Journal of Machine Learning Research 23 (2022) 1-50 Submitted 4/21; Revised 3/22; Published 5/22

Online Nonnegative CP-dictionary Learning for Markovian Data

Hanbaek Lyu* HLYU@MATH.WISC.EDU
Department of Mathematics
University of Wisconsin - Madison, WI 53709, USA

Christopher Strohmeier C.STROHMEIER@MATH.UCLA.EDU
Department of Mathematics
University of California, Los Angeles, CA 90095, USA

Deanna Needell DEANNA@MATH.UCLA.EDU
Department of Mathematics
University of California, Los Angeles, CA 90025, USA

Editor: Barbara Engelhardt

Abstract

Online Tensor Factorization (OTF) is a fundamental tool in learning low-dimensional interpretable features from streaming multi-modal data. While various algorithmic and theoretical aspects of OTF have been investigated recently, a general convergence guarantee to stationary points of the objective function without any incoherence or sparsity assumptions is still lacking even for the i.i.d. case. In this work, we introduce a novel algorithm that learns a CANDECOMP/PARAFAC (CP) basis from a given stream of tensor-valued data under general constraints, including nonnegativity constraints that induce interpretability of the learned CP basis. We prove that our algorithm converges almost surely to the set of stationary points of the objective function under the hypothesis that the sequence of data tensors is generated by an underlying Markov chain. Our setting covers the classical i.i.d. case as well as a wide range of application contexts including data streams generated by independent or MCMC sampling. Our result closes a gap between OTF and Online Matrix Factorization in global convergence analysis for CP-decompositions. Experimentally, we show that our algorithm converges much faster than standard algorithms for nonnegative tensor factorization tasks on both synthetic and real-world data. Also, we demonstrate the utility of our algorithm on a diverse set of examples from image, video, and time-series data, illustrating how one may learn qualitatively different CP-dictionaries from the same tensor data by exploiting the tensor structure in multiple ways.

Keywords: Online tensor factorization, CP-decomposition, dictionary learning, Markovian data, convergence analysis

1. Introduction

In modern signal processing applications, there is often a critical need to analyze and understand data that is high-dimensional (many variables), large-scale (many samples), and multi-modal (many attributes). For unimodal (vector-valued) data, *matrix factorization* provides a powerful tool for one to describe data in terms of a linear combination of factors

*. Corresponding author. All codes are available at `https://github.com/HanbaekLyu/OnlineCPDL`

or atoms. In this setting, we have a data matrix $X \in \mathbb{R}^{d\times n}$, and we seek a factorization of X into the product WH for $W \in \mathbb{R}^{d\times R}$ and $H \in \mathbb{R}^{R\times n}$. Including two classical matrix factorization algorithms of Principal Component Analysis (PCA) Wold et al. (1987) and Nonnegative Matrix Factorization (NMF) Lee and Seung (1999), this problem has gone by many names over the decades, each with different constraints: dictionary learning, factor analysis, topic modeling, component analysis. It has applications in text analysis, image reconstruction, medical imaging, bioinformatics, and many other scientific fields more generally Sitek et al. (2002); Berry and Browne (2005); Berry et al. (2007); Chen et al. (2011); Taslaman and Nilsson (2012); Boutchko et al. (2015); Ren et al. (2018).

A *tensor* is a multi-way array that is a natural generalization of a matrix (which is itself a 2-mode tensor) and is suitable for representing multi-modal data. As matrix factorization is for unimodal data, *tensor factorization* (TF) provides a powerful and versatile tool that can extract useful latent information out of multi-modal data tensors. As a result, tensor factorization methods have witnessed increasing popularity and adoption in modern data science. One of the standard tensor factorization paradigms is CANDECOMP/PARAFAC (CP) decomposition Tucker (1966); Harshman et al. (1970); Carroll and Chang (1970). In this setting, given a n-mode data tensor $\mathcal{X}$, one seeks n *loading matrices* $U^{(1)}, \ldots, U^{(n)}$, each with R columns, such that $\mathcal{X}$ is approximated by the sum of the outer products of the respective columns of U_i's. In other words, regarding the n-mode tensor $\mathcal{X}$ as the joint probability distribution of n random variables, the CP-decomposition approximates such a joint distribution as the sum of R product distributions, where the columns of the loading matrices give one-dimensional marginal distributions used to form the product distributions. A particular instance of CP-decomposition is when the data tensor and all of its loading matrices are required to have nonnegative entries. As pointed out in the seminal work of Lee and Seung Lee and Seung (1999) (in the matrix case), imposing a nonnegativity constraint in the decomposition problem helps one to learn interpretable features from multi-modal data.

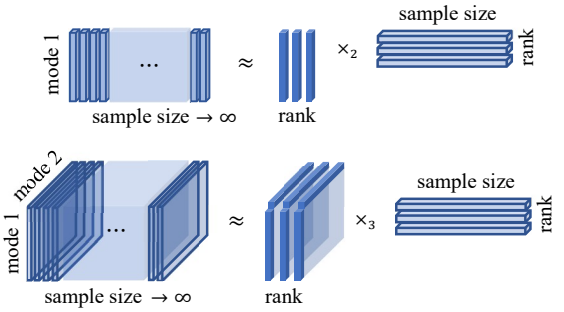


Figure 1: Illustration of online MF (top) and online CP-decomposition (bottom). n-mode tensors arrive sequentially and past data are not stored. One seeks n loading matrices that give approximate decomposition of all past data.

Besides being multi-modal, another unavoidable characteristic of modern data is its enormous volume and the rate at which new data are generated. *Online learning* algorithms permit incremental processing that overcomes the sample complexity bottleneck inherent to batch processing, which is especially important when storing the entire data set is cumbersome. Not only do online algorithms address capacity and accessibility, but they also have the ability to learn qualitatively different information than offline algorithms for data that admit such a "sequential" structure (see e.g. Lyu et al. (2020a)). In the literature, many "online" variants of more classical "offline" algorithms have been extensively studied — NMF Mairal et al. (2010); Guan et al. (2012); Lyu et al. (2020b), TF Zhou et al. (2016); Huang et al. (2015); Zhou et al. (2018); Du et al. (2018); Smith et al. (2018), and dictionary learning Rambhatla et al. (2019); Arora et al. (2015, 2014); Koppel et al. (2017). Online

Tensor Factorization (OTF) algorithms with suitable constraints (e.g., nonnegativity) can serve as valuable tools that can extract interpretable features from multi-modal data.

1.1 Contribution

In this work, we develop a novel algorithm and theory for the problem of *online CP-dictionary learning*, where the goal is to progressively learn a dictionary of rank-1 tensors (CP-dictionary) from a stream of tensor data. Namely, given n-mode nonnegative tensors $(\mathcal{X}_t)_{t\geq 0}$, we seek to find an adaptively changing sequence of nonnegative CP-dictionaries such that the current CP-dictionary can approximate all tensor-valued signals in the past as a suitable nonnegative linear combination of its CP-dictionary atoms (see Figure 1 in Section 1). Our framework is flexible enough to handle general situations of an arbitrary number of modes in the tensor data, arbitrary convex constraints in place of the nonnegativity constraint, and a sparse representation of the data using the learned rank-1 tensors. In particular, our problem setting includes online nonnegative CP-decomposition.

Furthermore, we rigorously establish that under mild conditions, our online algorithm produces a sequence of loading matrices that converge almost surely to the set of stationary points of the objective function. In particular, our convergence results hold not only when the sequence of input tensors $(\mathcal{X}_t)_{t\geq 0}$ are independent and identically distributed (i.i.d.), but also when they form a Markov chain or functions of some underlying Markov chain. Such a theoretical convergence guarantee for online NTF algorithms has not been available even under the i.i.d. assumption on the data sequences. The relaxation to the Markovian setting is particularly useful in practice since often the signals have to be sampled from some complicated or unknown distribution, and obtaining even approximately independent samples is difficult. In this case, the Markov Chain Monte Carlo (MCMC) approach provides a powerful sampling technique (e.g., sampling from the posterior in Bayesian methods Van Ravenzwaaij et al. (2018) or from the Gibbs measure for Cellular Potts models Voss-Böhme (2012), or motif sampling from sparse graphs Lyu et al. (2019)), where consecutive signals can be highly correlated.

1.2 Approach

Our algorithm combines the Stochastic Majorization-Minimization (SMM) framework Mairal (2013b), which has been used for online NMF algorithms Mairal et al. (2010); Guan et al. (2012); Zhao et al. (2017); Lyu et al. (2020b), and a recent work on block coordinate descent with diminishing radius (BCD-DR) Lyu (2020). In SMM, one iteratively minimizes a recursively defined surrogate loss function $\hat{f}_t$ that majorizes the empirical loss function f_t. A premise of SMM is that $\hat{f}_t$ is convex so that it is easy to minimize, which is the case for online matrix factorization problems in the aforementioned references. However, in the setting of factorizing n-mode tensors, $\hat{f}_t$ is only convex in each of the n loading matrices and nonconvex jointly in all loading matrices. Our main algorithm (Algorithm 1) only approximately minimizes $\hat{f}_t$ by a single round of cyclic block coordinate descent (BCD) in the n loading matrices. This additional layer of relaxation causes a number of technical difficulties in convergence analysis. One of our crucial innovations to handle them is to use a search radius restriction during this process Lyu (2020), which is reminiscent of restricting

step sizes in stochastic gradient descent algorithms and is in some sense 'dual' to proximal modifications of BCD Grippo and Sciandrone (2000); Xu and Yin (2013).

Our convergence analysis on dependent data sequences uses the technique of "conditioning on a distant past", which leverages the fact that while the one-step conditional distribution of a Markov chain may be a constant distance away from the stationary distribution π, the N-step conditional distribution is exponentially close to π in N. This technique has been developed in Lyu et al. (2020b) recently to handle dependence in data streams for online NMF algorithms.

1.3 Related work

We roughly divide the literature on TF into two classes depending on *structured* or *unstructured* TF problems. The *structured TF problem* concerns recovering exact loading matrices of a tensor, where a structured tensor decomposition with loading matrices satisfying some incoherence or sparsity conditions is assumed. A number of works address this problem in the offline setting Tang and Shah (2015); Anandkumar et al. (2015); Sharan and Valiant (2017); Sun et al. (2017); Barak et al. (2015); Ma et al. (2016); Schramm and Steurer (2017). Recently, Rambhatla et al. (2020) addresses an online structured TF problem by reducing it to an online MF problem using sparsity constraints on all but one loading matrices.

On the other hand, in the *unstructured TF problem*, one does not make any modeling assumption on the tensor subject to a decomposition so there are no true factors to be discovered. Instead, given an arbitrary tensor, one tries to find a set of factors (matrices or tensors) that gives the best fit of a chosen tensor decomposition model. In this case, convergence to a globally optimal solution cannot be expected, and global convergence to stationary points of the objective function is desired. For offline problems, global convergence to stationary points of the block coordinate descent method is known to hold under some regularity assumptions on the objective function Grippo and Sciandrone (2000); Grippof and Sciandrone (1999); Bertsekas (1999). The recent works Zhou et al. (2016); Huang et al. (2015); Zhou et al. (2018); Du et al. (2018); Smith et al. (2018) on online TF focus on computational considerations and do not provide a convergence guarantee. For online NMF, almost sure convergence to stationary points of a stochastic majorization-minimization (SMM) algorithm under i.i.d. data assumption is well-known Mairal et al. (2010), which has been recently extended to the Markovian case in Lyu et al. (2020b). Similar global convergence for online TF is not known even under the i.i.d. assumption. The main difficulty of extending a similar approach to online TF is that the recursively constructed surrogate loss functions are nonconvex and cannot be jointly minimized in all n loading matrices when $n \geq 2$.

There are several recent works improving standard CP-decomposition algorithms such as the alternating least squares (ALS) (see, e.g., Kolda and Bader (2009)). Battaglino et al. (2018) proposes a randomized ALS algorithm, that subsamples rows from each factor matrix update, which is an overdetermined least squares problem. A similar technique of row subsampling was used in the context of high-dimensional online matrix factorization Mensch et al. (2017). Ma et al. (2018) proposed a randomized algorithm for online CP-decomposition but no theoretical analysis was provided. Also, CP-decomposition with structured factor matrices has been investigated in Goulart et al. (2015). On the other hand, Vannieuwenhoven

et al. (2015) considers a more efficient version of gradient descent type algorithms for CP-decomposition.

In the context of dictionary learning, there is an interesting body of work considering tensor-dictionary learning based on the Tucker-decomposition Shakeri et al. (2016); Ghassemi et al. (2017); Shakeri et al. (2018); Ghassemi et al. (2019); Shakeri et al. (2019). When learning a reconstructive tensor dictionary for tensor-valued data, one can impose additional structural assumptions on the tensor dictionary in order to better exploit the tensor structure of the data and to gain reduced computational complexity. While in this work we consider the CP-decomposition model for the tensor-dictionary (also in an online setting), the aforementioned works consider the Tucker-decomposition instead and obtain various results on sample complexity, identifiability of a Tucker dictionary, and local convergence.

While our approach largely belongs to the SMM framework, there are related works using stochastic gradient descent (SGD). In Zhao et al. (2017), an online NMF algorithm based on projected SGD with general divergence in place of the squared ℓ_2-loss is proposed, and convergence to stationary points to the expected loss function for i.i.d. data samples is shown. In Sun et al. (2018), a similar convergence result for stochastic gradient descent algorithms for *unconstrained* nonconvex optimization problems with Markovian data samples is shown. While none of these results can be directly applied to our setting of online NTF for Markovian data, it may be possible to develop an SGD based approach for our setting, and it will be interesting to compare the performance of the algorithms based on SMM and SGD.

1.4 Organization

In Section 2 we first give a background discussion on NTF and CP-decomposition and then state the main optimization problem we address in this paper (see (13)). In Section 3, we provide the main algorithm (Algorithm 1) and give an overview of the main idea. Section 4 states the main convergence result in this paper, Theorem 1, together with a discussion on necessary assumptions and key lemmas used for the proof. In Section 5 we give the proof of the main result, Theorem 1. In Section 6, we compare the performance of our main algorithm on the offline nonnegative CP-decomposition problem against other baseline algorithms – Alternating Least Squares and Multiplicative Update. We then illustrate our approach on a diverse set of applications in Section 7; these applications are chosen to showcase the advantage of being able to flexibly reshape multi-modal tensor data and learn CP-dictionary atoms for any desired group of modes jointly.

In Appendix A, we provide some additional background on Markov chains and Markov chain Monter Carlo sampling. Appendix B contains some auxiliary lemmas. In Appendix C, we provide a memory-efficient implementation (Algorithm 2) of Algorithm 1 that uses bounded memory regardless of the length of the data stream.

1.5 Notation

For each integer $k \geq 1$, denote $[k] = \{1, 2, \ldots, k\}$. Fix integers $n, I_1, \ldots, I_n \geq 1$. An n-mode tensor $\mathbf{X}$ of shape $I_1 \times \cdots \times I_n$ is a map $(i_1, \ldots, i_n) \mapsto \mathbf{X}(i_1, \ldots, i_n) \in \mathbb{R}$ from the multi-index set $[I_1] \times \cdots \times [I_n]$ into the real line $\mathbb{R}$. We identify 2-mode tensors with matrices and 1-mode tensors with vectors, respectively. We do not distinguish between vectors and columns

of matrices. For two real matrices A and B, we denote their Frobenius inner product as $\langle A, B\rangle := \mathrm{tr}(B^T A)$ whenever the sizes match. If we have N n-mode tensors $\mathbf{X}_1, \ldots, \mathbf{X}_N$ of the same shape $I_1 \times \cdots \times I_n$, we identify the tuple $[\mathbf{X}_1, \ldots, \mathbf{X}_N]$ as the $(n+1)$-mode tensor $\mathcal{X}$ of shape $I_1 \times \cdots \times I_n \times N$, whose the i^{th} slice along the $(n+1)^{\text{st}}$ mode equals $\mathbf{X}_i$. For given n-mode tensors $\mathbf{A}$ and $\mathbf{B}$, denote by $\mathbf{A} \odot \mathbf{B}$ and $\mathbf{A} \otimes_{kr} \mathbf{B}$ their Hadamard (pointwise) product and Katri-Rao product, respectively. When $\mathbf{B}$ is a matrix, for each $1 \leq j \leq n$, we also denote their mode-j product by $\mathbf{A} \times_j \mathbf{B}$. (See Kolda and Bader (2009) for an excellent survey of tensor algorithms, albeit with notation that differs from our own).

2. Background and problem formulation

2.1 CP-dictionary learning and nonnegative tensor factorization

Assume that we are given N observed vector-valued signals $x_1, \ldots, x_N \in \mathbb{R}^d_{\geq 0}$. Fix an integer $R \geq 1$ and consider the following approximate factorization problem (see (2) for a precise statement)

$$[x_1, \ldots, x_N] \approx [u_1, \ldots, u_R] \times_2 H \qquad \Longleftrightarrow \qquad X \approx UH, \tag{1}$$

where $\times_2$ denotes the mode-2 product, $X = [x_1, \ldots, x_N] \in \mathbb{R}^{d \times N}_{\geq 0}$, $U = [u_1, \ldots, u_R] \in \mathbb{R}^{d \times R}_{\geq 0}$, and $H \in \mathbb{R}^{R \times N}_{\geq 0}$. The right hand side (1) is the well-known *nonnegative matrix factorization* (NMF) problem, where the use of nonnegativity constraint is crucial in obtaining a "parts-based" representation of the input signals Lee and Seung (1999). Such an approximate factorization learns R *dictionary atoms* $u_1, \ldots, u_R$ that together can approximate each observed signal x_j by using the nonnegative linear coefficients in the j^{th} column of H. The factors U and H in (1) above are called the *dictionary* and *code* of the data matrix X, respectively. They can be learned by solving the following optimization problem

$$\underset{U' \in \mathbb{R}^{d \times R}_{\geq 0}, H' \in \mathbb{R}^{R \times N}_{\geq 0}}{\arg\min} \left(\|X - U'H'\|_F^2 + \lambda \|H'\|_1 \right), \tag{2}$$

where $\lambda \geq 0$ is a regularization parameter that encourages a sparse representation of the columns of X over the columns of U. Note that (2) is also known as a *dictionary learning problem* Olshausen and Field (1997); Engan et al. (1999); Lewicki and Sejnowski (2000); Elad and Aharon (2006); Lee et al. (2005), especially when $R \geq d$.

Next, suppose we have N observed n-mode tensor-valued signals $\mathbf{X}_1, \ldots, \mathbf{X}_N \in \mathbb{R}^{I_1 \times \cdots \times I_n}_{\geq 0}$. A direct tensor analogue of the NMF problem (1) would be the following:

$$[\mathbf{X}_1, \ldots, \mathbf{X}_N] \approx [\mathbf{D}_1, \ldots, \mathbf{D}_R] \times_{n+1} H \qquad \Longleftrightarrow \qquad \mathcal{X} \approx \mathcal{D} \times_{n+1} H, \tag{3}$$

where $\mathcal{X} = [\mathbf{X}_1, \ldots, \mathbf{X}_N] \in \mathbb{R}^{I_1 \times \cdots \times I_n \times N}_{\geq 0}$, $\mathcal{D} = [\mathbf{D}_1, \ldots, \mathbf{D}_R] \in \mathbb{R}^{I_1 \times \cdots \times I_n \times R}_{\geq 0}$, and $H \in \mathbb{R}^{R \times N}_{\geq 0}$. As before, we call $\mathcal{D}$ and H above the dictionary and code of the data tensor $\mathcal{X}$, respectively. Note that this problem is equivalent to (1) since

$$\|\mathcal{X} - \mathcal{D} \times_{n+1} H\|_F^2 = \|\mathrm{MAT}(\mathcal{X}) - \mathrm{MAT}(\mathcal{D}) \times_2 H\|_F^2, \tag{4}$$

where $\mathrm{MAT}(\cdot)$ is the matricization operator that vectorizes (using lexicographic ordering of entries) each slice with respect to the last mode. For instance, $\mathrm{MAT}([\mathbf{X}_1, \ldots, \mathbf{X}_N])$ is a $(I_1 \cdots I_n) \times N$ matrix whose i^{th} column is the vectorization of $\mathbf{X}_i$.

Now, consider imposing an additional structural constraint on the dictionary atoms $\mathbf{D}_1, \dots, \mathbf{D}_N$ in (3). Specifically, suppose we want each $\mathbf{D}_i$ to be the sum of R rank 1 tensors. Equivalently, we assume that there exist *loading matrices* $[U^{(1)}, \dots, U^{(n)}] \in \mathbb{R}_{\geq 0}^{I_1 \times R} \times \cdots \times \mathbb{R}_{\geq 0}^{I_n \times R}$ such that

$$[\mathbf{D}_1, \dots, \mathbf{D}_R] = \texttt{Out}(U^{(1)}, \dots, U^{(n)}) \tag{5}$$

$$:= \left[\bigotimes_{k=1}^{n} U^{(k)}(:,1), \bigotimes_{k=1}^{n} U^{(k)}(:,2), \dots, \bigotimes_{k=1}^{n} U^{(k)}(:,R) \right] \in \mathbb{R}_{\geq 0}^{I_1 \times \cdots \times I_n \times R}, \tag{6}$$

where $U^{(k)}(:,j)$ denotes the i^{th} column of the $I_k \times R$ matrix $U^{(k)}$ and $\otimes$ denotes the outer product. Note that we are also defining the operator $\texttt{Out}(\cdot)$ here, which will be used throughout this paper. In this case, the tensor factorization problem in (3) becomes (a more precise statement is given in (13))

$$[\mathbf{X}_1, \dots, \mathbf{X}_N] \approx \texttt{Out}(U^{(1)}, \dots, U^{(n)}) \times_{n+1} H. \tag{7}$$

When $N = 1$ and $\lambda = 0$, then $H \in \mathbb{R}_{\geq 0}^{R \times 1}$, so by absorbing the i^{th} entry of H into the $\mathbf{D}_i$, we see that the above problem (7) reduces to

$$\mathbf{X} \approx \sum \texttt{Out}(U^{(1)}, \dots, U^{(n)}) := \sum_{i=1}^{R} \bigotimes_{k=1}^{n} U^{(k)}(:,i), \tag{8}$$

which is the nonnegative CANDECOMP/PARAFAC (CP) decomposition problem Tucker (1966); Harshman et al. (1970); Carroll and Chang (1970). On the other hand, if $n = 1$ so that $\mathbf{X}_i$ are vector-valued signals, then (7) reduces to the classical dictionary learning problem (2). For these reasons, we refer to (7) as the *CP-dictionary learning* (CPDL) problem. We call the $(n+1)$-mode tensor $\texttt{Out}(U^{(1)}, \dots, U^{(n)}) = [\mathbf{D}_1, \dots, \mathbf{D}_R]$ in $\mathbb{R}^{(I_1 \times \cdots \times I_n \times R)}$ a *CP-dictionary* and the matrix $H \in \mathbb{R}_{\geq 0}^{R \times N}$ the *code* of the dataset $\mathcal{X} = [\mathbf{X}_1, \dots, \mathbf{X}_N]$, respectively. Here we call the rank-1 tensors $\mathbf{D}_i$ the *atoms* of the CP-dictionary.

2.2 Online CP-dictionary learning

Next, we consider an *online* version of the CPDL problem we considered in (7). Given a continuously arriving sequence of data tensors $(\mathbf{X}_t)_{t \geq 0}$, can we find an adaptively changing sequence of CP-dictionaries such that the current CP-dictionary can approximate all tensor-valued signals in the past as a suitable nonnegative linear combination of its CP-dictionary atoms (see Figure 1 in Section 1)? This online problem can be explicitly formulated as an empirical loss minimization framework, and we will also state an equivalent stochastic program (under some modeling assumption) that we rigorously address.

Fix constraint sets for code and loading matrices $\mathcal{C}^{\text{code}} \subseteq \mathbb{R}^{R \times b}$ and $\mathcal{C}^{(i)} \subseteq \mathbb{R}^{I_i \times R}$, $i = 1, \dots, n$, respectively (generalizing the nonnegativity constraints in Subsection 2.1). Write $\mathcal{C}^{\text{dict}} := \mathcal{C}^{(1)} \times \cdots \times \mathcal{C}^{(n)}$. For each $\mathcal{X} \in \mathbb{R}_{\geq 0}^{I_1 \times \cdots \times I_n \times b}$, $\mathcal{D} := [U^{(1)}, \dots, U^{(n)}] \in \mathbb{R}^{I_1 \times R} \times \cdots \times \mathbb{R}^{I_n \times R}$, $H \in \mathbb{R}^{R \times b}$, define

$$\ell(\mathcal{X}, \mathcal{D}, H) := \|\mathcal{X} - \texttt{Out}(\mathcal{D}) \times_{n+1} H\|_F^2 + \lambda \|H\|_1, \tag{9}$$

$$\ell(\mathcal{X}, \mathcal{D}) := \inf_{H \in \mathcal{C}^{\text{code}}} \ell(\mathcal{X}, \mathcal{D}, H), \tag{10}$$

where $\lambda \geq 0$ is a regularization parameter. Fix a sequence of non-increasing weights $(w_t)_{t\geq 0}$ in $(0, 1]$. Here $\mathcal{X}$ denotes a minibatch of b tensors in $\mathbb{R}^{I_1 \times \cdots \times I_n}$, so minimizing $\ell(\mathcal{X}, \mathcal{D})$ with respect to $\mathcal{D}$ amounts to fitting the CP-dictionary $\mathcal{D}$ to the minibatch of b tensors in $\mathcal{X}$.

