of Multipliers (ADMM) [10], and can handle constraints. This new method is called AO-ADMM [6]. Applying AO-ADMM to our problem is straightforward, and its details are shown in Algorithm 1. The only (mild) limitation of Algorithm 1 is related to the uniqueness of the CP decomposition. From (7), the rank of $\boldsymbol{\mathcal{T}}$ is K, which should be less than the *expected rank* R^o to ensure uniqueness, or than R_s^o in the symmetric case [4], with:

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

Hence, we have the condition, $K = O(D^2)$ for the proposed algorithm. All in all, we only need moments of order three, *i.e.* $\boldsymbol{\mathcal{T}}$, to obtain probabilities of hidden multi-view variables, and addressing over-complete regimes up to $K = O(D^2)$ is possible (contrary to the Power Method).

6 Computer experiments

We explained in Section 2 how one can generate a set of multi-view variables that are related to the same hidden variable, with these two properties simultaneously:

- conditionally independent given the hidden variable
- having the same conditional distribution

In order to obtain an acceptable approximation of second and third order moments, we assume a large number of trials, say N=100,000. In each trial, we draw a random hidden variable, h, in the way described in Section 2. Then, for the chosen hidden variable, we draw three random variable, $(\boldsymbol{x}_p, \boldsymbol{x}_q, \boldsymbol{x}_r)$, such that they satisfy the two properties mentioned above (cf. Sections 1-2). At the end, by averaging arrays $\boldsymbol{P}$ and $\boldsymbol{\mathcal{T}}$ obtained in each trial, we have an approximation of second and third order moments respectively.

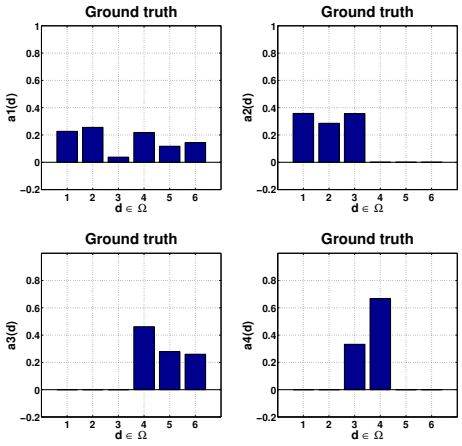

FIG. 1: Original $\boldsymbol{A}$

For the sake of representation, we consider $K = 4$, thereby we generate four distinct distributions which are actually the columns of $\boldsymbol{A}$. Because of the lack of space, we able to show only the comparison between various methods with only one single size of dictionary, $D = 6$, so that $\boldsymbol{A}$ is 6×4.

Figure 1 shows the values of the $D = 6$ conditional probabilities contained in each $\boldsymbol{a}_k$, $1 \leq k \leq K$. Fig. 2 reports the

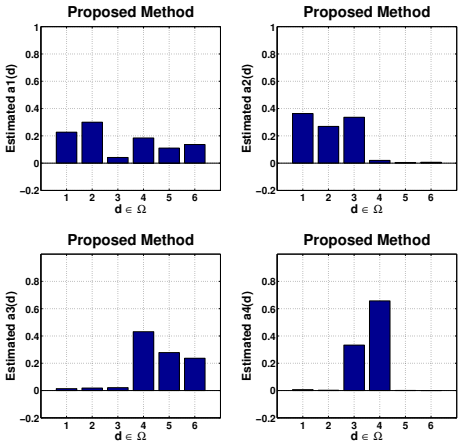

FIG. 2: Method proposed in Section 5.2

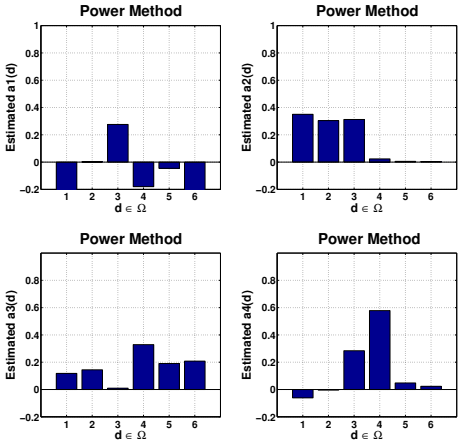

FIG. 3: The Robust tensor power method [1]

corresponding estimated values obtained by the algorithm described in Section 5.2 (after optimal permutation to get them in the same order). As shown in Fig. 3, the Power method fails in delivering the correct columns of $\boldsymbol{A}$, and suffers from negative values. This is the main difference between [1] and the proposed algorithm which imposes the nonnegative constraint. The performances in estimating vector $\boldsymbol{\varphi}$ have been compared in Fig. 4. It is clear that the result of proposed method is closer to the original vector $\boldsymbol{\varphi}$.