The *online CP-dictionary learning* (online CPDL) problem is the following empirical loss minimization problem:

$$\text{Upon arrival of } \mathcal{X}_t\text{:} \quad \mathcal{D}_t \in \operatorname*{arg\,min}_{\mathcal{D} \in \mathcal{C}^{\text{dict}}} \big(f_t(\mathcal{D}) := (1 - w_t) f_{t-1}(\mathcal{D}) + w_t \, \ell(\mathcal{X}_t, \mathcal{D}) \big), \tag{11}$$

where f_t is the *empirical loss function* recursively defined by the weighted average in (11) with $f_0 \equiv 0$. One can solve the recursion in (11) and obtain the more explicit formula for the empirical loss:

$$f_t(\mathcal{D}) = \sum_{k=1}^{t} \ell(\mathcal{X}_k, \mathcal{D}) \, w_k^t, \qquad w_k^t := w_k \prod_{i=k+1}^{t} (1 - w_i). \tag{12}$$

The weight w_t in (11) controls how much we want our new loading matrices in $\mathcal{D}_t$ to deviate from minimizing the previous empirical loss f_{t-1} to adapting to the newly observed tensor data $\mathcal{X}_t$. In one extreme case of $w_t \equiv 1$, $\mathcal{D}_t$ is a minimizer of the time-t loss $\ell(\mathcal{X}_t, \cdot)$ and ignores the past f_{t-1}. If $w_t \equiv \alpha \in (0, 1)$ then the history is forgotten exponentially fast, that is, $f_t(\cdot) = \sum_{s=1}^{t} \alpha(1-\alpha)^{t-s} \, \ell(\mathcal{X}_s, \cdot)$. On the other hand, the 'balanced weight' $w_t = 1/t$ makes the empirical loss to be the arithmetic mean: $f_t(\cdot) = \frac{1}{t} \sum_{s=1}^{t} \ell(\mathcal{X}_s, \cdot)$, which is the choice made for the online NMF problem in Mairal et al. (2010). Therefore, one can choose the sequence of weights $(w_t)_{t\geq 1}$ in Algorithm 1 in a desired way to control the sensitivity of the algorithm to the newly observed data. That is, make the weights decay fast for learning average features and decay slow (or keep it constant) for learning trending features. We mention that our theoretical convergence analysis covers only the former case.

We note that the online CPDL problem (11) involves solving a constrained optimization problem for each t, which is practically infeasible. Hence we may compute a sub-optimal sequence $(\mathcal{D}_t)_{t\geq 0}$ of tuples of loading matrices (see Algorithm 1) and assess its asymptotic fitness to the original problem (11). We seek to perform some rigorous theoretical analysis at the expense of some suitable but non-restrictive assumption on the data sequence $\mathcal{X}_t$ as well as the weight sequence $(w_t)_{t\geq 1}$. A standard modeling assumption in the literature is to assume the data sequence $\mathcal{X}_t$ are independent and identically distributed (i.i.d.) according to some distribution π Mairal et al. (2010); Mairal (2013b); Mensch et al. (2017); Zhao et al. (2017). We consider a more relaxed setting where $\mathcal{X}_t$ is given as a function of some underlying Markov chain (see (A1)) and π is the stationary distribution of $(\mathcal{X}_t)_{t\geq 1}$ viewed as a stochastic process. Under this assumption, consider the following stochastic program

$$\operatorname*{arg\,min}_{\mathcal{D} \in \mathcal{C}^{\text{dict}}} \big(f(\mathcal{D}) := \mathbb{E}_{\mathcal{X} \sim \pi} \left[\ell(\mathcal{X}, \mathcal{D}) \right] \big), \tag{13}$$

where the *random* tensor $\mathcal{X}$ is sampled from the distribution π, and we call the function f defined in (13) the *expected loss function*. The connection between the online (11) and the stochastic (13) formulation of the CPDL problem is that, if the parameter space $\mathcal{C}^{\text{dict}}$ is

compact and the weights w_t satisfy some condition, then $\sup_{\mathcal{D}} |f_t(\mathcal{D}) - f(\mathcal{D})| \to 0$ almost surely as $t \to \infty$. (see Lemma 19 in Appendix B). Hence, under this setting, we seek to find a sequence $(\mathcal{D}_t)_{t\geq 1}$ that converges to a solution to (13). In other words, fitness to the *single* expected loss function f is enough to deduce the asymptotic fitness to *all* empirical loss functions f_t.

Once we find an optimal CP-dictionary $\mathcal{D}^* = \texttt{Out}(U^{(1)}, \dots, U^{(n)})$ for (13), then we can obtain an optimal code matrix $H = H(\mathcal{X}) \in \mathbb{R}^{R\times b}$ for each realization of the random tensor $\mathcal{X}$ by solving the convex problem in (10). Demanding optimality of the CP-dictionary $\mathcal{D}^*$ and leaving the code matrix $H = H(\mathcal{X})$ adjustable in this way is a more flexible framework for CP-decomposition of a random (as well as online) tensor than seeking a pair of jointly optimal CP-dictionary $\mathcal{D}^*$ and code matrix H^*, especially when the variation of the random tensor is large. However, if we have a single deterministic tensor $\mathcal{X}$ to be factorized, then these two formulations are equivalent since $\min_{\mathcal{D},\mathcal{H}} \ell(\mathcal{X}, \mathcal{D}, H) = \min_{\mathcal{D}} \min_{H} \ell(\mathcal{X}, \mathcal{D}, H)$.

3. The Online CP-Dictionary Learning algorithm

In this section, we state our main algorithm (Algorithm 1). For simplicity, we first give a preliminary version for the case of $n = 2$ modes, minibatch size $b = 1$, with nonnegativity constraints. Suppose we have learned $n = 2$ loading matrices $\mathcal{D}_{t-1} := [U^{(1)}_{t-1}, U^{(2)}_{t-1}]$ from the sequence $\mathcal{X}_1, \dots, \mathcal{X}_{t-1}$ of data tensors, $\mathcal{X}_i \in \mathbb{R}^{I_1\times I_2\times 1}$. Then we compute the updated loading matrices $\mathcal{D}_t = [U^{(1)}_t, U^{(2)}_t]$ by

$$\begin{cases} H_t & \leftarrow \underset{H\in\mathbb{R}^{R\times 1}_{\geq 0}}{\arg\min} \ell(\mathcal{X}_t, \mathcal{D}_{t-1}, H) \\ \hat{f}_t(\mathcal{D}) & \leftarrow (1-w_t)\hat{f}_{t-1}(\mathcal{D}) + w_t \ell(\mathcal{X}_t, \mathcal{D}, H_t) \\ U^{(1)}_t & \leftarrow \underset{U\in\mathbb{R}^{I_i\times R}_{\geq 0},\, \|U-U^{(1)}_{t-1}\|_F \leq c' w_t}{\arg\min} \hat{f}_t(U, U^{(2)}_{t-1}) \\ U^{(2)}_t & \leftarrow \underset{U\in\mathbb{R}^{I_i\times R}_{\geq 0},\, \|U-U^{(2)}_{t-1}\|_F \leq c' w_t}{\arg\min} \hat{f}_t(U^{(1)}_t, U), \end{cases} \tag{14}$$

where $\lambda \geq 0$ is an absolute constant and $(w_t)_{t\geq 1}$ is a non-increasing sequence of weights in $(0, 1]$. The recursively defined function $\hat{f}_t : \mathcal{D} = [U^{(1)}, \dots, U^{(n)}] \mapsto [0, \infty)$ is called the *surrogate loss function*, which is quadratic in each factor $U^{(i)}$ but is not jointly convex. Namely, when the new tensor data $\mathcal{X}_t$ arrives, one computes the code $H_t \in \mathbb{R}^{R\times 1}_{\geq 0}$ for $\mathcal{X}_t$ with respect to the previous loading matrices in $\mathcal{D}_{t-1}$, updates the surrogate loss function $\hat{f}_t$, and then *sequentially* minimizes it to find updated loading matrices within diminishing search radius $c'w_t$.

Note that the surrogate loss function $\hat{f}_t$ in (16) is defined by the same recursion that defines the empirical loss function in (11). However, notice that the loss term $\ell(\mathcal{X}_t, \mathcal{D})$ in the definition of the empirical loss function f_t in (13) involves optimizing over the code matrices H in (10), which should be done for every $\mathcal{X}_s$, $1 \leq s \leq t$, in order to evaluate f_t. On the contrary, in the definition of the surrogate loss function $\hat{f}_t$ in (16), this term is replaced with the sub-optimal loss $\ell(\mathcal{X}_t, \mathcal{D}, H_t)$, which is sub-optimal since H_t was found by decomposing $\mathcal{X}_t$ using the previous CP-dictionary $\texttt{Out}(\mathcal{D}_{t-1})$. From this, it is clear that $\hat{f}_t \geq f_t$ for all $t \geq 0$. In other words, $\hat{f}_t$ is a majorizing surrogate of f_t.

Now we state our algorithm in the general mode case in Algorithm 1.

Algorithm 1 Online CP-Dictionary Learning (online CPDL)

Input: $(\mathcal{X}_t)_{1\le t\le T}$ (minibatches of data tensors in $\mathbb{R}^{I_1\times\cdots\times I_n\times b}$); $[U_0^{(1)},\dots,U_0^{(n)}]\in \mathbb{R}^{I_1\times R}\times\cdots\times\mathbb{R}^{I_n\times R}$ (initial loading matrices); $c'>0$ (search radius constant);

Constraints: $\mathcal{C}^{(i)}\subseteq\mathbb{R}^{I_i\times R}$, $1\le i\le n$, $\mathcal{C}^{\text{code}}\subseteq\mathbb{R}^{R\times b}$ (e.g., nonnegativity constraints)

Parameters: $R\in\mathbb{N}$ (# of dictionary atoms); $\lambda\ge 0$ (ℓ_1-regularization parameter); $(w_t)_{t\ge 1}$ (weights in $(0,1]$);

Initialize surrogate loss $\hat{f}_0\equiv 0$;

For $t=1,\dots,T$ **do:**

Coding: Compute the optimal code matrix

$$H_t \leftarrow \underset{H\in\mathcal{C}^{\text{code}}\subseteq\mathbb{R}^{R\times b}}{\arg\min}\ \ell(\mathcal{X}_t,U_{t-1}^{(1)},\dots,U_{t-1}^{(n)},H);\quad \text{(using Algorithm 4)} \tag{15}$$

Update surrogate loss function:

$$\hat{f}_t(U^{(1)},\dots,U^{(n)})\leftarrow(1-w_t)\hat{f}_{t-1}(U^{(1)},\dots,U^{(n)})+w_t\,\ell(\mathcal{X}_t,U^{(1)},\dots,U^{(n)},H_t) \tag{16}$$

Update loading matrices by restricted cyclic block coordinate descent:

For $i=1,\dots,n$ **do:**

$$\mathcal{C}_t^{(i)}\leftarrow\left\{U\in\mathcal{C}^{(i)}\,\middle|\,\|U-U_{t-1}^{(i)}\|_F\le c'w_t\right\};\ (\triangleright\ \textit{Restrict the search radius by } c'w_t) \tag{17}$$

$$U_t^{(i)}\leftarrow\underset{U\in\mathcal{C}_t^{(i)}}{\arg\min}\ \hat{f}_t\left(U_t^{(1)},\dots,U_t^{(i-1)},U,U_{t-1}^{(i+1)},\dots,U_{t-1}^{(n)}\right);\quad (\triangleright\textit{Update the } i^{\text{th}}\textit{ loading matrix}) \tag{18}$$

End for

End for

Return: $[U_T^{(1)},\dots,U_T^{(n)}]\in\mathcal{C}^{(1)}\times\cdots\times\mathcal{C}^{(n)}$;

Algorithm 1 combines two key elements: stochastic majorization-minimization (SMM) Mairal (2013b) and block coordinate descent with diminishing radius (BCD-DR) Lyu (2020). SMM amounts to iterating the following steps: sampling new data points, constructing a strongly convex surrogate loss, and then minimizing the surrogate loss to update the current estimate. This framework has been successfully applied to online matrix factorization problems Mairal et al. (2010); Mensch et al. (2017). However, the biggest bottleneck in using a similar approach in the tensor case is that the surrogate loss function $\hat{f}_t$ in (16) is only block multi-convex, meaning that it is convex in each block of coordinates but nonconvex jointly. Hence we cannot find an exact minimizer for $\hat{f}_t$ to update all n loading matrices at the same time.

In order to circumvent this issue, one could try to perform a few rounds of block coordinate descent (BCD) on the surrogate loss function $\hat{f}_t$, which can be easily done since $\hat{f}_t$ is convex in each loading matrix. However, this results in sub-optimal loading matrices in each

iteration, causing a number of difficulties in convergence analysis. Moreover, global convergence of BCD to stationary points is not guaranteed in general even for the deterministic tensor CP-decomposition problems without constraints Kolda and Bader (2009), and such a guarantee is known only with some additional regularity conditions Grippof and Sciandrone (1999); Grippo and Sciandrone (2000); Bertsekas (1999). There are other popular strategies of using proximal Grippo and Sciandrone (2000) or prox-linear Xu and Yin (2013) modifications of BCD to improve convergence properties. While these methods only ensure square-summability $\sum_{t=1}^{\infty}\|\mathcal{D}_t - \mathcal{D}_{t-1}\|_F^2 < \infty$ of changes (see, e.g., (Xu and Yin, 2013, Lem 2.2)), we find it crucial for our stochastic analysis that we are able to control the individual changes $\|\mathcal{D}_t - \mathcal{D}_{t-1}\|_F$ of the loading matrices in each iteration. This motivates to use BCD with diminishing radius in Lyu (2020). More discussions on technical points in convergence analysis are given in Subsection 4.2.

The coding step in (15) is a convex problem and can be easily solved by a number of known algorithms (e.g., LARS Efron et al. (2004), LASSO Tibshirani (1996), and feature-sign search Lee et al. (2007)). As we have noted before, the surrogate loss function $\hat{f}_t$ in (16) is quadratic in each block coordinate, so each of the subproblems in the factor matrix update step in (18) is a constrained quadratic problem and can be solved by projected gradient descent (see Mairal et al. (2010); Lyu et al. (2020b)).

Notice that implementing Algorithm 1 may seem to require unbounded memory as one needs to store all past data $\mathcal{X}_1, \ldots, \mathcal{X}_t$ to compute the surrogate loss function $\hat{f}_t$ in (16). However, it turns out that there are certain bounded-sized statistics that aggregates the past information that are sufficient to parameterize $\hat{f}_t$ and also to update the loading matrices. This bounded memory implementation of Algorithm 1 is given in Algorithm 2, and a detailed discussion on the memory efficiency is relegated to Appendix C.

4. Convergence results

In this section, we state our main convergence result of Algorithm 1. Note that all results that we state here also apply to Algorithm 2, which is a bounded-memory implementation of Algorithm 1.

4.1 Statement of main results

We first layout all technical assumptions required for our convergence results to hold.

(A1) *The observed minibatch of data tensors $\mathcal{X}_t = [\mathbf{X}_{t;1}, \ldots, \mathbf{X}_{t;b}]$ are given by $\mathcal{X}_t = \varphi(Y_t)$, where Y_t is an irreducible and aperiodic Markov chain defined on a finite state space Ω and $\varphi : \Omega \to \mathbb{R}^{I_1 \times \cdots \times I_n \times b}$ is a bounded function. Denote the transition matrix and the unique stationary distribution of Y_t by P and π, respectively.*

(A2) *For each $1 \le i \le n$, the i^{th} loading matrix is constrained to a compact and convex subset $\mathcal{C}^{(i)} \subseteq \mathbb{R}^{I_i \times R}$ that contains at least two points. Furthermore, the code matrices H_t belong to a compact and convex subset $\mathcal{C}^{\text{code}} \subseteq \mathbb{R}^{R \times b}$.*

(A3) *The sequence of non-increasing weights $w_t \in (0, 1]$ in Algorithm 1 $\sum_{t=1}^{\infty} w_t = \infty$, and $\sum_{t=1}^{\infty} w_t^2 \sqrt{t} < \infty$. Furthermore, $w_{t+1}^{-1} - w_t^{-1} \le 1$ for all sufficiently large t.*

(A4) *The expected loss function f defined in* (13) *is continuously differentiable and has Lipschitz gradient.*

It is standard to assume that the sequence of signals is drawn from a data distribution of compact support in an independent fashion Mairal et al. (2010); Mairal (2013a), which enables the processing of large data by using i.i.d. subsampling of minibatches. However, when the signals have to be sampled from some complicated or unknown distribution, obtaining even approximately independent samples is difficult. In this case, Markov Chain Monte Carlo (MCMC) provides a powerful sampling technique (e.g., sampling from the posterior in Bayesian methods Van Ravenzwaaij et al. (2018) or from the Gibbs measure for Cellular Potts models Voss-Böhme (2012), or motif sampling from sparse graphs Lyu et al. (2019)), where consecutive signals could be highly correlated. (See Appendix A for more background on Markov chains and MCMC.)

An important notion in MCMC sampling is "exponential mixing" of the Markov chain[1]. For a simplified discussion, suppose in (A1) that our data tensors $\mathcal{X}_t$ themselves form a Markov chain with unique stationary distribution π. Under the assumption of finite state space, irreducibility, and aperiodicity in (A1), the Markov chain $\mathcal{X}_t$ "mixes" to the stationary distribution π at an exponential rate. Namely, for any $\varepsilon > 0$, one can find a constant $\tau = \tau(\varepsilon) = O(\log \varepsilon^{-1})$, called the "mixing time" of $\mathcal{X}_t$, such that the conditional distribution of $\mathcal{X}_{t+\tau}$ given $\mathcal{X}_t$ is within total variation distance ε from π regardless of the distribution of $\mathcal{X}_t$ (see (87) for the definition of total variation distance). This mixing property of Markov chains is crucial both for practical applications of MCMC sampling as well as our theoretical analysis. For instance, a common practice of using MCMC sampling to obtain approximate i.i.d. samples is to first obtaining a long Markov chain trajectory $(\mathcal{X}_t)_{t\geq 1}$ and then thinning it to the subsequence $(\mathcal{X}_{k\tau})_{k\geq 1}$ (Brooks et al., 2011, Sec. 1.11). Due to the choice of mixing time τ, this forms an ε-approximate i.i.d. samples from π.

However, thinning a Markov chain trajectory does not necessarily produce truly independent samples, so classical stochastic analysis that relies crucially on independence between data samples is not directly applicable. For instance, if $\mathcal{X}_t$ is a reversible Markov chain then the correlation within the subsequence is nonzero and at least of order ε (see Appendix A.2 for the definition of reversibility and detailed discussion). In order to obtain truly independent samples, one may independently re-initialize a Markov chain trajectory, run it for τ iterations, and keep the last samples in each run (e.g., see the discussion in Sun et al. (2018)). However, in both approaches, only one out of τ Markov chain samples are used for optimization, which could be extremely wasteful if the Markov chain converges to the stationary distribution slowly so that the implied constant in $\tau(\varepsilon) = O(\log \varepsilon^{-1})$ is huge.

Instead, our assumption on input signals in (A1) allows us to use every single sample in the same MCMC trajectory without having to "burn" lots of samples. Such Markovian extension of the classical OMF algorithm in Mairal et al. (2010) has recently been achieved in Lyu et al. (2020b), which has applications in dictionary learning, denoising, and edge inference problems for network data Lyu et al. (2021).

Assumption (A2) is also standard in the literature of dictionary learning (see Mairal et al. (2010); Mairal (2013b)). A particular instance of interest is when they are confined to having

1. For our analysis, it is in fact sufficient to have a sufficiently fast polynomial mixing of the Markov chain. See (A1)' in Appendix A for a relaxed assumption using countable state-space.

nonnegative entries, in which case the learned dictionary components give a "parts-based" representation of the input signals Lee and Seung (1999).

Assumption (A3) states that the sequence of weights $w_t \in (0, 1]$ we use to recursively define the empirical loss (11) and surrogate loss (16) does not decay too fast so that $\sum_{t=1}^{\infty} w_t = \infty$ but decay fast enough so that $\sum_{t=1}^{\infty} w_t^2 \sqrt{t} < \infty$. This is analogous to requirements for stepsizes in stochastic gradient descent algorithms, where the stepsizes are usually required to be non-summable but square-summable (see, e.g., Sun et al. (2018)). The additioanl factor $\sqrt{t}$ is used in our analysis to deduce the uniform convergence of the empirical loss f_t to the expected loss f (see Lemma 19 in Appendix B), which was also the case in the literature Mairal et al. (2010); Mairal (2013b); Mensch et al. (2017); Lyu et al. (2020b). Also, the condition $w_t^{-1} - w_{t-1}^{-1} \leq 1$ for all suficiently large t is equivalent to saying the recursively defined weights w_k^t in (12) are non-decreasing in k for all sufficiently large k, which is required to use Lemma 19 in Appendix B. We also remark that (A3) is implied by the following simpler condition:

(A3') *The sequence of non-increasing weights $w_t \in (0, 1]$ in Algorithm 1 satisfy either $w_t = t^{-1}$ for $t \geq 1$ or $w_t = \Theta(t^{-\beta}(\log t)^{-\delta})$ for some $\delta \geq 1$ and $\beta \in [3/4, 1)$.*

For Assumption (A4), we remark that it follows from the uniqueness of the solution of (15) (see (Mairal et al., 2010, Prop. 1)). This can be enforced for example by the elastic net penalization Zou and Hastie (2005). Namely, we may add a quadratic regularizer $\lambda' \|H\|_F^2$ to the loss function ℓ in (9) for some $\lambda' > 0$. Then the resulting quadratic function is strictly convex and hence it has a unique minimizer in the convex constraint set $\mathcal{C}^{\text{code}}$. (See (Mairal et al., 2010, Sec. 4.1) and (Lyu et al., 2020b, Sec. 4.1) for more detailed discussion on this assumption).

The main result in this paper, which is stated below in Theorem 1, states that the sequence $\mathcal{D}_t$ of CP-dictionaries produced by Algorithm 1 converges to the set of stationary points of the expected loss function f defined in (13). To the best of our knowledge, Theorem 1 is the first convergence guarantee for any online *constrained* dictionary learning algorithm for tensor-valued signals or as an online *unconstrained* CP-factorization algorithm, which have not been available even under the classical i.i.d. assumption on input signals. Recall that f_t and $\hat{f}_t$ denote the empirical and surrogate loss function defined in (11) and (16), respectively.

Theorem 1 *Suppose (A1)-(A3) hold. Let $(\mathcal{D}_t)_{t\geq 1}$ be an output of Algorithm 1. Then the following hold.*

(i) $\lim_{t\to\infty} \mathbb{E}[f_t(\mathcal{D}_t)] = \lim_{t\to\infty} \mathbb{E}[\hat{f}_t(\mathcal{D}_t)] < \infty$.

(ii) $f_t(\mathcal{D}_t) - \hat{f}_t(\mathcal{D}_t) \to 0$ *and* $f(\mathcal{D}_t) - \hat{f}_t(\mathcal{D}_t) \to 0$ *as* $t \to \infty$ *almost surely.*

(iii) *Further assume (A4). Then the distance (measured by block-wise Frobenius distance) between $\mathcal{D}_t$ and the set of all stationary points of f in $\mathcal{C}^{\text{dict}}$ converges to zero almost surely.*

We acknowledge that a similar asymptotic convergence result under Markovian dependency in data samples has been recently obtained in (Sun et al., 2018, Thm. 2) in the

context of stochastic gradient descent for nonconvex optimization problems. The results are not directly comparable since (Sun et al., 2018, Thm. 2) only handles unconstrained problems.

4.2 Key lemmas and overview of the proof of Theorem 1.

In this subsection, we state the key lemmas we use to prove Theorem 1 and illustrate our contribution to techniques for convergence analysis.

As we mentioned in Section 3, there is a major difficulty in convergence analysis in the multi-modal case $n \geq 2$ as the surrogate loss function $\hat{f}_t$ (see (16)) to be minimized for updating the loading matrices is only multi-convex in n blocks. Note that we can view our algorithm (Algorithm 1) as a multi-modal extension of stochastic majorization-minimization (SMM) in the sense that it reduces to standard SMM in the case of vector-valued signals ($n = 1$). We first list the properties of SMM that have been critically used in convergence analysis in related works Mairal et al. (2010); Mairal (2013b); Mensch et al. (2017); Lyu et al. (2020b):

1 (Surrogate Optimality) $\mathcal{D}_t$ is a minimizer of $\hat{f}_t$ over $\mathcal{C}^{\text{dict}}$.

2 (Forward Monotonicity) $\hat{f}_t(\mathcal{D}_{t-1}) \geq \hat{f}_t(\mathcal{D}_t)$.

3 (Backward Monotonicity) $\hat{f}_{t-1}(\mathcal{D}_{t-1}) \leq \hat{f}_{t-1}(\mathcal{D}_t)$.

4 (Second-Order Growth Property) $\hat{f}_t(\mathcal{D}_{t-1}) - \hat{f}_t(\mathcal{D}_t) \geq c\|\mathcal{D}_t - \mathcal{D}_{t-1}\|_F^2$ for some constant $c > 0$.

5 (Stability of Estimates) $\|\mathcal{D}_t - \mathcal{D}_{t-1}\|_F = O(w_t)$.

6 (Stability of Errors) For $h_t := \hat{f}_t - f_t \geq 0$, $|h_t(\mathcal{D}_t) - h_{t-1}(\mathcal{D}_{t-1})| = O(w_t)$.

For $n = 1$, it is crucial that $\hat{f}_t$ is convex so that $\mathcal{D}_t$ is a minimizer of $\hat{f}_t$ in the convex constraint set $\mathcal{C}^{\text{dict}}$, as stated in **1**. From this the monotonicity properties **2** and **3** follow immediately. The second-order growth property **4** requires additional assumption that the surrogates $\hat{f}_t$ are strongly convex uniformly in t. Then **3** and **4** imply **5**, which then implies **6**. Lastly, **1** is also crucially used to conclude that every limit point of $(\mathcal{D}_t)_{t\geq 1}$ is a stationary point of f over $\mathcal{C}^{\text{dict}}$. Now in the multi-modal case $n \geq 2$, we do not have **1** so all of the implications mentioned above are not guaranteed. Hence the analysis in the multi-modal case requires a significant amount of technical innovation.

Now we state our key lemma that handles the nonconvexity of the surrogate loss $\hat{f}_t$ in the general multi-modal case $n \geq 1$.

Lemma 2 (Key Lemma) *Assume (A1)-(A3). Let $(\mathcal{D}_t)_{t\geq 1}$ be an output of Algorithm 1. For all $t \geq 1$, the following hold:*

(i) *(Forward Monotonicity)* $\hat{f}_t(\mathcal{D}_{t-1}) \geq \hat{f}_t(\mathcal{D}_t)$;

(ii) *(Stability of Estimates)* $\|\mathcal{D}_t - \mathcal{D}_{t-1}\|_F = O(w_t)$;

(iii) *(Stability of Errors)* $|h_t(\mathcal{D}_t) - h_{t-1}(\mathcal{D}_{t-1})| = O(w_t)$, *where* $h_t := \hat{f}_t - f_t$.

(iv) *(Asymptotic Surrogate Stationarity) Let $(t_k)_{k\geq 1}$ be any sequence such that $\mathcal{D}_{t_k}$ and $\hat{f}_{t_k}$ converges almost surely. Then $\mathcal{D}_\infty := \lim_{k\to\infty} \mathcal{D}_{t_k}$ is almost surely a stationary point of $\hat{f}_\infty := \lim_{k\to\infty} \hat{f}_{t_k}$ over $\mathcal{C}^{\mathrm{dict}}$.*

We show Lemma 2 **(i)** using a monotonicity property of block coordinate descent. One of our key observations is that we can directly ensure the stability properties **5** and **6** (Lemma 2 **(ii)** and **(iii)**) by using a search radius restriction (see (17) in Algorithm 1). In turn, we do not need the properties **3** and **4**. In particular, our analysis does not require strong convexity of the surrogate loss $\hat{f}_t$ in each loading matrices as opposed to the existing analysis (see, e.g., (Mairal et al., 2010, Assumption **B**) and (Mairal, 2013b, Def. 2.1)). Lastly, our analysis requires that estimates $\mathcal{D}_t$ are only asymptotically stationary to the limiting surrogate loss function along convergent subsequences, as stated in Lemma 2 **(iv)**. The proof of this statement is nontrivial and requires a substantial work. On a high level, the argument consists of demonstrating that the effect of search radius restriction by $O(w_t)$ vanishes in the limit, and the negative gradient $-\nabla \hat{f}_\infty(\mathcal{D}_\infty)$ is in the normal cone of $\mathcal{C}^{\mathrm{dict}}$ at $\mathcal{D}_\infty$.

The second technical challenge is to handle dependence on input signals, as stated in (A1). The theory of quasi-martingales Fisk (1965); Rao (1969) is a key ingredient in convergence analysis under i.i.d input in Mairal et al. (2010); Mairal (2013b); Agarwal et al. (2019). However, dependent signals do not induce quasi-martingale since conditional on the information $\mathcal{F}_t$ at time t, the following signal $\mathcal{X}_{t+1}$ could be heavily biased. We use the recently developed technique in Lyu et al. (2020b) to overcome this issue of data dependence. The essential fact is that for irreducible and aperiodic Markov chains on finite state space, the N-step conditional distribution converges exponentially in N to the unique stationary distribution regardless of the initial distribution (exponential mixing). The key insight in Lyu et al. (2020b) was that in the analysis, one can condition on "distant past" $\mathcal{F}_{t-\sqrt{t}}$, not on the present $\mathcal{F}_t$, in order to allow the underlying Markov chain to mix close enough to the stationary distribution π for $\sqrt{t}$ iterations. This is opposed to a common practice of thinning Markov chain samples in order to reduce the dependence between sample points we mentioned earlier in Subsection 4.1. We provide the estimate based on this technique in Lemma 3.

For the statement of Lemma 3, recall that under (A1), the data tensor at time t is given by $\mathcal{X}_t = \varphi(Y_t)$, where Y_t is an irreducible and aperiodic Markov chain on a finite state space Ω with transition matrix P. For $y, y' \in \Omega$ and $k \in \mathbb{N}$, $P^k(y, y')$ equals the k-step transition probability of Y_t from y to y', and $P^k(y, \cdot)$ equals the distribution of Y_k conditional on $Y_0 = y$. We also use the notation $a^+ = \max(0, a)$ for $a \in \mathbb{R}$.

Lemma 3 (Convergence of Positive Variation) *Let $(\mathcal{D}_t)_{t\geq 1}$ be an output of Algorithm 1. Suppose (A1)', (A2), and (A3) hold.*

(i) *Let $(a_t)_{t\geq 0}$ be a sequence of non-decreasing non-negative integers such that $a_t = o(t)$. Then there exists absolute constants $C_1, C_2, C_3 > 0$ such that for all sufficiently large $t \geq 0$,*

$$\mathbb{E}\left[\mathbb{E}\left[w_{t+1}\big(\ell(\mathcal{X}_{t+1}, \mathcal{D}_t) - f_t(\mathcal{D}_t)\big) \,\middle|\, \mathcal{F}_{t-a_t}\right]^+\right] \tag{19}$$

$$\leq C_1 w_{t-a_t}^2 \sqrt{t} + C_2 w_t^2 a_t + C_3 w_t \sup_{\mathbf{y}\in\Omega} \|P^{a_t+1}(\mathbf{y}, \cdot) - \pi\|_{TV}. \tag{20}$$

(ii) $\sum_{t=0}^{\infty} \left(\mathbb{E}\left[\hat{f}_{t+1}(\mathcal{D}_{t+1}) - \hat{f}_t(\mathcal{D}_t) \right] \right)^+ \leq \sum_{t=0}^{\infty} w_{t+1} \left(\mathbb{E}\left[(\ell(\mathcal{X}_{t+1}, \mathcal{D}_t) - f_t(\mathcal{D}_t)) \right] \right)^+ < \infty.$

We give some remarks on the statement of Lemma 3. According to Proposition 5 in Section 5, one of the main quantities we would like to bound is $\mathbb{E}[\ell(\mathcal{X}_{t+1}, \mathcal{D}_t) - \hat{f}_t(\mathcal{D})]^+$, which is the expected positive variation of the one-step difference between the one-point error $\ell(\mathcal{X}_{t+1}, \mathcal{D}_t)$ of factorizing the new data $\mathcal{X}_{t+1}$ using the current dictionary $\mathcal{D}_t$ and the empirical error $\hat{f}_t(\mathcal{D}_t)$. According to the recursive update of the empirical and surrogate losses in (11) and (16), we want the weighted sum of such expected positive variations in Lemma 3 **(ii)** is finite. This follows from the bound in Lemma 3 **(i)**, as long as $\sum_{t=1}^{\infty} w_t^2 \sqrt{t} < \infty$, $a_t = O(\sqrt{t})$, and the Markov chain Y_t modulating the data tensor $\mathcal{X}_t$ mixes fast enough (see (A1)). Such conditions are satisfied from the assumptions (A1) and (A3).

5. Proof of the main result

In this section, we prove our main convergence result, Theorem 1. Throughout this section, we assume the code matrices H_t and loading matrices $U_t^{(i)}$ belong to convex and compact constraint sets $H_t \in \mathcal{C}^{\text{code}} \subseteq \mathbb{R}^{R \times b}$, $U_t^{(i)} \in \mathcal{C}^{(i)} \subseteq \mathbb{R}^{I_i \times R}$ as in (A2) and denote $\mathcal{C}^{\text{dict}} = \mathcal{C}^{(1)} \times \cdots \times \mathcal{C}^{(n)} \subseteq \mathbb{R}^{I_1 \times R} \times \cdots \times \mathbb{R}^{I_n \times R}$.

5.1 Deterministic analysis

In this subsection, we provide some deterministic analysis of our online algorithm (Algorithm 1), which will be used in the forthcoming stochastic analysis.

First, we derive a parameterized form of Algorithm 1, where the surrogate loss function $\hat{f}_t$ is replaced with $\hat{g}_t$, which is a block-wise quadradtic function with recursively updating parameters. This will be critical in our proof of Lemma 2 **(iv)** as well as deriving the bounded-memory implementation of Algorithm 1 stated in Algorithm 2 in Appendix C. Consider the following block optimization problem

$$\text{Upon arrival of } \mathcal{X}_t\text{:} \quad \begin{cases} H_t = \arg\min_{H \in \mathcal{C}^{\text{code}} \subseteq \mathbb{R}^{R \times b}} \ \ell(\mathcal{X}_t, \mathcal{D}_{t-1}, H) \\ A_t = (1 - w_t) A_{t-1} + w_t H_t H_t^T \\ \mathbf{B}_t = (1 - w_t) \mathbf{B}_{t-1} + w_t (\mathcal{X}_t \times_{n+1} H_t^T) \\ \mathcal{D}_t = \underset{\substack{\mathcal{D} = [U^{(1)}, \ldots, U^{(n)}] \in \mathcal{C}^{\text{dict}} \\ \|U^{(i)} - U_{t-1}^{(i)}\|_F \leq c' w_t \ \forall i}}{\arg\min} \hat{g}_t(\mathcal{D}) \end{cases}, \tag{21}$$

where for each $\mathcal{D} = [U_1, \ldots, U_n] \in \mathcal{C}^{\text{dict}}$ (here we use subscripts to denote modes for taking their transpose) and $\hat{g}_t$ in (21) is defined as

$$\hat{g}_t(\mathcal{D}) := \text{tr}(A_t \, (U_n^T U_n \odot \ldots \odot U_1^T U_1)) - 2\text{tr}\left(\mathbf{B}_t^{(n+1)} (U_n \otimes_{kr} \ldots \otimes_{kr} U_1)^T \right), \tag{22}$$

where $\mathbf{B}_t^{(n+1)} \in \mathbb{R}^{I_1 \cdots I_n \times R}$ denotes the mode-$(n+1)$ unfolding of $\mathbf{B}_t \in \mathbb{R}^{I_1 \times \cdots \times I_n \times R}$, and $A_0 \in \mathbb{R}^{R \times R}$ and $\mathbf{B}_0 \in \mathbb{R}^{I_1 \times \cdots \times I_n \times R}$ are tensors of all zero entries. The following proposition shows that minimizing $\hat{f}_t$ in (16) is equivalent to minimizing $\hat{g}_t$ in (21). This shows that $\hat{f}_t$,

which requires the storage of all past tensors $\mathcal{X}_1, \dots, \mathcal{X}_t$ for its definition, can in fact be parameterized by an aggregate matrix $A_t \in \mathbb{R}^{R\times R}$ and an aggregate tensor $\mathbf{B}_t \in \mathbb{R}^{I_1\times\cdots\times I_n\times R}$, whose dimensions are independent of t. This implies that Algorithm 1 can be implemented using a bounded memory, without needing to store a growing number of full-dimensional data tensors $\mathcal{X}_1, \dots, \mathcal{X}_t$. See Appendix C for more details.

Proposition 4 *The following hold:*

(i) *Let $\hat{f}_t$ be as in* (16) *and $\hat{g}_t$ be in* (22)*. Then*

$$\hat{f}_t(\mathcal{D}) = \hat{g}_t(\mathcal{D}) + \sum_{s=1}^{t} \operatorname{tr}\left(\operatorname{MAT}(\mathcal{X}_s)\operatorname{MAT}(\mathcal{X}_s)^T\right) + \lambda\sum_{s=1}^{t}\|H_s\|_1, \tag{23}$$

(ii) *For each $1 \le j \le n$ and $t \ge 1$, let $\overline{A}_{t;j} \in \mathbb{R}^{R\times R}$, $\overline{B}_{t;j} \in \mathbb{R}^{I_j\times R}$ be the output of Algorithm 3 with input $A_t, \mathbf{B}_t, U_1, \dots, U_n$, and j. Then can rewrite $\hat{g}_t(\mathcal{D}) = \hat{g}_t(U_1, \dots, U_n)$ in* (22) *as*

$$\hat{g}_t(U_1, \dots, U_n) = \operatorname{tr}\left(U_i\overline{A}_{t;j}U_i^T\right) - 2\operatorname{tr}\left(U_i\overline{B}_{t;j}^T\right). \tag{24}$$

Proof Let $\operatorname{MAT}(\mathcal{X}_s) = [\operatorname{vec}(\mathcal{X}_{s;1}), \dots, \operatorname{vec}(\mathcal{X}_{s;b})] \in \mathbb{R}^{(I_1\dots I_n)\times b}$ denote the matrix whose i^{th} column is the vectorization $\operatorname{vec}(\mathcal{X}_{s;j})$ of the tensor $\mathcal{X}_{s;j} \in \mathbb{R}^{I_1\times\cdots\times I_n}$. The first assertion follows easily from observing that, for each $[U_1, \dots, U_n] \in \mathcal{C}^{\text{dict}}$ and $H \in \mathbb{R}^{R\times b}$,

$$\begin{aligned}
&\|\mathcal{X}_s - \texttt{Out}(U_1, \dots, U_n) \times_{n+1} H\|_F^2 \\
&\quad = \|\operatorname{MAT}(\mathcal{X}_s) - (U_n \otimes_{kr} \dots \otimes_{kr} U_1)H\|_F^2 \\
&\quad = \operatorname{tr}\left((U_n \otimes_{kr} \dots \otimes_{kr} U_1)HH^T(U_n \otimes_{kr} \dots \otimes_{kr} U_1)^T\right) \\
&\qquad - 2\operatorname{tr}\left(\operatorname{MAT}(\mathcal{X}_s)H^T(U_n \otimes_{kr} \dots \otimes_{kr} U_1)^T\right) + \operatorname{tr}\left(\operatorname{MAT}(\mathcal{X}_s)\operatorname{MAT}(\mathcal{X}_s)^T\right),
\end{aligned}$$

and also note that

$$\begin{aligned}
&\operatorname{tr}\left((U_n \otimes_{kr} \dots \otimes_{kr} U_1)HH^T(U_n \otimes_{kr} \dots \otimes_{kr} U_1)^T\right) \\
&\qquad = \operatorname{tr}(HH^T(U_n \otimes_{kr} \dots \otimes_{kr} U_1)^T(U_n \otimes_{kr} \dots \otimes_{kr} U_1)) \\
&\qquad = \operatorname{tr}(HH^T\,(U_n^TU_n \odot \dots \odot U_1^TU_1)).
\end{aligned}$$

Then the linearity of trace show

$$\begin{aligned}
\hat{f}_t(U_1, \dots, U_n) = \operatorname{tr}(A_t\,(U_n^TU_n \odot \dots \odot U_1^TU_1)) - 2\operatorname{tr}\left(\widetilde{\mathbf{B}}_t(U_n \otimes_{kr} \dots \otimes_{kr} U_1)^T\right) \\
+ \sum_{s=1}^{t} \operatorname{tr}\left(\operatorname{MAT}(\mathcal{X}_s)\operatorname{MAT}(\mathcal{X}_s)^T\right) + \lambda\sum_{s=1}^{t}\|H_s\|_1,
\end{aligned} \tag{25}$$

where A_t is recursively defined in (21) and $\widetilde{\mathbf{B}}_t \in \mathbb{R}^{(I_1\times\cdots\times I_n)\times b}$ is defined recursively by

$$\widetilde{\mathbf{B}}_s = (1 - w_t)\widetilde{\mathbf{B}}_{s-1} + w_t\operatorname{MAT}(\mathcal{X}_s)H_s^T.$$

By a simple induction argument, one can show that $\widetilde{B}_t$ equals the mode-$(n+1)$ unfolding $\mathbf{B}_t^{(n+1)}$ of $\mathbf{B}_t$ defined recursively in (21), as desired.

For **(ii)**, first note that

$$\begin{aligned}
&\operatorname{tr}(A\,(U_n^T U_n \odot \ldots \odot U_1^T U_1)) \\
&\quad = \operatorname{tr}((A \odot U_1^T U_1 \odot \ldots U_{i-1}^T U_{i-1} \odot U_{i+1}^T U_{i+1} \odot \ldots \odot U_n^T U_n)\, U_j^T U_j) \\
&\quad = \operatorname{tr}(U_j\,(A \odot U_1^T U_1 \odot \ldots U_{i-1}^T U_{i-1} \odot U_{i+1}^T U_{i+1} \odot \ldots \odot U_n^T U_n)\, U_j^T) \\
&\quad = \operatorname{tr}(U_j\, \overline{A}_{t;j}\, U_j^T).
\end{aligned}$$

Also, recall that $\mathbf{B}_t^{(n+1)}$ and $U_n \otimes_{kr} \ldots \otimes_{kr} U_1$ are $(\prod_{i=1}^n I_j) \times R$ matrices. Let $\mathbf{B}_t(,r) \in \mathbb{R}^{I_1 \times \cdots \times I_n}$ denote the r^{th} mode-$(n+1)$ slice of $\mathbf{B}_t$. We note that

$$\begin{aligned}
&\operatorname{tr}\left(\mathbf{B}_t^{(n+1)}(U_n \otimes_{kr} \ldots \otimes_{kr} U_1)^T\right) \\
&= \sum_{r=1}^{R} \operatorname{tr}\left(\mathbf{B}_t(,r) \times_1 U_1(:,r) \times_2 \cdots \times_{i-1} U_{i-1}(:,r) \times_i U_i(:,r) \times_{i+1} U_{i+1}(:,r) \times_{i+2} \cdots \times_n U_n(:,r)\right) \\
&= \operatorname{tr}\left(\sum_{r=1}^{R} [\mathbf{B}_t(,r) \times_1 U_1(:,r) \times_2 \cdots \times_{i-1} U_{i-1}(:,r) \times_{i+1} U_{i+1}(:,r) \times_{i+2} \cdots \times_n U_n(:,r)]\, U_i(:,r)^T\right) \\
&= \operatorname{tr}\left(U_i \overline{B}_{t;j}^T\right),
\end{aligned}$$

where $\overline{B}_{t;j}^T$ is as in the assertion. Then the assertion follows. ■

Proof of Lemma 2 (i)-(iii). First, we show **(i)**. Write $\mathcal{D}_{t-1} = [U_1, \ldots, U_n]$ and $\mathcal{D}_t = [U_1', \ldots, U_n']$ (here we use subscripts to denote modes). Using Proposition 4 **(i)**, we write

$$\begin{aligned}
&\hat{f}_t(\mathcal{D}_{t-1}) - \hat{f}_t(\mathcal{D}_t) && (26)\\
&\quad = \hat{f}_t([U_1, \ldots, U_n]) - \hat{f}_t([U_1', \ldots, U_n']) && (27)\\
&\quad = \sum_{i=1}^{n} \hat{f}_t([U_1', \ldots, U_{i-1}', U_i, U_{i+1}, \ldots, U_n]) - \hat{f}_t([U_1', \ldots, U_{i-1}', U_i', U_{i+1}, \ldots, U_n]). && (28)
\end{aligned}$$

Recall that U_i' is a minimizer of the function $U \mapsto \hat{f}_t([U_1', \ldots, U_{i-1}', U, U_{i+1}, \ldots, U_n])$ (which is convex by Proposition 4) over the convex set $\mathcal{C}_i$ defined in Algorithm 1. Also, U_i' belongs to $\mathcal{C}_i$. Hence each summand in the last expression above is nonnegative. This shows $\hat{f}_t(\mathcal{D}_{t-1}) - \hat{f}_t(\mathcal{D}_t) \geq 0$, as desired. Also note that **(ii)** is trivial by the search radius restriction in Algorithm 1.

Lastly, we show **(iii)**. Both $\hat{f}_t$ and f_t are uniformly bounded and Lipschitz by Lemma 16 in Appendix B. Hence $h_t = \hat{f}_t - f_t$ is also Lipschitz with some constant $C > 0$ independent of t. Then by the recursive definitions of $\hat{f}_t$ and f_t (see (16) and (11)) and noting that $\ell(\mathcal{X}_t, \mathcal{D}_{t-1}, H_t) = \ell(\mathcal{X}_t, \mathcal{D}_{t-1})$, we have

$$\begin{aligned}
&|h_t(\mathcal{D}_t) - h_{t-1}(\mathcal{D}_{t-1})| && (29)\\
&\quad \leq |h_t(\mathcal{D}_t) - h_t(\mathcal{D}_{t-1})| + |h_t(\mathcal{D}_{t-1}) - h_{t-1}(\mathcal{D}_{t-1})| && (30)
\end{aligned}$$

$$\leq C\|\mathcal{D}_t - \mathcal{D}_{t-1}\|_F + \left|\left(\hat{f}_t(\mathcal{D}_{t-1}) - \hat{f}_{t-1}(\mathcal{D}_{t-1})\right) - (f_t(\mathcal{D}_{t-1}) - f_{t-1}(\mathcal{D}_{t-1}))\right| \tag{31}$$

$$= C\|\mathcal{D}_t - \mathcal{D}_{t-1}\|_F + w_t|\hat{f}_{t-1}(\mathcal{D}_{t-1}) - f_{t-1}(\mathcal{D}_{t-1})|. \tag{32}$$

Hence this and **(ii)** show $|h_t(\mathcal{D}_t) - h_{t-1}(\mathcal{D}_{t-1})| = O(w_t)$, as desired. ■

Next, we establish two elementary yet important inequalities connecting the empirical and surrogate loss functions. This is trivial in the case of vector-valued signals, in which case we can directly minimize $\hat{f}_t$ over a convex constraint set $\mathcal{C}^{\text{dict}}$ to find $\mathcal{D}_t$ so we have the 'forward monotonicity' $\hat{f}_t(\mathcal{D}_t) \leq \hat{f}_t(\mathcal{D}_{t-1})$ immediately from the algorithm design. In the tensor case, this still holds since we use block coordinate descent to progressively minimize $\hat{f}_t$ in each loading matrix.

Proposition 5 *Let $(\mathcal{D}_t)_{t\geq 1}$ be an output of Algorithm 1. Then for each $t \geq 0$, the following hold:*

(i) $\hat{f}_{t+1}(\mathcal{D}_{t+1}) - \hat{f}_t(\mathcal{D}_t) \leq w_{t+1}\left(\ell(\mathcal{X}_{t+1}, \mathcal{D}_t) - f_t(\mathcal{D}_t)\right)$.

(ii) $0 \leq w_{t+1}\left(\hat{f}_t(\mathcal{D}_t) - f_t(\mathcal{D}_t)\right) \leq w_{t+1}\left(\ell(\mathcal{X}_{t+1}, \mathcal{D}_t) - f_t(\mathcal{D}_t)\right) + \hat{f}_t(\mathcal{D}_t) - \hat{f}_{t+1}(\mathcal{D}_{t+1})$.

Proof We begin by observing that

$$\hat{f}_{t+1}(\mathcal{D}_t) = (1 - w_{t+1})\hat{f}_t(\mathcal{D}_t) + w_{t+1}\ell_{t+1}(\mathcal{X}_{t+1}, \mathcal{D}_t, H_{t+1}) \tag{33}$$

$$= (1 - w_{t+1})\hat{f}_t(\mathcal{D}_t) + w_{t+1}\ell_{t+1}(\mathcal{X}_{t+1}, \mathcal{D}_t) \tag{34}$$

for all $t \geq 0$. The first equality above uses the definition of $\hat{f}_t$ in (16) and the second equality uses the fact that H_{t+1} is a minimizer of $\ell(\mathcal{X}_{t+1}, \mathcal{D}_t, H)$ over $\mathcal{C}^{\text{code}}$. Hence

$$\begin{aligned}
&\hat{f}_{t+1}(\mathcal{D}_{t+1}) - \hat{f}_t(\mathcal{D}_t) \qquad (35)\\
&= \hat{f}_{t+1}(\mathcal{D}_{t+1}) - \hat{f}_{t+1}(\mathcal{D}_t) + \hat{f}_{t+1}(\mathcal{D}_t) - \hat{f}_t(\mathcal{D}_t)\\
&= \hat{f}_{t+1}(\mathcal{D}_{t+1}) - \hat{f}_{t+1}(\mathcal{D}_t) + (1 - w_{t+1})\hat{f}_t(\mathcal{D}_t) + w_{t+1}\ell(\mathcal{X}_{t+1}, \mathcal{D}_t) - \hat{f}_t(\mathcal{D}_t)\\
&= \hat{f}_{t+1}(\mathcal{D}_{t+1}) - \hat{f}_{t+1}(\mathcal{D}_t) + w_{t+1}(\ell(\mathcal{X}_{t+1}, \mathcal{D}_t) - f_t(\mathcal{D}_t)) + w_{t+1}(f_t(\mathcal{D}_t) - \hat{f}_t(\mathcal{D}_t)).
\end{aligned}$$

Now note that $f_t \leq \hat{f}_t$ by definition. Furthermore, $\hat{f}_{t+1}(\mathcal{D}_{t+1}) - \hat{f}_{t+1}(\mathcal{D}_t) \leq 0$ by Lemma 2 **(i)**, so the inequalities in both **(i)** and **(ii)** follow. ■

5.2 Stochastic analysis

In this subsection, we develop stochastic analysis on our online algorithm, a major portion of which is devoted to handling Markovian dependence in signals as stated in assumption (A1). The analysis here is verbatim as the one developed in Lyu et al. (2020b) for the vector-valued signal (or matrix factorization) case, which we present some of the important arguments in detail here for the sake of completeness. However, the results in this subsection crucially rely on the deterministic analysis in the previous section that was necessary to handle difficulties arising in the tensor-valued signal case.