7 Discussion

Estimates of $\boldsymbol{P}$ and $\boldsymbol{\mathcal{T}}$ from data via (4) and (5) may not yield symmetric arrays, which raises serious problems. It can be fixed by forcing the symmetry of the estimate of $\boldsymbol{P}$ before running the Power Method. Next, the Power Method is not the most adequate algorithm to compute a CP decomposition, even under the orthogonal constraints; in fact, it is very sensitive to initialization[3], and may output several times the same dominant eigenvector. This can be faced by the re-orthogonalization procedure described in Section 5.1. A more serious limitation of the algorithms described in Section 4 is the output of negative values. The algorithm proposed in Section 5 fixes the problem, as demonstrated in Section 6. In addition, it can handle overcomplete cases, where $K > D$, provided K is not larger than $O(D^2)$, whereas the Power Method cannot.

[3] In [1], many initial values are tried, and the best one is picked. In Section 6, the Power Method has been initialized with "only" 10 initial values for each eigenvector, to keep a reasonable computational complexity.

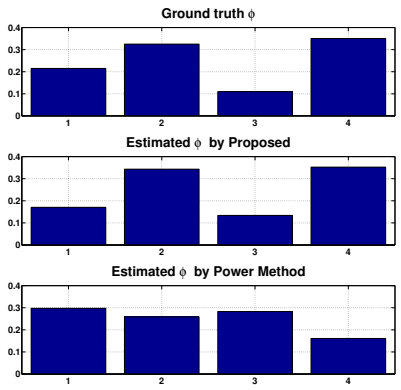

FIG. 4: Performance of methods in estimating $\boldsymbol{\varphi}$

References

[1] A. Anandkumar, R. Ge, D. Hsu, S. M. Kakade, and M. Telgarsky, "Tensor decompositions for learning latent variable models," *J. Machine Learning Research*, vol. 15, pp. 2773–2832, Aug. 2014.

[2] A. Anandkumar, D. Hsu, and S. M. Kakade, "A method of moments for mixture models and hidden markov models," in *25th Annual Conference on learning Theory*, vol. 23, 2012, pp. 33.1–33.34. [Online]. Available: arxiv:1203.0683

[3] P. Comon and C. Jutten, Eds., *Handbook of Blind Source Separation*. Oxford UK, Burlington USA: Academic Press, 2010.

[4] P. Comon, "Tensors: a brief introduction," *IEEE Signal Processing Magazine*, vol. 31, no. 3, pp. 44–53, 2014.

[5] L. DeLathauwer, P. Comon *et al.*, "Higher-order power method, application in Independent Component Analysis," in *NOLTA*, Las Vegas, Dec. 1995, pp. 91–96.

[6] K.Huang, N. Sidiropoulos, and A. P. Liavas, "A flexible and efficient algorithmic framework for constrained matrix and tensor factorization," *IEEE Trans. Signal proc.*, vol. 64, no. 19, pp. 5052–5065, 2016.

[7] E. Sobhani, M. Sadeghi, M. Babaie-Zadeh, and C. Jutten, "A robust ellipse fitting algorithm based on sparsity of outliers," in *2017 25th European Signal Processing Conference (EUSIPCO)*. IEEE, 2017, pp. 1195–1199.

[8] P. Comon, X. Luciani, and A. L. De Almeida, "Tensor decompositions, alternating least squares and other tales," *Jour. Chemometrics*, vol. 23, no. 7-8, pp. 393–405, 2009.

[9] A. Beck and L. Tetruashvili, "On the convergence of block coordinate descent type methods," *SIAM journal on Optimization*, vol. 23, no. 4, pp. 2037–2060, 2013.

[10] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.