Recall that under our assumption (A1), the signals $(\mathcal{X}_t)_{t\geq 0}$ are given as $\mathcal{X}_t = \varphi(Y_t)$ for a fixed function φ and a Markov chain $(Y_t)_{t\geq 0}$. Note that Proposition 5 gives a bound on the change in surrogate loss $\hat{f}_t(\mathcal{D}_t)$ in one iteration, which allows us to control its *positive variation* in terms of difference $\ell(\mathcal{X}_{t+1}, \mathcal{D}_t) - f_t(\mathcal{D}_t)$. The core of the stochastic analysis in this subsection is to show that $w_{t+1}\mathbb{E}[\ell(\mathcal{X}_{t+1}, \mathcal{D}_t) - f_t(\mathcal{D}_t)]^+$ is summable. In the classical setting when Y_t's are i.i.d., our signals $\mathcal{X}_t = \varphi(Y_t)$ are also i.i.d., so we can condition on the information $\mathcal{F}_t$ up to time t so that

$$\mathbb{E}\left[\ell(\mathcal{X}_{t+1}, \mathcal{D}_t) - f_t(\mathcal{D}_t) \,\middle|\, \mathcal{F}_t\right] = f(\mathcal{D}_t) - f_t(\mathcal{D}_t). \tag{36}$$

Note that for each fixed $\mathcal{D} \in \mathcal{C}^{\text{dict}}$, $f_t(\mathcal{D}) \to f(\mathcal{D})$ almost surely as $t \to \infty$ by the strong law of large numbers. To handle time dependence of the evolving dictionaries $\mathcal{D}_t$, one can instead look that the convergence of the supremum $\|f_t - f\|_\infty$ over the compact set $\mathcal{C}^{\text{dict}}$, which is provided by the classical Glivenko-Cantelli theorem. This is the approach taken in Mairal et al. (2010); Mairal (2013b) for i.i.d. input.

However, the same approach is not applicable for dependent signals, for instance, when $(Y_t)_{t\geq 0}$ is a Markov chain. This is because, in this case, conditional on $\mathcal{F}_t$, the distribution of Y_{t+1} is not necessarily the stationary distribution π. In fact, when Y_t's form a Markov chain with transition matrix P, Y_t given Y_{t-1} has distribution $P(Y_{t-1}, \cdot)$, and this conditional distribution is a constant distance away from the stationary distribution π. (For instance, consider the case when Y_t takes two values and it differs from Y_{t-1} with probability $1 - \varepsilon$. Then $\pi = [1/2, 1/2]$ and the distribution of Y_t converges exponentially fast to π, but $P(Y_{t-1}, \cdot)$ is either $[1-\varepsilon, \varepsilon]$ or $[\varepsilon, 1-\varepsilon]$ for all $t \geq 1$.)

To handle dependence in data samples, we adopt the strategy developed in Lyu et al. (2020b) in order to handle a similar issue for vector-valued signals (or matrix factorization). The key insight in Lyu et al. (2020b) is that, while the 1-step conditional distribution $P(X_{t-1}, \cdot)$ may be far from the stationary distribution π, the N-step conditional distribution $P^N(X_{t-N}, \cdot)$ is exponentially close to π under mild conditions. Hence we can condition much early on – at time $t-N$ for some suitable $N = N(t)$. Then the Markov chain runs $N+1$ steps up to time $t+1$, so if N is large enough for the chain to mix to its stationary distribution π, then the distribution of Y_{t+1} conditional on $\mathcal{F}_{t-N}$ is close to π. The error of approximating the stationary distribution by the $N+1$ step distribution can be controlled using total variation distance and Markov chain mixing bound. This is stated more precisely in the proposition below.

Proposition 6 *Suppose (A1) hold. Fix a CP-dictionary $\mathcal{D}$. Then for each $t \geq 0$ and $0 \leq N < t$, conditional on the information $\mathcal{F}_{t-N}$ up to time $t - N$,*

$$\left(\mathbb{E}\left[\ell(\mathcal{X}_{t+1}, \mathcal{D}) - f_t(\mathcal{D}) \,\middle|\, \mathcal{F}_{t-N}\right]\right)^+ \leq |f(\mathcal{D}) - f_{t-N}(\mathcal{D})| + N w_t f_{t-N}(\mathcal{D}) \tag{37}$$

$$+ 2\|\ell(\cdot, \mathcal{D})\|_\infty \sup_{\mathbf{y}\in\Omega} \|P^{N+1}(\mathbf{y}, \cdot) - \pi\|_{TV}. \tag{38}$$

Proof The proof is identical to that of (Lyu et al., 2020b, Prop. 7.5). ■

Proof of Lemma 3. Part **(i)** can be derived from Proposition 6 and Lemma 19 in Appendix B. See the proof of (Lyu et al., 2020b, Prop. 7.8 **(i)**) for details. Next, part **(ii)** can be derived from part **(i)** with Proposition 5 **(i)**. See the proof of (Lyu et al., 2020b, Prop. 7.8 **(ii)**) for details. ■

Lemma 7 *Let $(\mathcal{D}_t)_{t\geq 1}$ be the output of Algorithm 1. Suppose (A1)-(A3) hold. Then the following hold.*

(i) $\sum_{t=0}^{\infty} \mathbb{E}\left[w_{t+1}\left(\ell(\mathcal{X}_{t+1}, \mathcal{D}_t) - f_t(\mathcal{D}_t)\right)\right]^+ < \infty$;

(ii) $\mathbb{E}[\hat{f}_t(\mathcal{D}_t)]$ *converges as* $t \to \infty$;

(iii) $\mathbb{E}\left[\sum_{t=0}^{\infty} w_{t+1}\left(\hat{f}_t(\mathcal{D}_t) - f_t(\mathcal{D}_t)\right)\right] = \sum_{t=0}^{\infty} w_{t+1}\left(\mathbb{E}[\hat{f}_t(\mathcal{D}_t)] - \mathbb{E}[f_t(\mathcal{D}_t)]\right) < \infty$;

(iv) $\sum_{t=0}^{\infty} w_{t+1}\left(\hat{f}_t(\mathcal{D}_t) - f_t(\mathcal{D}_t)\right) < \infty$ *almost surely.*

Proof Part **(i)** can be derived from Proposition 6 and Jensen's inequality. See the proof of (Lyu et al., 2020b, Lem. 12 **(ii)**) for details. Parts **(ii)**-**(iv)** can be shown by using Propositions 5, 6, and part **(i)**. See the proof of (Lyu et al., 2020b, Lem. 13) for details. ■

5.3 Asymptotic surrogate stationarity

In this subsection, we prove Lemma 2 **(iv)**, which requires one of the most nontrivial arguments we give in this work. Throughout this subsection, we will denote by $(\mathcal{D}_t)_{t\geq 1}$ the output of Algorithm 1 and $\Lambda := \{\mathcal{D}_t \,|\, t \geq 1\} \subseteq \mathcal{C}^{\text{dict}}$. Note that by Proposition 4, $\hat{f}_{t_k}$ converges almost surely if and only if $A_{t_k}, \mathbf{B}_{t_k}, \mathcal{X}_{t_k}, H_{t_k}$ converge a.s. as $k \to \infty$. In what follows, we say $\mathcal{D}_\infty \in \mathcal{C}^{\text{dict}}$ a *stationary point* of Λ if it is a limit point $\mathcal{D}_\infty$ of Λ and there exists a sequence $t_k \to \infty$ such that $\mathcal{D}_{t_k} \to \mathcal{D}_\infty$ and $\hat{f}_\infty := \lim_{k\to\infty} \hat{f}_{t_k}$ exists almost surely and $\mathcal{D}_\infty$ is a stationary point of $\hat{f}_\infty$ over $\mathcal{C}^{\text{dict}}$. Our goal is to show that every limit point of Λ is stationary.

The following observation is key to our argument.

Proposition 8 *Assume (A1)-(A3) hold. Let $(\mathcal{D}_t)_{t\geq 1}$ be an output of Algorithm 1. Then almost surely,*

$$\sum_{t=1}^{\infty} \left| \left(\nabla \hat{f}_{t+1}(\mathcal{D}_{t+1})^T (\mathcal{D}_t - \mathcal{D}_{t+1}) \right) \right| < \infty.$$

Proof Since $\mathcal{C}^{\text{dict}}$ is compact by (A2) and the aggregate tensors $A_t, \mathbf{B}_t$ are uniformly bounded by Lemma 15 in Appendix B, we can see from Proposition 4 that $\nabla \hat{f}_{t+1}$ over $\mathcal{C}^{\text{dict}}$

is Lipschitz with some uniform constant $L > 0$. Hence by Lemma 14 in Appendix B, for all $t \geq 1$,

$$\left|\hat{f}_{t+1}(\mathcal{D}_t) - \hat{f}_{t+1}(\mathcal{D}_{t+1}) - \operatorname{tr}\left(\nabla \hat{f}_{t+1}(\mathcal{D}_{t+1})^T(\mathcal{D}_t - \mathcal{D}_{t+1})\right)\right| \leq \frac{L}{2}\|\mathcal{D}_t - \mathcal{D}_{t+1}\|_F^2.$$

Also note that $\hat{f}_{t+1}(\mathcal{D}_t) \geq \hat{f}_{t+1}(\mathcal{D}_{t+1})$ by Lemma 2 **(i)**. Hence it follows that

$$\left|\operatorname{tr}\left(\nabla \hat{f}_{t+1}(\mathcal{D}_{t+1})^T(\mathcal{D}_t - \mathcal{D}_{t+1})\right)\right| \leq \frac{L}{2}\|\mathcal{D}_t - \mathcal{D}_{t+1}\|_F^2 + \hat{f}_{t+1}(\mathcal{D}_t) - \hat{f}_{t+1}(\mathcal{D}_{t+1}) \tag{39}$$

On the other hand, (35) and $\hat{f}_t \geq f_t$ yields

$$0 \leq \hat{f}_{t+1}(\mathcal{D}_t) - \hat{f}_{t+1}(\mathcal{D}_{t+1}) \leq \hat{f}_t(\mathcal{D}_t) - \hat{f}_{t+1}(\mathcal{D}_{t+1}) + w_{t+1}(\ell(\mathcal{X}_{t+1}, \mathcal{D}_t) - f_t(\mathcal{D}_t)).$$

Hence using Lemma 7, we have

$$\sum_{t=1}^{\infty} \mathbb{E}\left[\hat{f}_{t+1}(\mathcal{D}_t) - \hat{f}_{t+1}(\mathcal{D}_{t+1})\right] < \infty.$$

Then from (39) and noting that $\|\mathcal{D}_t - \mathcal{D}_{t+1}\|_F^2 = O(w_{t+1}^2)$ and $\sum_{t=1}^{\infty} w_t^2 < \infty$ (see (A3)), it follows that

$$\begin{aligned}\sum_{t=1}^{\infty} \mathbb{E}\left[\left|\operatorname{tr}\left(\nabla \hat{f}_{t+1}(\mathcal{D}_{t+1})^T(\mathcal{D}_t - \mathcal{D}_{t+1})\right)\right|\right] &= \frac{L}{2}\sum_{t=1}^{\infty} \mathbb{E}\left[\|\mathcal{D}_t - \mathcal{D}_{t+1}\|_F^2\right] \\ &\quad + \sum_{t=1}^{\infty} \mathbb{E}\left[\hat{f}_{t+1}(\mathcal{D}_t) - \hat{f}_{t+1}(\mathcal{D}_{t+1})\right] < \infty.\end{aligned}$$

Then the assertion follows by Fubini's theorem and the fact that $\mathbb{E}[|X|] < \infty$ implies $|X| < \infty$ almost surely for any random variable X, where $|\cdot|$ denotes the largest absolute value among the entries of X. ■

Next, we show that the block coordinate descent we use to obtain $\mathcal{D}_{t+1}$ should always give the optimal first-order descent up to a small additive error.

Proposition 9 (Asymptotic first-order optimality) *Assume (A1)-(A3) and $w_t = o(1)$. Then there exists a constant $c_1 > 0$ such that for all $t \geq 1$,*

$$\operatorname{tr}\left(\nabla \hat{f}_{t+1}(\mathcal{D}_{t+1})^T \frac{(\mathcal{D}_{t+1} - \mathcal{D}_t)}{\|\mathcal{D}_{t+1} - \mathcal{D}_t\|_F}\right) \leq \inf_{\mathcal{D} \in \mathcal{C}^{\text{dict}}} \operatorname{tr}\left(\nabla \hat{f}_{t+1}(\mathcal{D}_t)^T \frac{(\mathcal{D} - \mathcal{D}_t)}{\|\mathcal{D} - \mathcal{D}_t\|_F}\right) \tag{40}$$

$$+ c_1\|\mathcal{D}_{t+1} - \mathcal{D}_t\|_F^2. \tag{41}$$

Proof Fix a sequence $(b_t)_{t\geq 1}$ such that $0 < b_t \leq c' w_t$ for all $t \geq 1$. Write $\mathcal{D}_t = [U_t^{(1)}, \ldots, U_t^{(n)}]$ for $t \geq 1$ and denote

$$\hat{f}_{t+1;i} : U \mapsto \hat{f}_{t+1}(U_{t+1}^{(1)}, \ldots, U_{t+1}^{(i-1)}, U, U_t^{(i+1)}, \ldots, U_t^{(n)}) \tag{42}$$

for $U \in \mathbb{R}^{I_i}$ and $i = 1, \dots, n$. Recall that $U_{t+1}^{(i)}$ is a minimizer of $\hat{f}_{t+1;i}$ over the convex set $\mathcal{C}_{t+1}^{(i)}$ defined in (17). Fix arbitrary $\mathcal{D} = [U^{(1)}, \dots, U^{(n)}] \in \mathcal{C}^{\text{dict}}$ such that $\|\mathcal{D} - \mathcal{D}_t\|_F \le b_{t+1}$. Then $\|U^{(i)} - U_t^{(i)}\|_F \le b_{t+1}$ for all $1 \le i \le n$. By convexity of $\mathcal{C}^{(i)}$, note that for each $U^{(i)} \in \mathcal{C}^{(i)}$, $U_t^{(i)} + a(U^{(i)} - U_t^{(i)}) \in \mathcal{C}^{(i)}$ for all $a \in [0, 1]$. Then by the definition of $\mathcal{C}_{t+1}^{(i)}$ and the choice of $U_{t+1}^{(i)}$, we have that for all $t \ge 1$,

$$\hat{f}_{t+1;i}(U_{t+1}^{(i)}) - \hat{f}_{t+1;i}(U_t^{(i)}) \le \hat{f}_{t+1;i}\left(U_t^{(i)} + a(U^{(i)} - U_t^{(i)})\right) - \hat{f}_{t+1;i}(U_t^{(i)}). \tag{43}$$

Recall that $\nabla \hat{f} = [\nabla \hat{f}_{t+1;1}, \dots, \nabla \hat{f}_{t+1;n}]$ is Lipschitz with uniform Lipschitz constant $L > 0$. Hence by Lemma 14 in Appendix B, there exists a constant $c_1 > 0$ such that for all $t \ge 1$,

$$\operatorname{tr}\left(\nabla \hat{f}_{t+1;i}(U_t^{(i)})^T (U_{t+1}^{(i)} - U_t^{(i)})\right) - \frac{L}{2}\|U_{t+1}^{(i)} - U_t^{(i)}\|^2 \tag{44}$$

$$\le a \operatorname{tr}\left(\nabla \hat{f}_{t+1;i}(U_t^{(i)})^T (U^{(i)} - U_t^{(i)})\right) + \frac{La^2\|U^{(i)} - U_t^{(i)}\|}{2}. \tag{45}$$

Adding up these inequalities for $i = 1, \dots, n$, we get

$$\operatorname{tr}\left(\left[\nabla \hat{f}_{t+1;1}(U_t^{(1)}), \dots, \nabla \hat{f}_{t+1;n}(U_t^{(n)})\right]^T (\mathcal{D}_{t+1} - \mathcal{D}_t)\right) \tag{46}$$

$$\le a \operatorname{tr}\left(\left[\nabla \hat{f}_{t+1;1}(U_t^{(1)}), \dots, \nabla \hat{f}_{t+1;n}(U_t^{(n)})\right]^T (\mathcal{D} - \mathcal{D}_t)\right) \tag{47}$$

$$+ \frac{L}{2}\|\mathcal{D}_{t+1} + \mathcal{D}_t\|_F^2 + \frac{La^2}{2}\|\mathcal{D} - \mathcal{D}_t\|_F^2. \tag{48}$$

Since $\nabla \hat{f}_{t+1}$ is L-Lipschitz, using Cauchy-Schwarz inequality,

$$\operatorname{tr}\left(\nabla \hat{f}_{t+1}(\mathcal{D}_{t+1})^T (\mathcal{D}_{t+1} - \mathcal{D}_t)\right) \tag{49}$$

$$\le a \operatorname{tr}\left(\nabla \hat{f}_{t+1}(\mathcal{D}_t)^T (\mathcal{D} - \mathcal{D}_t)\right) + a\|\mathcal{D}_{t+1} - \mathcal{D}_t\|_F \|\mathcal{D} - \mathcal{D}_t\|_F \tag{50}$$

$$+ \frac{3L}{2}\|\mathcal{D}_{t+1} - \mathcal{D}_t\|_F^2 + \frac{La^2}{2}\|\mathcal{D} - \mathcal{D}_t\|_F^2 \tag{51}$$

$$\le a \operatorname{tr}\left(\nabla \hat{f}_{t+1}(\mathcal{D}_t)^T (\mathcal{D} - \mathcal{D}_t)\right) + a\|\mathcal{D}_{t+1} - \mathcal{D}_t\|_F \|\mathcal{D} - \mathcal{D}_t\|_F \tag{52}$$

$$+ \frac{3L}{2}\|\mathcal{D}_{t+1} - \mathcal{D}_t\|_F^2 + ca^2\|\mathcal{D} - \mathcal{D}_t\|_F^2 \tag{53}$$

for some constant $c > 0$ for all $t \ge 1$. Recall that the above holds for all $a \in [0, 1]$. Note that since $\|\nabla \hat{f}_t\|$ is uniformly bounded and $\mathcal{D}^{\text{dict}}$ is compact (see (A2)), the last expression above, viewed as a quadratic function in a, is strictly increasing in a for all $t \ge 1$ when $c > 0$ is sufficiently large. We make such choice for c_3. Hence, the above holds for all $a \ge 0$. Now we may choose $a = b_{t+1}/\|\mathcal{D} - \mathcal{D}_t\|$ and bound the last expression by its first term plus $c_1(\|\mathcal{D}_{t+1} - \mathcal{D}_t\|_F + b_{t+1})^2$ for some constant $c_1 > 0$. Finally, by the radius restriction $\|\mathcal{D}_{t+1} - \mathcal{D}_t\|_F \le c' w_{t+1}$, we may choose $b_{t+1} = \|\mathcal{D}_{t+1} - \mathcal{D}_t\|_F$. Then the assertion follows by dividing both sides of the resulting inequality by $\|\mathcal{D}_{t+1} - \mathcal{D}_t\|$. ■

Proposition 10 *Assume (A1)-(A3). Suppose there exists a subsequence $(\mathcal{D}_{t_k})_{k\geq 1}$ such that either*

$$\sum_{k=1}^{\infty}\|\mathcal{D}_{t_k}-\mathcal{D}_{t_k+1}\|_F=\infty \quad \text{or} \quad \liminf_{k\to\infty}\left|\mathrm{tr}\left(\nabla\hat{f}_{t_k+1}(\mathcal{D}_{t_k+1})^T\frac{\mathcal{D}_{t_k}-\mathcal{D}_{t_k+1}}{\|\mathcal{D}_{t_k}-\mathcal{D}_{t_k+1}\|_F}\right)\right|=0. \quad (54)$$

There exists a further subsequence $(s_k)_{k\geq 1}$ of $(t_k)_{k\geq 1}$ such that $\mathcal{D}_\infty:=\lim_{k\to\infty}\mathcal{D}_{s_k}$ exists and is a stationary point of Λ.

Proof By Proposition 8, we have

$$\sum_{k=1}^{\infty}\|\mathcal{D}_{t_k}-\mathcal{D}_{t_k+1}\|_F\left|\mathrm{tr}\left(\nabla\hat{f}_{t_k+1}(\mathcal{D}_{t_k+1})^T\frac{\mathcal{D}_{t_k}-\mathcal{D}_{t_k+1}}{\|\mathcal{D}_{t_k}-\mathcal{D}_{t_k+1}\|_F}\right)\right|<\infty. \quad (55)$$

Hence the former condition implies the latter condition in (54). Thus it suffices to show that this latter condition implies the assertion. Assume this condition, and let $(s_k)_{k\geq 1}$ be a subsequence of $(t_k)_{k\geq 1}$ for which the liminf in (54) is achieved. By taking a subsequence, we may assume that $\mathcal{D}'_\infty=\lim_{k\to\infty}\mathcal{D}_{s_k}$ and $\hat{f}_\infty:=\lim_{k\to\infty}\hat{f}_{s_k}$ exist.

Now suppose for contradiction that $\mathcal{D}_\infty$ is not a stationary point of $\hat{f}_\infty$ over $\mathcal{C}^{\mathrm{dict}}$. Then there exists $\mathcal{D}^\star\in\mathcal{C}^{\mathrm{dict}}$ and $\delta>0$ such that

$$\mathrm{tr}\left(\nabla\hat{f}_\infty(\mathcal{D}_\infty)^T(\mathcal{D}^\star-\mathcal{D}_\infty)\right)<-\delta<0. \quad (56)$$

By triangle inequality, write

$$\begin{aligned}&\|\nabla\hat{f}_{s_k+1}(\mathcal{D}_{s_k})^T(\mathcal{D}^\star-\mathcal{D}_{s_k})-\nabla\hat{f}_\infty(\mathcal{D}_\infty)^T(\mathcal{D}^\star-\mathcal{D}_\infty)\|_F \quad (57)\\ &\quad\leq\|\nabla\hat{f}_{s_k+1}(\mathcal{D}_{s_k})-\nabla\hat{f}_\infty(\mathcal{D}_\infty)\|_F\cdot\|\mathcal{D}^\star-\mathcal{D}_{s_k}\|_F+\|\nabla\hat{f}_\infty(\mathcal{D}_\infty)\|_F\cdot\|\mathcal{D}_\infty-\mathcal{D}_{s_k}\|_F. \quad (58)\end{aligned}$$

Noting that $\|\mathcal{D}_t-\mathcal{D}_{t-1}\|_F=O(w_t)=o(1)$, we see that the right hand side goes to zero as $k\to\infty$. Hence for all sufficiently large $k\geq 1$, we have

$$\mathrm{tr}\left(\nabla\hat{f}_{s_k+1}(\mathcal{D}_{s_k})^T(\mathcal{D}^\star-\mathcal{D}_{s_k})\right)<-\delta/2. \quad (59)$$

Then by Proposition 9, denoting $\|\mathcal{C}^{\mathrm{dict}}\|_F:=\sup_{\mathcal{D},\mathcal{D}'\in\mathcal{C}^{\mathrm{dict}}}\|\mathcal{D}-\mathcal{D}'\|_F<\infty$,

$$\liminf_{k\to\infty}\ \mathrm{tr}\left(\nabla\hat{f}_{s_k+1}(\mathcal{D}_{s_k+1})^T\frac{\mathcal{D}_{s_k}-\mathcal{D}_{s_k+1}}{\|\mathcal{D}_{s_k}-\mathcal{D}_{s_k+1}\|_F}\right)\leq-\frac{c_1\delta}{2\|\mathcal{C}^{\mathrm{dict}}\|_F}<0, \quad (60)$$

which contradicts the choice of the subsequence $(\mathcal{D}_{s_k})_{k\geq 1}$. This shows the assertion. ■

Recall that during the update $\mathcal{D}_{t-1}\mapsto\mathcal{D}_t$ in (18) each factor matrix of $\mathcal{D}_{t-1}$ changes by at most w_t in Frobenius norm. For each $t\geq 1$, we say $\mathcal{D}_t$ is a *long point* if none of the factor matrices of $\mathcal{D}_{t-1}$ change by w_t in Frobenius norm and *short point* otherwise. Observe that if $\mathcal{D}_t$ is a long point, then imposing the search radius restriction in 17 has no effect and $\mathcal{D}_t$ is obtained from $\mathcal{D}_{t-1}$ by a single cycle of block coordinate descent on $\hat{f}_t$ over $\mathcal{C}^{\mathrm{dict}}$.

Proposition 11 *Assume (A1)-(A3) hold. If $(\mathcal{D}_{t_k})_{k\geq 1}$ is a convergent subsequence of $(\mathcal{D}_t)_{t\geq 1}$ consisting of long points, then the $\mathcal{D}_\infty = \lim_{k\to\infty} \mathcal{D}_{s_k}$ is stationary.*

Proof For each $A \in \mathbb{R}^{R\times R}$, $B \in \mathbb{R}^{I_1\times\cdots\times I_n\times b}$, $\mathcal{D} = [U^{(1)}, \ldots, U^{(n)}] \in \mathbb{R}^{I_1\times R} \times \cdots \times \mathbb{R}^{I_n\times R}$, define

$$\hat{g}(A, B, \mathcal{D}) = \mathrm{tr}(A\,((U^{(n)})^T U^{(n)} \odot \ldots \odot (U^{(1)})^T U^{(1)})) \tag{61}$$

$$- 2\mathrm{tr}\left(B^{(n+1)}(U^{(n)} \otimes_{kr} \ldots \otimes_{kr} U^{(1)})^T\right), \tag{62}$$

where $B^{(n+1)}$ denotes the mode-$(n+1)$ unfolding of B (see also (22)). By taking a subsequence of $(t_k)_{k\geq 1}$, we may assume that $A_\infty := \lim_{k\to\infty} A_{t_k}$ and $\mathbf{B}_\infty := \lim_{k\to\infty} \mathbf{B}_{t_k}$ exist. Hence the function $\hat{g}_\infty := \lim_{k\to\infty} \hat{g}_{t_k} = \hat{g}(A_\infty, \mathbf{B}_\infty, \cdot)$ is well-defined. Noting that since $\nabla\hat{f}_t = \nabla\hat{g}_t$ for all $t \geq 1$ by Proposition 4, it suffices to show that $\mathcal{D}_\infty$ is a stationary point of $\hat{g}_\infty$ over $\mathcal{C}^{\mathrm{dict}}$ almost surely.

The argument is similar to that of (Bertsekas, 1997, Prop. 2.7.1). However, here we do not need to assume uniqueness of solutions to minimization problems of $\hat{f}_t$ in each block coordinate due to the added search radius restriction. Namely, write $\mathcal{D}_\infty = [U_\infty^{(1)}, \ldots, U_\infty^{(n)}]$. Then for each $k \geq 1$,

$$\hat{g}_{t_k+1}(U_{t_k+1}^{(1)}, U_{t_k}^{(2)}, \ldots, U_{t_k}^{(n)}) \leq \hat{g}_{t_k+1}(U^{(1)}, U_{t_k}^{(2)}, \ldots, U_{t_k}^{(n)}) \tag{63}$$

for all $U^{(1)} \in \mathcal{C}^{(1)} \cap \{U\colon \|U - U_{t_k}^{(1)}\|_F \leq c' w_{t_k+1}\}$. In fact, since $\mathcal{D}_{t_k}$ is a long point by the assumption, (63) holds for all $U^{(1)} \in \mathcal{C}^{(1)}$. Taking $k \to \infty$ and using the fact that $\|U_{t_k+1}^{(1)} - U_{t_k}^{(1)}\|_F \leq c' w_{t_k+1} = o(1)$,

$$\hat{g}_\infty(U_\infty^{(1)}, U_\infty^{(2)}, \ldots, U_\infty^{(n)}) \leq \hat{g}_\infty(U^{(1)}, U_\infty^{(2)}, \ldots, U_\infty^{(n)}) \quad \text{for all } U_1 \in \mathcal{C}^{(1)}. \tag{64}$$

Since $\mathcal{C}^{(1)}$ is convex, it follows that

$$\nabla_1\hat{g}_\infty(\mathcal{D}_\infty)^T(U_1 - U_1^{(\infty)}) \geq 0 \qquad \text{for all } U_1 \in \mathcal{C}^{(1)}, \tag{65}$$

where ∇_1 denotes the partial gradient with respect to the first block $U^{(1)}$. By using a similar argument for other coordinates of $\mathcal{D}_\infty$, it follows that $\nabla\hat{g}_\infty(\mathcal{D}_\infty)^T(\mathcal{D} - \mathcal{D}_\infty) \geq 0$ for all $\mathcal{D} \in \mathcal{C}^{\mathrm{dict}}$. This shows the assertion. ■

Proposition 12 *Assume (A1)-(A3) hold. Suppose there exists a non-stationary limit point $\mathcal{D}_\infty$ of Λ. Then there exists $\varepsilon > 0$ such that the ε-neighborhood $B_\varepsilon(\mathcal{D}_\infty) := \{\mathcal{D} \in \mathcal{C}^{\mathrm{dict}} \mid \|\mathcal{D} - \mathcal{D}_\infty\|_F < \varepsilon\}$ with the following properties:*

(a) *$B_\varepsilon(\mathcal{D}_\infty)$ does not contain any stationary points of Λ.*

(b) *There exists infinitely many $\mathcal{D}_t$'s outside of $B_\varepsilon(\mathcal{D}_\infty)$.*

Proof We will first show that there exists an ε-neighborhood $B_\varepsilon(\mathcal{D}_\infty)$ of $\mathcal{D}_\infty$ that does not contain any long points of Λ. Suppose for contradiction that for each $\varepsilon > 0$, there exists a long point Λ in $B_\varepsilon(\mathcal{D}_\infty)$. Then one can construct a sequence of long points converging to $\mathcal{D}_\infty$. But then by Proposition 11, $\mathcal{D}_\infty$ is a stationary point, a contradiction.

Next, we show that there exists $\varepsilon > 0$ such that $B_\varepsilon(\mathcal{D}_\infty)$ satisfies **(a)**. Suppose for contradiction that there exists no such $\varepsilon > 0$. Then we have a sequence $(\mathcal{D}_{\infty;k})_{k\geq 1}$ of stationary points of Λ that converges to $\mathcal{D}_\infty$. Denote the limiting surrogate loss function associated with $\mathcal{D}_{\infty;k}$ by $\hat{f}_{\infty;k}$. Recall that each $\hat{f}_{\infty;k}$ is parameterized by elements in a compact set (see (A1), Proposition 4, and Lemma 16) in Appendix B. Hence by choosing a subsequence, we may assume that $\hat{f}_\infty := \lim_{k\to\infty} \hat{f}_{\infty;k}$ is well-defined. Fix $\mathcal{D} \in \mathcal{C}^{\text{dict}}$ and note that by Cauchy-Schwarz inequality,

$$\nabla \hat{f}_\infty(\mathcal{D}_\infty)^T(\mathcal{D} - \mathcal{D}_\infty) \geq -\|\nabla \hat{f}_\infty(\mathcal{D}_\infty) - \nabla \hat{f}_{\infty;k}(\mathcal{D}_{\infty;k})\|_F \cdot \|\mathcal{D} - \mathcal{D}_\infty\|_F \tag{66}$$

$$- \|\nabla \hat{f}_{\infty;k}(\mathcal{D}_{\infty;k})\|_F \cdot \|\mathcal{D}_\infty - \mathcal{D}_{\infty;k}\|_F \tag{67}$$

$$+ \nabla \hat{f}_{\infty;k}(\mathcal{D}_{\infty;k})^T(\mathcal{D} - \mathcal{D}_{\infty;k}). \tag{68}$$

Note that $\nabla \hat{f}_{\infty;k}(\mathcal{D}_{\infty;k})^T(\mathcal{D} - \mathcal{D}_{\infty;k}) \geq 0$ since $\mathcal{D}_{\infty;k}$ is a stationary point of $\hat{f}_{\infty;k}$ over $\mathcal{C}^{\text{dict}}$. Hence by taking $k \to \infty$, this shows $\nabla \hat{f}_\infty(\mathcal{D}_\infty)^T(\mathcal{D} - \mathcal{D}_\infty) \geq 0$. Since $\mathcal{D} \in \mathcal{D}^{\text{dict}}$ was arbitrary, this shows that $\mathcal{D}_\infty$ is a stationary point of $\hat{f}_\infty$ over $\mathcal{C}^{\text{dict}}$, a contradiction.

Lastly, from the earlier results, we can choose $\varepsilon > 0$ such that $B_\varepsilon(\mathcal{D}_\infty)$ has no long points of Λ and also satisfies **(b)**. We will show that $B_{\varepsilon/2}(\mathcal{D}_\infty)$ satisfies **(c)**. Then $B_{\varepsilon/2}(\mathcal{D}_\infty)$ satisfies **(a)**-**(b)**, as desired. Suppose for contradiction there are only finitely many $\mathcal{D}_t$'s outside of $B_{\varepsilon/2}(\mathcal{D}_\infty)$. Then there exists an integer $M \geq 1$ such that $\mathcal{D}_t \in B_{\varepsilon/2}(\mathcal{D}_\infty)$ for all $t \geq M$. Then each $\mathcal{D}_t$ for $t \geq M$ is a short point of Λ. By definition, it follows that $\|\mathcal{D}_t - \mathcal{D}_t\|_F \geq w_t$ for all $t \geq M$, so $\sum_{t=1}^\infty \|\mathcal{D}_t - \mathcal{D}_t\|_F \geq \sum_{t=1}^\infty w_t = \infty$. Then by Proposition 10, there exists a subsequence $(s_k)_{k\geq 1}$ such that $\mathcal{D}'_\infty := \lim_{k\to\infty} \mathcal{D}_{t_k}$ exists and is stationary. But since $\mathcal{D}'_\infty \in B_\varepsilon(\mathcal{D})$, this contradicts **(a)** for $B_\varepsilon(\mathcal{D})$. This shows the assertion. ■

We are now ready to give a proof of Lemma 2 **(iv)**.

Proof of Lemma 2 (iv). Assume (A1)-(A3) hold. Suppose there exists a non-stationary limit point $\mathcal{D}_\infty$ of Λ. By Proposition 12, we may choose $\varepsilon > 0$ such that $B_\varepsilon(\mathcal{D}_\infty)$ satisfies the conditions **(a)**-**(b)** of Proposition 12. Choose $M \geq 1$ large enough so that $c'w_t < \varepsilon/4$ whenever $t \geq M$. We call an integer interval $I := [\ell, \ell')$ a *crossing* if $\mathcal{D}_\ell \in B_{\varepsilon/3}(\mathcal{D}_\infty)$, $\mathcal{D}_{\ell'} \notin B_{2\varepsilon/3}(\mathcal{D}_\infty)$, and no proper subset of I satisfies both of these conditions. By definition, two distinct crossings have empty intersection. Fix a crossing $I = [\ell, \ell')$. Then it follows that by triangle inequality,

$$\sum_{t=\ell}^{\ell'-1} \|\mathcal{D}_{t+1} - \mathcal{D}_t\|_F \geq \|\mathcal{D}_{\ell'} - \mathcal{D}_\ell\|_F \geq \varepsilon/3. \tag{69}$$

Note that since $\mathcal{D}_\infty$ is a limit point of Λ, $\mathcal{D}_t$ visits $B_{\varepsilon/3}(\mathcal{D}_\infty)$ infinitely often. Moreover, by condition **(a)** of Proposition 12, $\mathcal{D}_t$ also exits $B_\varepsilon(\mathcal{D}_\infty)$ infinitely often. It follows that there are infinitely many crossings. Let t_k denote the k^{th} smallest integer that appears in some

crossing. By definition, $\mathcal{D}_{t_k} \in B_{2\varepsilon/3}(\mathcal{D}_\infty)$ for $k \geq 1$. Then $t_k \to \infty$ as $k \to \infty$, and by (69),

$$\sum_{k=1}^{\infty} \|\mathcal{D}_{t_k+1} - \mathcal{D}_{t_k}\|_F \geq (\#\text{ of crossings})\,\frac{\varepsilon}{3} = \infty. \tag{70}$$

Then by Proposition 10, there is a further subsequence $(s_k)_{k\geq 1}$ of $(t_k)_{k\geq 1}$ such that $\mathcal{D}'_\infty := \lim_{k\to\infty} \mathcal{D}_{s_k}$ exists and is stationary. However, since $\mathcal{D}_{t_k} \in B_{2\varepsilon/3}(\mathcal{D}_\infty)$ for $k \geq 1$, we have $\mathcal{D}'_\infty \in B_\varepsilon(\mathcal{D}_\infty)$. This contradicts the condition **(b)** of Proposition 12 for $B_\varepsilon(\mathcal{D}_\infty)$ that it cannot contain any stationary point of Λ. This shows the assertion. ■

5.4 Proof of the main result

Now we prove the main result in this paper, Theorem 1.

Proof of Theorem 1. Suppose (A1)-(A3) hold. We first show **(i)**. Recall that $\mathbb{E}[\hat{f}_t(\mathcal{D}_t)]$ converges by Lemma 7. Jensen's inequality and Lemma 2 **(iv)** imply

$$|\mathbb{E}[h_{t+1}(\mathcal{D}_{t+1})] - \mathbb{E}[h_t(\mathcal{D}_t)]| \leq \mathbb{E}\left[|h_{t+1}(\mathcal{D}_{t+1}) - h_t(\mathcal{D}_t)|\right] = O(w_{t+1}). \tag{71}$$

Since $\mathbb{E}[\hat{f}_t(\mathcal{D}_t)] \geq \mathbb{E}[f_t(\mathcal{D}_t)]$, Lemma 7 **(ii)**-**(iii)** and Lemma 17 in Appendix B give

$$\lim_{t\to\infty} \mathbb{E}[f_t(\mathcal{D}_t)] = \lim_{t\to\infty} \mathbb{E}[\hat{f}_t(\mathcal{D}_t)] + \lim_{t\to\infty} \Big(\mathbb{E}[f_t(\mathcal{D}_t)] - \mathbb{E}[\hat{f}_t(\mathcal{D}_t)]\Big) \tag{72}$$

$$= \lim_{t\to\infty} \mathbb{E}[\hat{f}_t(\mathcal{D}_t)] \in (1, \infty). \tag{73}$$

This shows **(i)**.

Next, we show **(ii)**. Triangle inequality gives

$$|f(\mathcal{D}_t) - \hat{f}_t(\mathcal{D}_t)| \leq \left(\sup_{\mathcal{D}\in\mathcal{C}^{\mathrm{dict}}} |f(\mathcal{D}) - f_t(\mathcal{D})|\right) - h_t(\mathcal{D}_t). \tag{74}$$

Note that $|h_{t+1}(\mathcal{D}_{t+1}) - h_t(\mathcal{D}_t)| = O(w_{t+1})$ by Lemma 2 **(iii)**. Hence Lemma 7 **(iv)** and Lemma 17 in Appendix B show that $h_t(\mathcal{D}_t) \to 0$ almost surely. Furthermore, (74) and Lemma 19 in Appendix B show that $|f(\mathcal{D}_t) - \hat{f}_t(\mathcal{D}_t)| \to 0$ almost surely. This completes the proof of **(ii)**.

Lastly, we show **(iii)**. Further assume (A4). Let $\mathcal{D}_\infty \in \mathcal{C}^{\mathrm{dict}}$ be an arbitrary limit point of the sequence $(\mathcal{D}_t)_{t\geq 1}$. Recall that $\Sigma_t := (\mathcal{D}_t, A_t, \mathbf{B}_t, r_t)_{t\geq 0}$ is bounded by Lemma 15(in Appendix B) and (A1) and (A2). Hence we may choose a random subsequence $(t_k)_{k\geq 1}$ so that $\mathcal{D}_{t_k} \to \mathcal{D}_\infty$. By taking a further subsequence, we may also assume that Σ_{t_k} converges to some random element $(\mathcal{D}_\infty, A_\infty, \mathbf{B}_\infty, r_\infty)$ a.s. as $k \to \infty$. Then $\hat{f}_\infty := \lim_{k\to\infty} \hat{f}_{t_k}$ exists almost surely. It is important to note that $\mathcal{D}_\infty$ is a stationary point of $\hat{f}_\infty$ over $\mathcal{C}^{\mathrm{dict}}$ by Lemma 2 **(iv)**.

Recall that $\hat{f}_t(\mathcal{D}_t) - f_t(\mathcal{D}_t) \to 0$ as $t \to \infty$ almost surely by part **(ii)**. By using continuity of $\hat{f}_t$, f_t, f in parameters (see (A4)), it follows that

$$\left|\hat{f}_\infty(\mathcal{D}_\infty) - f(\mathcal{D}_\infty)\right| = \lim_{k\to\infty} \left|\hat{f}_{t_k}(\mathcal{D}_{t_k}) - f_{t_k}(\mathcal{D}_{t_k})\right| \tag{75}$$

$$\leq \lim_{k\to\infty} \left(\sup_{\mathcal{D}\in\mathcal{C}^{\text{dict}}} |f - f_{t_k}(\mathcal{D})| - h_{t_k}(\mathcal{D}_{t_k}) \right) = 0, \tag{76}$$

where the last equality also uses Lemma 19 in Appendix B.

Fix $\varepsilon > 0$ and $\mathcal{D} \in \mathbb{R}^{I_1\times R} \times \cdots \times \mathbb{R}^{I_n\times R}$. Hence, almost surely,

$$\hat{f}_\infty(\mathcal{D}_\infty + \mathcal{D}) = \lim_{k\to\infty} \hat{f}_{s_k}(\mathcal{D}_{s_k} + \mathcal{D}) \geq \lim_{k\to\infty} f_{s_k}(\mathcal{D}_{s_k} + \mathcal{D}) = f(\mathcal{D}_\infty + \mathcal{D}), \tag{77}$$

where the last equality follows from Lemma 19. Since $\nabla\hat{f}$ and ∇f are both Lipschitz (see (A4) for the latter), by Lemma 14 in Appendix B, we have

$$\left| \hat{f}_\infty(\mathcal{D}_\infty + \varepsilon\mathcal{D}) - \hat{f}_\infty(\mathcal{D}_\infty) - \operatorname{tr}\left(\nabla\hat{f}_\infty(\mathcal{D}_\infty)^T(\varepsilon\mathcal{D}) \right) \right| \leq c_1\varepsilon^2\|\mathcal{D}\|_F^2, \tag{78}$$

$$\left| f(\mathcal{D}_\infty + \varepsilon\mathcal{D}) - f(\mathcal{D}_\infty) - \operatorname{tr}\left(\nabla f(\mathcal{D}_\infty)^T(\varepsilon\mathcal{D}) \right) \right| \leq c_1\varepsilon^2\|\mathcal{D}\|_F^2, \tag{79}$$

for some constant $c_1 > 0$ for all $\varepsilon > 0$. Recall that $\hat{f}_\infty(\mathcal{D}_\infty) = f(\mathcal{D}_\infty)$ a.s. by (75). Hence it follows that there exists some constant $c_2 > 0$ such that almost surely

$$\operatorname{tr}\left(\left(\nabla\hat{f}_\infty(\mathcal{D}_\infty) - \nabla f(\mathcal{D}_\infty) \right)^T (\varepsilon\mathcal{D}) \right) \geq -c_2\varepsilon^2\|\mathcal{D}\|_F^2. \tag{80}$$

After canceling out $\varepsilon > 0$ and letting $\varepsilon \searrow 0$ in (80),

$$\operatorname{tr}\left(\left(\nabla\hat{f}_\infty(\mathcal{D}_\infty) - \nabla f(\mathcal{D}_\infty) \right)^T \mathcal{D} \right) \geq 0 \qquad \text{a.s.} \tag{81}$$

Since this holds for all $\mathcal{D} \in \mathbb{R}^{I_1\times R} \cdots \times \mathbb{R}^{I_n\times R}$, it follows that $\nabla\hat{f}_\infty(\mathcal{D}_\infty) = \nabla f(\mathcal{D}_\infty)$ almost surely. But since $\mathcal{D}_\infty$ is a stationary point of $\hat{f}_\infty$ over $\mathcal{C}^{\text{dict}}$ by Lemma 2 **(iv)**, it follows that $\nabla\hat{f}_\infty(\mathcal{D}_\infty)$ is in the normal cone of $\mathcal{C}^{\text{dict}}$ at $\mathcal{D}_\infty$ (see., e.g., Boyd et al. (2004)). The same holds for $\nabla f(\mathcal{D}_\infty)$. This means that $\mathcal{D}_\infty$ is a stationary point of f over $\mathcal{C}^{\text{dict}}$. Since $\mathcal{D}_\infty$ is an arbitrary limit point of $\mathcal{D}_t$, the desired conclusion follows. ■

6. Experimental validation

In this section, we compare the performance of our proposed online CPDL algorithm (Algorithm 1) for the standard (offline) NCPD problem (8) against the two most popular algorithms of Alternating Least Squares (ALS), which is a special instance of Block Coordinate Descent, and Multiplicative Update (MU) (see Shashua and Hazan (2005)) for this task. See Algorithms 6 and 7 for implementations of ALS and MU.

We give a more precise statement of the NCPD problem we consider here. Given a 3-mode data tensor $\mathbf{X} \in \mathbb{R}_{\geq 0}^{d_1\times d_2\times d_3}$ and an integer $R \geq 1$, we want to find three nonnegative factor matrices $U^{(k)} \in \mathbb{R}_{\geq 0}^{d_k\times R}$, $k = 1, 2, 3$, that minimize the following CP-reconstruction error:

$$\min_{[U^{(1)},U^{(2)},U^{(3)}]\in\mathcal{C}_M^{\text{dict}}} \left\| \mathbf{X} - \sum_{i=1}^{R} \bigotimes_{k=1}^{3} U^{(k)}(:, i) \right\|_F, \tag{82}$$

where $\mathcal{C}_M^{\text{dict}}$ is the subset of $\mathbb{R}_{\geq 0}^{d_1 \times R} \times \mathbb{R}_{\geq 0}^{d_2 \times R} \times \mathbb{R}_{\geq 0}^{d_3 \times R}$ consisting of factor matrices of Frobenius norm bounded by a fixed constant $M \geq \sqrt{R}\|\mathbf{X}\|_F^{1/3}$. Note that the constraint set $\mathcal{C}_M^{\text{dict}}$ is convex and compact, as required in (A2) for Theorem 1 to apply. We claim that the additional bounded norm constraint on the factor matrices does not lose any generality, in the sense that an optimal solution of (82) with $M \geq \sqrt{R}\|\mathbf{X}\|_F^{1/3}$ has the same objective value as the optimal solution of (82) with $M = \infty$:

$$\left(\text{optimal value of (82) for } M \geq \sqrt{R}\|\mathbf{X}\|_F^{1/3}\right) = (\text{optimal value of (82) for } M = \infty)\,. \quad (83)$$

In order to maintain the flow, we justify this claim at the end of this section.

We consider one synthetic and three real-world tensor data derived from text data and that were used for dynamic topic modeling experiments in Kassab et al. (2021). Each document is encoded as a 5000 or 7000 dimensional word frequency vector using tf-idf vectorizer Rajaraman and Ullman (2011).

1. $\mathbf{X}_{\text{synth}} \in \mathbb{R}_{\geq 0}^{100 \times 100 \times 100}$ is generated by $\mathbf{X}_{\text{synth}} = 0.01 * \texttt{Out}(V_1, V_2, V_3)$, where the loading matrices $V_1, V_2, V_3 \in \mathbb{R}_{\geq 0}^{100 \times 50}$ are generated by sampling each of their entries uniformly and independently from the unit interval $[0, 1]$.

2. $\mathbf{X}_{\text{20News}} \in \mathbb{R}_{\geq 0}^{40 \times 5000 \times 26}$ (41.6MB) is a tensor representing semi-synthetic text data based on 20 Newsgroups dataset Rennie (2008) synthesized in Kassab et al. (2021) for dynamic topic modeling, consisting of 40 stacks of 26 documents encoded in 5000 dimensional word space.

3. $\mathbf{X}_{\text{Twitter}} \in \mathbb{R}_{\geq 0}^{90 \times 5000 \times 1000}$ (3.6GB) is an anonymized Twitter text data related to the COVID-19 pandemic from Feb. 1 to May 1 of 2020. The three modes correspond to days, words, and tweets, in order. Each day, the top 1000 most retweeted English tweets are collected. The original data was collected in Kassab et al. (2021).[2]

4. $\mathbf{X}_{\text{Headlines}} \in \mathbb{R}_{\geq 0}^{203 \times 7000 \times 700}$ (8.0GB) is a tensor derived in Kassab et al. (2021) from news headlines published over a period of 17 years sourced from the Australian news source ABC Kulkarni (2018). The three modes correspond to months, words, and headlines, in order. In each month, 700 headlines are chosen uniformly at random.

For all datasets, we used all algorithms to learn the loading matrices $U^{(1)}, U^{(2)}, U^{(3)}$ with $R = 5$ columns, that evolve in time as the algorithm proceeds. The choice of $R = 5$ is arbitrary and is not ideal especially for the real data tensors, but it suffices for the purpose of this experiment as a benchmark of our online CPDL against ALS and MU. We plot the reconstruction error $\|\mathbf{X} - \texttt{Out}(U^{(1)}, U^{(2)}, U^{(3)})\|_F$ against elapsed time in both cases in Figure 2. Since these benchmark algorithms are also iterative (see Algorithms 6 and 7), we can measure how reconstruction error drops as the three algorithms proceed. In order to make a fair comparison, we compare the reconstruction error against CPU times with the same machine, not against iteration counts, since a single iteration may have different computational costs across different algorithms. For ALS and MU, we disregarded

2. For code repository, see `https://github.com/lara-kassab/dynamic-tensor-topic-modeling`

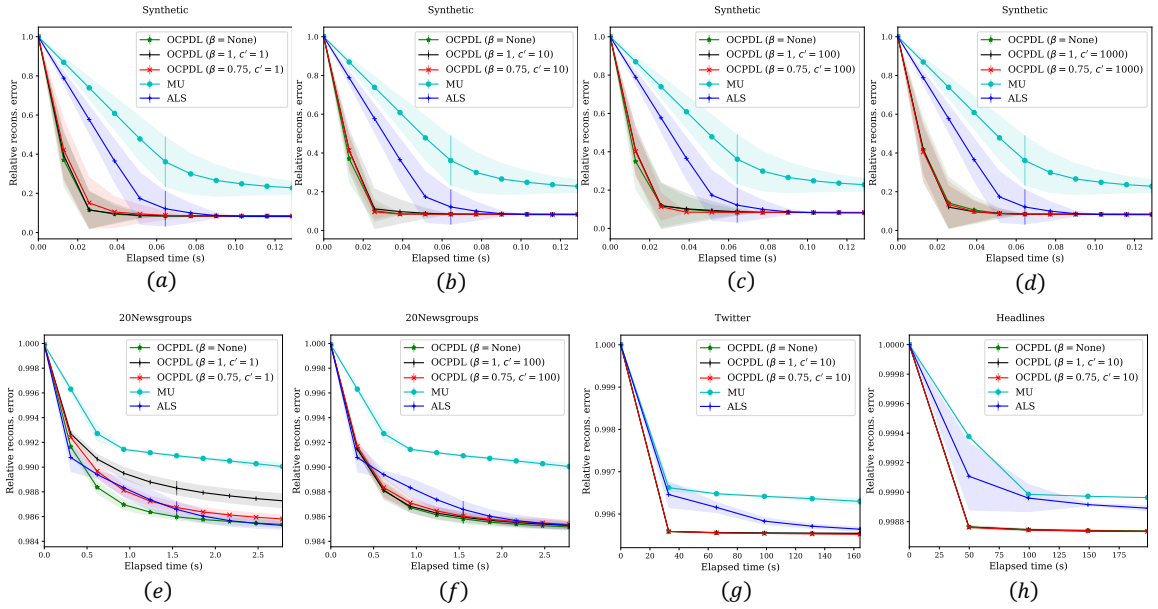


Figure 2: Comparison of performance of online CPDL for the nonnegative tensor factorization problem against Alternating Least Squares (ALS) and Multiplicative Update (MU). For each data tensor, we apply each algorithm to find nonnegative loading matrices $U^{(1)}, U^{(2)}, U^{(3)}$ of $R = 5$ columns. We repeat this multiple times (50 for synthetic, 20 for 20Newsgroups, and 10 for the other two) and the average reconstruction error with 1 standard deviation are shown by the solid lines and shaded regions of respective colors.

the bounded norm constraint in (82), which makes it only favorable to those benchmark methods so it is still a fair comparison of our method.

We give some implementation details of online CPDL (Algorithm 1) for the offline NCPD problem in (82). From the given data tensor $\mathbf{X}$, we obtain a sequence of tensors $\mathbf{X}_1, \ldots, \mathbf{X}_T$ obtained by subsampling 1/5 of the coordinates from the last mode. Hence while ALS and MU require loading the entire tensors into memory, only 1/5 of the data needs to be loaded to execute online CPDL. In Figure 2, OCPDL (β) for for $\beta \in \{0.75, 1\}$ denotes Algorithm 1 with weights $w_t = c' t^{-\beta} / \log t$ (with $w_1 = c'$), where 0.75 and 1 for β correspond to the two extreme values that satisfy the assumption (A3) (see also (A3')) of Theorem 1; the case of $\beta = \text{None}$ uses $c' = \infty$ and $w_t \equiv t^{-1}/(\log t)$ (with $w_1 = 1$). In all cases, the weights satisfy (A3) so the algorithm is guaranteed to converge to the stationary points of the objective function almost surely by Theorem 1. The constant c' is chosen from $\{1, 10, 100, 1000\}$ for $\beta \in \{0.75, 1\}$. Initial loading matrices for Algorithm 1 are chosen with i.i.d. entries drawn from the uniform distribution on $[0, 1]$.

Note that since the last mode of the full tensors is subsampled, the loading matrices we learn from online CPDL have sizes $(d_1 \times R)$, $(d_2 \times R)$, and $(d_3' \times R)$, where $d_3' < d_3$ equals the size of the last mode of the subsampled tensors. In order to compute the reconstruction error for the full $d_1 \times d_2 \times d_3$ tensor, we recompute the last factor matrix of size $d_3 \times R$ by using the first two factor matrices with the sparse coding algorithm (Algorithm 4). This last step of computing a single loading matrix while fixing all the others is equivalent to a single step of ALS in Algorithm 6.

In Figure 2, each algorithm is used multiple times (50 for Synthetic, 20 for 20Newsgroups, and 10 for Twitter and Headlines) for the same data, and the plot shows the average reconstruction errors together with their standard deviation (shading). In all cases except the smallest initial radius $c' = 1$ on the densest tensor $\mathbf{X}_{\text{20News}}$, online CPDL is able to obtain significantly lower reconstruction error much more rapidly than the other two algorithms and maintains low average reconstruction accuracy.

For $\mathbf{X}_{\text{20News}}$ with $c' = 1$ (Figure 2 (e)), we observe some noticible difference in the performance of online CPDL depending on β, where larger values of β (faster decaying radii) give slower convergence. This seems to be due to the fact that $\mathbf{X}_{\text{20News}}$ is the densest among the four tensors by orders of magnitude and $c' = 1$ gives too small of an initial radius. Namely, the average Frobeinus norm, $\|\mathbf{X}\|_F/(d_1 d_2 d_3)$ equals 6.26×10^{-5} for $\mathbf{X}_{\text{Synthetic}}$, 1.10×10^{-3} for $\mathbf{X}_{\text{20News}}$, 6.65×10^{-7} for $\mathbf{X}_{\text{Twitter}}$, and 3.78×10^{-7} and $\mathbf{X}_{\text{Headlines}}$. However, we did not observe any significant difference in all other cases. In general, when c' is large enough, it appears that the radius restriction in Algorithm 1 enables the theoretical convergence guarantee in Theorem 1 without any compromise in practical performance, which did not depend significantly on the decay rate paramter β. In our experiments, $c' = \|\mathbf{X}\|_F$ was sufficiently large, where $\|\mathbf{X}_{\text{Synthetic}}\|_F = 62.61$, $\|\mathbf{X}_{\text{20News}}\|_F = 32.74$, $\|\mathbf{X}_{\text{Twitter}}\|_F = 299.37$, and $\|\mathbf{X}_{\text{Synthetic}}\|_F = 376.38$.

Proof of claim (83)**.** Suppose $\mathcal{D}_\infty := [U^{(1)}, U^{(2)}, U^{(3)}]$ is an optimal solution of (82) without norm restriction (i.e., $M = \infty$). Fix a column index $i \in \{1, \dots, R\}$ and positive scalars $\alpha_1, \alpha_2, \alpha_3$ such that $\alpha_1\alpha_2\alpha_3 = 1$. The objective in (82) is invariant under rescaling the three respective columns: $U^{(k)}(:, i) \mapsto \alpha_k U^{(k)}(:, i)$ for $k \in \{1, 2, 3\}$. At the optimal factor matrices, the objective in (82) should be at most $\|\mathbf{X}\|_F$. Since all factors are nonnegative, it follows that

$$\prod_{k=1}^{3} \|\alpha_k U^{(k)}(:, i)\|_F = \left\| \bigotimes_{k=1}^{3} \alpha_k U^{(k)}(:, i) \right\|_F \leq \|\mathbf{X}\|_F. \tag{84}$$

Then we can choose α_k's in a way that $\|\alpha_k U^{(k)}(:, i)\|_F$ is constant in k[3], in which case $\|\alpha_k U^{(k)}(:, i)\|_F \leq \|\mathbf{X}\|_F^{1/3}$. This argument shows that we can rescale the ith columns of the optimal factor matrices in $\mathcal{D}_\infty$ in a way that the objective value does not change and the columns have norms bounded by $\|\mathbf{X}\|_F^{1/3}$. This holds for all columns i, so we can find a tuple of factor matrices $[V^{(1)}, V^{(2)}, V^{(3)}]$ in $\mathcal{C}_M^{\text{dict}}$ that has the same objective value as $\mathcal{D}_\infty$ as long as $M \geq R\|\mathbf{X}\|_F^{1/3}$. ■

7. Applications

For all our applications in this section, we take the constraint sets $\mathcal{C}^{\text{code}}$ and $\mathcal{C}^{\text{dict}}$ in Algorithm 1 to consists of *nonnegative* matrices so that the learned CP-dictionary gives a "parts-based representation" of the subject data as in classical NMF (see Lee and Seung

3. e.g., $\alpha_k = a_j a_l / a_k^2$, where $a_k := \|U^{(k)}(:, i)\|_F$ and $j, k, l \in \{1, 2, 3\}$ are distinct

(1999, 2001); Lee et al. (2009)). In all our experiments in this section, we used the balanced weight $w_t = 1/t$, which satisfies the assumption (A3).

7.1 Reshaping tensors before CP-decomposition to preserve joint features

Before we discuss our real-world applications of the online CPDL method, we first give some remarks on reshaping tensor data before factorization and why it would be useful in applications.

One may initially think that concatenating some modes of a tensor into a single mode before applying CP-decomposition loses joint features corresponding to the concatenated modes. In fact, if we *undo the unfolding* after the decomposition, it actually *preserves* the joint features. Hence in practice, one can exploit the tensor structure in multiple ways before CP-decomposition to disentangle a select set of features in the desired way, which we demonstrate through analyzing a diverse set of examples from image, video, and time-series in Section 7.

To better illustrate our point, suppose we have three discrete random variables X_1, X_2, X_3, where X_i takes n_i distinct values for $1 \leq i \leq 3$. Denote their 3-dimensional joint distribution as a 3-mode tensor $\mathbf{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$. Suppose we have its CP-decomposition $\mathbf{X} \approx \texttt{Out}(U^{(1)}, U^{(2)}, U^{(3)})$. We can interpret this as the sum of R product distributions of the marginal distributions given by the respective columns in the three factor matrices. In an extreme case of a $R = 1$ CP-decomposition, any kind of joint features among multiple random variables will be lost in the single product distribution.

On the other hand, consider combining the first two random variables (X_1, X_2) into a single random variable, say, Y_1, which takes $n_1 n_2$ distinct values. Then the joint distribution of (Y_1, X_3) will be represented as a 2-dimensional tensor $\mathbf{X}^{(12)} \in \mathbb{R}^{n_1 n_2 \times n_3}$, which corresponds to the tensor obtained by concatenating the first two modes of $\mathbf{X}$. Suppose we have its CP-decomposition $\mathbf{X}^{(12)} \approx \texttt{Out}(V^{(1)}, V^{(2)})$, where $V^{(1)} \in \mathbb{R}^{n_1 n_2 \times R}$ and $V^{(2)} \in \mathbb{R}^{n_3 \times R}$. Then we can reshape each column $V^{(1)}(:, i)$ to a 2-dimensional tensor $V^{(1)}_{n_1 \times n_2}(:, i) \in \mathbb{R}^{n_1 \times n_2}$ by using the ordering of entries in $[n_1] \times [n_2]$ we used to concatenate X_1 and X_2 into Y_1. In this way, we have approximated the full joint distribution $\mathbf{X}$ as the sum of the product between two- and one-dimensional distributions $V^{(1)}_{n_1 \times n_2}(:, i) \otimes V^{(2)}(:, i)$. In this factorization, the joint features of X_1 and X_2 can still be encoded in the 2-dimensional joint distributions $V^{(1)}_{n_1 \times n_2}(:, i)$, and only the joint features between (X_1, X_2) and X_3 are disentangled.

For instance, the tensor for the mouse brain activity video in Subsection 7.3 has four modes $(\texttt{time}, \texttt{horizontal}, \texttt{vertical}, \texttt{color})$. There is almost no change in the shape of the brain in the video and only the color changes indicate neuronal activation in time. Hence, we do not want to disentangle the horizontal, vertical, and color modes, but instead, concatenate them to maintain the joint feature of the spatial activation pattern (see Figure 4). See also Figures 3 and 5 for the effect of various tensor reshaping before factorization in the context of image and time-series data.

7.2 Image processing applications

We first apply our algorithm to patch-based image processing. A workflow for basic patch-based image analysis is to extract small overlapping patches from some large images, vectorize these patches, apply some standard dictionary learning algorithm, and reshape back.

Dictionaries obtained from this general procedure have a wide variety of uses, including image compression, denoising, deblurring, and inpainting Elad (2010); Dong et al. (2011); Papyan et al. (2017); Ma et al. (2013).

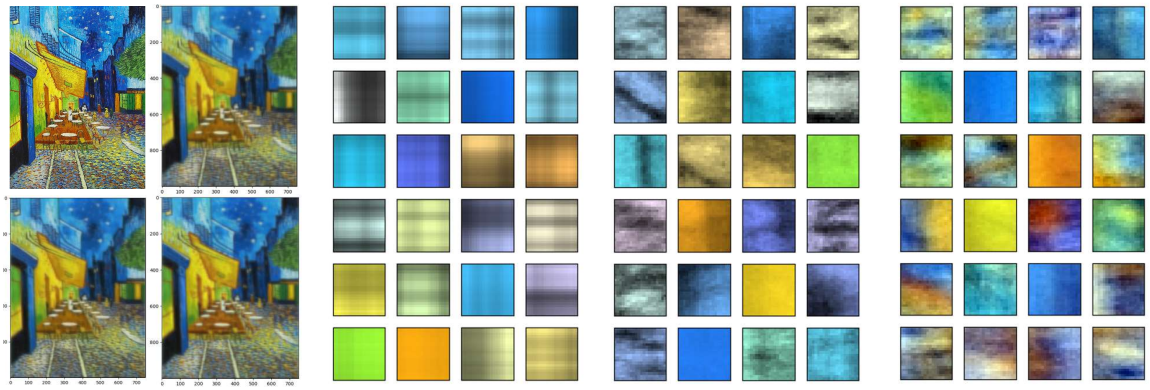


Figure 3: Color image reconstruction by online CPDL. The original image is shown in the top left of (a). The top right reconstruction in (a) is derived from the dictionary learned from the unmodified tensor decomposition of color image patches, which is exemplified in (b). The bottom left reconstruction in (a) uses the dictionary in (c) learned by tensor decomposition of color image patches whose spatial modes are vectorized. The bottom right reconstruction in (a) uses the dictionary learned by a tensor decomposition of fully vectorized color image patches, which is shown in (d).

Although this procedure has produced countless state-of-the-art results, a major drawback to such methods is that vectorizing image patches can greatly slow down the learning process by increasing the effective dimension of the dictionary learning problem. Moreover, by respecting the natural tensor structure of the data, we find that our learned dictionary atoms display a qualitative difference from those trained on reshaped color image patch data. We illustrate this phenomenon in Figure 3. Our experiment is as follows. Figure 3 (a) top left is a famous painting (Van Gogh's *Café Terrace at Night*) from which we extracted 1000 color patches of shape $(\texttt{vertical} \times \texttt{horizontal} \times \texttt{color}) = (20 \times 20 \times 3)$. We applied our online CPDL algorithm (Algorithm 1 for 400 iterations with $\lambda = 1$) to various reshapings of such patches to learn three separate dictionaries, each consisting of 24 atoms.

The first dictionary, displayed in Figure 3 (b), is obtained by applying online CPDL without reshaping the patches. Due to the rank-1 restriction on the atoms as a 3-mode tensor, the spatial features are parallel to the vertical or horizontal axes, and also color variation within each atom is only via scalar multiple (a.k.a. 'saturation'). The second dictionary, Figure 3 (c), was trained by vectorizing the color image patches along the spatial axes, applying online CPDL to the resulting 2-mode data tensors of shape $(\texttt{space} \times \texttt{color}) = (400 \times 3)$, and reshaping back. Here, the rank-1 restriction on the atoms as a 2-mode tensor separates the spatial and color features, but now the spatial features in the atoms are more 'generic' as they do not have to be parallel to the vertical or horizontal axes. Note that the color variation within each atom is still via a scalar multiple. Lastly, the third dictionary, Figure 3 (d), is obtained by applying our online CPDL to the fully vectorized image patch data. Here the features in the atoms do not have any rank-1 restriction along with any mode so that they exhibit 'fully entangled' spatial and color features. Although dictionary

(b) requires much less storage, the reconstructed images from all three dictionaries shown in Figure 3 (a) show that it still performs adequately for the task of image reconstruction.

7.3 Learning spatial and temporal activation patterns in cortex

In this subsection, we demonstrate our method on video data of brain activity across a mouse cortex, and how our online CPDL learns dictionaries for the spatial and temporal activation patterns simultaneously. The original video is due to Barson et al. Barson et al. (2020) by using genetically encoded calcium indicators to image brain activity transcranially. Simultaneous cellular-resolution two-photon calcium imaging of a local microcircuit as well as mesoscopic widefield calcium imaging of the entire cortical mantle in awake mice are used to capture the video (see Barson et al. (2020) for more details.)

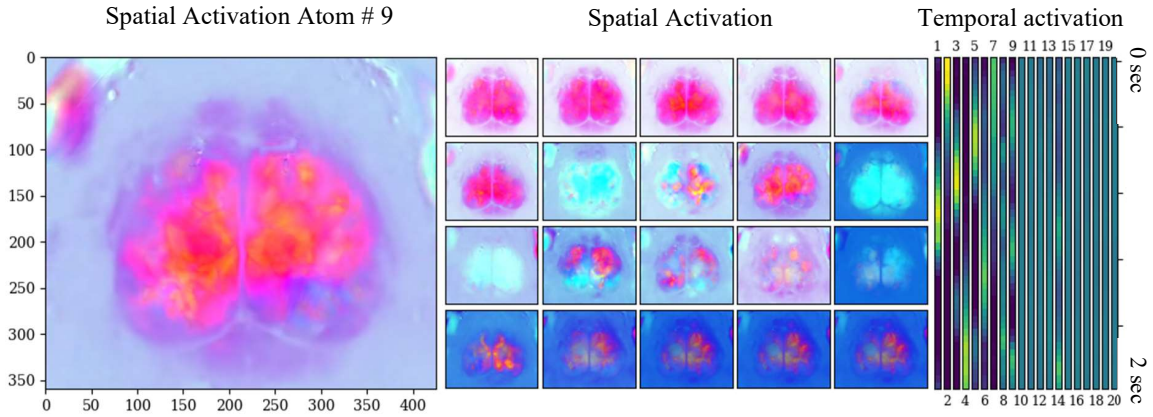


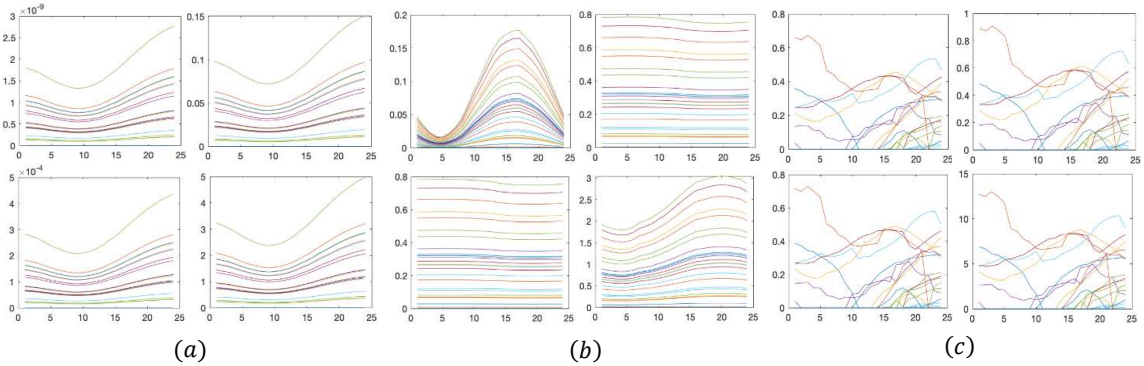
Figure 4: Learning 20 CP-dictionary atoms from video frames on brain activity across the mouse cortex.

The original video frame is a tensor of shape $((1501, 360, 426, 3)$ corresponding to the four modes $(\texttt{time}, \texttt{horizontal}, \texttt{vertical}, \texttt{color})$, where frames are 0.04 sec apart, which spans total 60.04 seconds. We intend to learn weakly periodic patterns of spatial and temporal activation patterns of duration at most 2 seconds. To this end, we sample 50-frame (2 sec. long) clips uniformly at random for 200 times. Each sampled tensor is reshaped into $(\texttt{time}, \texttt{space} * \texttt{color}) = (50,\ 360 * 426 * 3)$ matrix, and then sequentially fed into the online CPDL algorithm with $w_t = c'/t$, $\lambda = 2$, and $c' = 10^5$. Note that we vectorize the horizontal, vertical and color modes into a single mode before factorization in order to *preserve* the spatial structure learned in the loading matrix. Namely, for spatial activation patterns, we desire dictionary atoms of the form of Figure 3 (d) rather than (b) or (c).

Our algorithm learns a CP-dictionary in the space-color mode that shows spatial activation patterns and the corresponding time mode shows their temporal activation pattern, as seen in Figure 4. Due to the nonnegativity constraint, spatial activation atoms representing localized activation regions in the cortex are learned, while the darker ones represent the background brain shape without activation. On the other hand, the activation frequency is simultaneously learned by the temporal activation atoms shown in Figure 4 (right). For instance, the spacial activation atom # 9 (numbered lexicographically) activates three times in its corresponding temporal activation atom in the right, so such activation pattern has an approximate period of 2/3 sec.

7.4 Joint time series dictionary learning

A key advantage of online algorithms is that they are well-suited to applications in which data are arriving in real-time. We apply our algorithm to a weather dataset obtained from Beniaguev (2017). Beginning with a $(36 \times 2998 \times 4)$ tensor where the first mode corresponds to cities, the second mode to time in hours, and the third mode to weather data such that the frontal slices correspond to temperature, humidity, pressure, and wind speed. We regularized the data by taking a moving average over up to four hours (in part to impute missing data values), and by applying a separate rescaling of each frontal slice to normalize the magnitudes of the entries.



(a) (b) (c)

Figure 5: Display of one atom from three different dictionaries of 25 atoms which were obtained from online CPDL on weather data: (a) no reshaping, (b) data which was reshaped to $36 \times (24 \times 4)$, and (c) data which was reshaped to $(36 \times 24) \times 4$. For each subplot, the four subplots represent the evolution of four measurements (temperature (top left), humidity (bottom left), pressure (top right), and wind speed (bottom right)) in time for 24 hours (horizontal axis) in 36 cities (in different colors).

From this large data tensor, we sequentially extracted smaller $(36 \times 24 \times 4) = (\texttt{cities} \times \texttt{time} \times \texttt{measurements})$ tensors by dividing time into overlapping segments of length 24 hours, with overlap size 4 hours. Our experiment consisted of applying the online CPDL (Algorithm 1) to this dataset to learn a single CP-dictionary atom ($R = 1$), say $\mathbf{D} \in \mathbb{R}^{36 \times 24 \times 4}$, with three different reshaping schemes to preprocess the input tensors of shape $(36 \times 24 \times 4)$: no reshaping ($\texttt{cities} \times \texttt{time} \times \texttt{measurements}$) (Figure 5 (a)); concatenating time and measurements ($\texttt{cities} \times (\texttt{time} * \texttt{measurements})$) (Figure 5 (b)); and concatenating cities and time (($\texttt{cities} * \texttt{time}) \times \texttt{measurements}$) (Figure 5 (c)). (See Subsection 7.1 for a discussion on reshaping and CP-dictionary learning). Roughly speaking, the single CP-dictionary atom we learn is a 3-mode tensor of shape $(36 \times 24 \times 4)$ that best approximates the evolution of four weather measurements during a randomly chosen 24-hour period from the original 2998-hour-long data subject to different constraints on the three modes depending how we reshape the input tensors.

In Figure 5, for each atom, the top left corner represents the first frontal slice (temperature), the bottom left the second frontal slice (humidity), the top right the third frontal slice (pressure), and the bottom right the fourth frontal slice (wind speed). The horizontal axis corresponds to time (in hours), each individual time series to a "row" in the first mode, and the vertical axis to the value of the corresponding entry in the CP-dictionary atom.

We emphasize the qualitative difference in the corresponding learned dictionaries. In the first example in Figure 5 (a), the CP-constraint is applied between all modes, so the single CP-dictionary atom $\mathbf{D} \in \mathbb{R}^{36\times 24\times 4}$ is given by the outer product of three marginal vectors, one for each of the three mode (analogous to Figure 3 (a)). Namely, let $\mathbf{D} = u_1 \otimes u_2 \otimes u_3$, where $u_1 \in \mathbb{R}^{36}$, $u_2 \in \mathbb{R}^{24}$, and $u_3 \in \mathbb{R}^4$. Then u_2 represents a 24-hour long time series, and for example, the humidity of the first city is approximated by $(u_1(1)u_3(2))u_2$. This makes the variability of the time-series across different cities and measurements restrictive, as shown in Figure 5 (a).

Next, in the second example in Figure 5 (b), the CP-constraint is applied only between cities and the other two modes combined, so the single CP-dictionary atom $\mathbf{D} \in \mathbb{R}^{36\times 24\times 4}$ is given by $\mathbf{D} = u_1 \otimes U_{23}$, where $u_1 \in \mathbb{R}^{36}$ and $U_{23} \in \mathbb{R}^{24\times 4}$. Thus, for each city, the 24-hour evolution of the four measurements need not be some scalar multiple of a single time evolution vector as before, as we can use the full 24×4 entries in U_{23} to encode such information. On the other hand, the variability of the joint 24-hour evolution of the four measurements across the cities should only be given by a scalar multiple, as we can observe in Figure 5 (b).

Lastly, in the third example in Figure 5 (c), the CP-constraint is applied only between the measurements (last mode) and the other two modes combined, so the single CP-dictionary atom $\mathbf{D} \in \mathbb{R}^{36\times 24\times 4}$ is given by $\mathbf{D} = U_{12} \otimes u_3$, where $U_{12} \in \mathbb{R}^{36\times 24}$ and $u_3 \in \mathbb{R}^4$. Thus, the 24-hour evolution of a latent measurement of the 36 cities can be encoded by the 36×24 matrix U_{12} without rank restriction. For each of the four measurements, this joint evolution pattern encoded in U_{12} is multiplied by a scalar. For example, the temperature evolution across 36 cities is modeled by $u_3(1)U_{12}$.

Acknowledgement

HL is partially supported by NSF DMS #2206296 and NSF DMS #2010035. CS and DN are grateful to and were partially supported by NSF BIGDATA #1740325, NSF DMS #2011140 and NSF DMS #2108479.

References

Abhishek Agarwal, Jianhao Peng, and Olgica Milenkovic. Online convex dictionary learning. In *Advances in Neural Information Processing Systems*, pages 13242–13252, 2019.

Animashree Anandkumar, Rong Ge, and Majid Janzamin. Learning overcomplete latent variable models through tensor methods. In *Conference on Learning Theory*, pages 36–112, 2015.

Sanjeev Arora, Rong Ge, and Ankur Moitra. New algorithms for learning incoherent and overcomplete dictionaries. In *Conference on Learning Theory*, pages 779–806, 2014.

Sanjeev Arora, Rong Ge, Tengyu Ma, and Ankur Moitra. Simple, efficient, and neural algorithms for sparse coding. In *Conference on learning theory*, pages 113–149. PMLR, 2015.

Boaz Barak, Jonathan A Kelner, and David Steurer. Dictionary learning and tensor decomposition via the sum-of-squares method. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 143–151, 2015.

Daniel Barson, Ali S Hamodi, Xilin Shen, Gyorgy Lur, R Todd Constable, Jessica A Cardin, Michael C Crair, and Michael J Higley. Simultaneous mesoscopic and two-photon imaging of neuronal activity in cortical circuits. *Nature methods*, 17(1):107–113, 2020.

Casey Battaglino, Grey Ballard, and Tamara G Kolda. A practical randomized cp tensor decomposition. *SIAM Journal on Matrix Analysis and Applications*, 39(2):876–901, 2018.

David Beniaguev. Historical hourly weather data 2012-2017, version 2. *https://www.kaggle.com/selfishgene/historical-hourly-weather-data*, 2017.

Michael W Berry and Murray Browne. Email surveillance using non-negative matrix factorization. *Computational & Mathematical Organization Theory*, 11(3):249–264, 2005.

Michael W Berry, Murray Browne, Amy N Langville, V Paul Pauca, and Robert J Plemmons. Algorithms and applications for approximate nonnegative matrix factorization. *Computational statistics & data analysis*, 52(1):155–173, 2007.

Dimitri P Bertsekas. Nonlinear programming. *Journal of the Operational Research Society*, 48(3): 334–334, 1997.

Dimitri P Bertsekas. *Nonlinear programming*. Athena scientific Belmont, 1999.

Rostyslav Boutchko, Debasis Mitra, Suzanne L Baker, William J Jagust, and Grant T Gullberg. Clustering-initiated factor analysis application for tissue classification in dynamic brain positron emission tomography. *Journal of Cerebral Blood Flow & Metabolism*, 35(7):1104–1111, 2015.

Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Steve Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng. *Handbook of markov chain monte carlo*. CRC press, 2011.

J Douglas Carroll and Jih-Jie Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of "Eckart-Young" decomposition. *Psychometrika*, 35(3):283–319, 1970.

Yang Chen, Xiao Wang, Cong Shi, Eng Keong Lua, Xiaoming Fu, Beixing Deng, and Xing Li. Phoenix: A weight-based network coordinate system using matrix factorization. *IEEE Transactions on Network and Service Management*, 8(4):334–347, 2011.

Weisheng Dong, Xin Li, Lei Zhang, and Guangming Shi. Sparsity-based image denoising via dictionary learning and structural clustering. In *Conference on Computer Vision and Pattern Recognition*, pages 457–464. IEEE, 2011.

Yishuai Du, Yimin Zheng, Kuang-chih Lee, and Shandian Zhe. Probabilistic streaming tensor decomposition. In *2018 IEEE International Conference on Data Mining*, pages 99–108. IEEE, 2018.

Bradley Efron, Trevor Hastie, Iain Johnstone, Robert Tibshirani, et al. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004.

Michael Elad. *Sparse and redundant representations: from theory to applications in signal and image processing*. Springer Science & Business Media, 2010.

Michael Elad and Michal Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image processing*, 15(12):3736–3745, 2006.

Kjersti Engan, Sven Ole Aase, and John Hakon Husoy. Frame based signal compression using method of optimal directions (mod). In *Proceedings of the IEEE International Symposium on Circuits and Systems VLSI*, volume 4, pages 1–4. IEEE, 1999.

Donald L Fisk. Quasi-martingales. *Transactions of the American Mathematical Society*, 120(3): 369–389, 1965.

Mohsen Ghassemi, Zahra Shakeri, Anand D Sarwate, and Waheed U Bajwa. Stark: Structured dictionary learning through rank-one tensor recovery. In *Proceedings of the IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, pages 1–5. IEEE, 2017.

Mohsen Ghassemi, Zahra Shakeri, Anand D Sarwate, and Waheed U Bajwa. Learning mixtures of separable dictionaries for tensor data: Analysis and algorithms. *IEEE Transactions on Signal Processing*, 68:33–48, 2019.

José Henrique de M Goulart, Maxime Boizard, Rémy Boyer, Gérard Favier, and Pierre Comon. Tensor cp decomposition with structured factor matrices: Algorithms and performance. *IEEE Journal of Selected Topics in Signal Processing*, 10(4):757–769, 2015.

Luigi Grippo and Marco Sciandrone. On the convergence of the block nonlinear gauss–seidel method under convex constraints. *Operations research letters*, 26(3):127–136, 2000.

Luigi Grippof and Marco Sciandrone. Globally convergent block-coordinate techniques for unconstrained optimization. *Optimization methods and software*, 10(4):587–637, 1999.

Naiyang Guan, Dacheng Tao, Zhigang Luo, and Bo Yuan. Online nonnegative matrix factorization with robust stochastic approximation. *IEEE Transactions on Neural Networks and Learning Systems*, 23(7):1087–1099, 2012.

Richard A Harshman et al. Foundations of the parafac procedure: Models and conditions for an "explanatory" multimodal factor analysis. 1970.

Furong Huang, UN Niranjan, Mohammad Umar Hakeem, and Animashree Anandkumar. Online tensor methods for learning latent variable models. *The Journal of Machine Learning Research*, 16(1):2797–2835, 2015.

Lara Kassab, Alona Kryshchenko, Hanbaek Lyu, Denali Molitor, Deanna Needell, and Elizaveta Rebrova. Detecting short-lasting topics using nonnegative tensor decomposition (preprint). *arXiv preprint arXiv:2010.01600*, 2021.

Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51 (3):455–500, 2009.

Alec Koppel, Garrett Warnell, Ethan Stump, and Alejandro Ribeiro. D4l: Decentralized dynamic discriminative dictionary learning. *IEEE Transactions on Signal and Information Processing over Networks*, 3(4):728–743, 2017.

Rohit Kulkarni. A Million News Headlines, 2018. URL `https://doi.org/10.7910/DVN/SYBGZL`.

Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788, 1999.

Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems*, pages 556–562, 2001.

Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y Ng. Efficient sparse coding algorithms. In *Advances in Neural Information Processing Systems*, pages 801–808, 2007.

Hyekyoung Lee, Jiho Yoo, and Seungjin Choi. Semi-supervised nonnegative matrix factorization. *IEEE Signal Processing Letters*, 17(1):4–7, 2009.

Kuang-Chih Lee, Jeffrey Ho, and David J Kriegman. Acquiring linear subspaces for face recognition under variable lighting. *IEEE Transactions on pattern analysis and machine intelligence*, 27(5): 684–698, 2005.

David A Levin and Yuval Peres. *Markov chains and mixing times*, volume 107. American Mathematical Soc., 2017.

Michael S Lewicki and Terrence J Sejnowski. Learning overcomplete representations. *Neural computation*, 12(2):337–365, 2000.

H Lyu, F Memoli, and D Sivakoff. Sampling random graph homomorphisms and applications to network data analysis. *arXiv:1910.09483*, 2019.

Hanbaek Lyu. Convergence of block coordinate descent with diminishing radius for nonconvex optimization. *arXiv preprint arXiv:2012.03503*, 2020.

Hanbaek Lyu, Georg Menz, Deanna Needell, and Christopher Strohmeier. Applications of online nonnegative matrix factorization to image and time-series data. In *Information Theory and Applications Workshop*, pages 1–9. IEEE, 2020a.

Hanbaek Lyu, Deanna Needell, and Laura Balzano. Online matrix factorization for markovian data and applications to network dictionary learning. *Journal of Machine Learning Research*, 21(251): 1–49, 2020b.

Hanbaek Lyu, Yacoub H Kureh, Joshua Vendrow, and Mason A Porter. Learning low-rank latent mesoscale structures in networks. *arXiv preprint arXiv:2102.06984*, 2021.

Congbo Ma, Xiaowei Yang, and Hu Wang. Randomized online CP decomposition. In *Proceedings of International Conference on Advanced Computational Intelligence*, pages 414–419. IEEE, 2018.

Liyan Ma, Lionel Moisan, Jian Yu, and Tieyong Zeng. A dictionary learning approach for poisson image deblurring. *IEEE Transactions on medical imaging*, 32(7):1277–1289, 2013.

Tengyu Ma, Jonathan Shi, and David Steurer. Polynomial-time tensor decompositions with sum-of-squares. In *IEEE 57th Annual Symposium on Foundations of Computer Science*, pages 438–446. IEEE, 2016.

Julien Mairal. Optimization with first-order surrogate functions. In *International Conference on Machine Learning*, pages 783–791. PMLR, 2013a.

Julien Mairal. Stochastic majorization-minimization algorithms for large-scale optimization. In *Advances in Neural Information Processing Systems*, pages 2283–2291, 2013b.

Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11(1):19–60, 2010.

Arthur Mensch, Julien Mairal, Bertrand Thirion, and Gaël Varoquaux. Stochastic subsampling for factorizing huge matrices. *IEEE Transactions on Signal Processing*, 66(1):113–128, 2017.

Sean P Meyn and Richard L Tweedie. *Markov chains and stochastic stability*. Springer Science & Business Media, 2012.

Yurii Nesterov. Introductory lectures on convex programming volume i: Basic course. *Lecture notes*, 3(4):5, 1998.

Bruno A Olshausen and David J Field. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision research*, 37(23):3311–3325, 1997.

Vardan Papyan, Yaniv Romano, Jeremias Sulam, and Michael Elad. Convolutional dictionary learning via local processing. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5296–5304, 2017.

Anand Rajaraman and Jeffrey David Ullman. *Mining of massive datasets*. Cambridge University Press, 2011.

Sirisha Rambhatla, Xingguo Li, and Jarvis Haupt. Noodl: Provable online dictionary learning and sparse coding. *In 7th International Conference on Learning Representations*, 2019.

Sirisha Rambhatla, Xingguo Li, and Jarvis Haupt. Provable online cp/parafac decomposition of a structured tensor via dictionary learning. *Advances in Neural Information Processing Systems*, 33, 2020.

K Murali Rao. Quasi-martingales. *Mathematica Scandinavica*, 24(1):79–92, 1969.

Bin Ren, Laurent Pueyo, Guangtun Ben Zhu, John Debes, and Gaspard Duchêne. Non-negative matrix factorization: robust extraction of extended structures. *The Astrophysical Journal*, 852 (2):104, 2018.

J. Rennie. 20 Newsgroups, 2008. URL `http://qwone.com/~jason/20Newsgroups/`.

Tselil Schramm and David Steurer. Fast and robust tensor decomposition with applications to dictionary learning. *arXiv preprint arXiv:1706.08672*, 2017.

Zahra Shakeri, Waheed U Bajwa, and Anand D Sarwate. Minimax lower bounds for kronecker-structured dictionary learning. In *IEEE International Symposium on Information Theory*, pages 1148–1152. IEEE, 2016.

Zahra Shakeri, Anand D Sarwate, and Waheed U Bajwa. Identifiability of kronecker-structured dictionaries for tensor data. *IEEE Journal of Selected Topics in Signal Processing*, 12(5):1047–1062, 2018.

Zahra Shakeri, Anand D Sarwate, Waheed U Bajwa, M Rodrigues, and Y Eldar. Sample complexity bounds for dictionary learning from vector-and tensor-valued data. In *Information Theoretic Methods in Data Science*. Cambridge Univ. Press Cambridge, UK, 2019.

Vatsal Sharan and Gregory Valiant. Orthogonalized als: A theoretically principled tensor decomposition algorithm for practical use. In *International Conference on Machine Learning*, pages 3095–3104. PMLR, 2017.

Amnon Shashua and Tamir Hazan. Non-negative tensor factorization with applications to statistics and computer vision. In *Proceedings of the 22nd international conference on Machine learning*, pages 792–799. ACM, 2005.

Arkadiusz Sitek, Grant T Gullberg, and Ronald H Huesman. Correction for ambiguous solutions in factor analysis using a penalized least squares objective. *IEEE transactions on medical imaging*, 21(3):216–225, 2002.

Shaden Smith, Kejun Huang, Nicholas D Sidiropoulos, and George Karypis. Streaming tensor factorization for infinite data sources. In *Proceedings of the SIAM International Conference on Data Mining*, pages 81–89. SIAM, 2018.

Tao Sun, Yuejiao Sun, and Wotao Yin. On markov chain gradient descent. In *Advances in Neural Information Processing Systems*, pages 9896–9905, 2018.

Will Wei Sun, Junwei Lu, Han Liu, and Guang Cheng. Provable sparse tensor decomposition. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 3(79):899–916, 2017.

Gongguo Tang and Parikshit Shah. Guaranteed tensor decomposition: A moment approach. In *International Conference on Machine Learning*, pages 1491–1500. PMLR, 2015.

Leo Taslaman and Björn Nilsson. A framework for regularized non-negative matrix factorization, with application to the analysis of gene expression data. *PloS one*, 7(11):e46331, 2012.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

Ledyard R Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3): 279–311, 1966.

Don Van Ravenzwaaij, Pete Cassey, and Scott D Brown. A simple introduction to markov chain monte–carlo sampling. *Psychonomic bulletin & review*, 25(1):143–154, 2018.

Nick Vannieuwenhoven, Karl Meerbergen, and Raf Vandebril. Computing the gradient in optimization algorithms for the cp decomposition in constant memory through tensor blocking. *SIAM Journal on Scientific Computing*, 37(3):C415–C438, 2015.

Anja Voss-Böhme. Multi-scale modeling in morphogenesis: a critical analysis of the cellular potts model. *PloS one*, 7(9), 2012.

Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

Yangyang Xu and Wotao Yin. A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion. *SIAM Journal on imaging sciences*, 6(3):1758–1789, 2013.

Renbo Zhao, Vincent Tan, and Huan Xu. Online nonnegative matrix factorization with general divergences. In *Artificial Intelligence and Statistics*, pages 37–45. PMLR, 2017.

Shuo Zhou, Nguyen Xuan Vinh, James Bailey, Yunzhe Jia, and Ian Davidson. Accelerating online cp decompositions for higher order tensors. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1375–1384, 2016.

Shuo Zhou, Sarah Erfani, and James Bailey. Online cp decomposition for sparse tensors. In *IEEE International Conference on Data Mining*, pages 1458–1463. IEEE, 2018.

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.

Appendix A. Background on Markov chains and MCMC

A.1 Markov chains

Here we give a brief account on Markov chains on countable state space (see, e.g., Levin and Peres (2017)). Fix a countable set Ω. A function $P : \Omega^2 \to [0, \infty)$ is called a *Markov transition matrix* if every row of P sums to 1. A sequence of Ω-valued random variables $(X_t)_{t\geq 0}$ is called a *Markov chain* with transition matrix P if for all $x_0, x_1, \ldots, x_n \in \Omega$,

$$\mathbb{P}(X_n = x_n \,|\, X_{n-1} = x_{n-1}, \ldots, X_0 = x_0) = \mathbb{P}(X_n = x_n \,|\, X_{n-1} = x_{n-1}) = P(x_{n-1}, x_n). \tag{85}$$

We say a probability distribution π on Ω a *stationary distribution* for the chain $(X_t)_{t\geq 0}$ if $\pi = \pi P$, that is,

$$\pi(x) = \sum_{y\in\Omega} \pi(y)P(y, x). \tag{86}$$

We say the chain $(X_t)_{t\geq 0}$ is *irreducible* if for any two states $x, y \in \Omega$ there exists an integer $t \geq 0$ such that $P^t(x, y) > 0$. For each state $x \in \Omega$, let $\mathcal{T}(x) = \{t \geq 1 \,|\, P^t(x, x) > 0\}$ be the set of times when it is possible for the chain to return to starting state x. We define the *period* of x by the greatest common divisor of $\mathcal{T}(x)$. We say the chain X_t is *aperiodic* if all states have period 1. Furthermore, the chain is said to be *positive recurrent* if there exists a state $x \in \Omega$ such that the expected return time of the chain to x started from x is finite. Then an irreducible and aperiodic Markov chain has a unique stationary distribution if and only if it is positive recurrent (Levin and Peres, 2017, Thm 21.21).

Given two probability distributions μ and ν on Ω, we define their *total variation distance* by

$$\|\mu - \nu\|_{TV} = \sup_{A\subseteq\Omega} |\mu(A) - \nu(A)|. \tag{87}$$

If a Markov chain $(X_t)_{t\geq 0}$ with transition matrix P starts at $x_0 \in \Omega$, then by (85), the distribution of X_t is given by $P^t(x_0, \cdot)$. If the chain is irreducible and aperiodic with stationary distribution π, then the convergence theorem (see, e.g., (Levin and Peres, 2017, Thm 21.14)) asserts that the distribution of X_t converges to π in total variation distance: As $t \to \infty$,

$$\sup_{x_0\in\Omega} \|P^t(x_0, \cdot) - \pi\|_{TV} \to 0. \tag{88}$$

See (Meyn and Tweedie, 2012, Thm 13.3.3) for a similar convergence result for the general state space chains. When Ω is finite, then the above convergence is exponential in t (see., e.g., (Levin and Peres, 2017, Thm 4.9))). Namely, there exists constants $\lambda \in (0, 1)$ and $C > 0$ such that for all $t \geq 0$,

$$\max_{x_0\in\Omega} \|P^t(x_0, \cdot) - \pi\|_{TV} \leq C\lambda^t. \tag{89}$$

Markov chain mixing refers to the fact that, when the above convergence theorems hold, then one can approximate the distribution of X_t by the stationary distribution π.

Remark 13 Our main convergence result in Theorem 3.1 assumes that the underlying Markov chain Y_t is irreducible, aperiodic, and defined on a finite state space Ω, as stated in (A1). This can be relaxed to countable state space Markov chains. Namely, Theorem 3.1 holds if we replace (A1) by

(A1)' *The observed data tensors $\mathbf{X}_t$ are given by $\mathbf{X}_t = \varphi(Y_t)$, where Y_t is an irreducible, aperiodic, and positive recurrent Markov on a countable and compact state space Ω and $\varphi : \Omega \to \mathbb{R}^{d\times n}$ is a bounded function. Furthermore, there exist constants $\beta \in (3/4, 1]$ and $\gamma > 2(1 - \beta)$ such that*

$$w_t = O(t^{-\beta}), \qquad \sup_{\mathbf{y}\in\Omega} \|P^t(\mathbf{y}, \cdot) - \pi\|_{TV} = O(t^{-\gamma}), \tag{90}$$

where P and π denote the transition matrix and unique stationary distribution of the chain Y_t.

Note that the polynomial mixing condition in (A1)' is automatically satisfied when Ω is finite due to (89). Polynomial mixing rate is available in most MCMC algorithms used in practice.

A.2 Markov chain Monte Carlo Sampling

Suppose we have a finite sample space Ω and probability distribution π on it. We would like to sample a random element $\omega \in \Omega$ according to the distribution π. *Markov chain Monte Carlo (MCMC)* is a sampling algorithm that leverages the properties of Markov chains we mentioned in Subsection A.1. Namely, suppose that we have found a Markov chain $(X_t)_{t\geq 0}$ on state space Ω that is irreducible, aperiodic[4], and has π as its unique stationary distribution. Denote its transition matrix as P. Then by (89), for any $\varepsilon > 0$, one can find a constant $\tau = \tau(\varepsilon) = O(\log \varepsilon^{-1})$ such that the conditional distribution of $X_{t+\tau}$ given X_t is within total variation distance ε from π regardless of the distribution of X_t. Recall such $\tau = \tau(\varepsilon)$ is called the *mixing time* of the Markov chain $(X_t)_{t\geq 1}$. Then if one samples a long Markov chain trajectory $(X_t)_{t\geq 1}$, the subsequence $(X_{k\tau})_{k\geq 1}$ gives approximate i.i.d. samples from π.

We can further compute how far the thinned sequence $(X_{k\tau})_{k\geq 1}$ is away from being independent. Namely, observe that for any two nonempty subsets $A, B \subseteq \Omega$,

$$
\begin{aligned}
&|\mathbb{P}(X_{k\tau} \in A,\, X_\tau \in B) - \mathbb{P}(X_{k\tau} \in A)\mathbb{P}(X_\tau \in B)| && (91)\\
&\quad = |\mathbb{P}(X_{k\tau} \in A) - \mathbb{P}(X_{k\tau} \in A \,|\, X_\tau \in B)|\; |\mathbb{P}(X_\tau \in B)| && (92)\\
&\quad \leq |\mathbb{P}(X_{k\tau} \in A) - \mathbb{P}(X_{k\tau} \in A \,|\, X_\tau \in B)| && (93)\\
&\quad \leq |\mathbb{P}(X_{k\tau} \in A) - \pi(A)| + |\pi(A) - \mathbb{P}(X_{k\tau} \in A \,|\, X_\tau \in B)| \leq \lambda^{k\tau} + \lambda^{(k-1)\tau}. && (94)
\end{aligned}
$$

Hence the correlation between $X_{k\tau}$ and X_τ is $O(\lambda^{(k-1)\tau})$.

For the lower bound, let us assume that X_t is *reversible* with respect to π, that is, $\pi(x)P(x,y) = \pi(y)P(y,x)$ for $x, y \in \Omega$ (e.g., random walk on graphs). Then $\tau(\varepsilon) = \Theta(\log \varepsilon^{-1})$ (see (Levin and Peres, 2017, Thm. 12.5)), which yields $\sup_{x\in\Omega} \|P^t(x,\cdot) - \pi\|_{TV} = \Theta(\lambda^t)$. Also, $\mathbb{P}(X_\tau \in B) > \delta > 0$ for some $\delta > 0$ whenever τ is large enough under the hypothesis. Hence

$$
\begin{aligned}
&|\mathbb{P}(X_{k\tau} \in A,\, X_\tau \in B) - \mathbb{P}(X_{k\tau} \in A)\mathbb{P}(X_\tau \in B)| && (95)\\
&\quad \geq \delta^{-1}\, |\mathbb{P}(X_{k\tau} \in A) - \mathbb{P}(X_{k\tau} \in A \,|\, X_\tau \in B)| && (96)\\
&\quad \geq \big|\, |\mathbb{P}(X_{k\tau} \in A) - \pi(A)| - |\pi(A) - \mathbb{P}(X_{k\tau} \in A \,|\, X_\tau \in B)| \,\big| \geq c\lambda^{(k-1)\tau} && (97)
\end{aligned}
$$

for some constant $c > 0$. Hence the correlation between $X_{k\tau}$ and X_τ is $\Theta(\lambda^{(k-1)\tau})$. In particular, the correlation between two consecutive terms in $(X_{k\tau})_{k\geq 1}$ is of $\Theta(\lambda^\tau) = \Theta(\varepsilon)$. Thus, we can make the thinned sequence $(X_{k\tau})_{k\geq 1}$ arbitrarily close to being i.i.d. for π, but if X_t is reversible with respect to π, the correlation within the thinned sequence is always nonzero.

In practice, one may not know how to estimate the mixing time $\tau = \tau(\varepsilon)$. In order to empirically assess that the Markov chain has mixed to the stationary distribution, multiple chains are run for diverse mode exploration, and their empirical distribution is compared to the stationary distribution (a.k.a. multistart heuristic). See Brooks et al. (2011) for more details on MCMC sampling.

Appendix B. Auxiliary lemmas

4. Aperiodicity can be easily obtained by making a given Markov chain lazy, that is, adding a small probability ε of staying at the current state. Note that this is the same as replacing the transition matrix P by $P_\varepsilon := (1-\varepsilon)P + \varepsilon I$ for some $\varepsilon > 0$. This 'lazyfication' does not change stationary distributions, as $\pi P = \pi$ implies $\pi P_\varepsilon = \pi$.

Lemma 14 (Convex Surrogate for Functions with Lipschitz Gradient) *Let $f : \mathbb{R}^p \to \mathbb{R}$ be differentiable and ∇f be L-Lipschitz continuous. Then for each $\theta, \theta' \in \mathbb{R}^p$,*

$$\left| f(\theta') - f(\theta) - \nabla f(\theta)^T(\theta' - \theta) \right| \leq \frac{L}{2}\|\theta - \theta'\|_F^2. \tag{98}$$

Proof This is a classical Lemma. See (Nesterov, 1998, Lem 1.2.3). ■

For each $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_n \times b}$ and $\mathcal{D} \in \mathbb{R}^{I_1 \times R} \times \ldots \mathbb{R}^{I_n \times R}$, denote

$$H^\star(\mathcal{X}, \mathcal{D}) \in \operatorname*{arg\,min}_{H \in \mathcal{C}^{\text{code}}} \ell(\mathcal{X}, \mathcal{D}, H). \tag{99}$$

Recall Assumption (A1)'. For each subset S of a Euclidean space, denote $\|S\|_F = \sup_{x \in S}\|x\|_F$. The following boundedness results for the codes H_t and aggregate tensors $A_t, \mathbf{B}_t$ are easy to derive.

Lemma 15 *Assume (A1)' and (A2). Then the following hold:*

(i) *For all $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_n \times b}$ and $\mathcal{D} \in \mathcal{C}^{\text{dict}}$,*

$$\|H^\star(\mathcal{X}, \mathcal{D})\|_F^2 \leq \lambda^{-2}\|\varphi(\Omega)\|_F^4 < \infty. \tag{100}$$

(ii) *For any sequences $(\mathcal{X}_t)_{t \geq 1}$ in $\mathbb{R}^{I_1 \times \cdots \times I_n \times b}$ and $(\mathcal{D}_t)_{t \geq 1}$ in $\mathcal{C}$, define A_t and $\mathbf{B}_t$ recursively as in Algorithm 1. Then for all $t \geq 1$, we have*

$$\|A_t\|_F \leq \lambda^{-2}\|\varphi(\Omega)\|_F^4, \qquad \|\mathbf{B}_t\|_F \leq \lambda^{-1}\|\varphi(\Omega)\|_F^3. \tag{101}$$

Proof Omitted. See (Lyu et al., 2020b, Prop. 7.2). ■

The following lemma shows Lipschitz continuity of the loss function $\ell(\varphi(\cdot), \cdot)$ defined in (10). Since Ω and $\mathcal{C}^{\text{code}}$ are both compact, this also implies that $\mathcal{D} \mapsto \hat{f}_t(\mathcal{D})$ and $\mathcal{D} \mapsto f_t(\mathcal{D})$ are L-Lipschitz for some $L > 0$ uniformly for all $t \geq 0$.

Lemma 16 *Suppose (A1)' and (A2) hold, and let $M = 2\|\varphi(\Omega)\|_F + 2\|\mathcal{C}^{\text{dict}}\|_F\|\varphi(\Omega)\|_F^2/\lambda$. Then for each $Y_1, Y_2 \in \Omega$ and $\mathcal{D}_1, \mathcal{D}_2 \in \mathcal{C}^{\text{dict}}$,*

$$|\ell(\varphi(Y_1), \mathcal{D}_1) - \ell(\varphi(Y_2), \mathcal{D}_2)| \leq M\left(\|Y_1 - Y_2\|_F + \lambda^{-1}\|\varphi(\Omega)\|_F\|\mathcal{D}_1 - \mathcal{D}_2\|_F\right). \tag{102}$$

Proof Omitted. See (Lyu et al., 2020b, Prop. 7.3). ■

The following deterministic statement on converging sequences is due to (Mairal et al., 2010).

Lemma 17 *Let $(a_n)_{n \geq 0}$ and $(b_n)_{\geq 0}$ be non-negative real sequences such that*

$$\sum_{n=0}^{\infty} a_n = \infty, \qquad \sum_{n=0}^{\infty} a_n b_n < \infty, \qquad |b_{n+1} - b_n| = O(a_n). \tag{103}$$

Then $\lim_{n \to \infty} b_n = 0$.

Proof Omitted. See (Mairal, 2013b, Lem. A.5). ■

Lemma 18 *Under the assumptions (A1)' and (A2),*

$$\mathbb{E}\left[\sup_{W\in\mathcal{C}^{\mathrm{dict}}}\sqrt{t}\left|f(\mathcal{D})-\frac{1}{t}\sum_{s=1}^{t}\ell(\mathcal{X}_s,\mathcal{D})\right|\right]=O(1). \tag{104}$$

Furthermore, $\sup_{W\in\mathcal{C}}\left|f(\mathcal{D})-\frac{1}{t}\sum_{s=1}^{t}\ell(\mathcal{X}_s,\mathcal{D})\right|\to 0$ *almost surely as* $t\to\infty$.

Proof Omitted. See (Lyu et al., 2020b, Lem. 7.8). ∎

In the following lemma, we generalize the uniform convergence results in Lemma 18 for general weights $w_t\in(0,1)$ (not only for the 'balanced weights' $w_t=1/t$). The original lemma is due to Mairal (Mairal, 2013b, Lem B.7), which originally extended the uniform convergence result to weighted empirical loss functions with respect to i.i.d. input signals. A similar argument gives the corresponding result in our Markovian case (A1)', which was also used in Lyu et al. (2020b). In Lemma 18 below, we also generalize the statement for the weights w_t satisfy the required monotonicity property $w_{t+1}^{-1}-w_t^{-1}\le 1$ only asymptotically. See Remark 21 for more discussion.

Lemma 19 *Suppose (A1)'-(A2) hold, and assume that there exist an integer* $T\ge 1$ *such that* $w_{t+1}^{-1}-w_t^{-1}\le 1$ *for all* $t\ge T$. *Also assume that there are some constants* $c>0$ *and* $\gamma\in(0,1]$ *such that* $w_t\ge ct^{-\gamma}$ *for all* $t\ge 1$. *Further assume that if* $T\ge 1$ *and* $\gamma=1$, *then* $c\ge 1/2$. *Then there exists a constant* $C=C(T)>0$ *such that*

$$\mathbb{E}\left[\sup_{W\in\mathcal{C}^{\mathrm{dict}}}|f(\mathcal{D})-f_t(\mathcal{D})|\right]\le Cw_t\sqrt{t}. \tag{105}$$

Furthermore, if $\sum_{t=1}^{\infty}w_t=\infty$, $\sum_{t=1}^{\infty}w_t^2\sqrt{t}<\infty$, *then* $\sup_{\mathcal{D}\in\mathcal{C}^{\mathrm{dict}}}|f(\mathcal{D})-f_t(\mathcal{D})|\to 0$ *almost surely as* $t\to\infty$.

Proof Fix $t\in\mathbb{N}$. Recall the weighted empirical loss $f_t(\mathcal{D})$ defined recursively using the weights $(w_s)_{s\ge 0}$ in (11). For each $0\le s\le t$, denote $w_s^t=w_s\prod_{j=s}^{t}(1-w_j)$ and set $w_0^t=0$. Then for each $t\in\mathbb{N}$, we can write $f_t(\mathcal{D})=\sum_{s=1}^{t}\ell(X_s,W)w_s^t$. Moreover, note that $w_1^t,\dots,w_t^t>0$ and $w_1^t+\cdots+w_t^t=1$. Define $F_i(\mathcal{D})=(t-i+1)^{-1}\sum_{j=1}^{t}\ell(X_i,W)$ for each $1\le i\le t$. By Lemma 18, there exists a constant $c_1>0$ such that

$$\mathbb{E}\left[\sup_{W\in\mathcal{C}}|F_i(\mathcal{D})-f(\mathcal{D})|\right]\le\frac{c_1}{\sqrt{t-i+1}} \tag{106}$$

for all $t\ge 1$ and $1\le i\le t$. Noting that $[w_1^t,\dots,w_t^t]$ is a probability distribution on $\{1,\dots,t\}$, a simple calculation shows the following important identity

$$f_t-f=\sum_{i=1}^{t}(w_i^t-w_{i-1}^t)(t-i+1)(F_i-f), \tag{107}$$

with the convention of $w_0^t=0$. Also, suppose $T\ge 1$ is such that $w_k^{-1}-w_{k-1}^{-1}\le 1$ for $k\ge T$. Note that for $i\ge 2$, $w_{i-1}^t\le w_i^t$ if and only if $w_{i-1}(1-w_i)\le w_i$ if and only if $w_i^{-1}-w_{i-1}^{-1}\le 1$. Hence for each $n>T$ and $k\ge T$, we have $w_k^t\le w_{k+1}^t\le\cdots\le w_t^t=w_t$. Then observe that

$$\mathbb{E}\left[\sup_{\mathcal{D}\in\mathcal{C}^{\mathrm{dict}}}|f_t(\mathcal{D})-\bar{\psi}(\mathcal{D})|\right]\le\mathbb{E}\left[\sum_{i=1}^{t}|w_i^t-w_{i-1}^t|(t-i+1)\sup_{W\in\mathcal{C}^{\mathrm{dict}}}|f_i(\mathcal{D})-f(\mathcal{D})|\right] \tag{108}$$

$$= \sum_{i=1}^{t} |w_i^t - w_{i-1}^t|(t-i+1)\,\mathbb{E}\left[\sup_{\mathcal{D}\in\mathcal{C}^{\mathrm{dict}}} |f_i(\mathcal{D}) - f(\mathcal{D})|\right] \tag{109}$$

$$\leq \sum_{i=1}^{t} |w_i^t - w_{i-1}^t| c_1\sqrt{t-i+1} \tag{110}$$

$$\leq c_1\sqrt{t}\left(\sum_{i=1}^{T} |w_i^t - w_{i-1}^t| + \sum_{i=T}^{t} (\hat{w}_i^t - \hat{w}_{i-1}^t)\right) \tag{111}$$

$$\leq c_1\sqrt{t}\left(w_t + \sum_{i=1}^{T} w_i^t\right). \tag{112}$$

By using Lemma 20, we have $\sum_{i=1}^{T} w_i^t = O(1/n)$. Furthermore, since $w_{t+1}^{-1} - w_t^{-1} \leq 1$ for all $t \geq T$, we deduce $w_t^{-1} - w_T^{-1} \leq t - T$ for all $t \geq T$, so $w_t \geq \frac{1}{t-T+w_T^{-1}}$. Thus for some constant $c > 0$, we have $w_t \geq \frac{1}{t+c}$ for all $t \geq T$. Thus the last displayed expression above is of $O(w_t\sqrt{t})$. This shows (105). We can show the part by using Lemma 17 in Appendix B, following the argument in the proof of (Mairal, 2013b, Lem. B7). See the reference for more details. ■

The following lemma was used in the proof of Lemma 19.

Lemma 20 *Fix a sequence $(w_n)_{n\geq 1}$ of numbers in $(0, 1]$. Denote $w_k^n := w_k \prod_{i=k+1}^{n}(1 - w_i)$ for $1 \leq k \leq n$. Suppose $w_n^{-1} - w_{n-1}^{-1} \leq 1$ for all sufficiently large $n \geq 1$. Fix $T \geq 1$. Then for all $n \geq T$,*

$$\sum_{i=1}^{T} w_i^n = O(1/n). \tag{113}$$

Proof Suppose $w_n^{-1} - w_{n-1}^{-1} \leq 1$ for all $n \geq N$ for some $N \geq 1$. It follows that $w_n^{-1} - w_N^{-1} \leq n - N$, so $w_n \geq \frac{1}{n-N+w_N^{-1}}$. Hence for some constant $c > 0$, $w_n \geq \frac{1}{n+c}$ for all $n \geq N$. Denote $a \vee b = \max(a, b)$. Then note that

$$w_k^n = w_k \exp\left(\sum_{i=k+1}^{n} \log(1 - w_i)\right) \leq \exp\left(-\sum_{i=k+1}^{n} w_i\right) \tag{114}$$

$$\leq \exp\left(-\int_{N\vee(k+1)}^{n} \frac{1}{x+c}\,dx\right) = \frac{[N \vee (k+1)] + c}{n+c}, \tag{115}$$

where the second inequality uses $w_k \leq 1$ and the following inequality uses $\log(1-a) \leq -a$ for $a < 1$. Hence for each fixed $1 \leq T \leq n$, we have

$$\sum_{k=1}^{T} w_k^n \leq T\left((N \vee (T+1)) + c\right)\frac{1}{n+c}. \tag{116}$$

This shows the assertion. ■

Remark 21 In the original statement of (Mairal, 2013b, Lem B.7), the assumption that $w_{t+1}^{-1} - w_t^{-1} \leq 1$ for sufficiently large t was not used, and it seems that the argument in Mairal (2013b) needs this assumption. To give more detail, the argument begins with writing the empirical loss $f_t(\cdot) = \sum_{k=1}^{t} w_k^t\,\ell(\mathcal{X}_k, \cdot)$, where $w_k^t := w_k(1 - w_{k-1})\cdots(1 - w_t)$, and proceeds with assuming the

monotonicity $w_1^t \leq \cdots \leq w_t^t$, which is equivalent to $w_k \geq w_{k-1}(1-w_k)$ for $2 \leq k \leq t$. In turn, this is equivalent to $w_k^{-1} - w_{k-1}^{-1} \leq 1$ for $2 \leq k \leq t$. Note that this condition implies $w_k^{-1} \leq (k-1) + w_1^{-1}$, or $w_k \geq \frac{1}{k-1+w_1^{-1}}$, where $w_1 \in [0,1]$ is a fixed constant. This means that, asymptotically, w_k cannot decay faster than the balanced weight $1/k$, which gives $w_k^t \equiv 1/t$ for $k \in \{1, \ldots, t\}$. Note that we proved Lemma 20 with requiring $w_{t+1}^{-1} - w_t^{-1} \leq 1$ for all sufficiently large t.

Next, we will argue that (A3') implies (A3). It is clear that if the sequence $w_t \in (0,1]$ satisfies (A3'), then $\sum_{t=1}^{\infty} w_t = \infty$ and $\sum_{t=1}^{\infty} w_t^2 \sqrt{t} < \infty$. So it remains to verify $w_t^{-1} - w_{t-1}^{-1} \leq 1$ for sufficiently large t. Suppose $w_t = \Theta(t^{-\beta}(\log t)^{-\delta})$ for some $\beta \in [0,1]$ and $\delta \geq 0$. Let $c_1, c_2 > 0$ be constants such that $w_t t^{\beta} (\log t)^{\delta} \in [c_1, c_2]$ for all $t \geq 1$. Then by the mean value theorem,

$$w_{t+1}^{-1} - w_t^{-1} \leq c_2 \left((t+1)^{\beta} (\log(t+1))^{\delta} - t^{\beta} (\log t)^{\delta} \right) \tag{117}$$

$$\leq c_2 \sup_{t \leq s \leq t+1} \left(\beta s^{\beta-1} (\log s)^{\delta} + \delta s^{\beta-1} (\log s)^{\delta-1} \right) \tag{118}$$

$$\leq c_2 \sup_{t \leq s \leq t+1} s^{\beta-1} (\log s)^{\delta-1} \left((\log s) + \delta \right). \tag{119}$$

Since $t \geq 1$, the last expression is of $o(1)$ if $\beta < 1$. Otherwise, $w_t = t^{-1}$ for $t \geq 1$ by (A3'). Then $w_{t+1}^{-1} - w_t^{-1} \equiv 1$ for all $t \geq 1$.

Appendix C. Bounded memory implementation of Algorithm 1

In this section, we introduce an alternative implementation of Algorithm 1 that uses bounded memory that is independent of the number T of minibatches of data tensors being processed. This will be done by replacing the step for computing the surrogate loss function $\hat{f}_t$ with computing two 'aggregate tensors' based on our deterministic analysis in Proposition 4. The total amount of information fed in to the algorithm is $O(T \prod_{i=1}^{n} I_n)$ and $T \to \infty$, whereas Algorithm 2 stores only $O(R \prod_{i=1}^{n} I_n)$ (recall that R is the number of dictionary atoms to be learned and T is the number of minibatches of data tensors that have arrived). This is an inherent memory efficiency of online algorithms against non-online algorithms (see, e.g., Mairal et al. (2010)).

We describe how Algorithm 2 is derived and why it is equivalent to Algorithm 1. By the time that the new data tensor $\mathcal{X}_t$ arrives, the algorithm have computed previous loading matrices $U_{t-1}^{(1)}, \ldots, U_{t-1}^{(n)}$ and two aggregate tensors $A_{t-1} \in \mathbb{R}^{R \times R}$ and $\mathbf{B}_{t-1} \in \mathbb{R}^{I_1 \times \cdots \times I_n \times R}$. Then one computes the code matrix $H_t \in \mathcal{C}^{\text{code}} \subseteq \mathbb{R}^{R \times b}$ by solving the convex optimization problem in (120), and then updates the aggregate tensors $A_t \leftarrow A_{t-1}$ and $\mathbf{B}_t \leftarrow \mathbf{B}_{t-1}$. In order to perform the block coordinate descent to update the loading matrices $U_i^{(t)}$ in eq. (18) of Algorithm 1, we appropriately recompute intermediate aggregate matrices $\overline{A}_i$ and $\overline{B}_i$ using Algorithm 3 so that we are correctly minimizing the surrogate loss function $\hat{f}_t$ in (21) marginally according to Proposition 4 **(ii)**.

Appendix D. Auxiliary Algorithms

In this section, we give auxiliary algorithms that are used to solve convex sub-problems in coding and loading matrix updates for the main algorithm (Algorithm 1 for online CPDL). We denote by Π_S the projection operator onto the given subset S defined on the respective ambient space. For each matrix A, denote by $[A]_{\bullet i}$ (resp., $[A]_{i\bullet}$) the ith column (resp., row) of A.

Algorithm 2 Online CP-Dictionary Learning (Bounded Memory Implementation)

Input: $(\mathcal{X}_t)_{1\le t\le T}$ (minibatches of data tensors in $\mathbb{R}_{\ge 0}^{I_1\times\cdots\times I_n\times b}$); $[U_0^{(1)},\dots,U_0^{(n)}]\in \mathbb{R}_{\ge 0}^{I_1\times R}\times\cdots\times\mathbb{R}_{\ge 0}^{I_n\times R}$ (initial loading matrices); $c'>0$ (search radius constant);

Constraints: $\mathcal{C}^{(i)}\subseteq\mathbb{R}^{I_i\times R}$, $1\le i\le n$, $\mathcal{C}^{\text{code}}\subseteq\mathbb{R}^{R\times b}$ (e.g., nonnegativity constraints)

Parameters: $R\in\mathbb{N}$ (# of dictionary atoms); $\lambda\ge 0$ (ℓ_1-regularizer); $(w_t)_{t\ge 1}$ (weights in $(0,1]$);

Initialize aggregate tensors $A_0\in\mathbb{R}^{R\times R}$, $\mathbf{B}_0\in\mathbb{R}^{I_1\times\cdots\times I_n\times R}$;

For $t=1,\dots,T$ **do:**

Coding: Compute the optimal code matrix

$$H_t\leftarrow\underset{H\in\mathcal{C}^{\text{code}}\subseteq\mathbb{R}^{R\times b}}{\arg\min}\ \ell(\mathcal{X}_t,U_{t-1}^{(1)},\dots,U_{t-1}^{(n)},H);\quad\text{(using Algorithm 4)}\tag{120}$$

Update aggregate tensors:

$$A_t\leftarrow(1-w_t)A_{t-1}+w_tH_tH_t^T\in\mathbb{R}^{R\times R};\tag{121}$$

$$\mathbf{B}_t\leftarrow(1-w_t)\mathbf{B}_{t-1}+w_t(\mathcal{X}_t\times_{n+1}H_t^T)\in\mathbb{R}^{I_1\times\cdots\times I_n\times R};\tag{122}$$

Update dictionary:

For $i=1,\dots,n$ **do:**

$\overline{A}_{t;i}\in\mathbb{R}^{R\times R}$, $\overline{B}_{t;i}\in\mathbb{R}^{I_i\times R}$

$\leftarrow$ Algorithm 3 with input $A_t,\mathbf{B}_t,U_t^{(1)},\dots,U_t^{(i-1)},U_t^{(i)},U_{t-1}^{(i+1)},\dots,U_{t-1}^{(n)},i$;

$\mathcal{C}_t^{(i)}\leftarrow\left\{U\in\mathcal{C}^{(i)}\,\middle|\,\|U-U_{t-1}^{(i)}\|_F\le c'w_t\right\}$; (Restrict the search radius by w_t)

$U_t^{(i)}\leftarrow\arg\min_{U\in\mathcal{C}_t^{(i)}}\left[\text{tr}(U\overline{A}_{t;i}U^T)-2\text{tr}(U\,\overline{B}_{t;i}^T)\right]$; (Using Algorithm 5)

End for

End for

Return: $[U_T^{(1)},\dots,U_T^{(n)}]\in\mathcal{C}^{(1)}\times\cdots\times\mathcal{C}^{(n)}$;

Algorithm 6 Alternating Least Squares for NCPD

Input: $\mathbf{X}\in\mathbb{R}_{\ge 0}^{I_1\times\cdots\times I_m}$ (data tensor); $R\in\mathbb{N}$ (rank parameter); $\mathcal{D}_0=(U_0^{(1)},\cdots,U_0^{(m)})\in\mathbb{R}_{\ge 0}^{I_1\times R}\times\cdots\times\mathbb{R}_{\ge 0}^{I_m\times R}$ (initial loading matrices); N (number of iterations);

for $n=1,\dots,N$ **do**:

Update loading matrices $\mathcal{D}_n=[U_n^{(1)},\cdots,U_n^{(m)}]$ by

For $i=1,\cdots,m$ **do**:

$$\mathbf{A}\leftarrow\texttt{Out}(U_{n-1}^{(1)},\dots,U_{n-1}^{(i-1)},U_{n-1}^{(i+1)},\dots,U_{n-1}^{(m-1)})^{(m)}\in\mathbb{R}^{(I_1\times\cdots\times I_{i-1}\times I_{i+1}\times\cdots\times I_m)\times R}\tag{129}$$

$$B\leftarrow\text{unfold}(\mathbf{A},m)\in\mathbb{R}^{(I_1\cdots I_{i-1}I_{i+1}\cdots I_m)\times R}\tag{130}$$

$$U_n^{(i)}\in\underset{U\in\mathbb{R}_{\ge 0}^{I_i}}{\arg\min}\|\text{unfold}(\mathbf{X},i)-B(U^{(i)})^T\|^2\tag{131}$$

$\triangleright$ (unfold$(\cdot,i)$ denotes the mode-i tensor unfolding (see Kolda and Bader (2009))) (132)

end for

end for

output: $\mathcal{D}_N$

Algorithm 3 Intermediate Aggregation

Input: $A \in \mathbb{R}^{R\times R}$, $B \in \mathbb{R}^{I_1\times\cdots\times I_n\times R}$, $[U_1,\ldots,U_n] \in \mathbb{R}^{I_1\times R}\times\ldots\times\mathbb{R}^{I_n\times R}$, $1\le j\le n$

Do:

$$\overline{A}_i = A \odot U_1^T U_1 \odot \ldots \odot U_{i-1}^T U_{i-1} \odot U_{i+1}^T U_{i+1} \odot \ldots \odot U_n^T U_n \in \mathbb{R}^{R\times R} \tag{123}$$

For $r = 1,\ldots,R$ **do:**

$$B(,r) := \text{mode-}(n+1)\text{ slice of } B \text{ at coordinate } r \tag{124}$$

$$b_{i;r} = B(,r) \times_1 U_1(:,r) \times_2 \cdots \times_{i-1} U_{i-1}(:,r) \times_{i+1} U_{i+1}(:,r) \times_{i+2} \cdots \times_n U_n(:,r) \in \mathbb{R}^{I_i} \tag{125}$$

$$\overline{B}_{t;i} = I_i \times R \text{ matrix whose } r\text{th column is } b_{i;r} \tag{126}$$

End for

Return:

$$\overline{A}_i = \overline{A}_i(A, U_1,\ldots,U_{i-1},U_{i+1},\ldots,U_n)$$
$$\overline{B}_i = \overline{B}_i(B, U_1,\ldots,U_{i-1},U_{i+1},\ldots,U_n)$$

Algorithm 4 Coding

Input: $X \in \mathbb{R}^{M\times b}$: data matrix, $W \in \mathbb{R}^{M\times R}$: dictionary matrix

$\lambda \ge 0$: sparsity regularizer

$\mathcal{C}^{\text{code}} \subseteq \mathbb{R}^{R\times b}$: constraint set of codes

Repeat until convergence:

Do

$$H \leftarrow \Pi_{\mathcal{C}^{\text{code}}}\left(H - \frac{1}{\operatorname{tr}(W^TW)}(W^TWH - W^TX + \lambda J)\right), \tag{127}$$

where $J \subseteq \mathbb{R}^{R\times b}$ is all ones matrix.

Return $H \in \mathcal{C}^{\text{code}} \subseteq \mathbb{R}^{R\times b}$

Algorithm 5 Loading matrix update

Variables:

$U \in \mathcal{C}^{(i)} \subseteq \mathbb{R}^{I_i\times R}$: previous jth loading matrix

$(\overline{A}_i, \overline{B}_{t;j}) \in \mathbb{R}^{R\times R}\times\mathbb{R}^{R\times(I_1\ldots I_n)}$: intermediate loading matrices computed previously

Repeat until convergence:

For $i = 1$ **to** R**:**

$$[U]_{\bullet i} \leftarrow \Pi_{\mathcal{C}^{(i)}}\left([U]_{\bullet i} - \frac{1}{[\overline{A}_t]_{ii}+1}(U[\overline{A}_i]_{\bullet i} - [\overline{B}_{t;j}^T]_{\bullet i})\right) \tag{128}$$

Return $U \in \mathcal{C}^{(i)} \subseteq \mathbb{R}^{I_i\times R}$

Algorithm 7 Multiplicative Update for NCPD

Input: $\mathbf{X} \in \mathbb{R}_{\geq 0}^{I_1 \times \cdots \times I_m}$ (data tensor); $R \in \mathbb{N}$ (rank parameter); $\mathcal{D}_0 = (U_0^{(1)}, \cdots, U_0^{(m)}) \in \mathbb{R}_{\geq 0}^{I_1 \times R} \times \cdots \times \mathbb{R}_{\geq 0}^{I_m \times R}$ (initial loading matrices); N (number of iterations);

for $n = 1, \ldots, N$ **do**:

Update loading matrices $\mathcal{D}_n = [U_n^{(1)}, \cdots, U_n^{(m)}]$ by

For $i = 1, \cdots, m$ **do**:

$$\mathbf{A} \leftarrow \texttt{Out}(U_{n-1}^{(1)}, \ldots, U_{n-1}^{(i-1)}, U_{n-1}^{(i+1)}, \ldots, U_{n-1}^{(m-1)})^{(m)} \in \mathbb{R}^{(I_1 \times \cdots \times I_{i-1} \times I_{i+1} \times \cdots \times I_m) \times R} \tag{133}$$

$$B \leftarrow \text{unfold}(\mathbf{A}, m) \in \mathbb{R}^{(I_1 \cdots I_{i-1} I_{i+1} \cdots I_m) \times R} \tag{134}$$

$$U_n^{(i)} \leftarrow U_{n-1}^{(i)} \odot (\text{unfold}(\mathbf{X}, i))^T B \oslash (U B^T B) \tag{135}$$

$$\triangleright \text{ (unfold}(\cdot, i) \text{ denotes the mode-}i \text{ tensor unfolding (see Kolda and Bader (2009)))} \tag{136}$$

$$\triangleright \text{ } (\odot \text{ and } \oslash \text{ denote entrywise product and division)} \tag{137}$$

end for

end for

output: $\mathcal{D}_N$