ONLINE SKETCHING FOR BIG DATA SUBSPACE LEARNING

Morteza Mardani and Georgios B. Giannakis

Dept. of ECE and Digitial Technology Center, University of Minnesota

ABSTRACT

Sketching (a.k.a. subsampling) high-dimensional data is a crucial task to facilitate data acquisition process e.g., in magnetic resonance imaging, and to render affordable 'Big Data' analytics. *Multidimensional nature* and the need for *real-time processing* of data however pose major obstacles. To cope with these challenges, the present paper brings forth a novel *real-time sketching* scheme that exploits the correlations across data stream to learn a latent subspace based upon tensor PARAFAC decomposition 'on the fly.' Leveraging the online subspace updates, we introduce a notion of *importance score*, which is subsequently adapted into a randomization scheme to *predict* a minimal subset of important features to acquire in the next time instant. Preliminary tests with synthetic data corroborate the effectiveness of the novel scheme relative to uniform sampling.

Index Terms— Tensor, randomization, streaming data, subspace learning.

1. INTRODUCTION

Extracting latent low dimensional structure from high dimensional datasets is of paramount importance in timely inference tasks encountered with 'Big Data' analytics. Modern datasets are oftentimes indexed by three or more variables giving rise to a tensor, that is a data cube or a mutliway array [4]. For instance, in magnetic resonance imaging (MRI) one can form a three-way tensor by stacking high-resolution MR snapshots as tensor slices [10]. Unraveling structure of a large-scale tensor is a daunting task in nowadays 'Big Data' applications primarily due to: (c1) the sheer volume of data, and (c2) the need for real-time processing. Sketching as a method of choice to cope with (c1) typically needs the entire dataset to rank individual each datum according to its level of innovation towards learning. This is however against the concern of (c2).

The past work on sketching of high-dimensional data predominantly focus on one- and two-way arrays, where data samples are ranked according to a certain importance (or so-termed leverage) score that depends on the entire *batch* of data; see e.g., [8]. Given a sketch of data, our precursors in [6, 7] have paved the roadblocks associated with the growing scale and streaming nature of data. In essence, [6,7] introduces a novel framework for online tensor subspace learning with rank regularization from incomplete (subsampled) data. To date, *real-time sketching* for learning from streaming data is still in infancy. Being computationally appealing, this task is also well motivated for design of experiments in numerous applications such as dynamic (cardiac) MRI, where the slow acquisition speed and the moving nature of a patient's heart dictates acquiring only a small fraction of data per cardiac snapshot [10].

The present paper bridges this gap by leveraging the correlations across the data streams. Focusing on three-way tensors, seen as a stream of correlated slices (matrices), PARAFAC decomposition is adapted to effect low rank for the latent subspace. The sketch here refers to a (small) subset of slice features, to be used to interpolate the missing ones. Broadening the scope of our precursor subspace learning and imputation schemes in [6, 7], our basic idea is to leverage the intermediate estimates of the subspace bases revealed by the online scheme to *predict* a subset of most important features to acquire in the *next* slice. Towards this end, we adopt a measure of importance based upon the bases, that is then adapted into a randomization scheme to collect a subset of important features. Simulated tests with synthetic data confirm the convergence and effectiveness of the novel scheme upon running the algorithm with a warm initialization.

Notation: Operators $()^\top$, and $\circ$ denote transposition, and outer product, respectively; $\mathrm{diag}(\mathbf{x})$ is a diagonal matrix with diagonal entries $\mathbf{x}$; $|\cdot|$ is the cardinality of a set; $\|\cdot\|_2$ the ℓ_2-norm of a vector; $\|\cdot\|_F$ the Frobenius norm of a matrix; and the set $[K] := \{1, 2, \ldots, K\}$.

2. PRELIMINARIES ON TENSORS

For vectors $\mathbf{a} \in \mathbb{R}^M$, $\mathbf{b} \in \mathbb{R}^N$, and $\mathbf{c} \in \mathbb{R}^T$, the outer product $\mathbf{a} \circ \mathbf{b} \circ \mathbf{c}$ is an $M \times N \times T$ rank-one three-way array with (m, n, t)-th entry given by $\mathbf{a}(m)\mathbf{b}(n)\mathbf{c}(t)$. Note that this comprises a generalization to the two vector (matrix) case, where $\mathbf{a} \circ \mathbf{b} := \mathbf{a}\mathbf{b}^\top$ is a rank-one matrix. The PARAFAC model is arguably the most basic tensor model because of its connection to tensor rank [4]. Based on the previous discussion it is natural to form a *low-rank approximation* of tensor $\underline{\mathbf{X}} \in \mathbb{R}^{M \times N \times T}$ as $\underline{\mathbf{X}} \approx \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r$.

When the decomposition is exact, this offers the PARAFAC decomposition of $\underline{\mathbf{X}}$. Accordingly, the minimum value R for

Supported by the MURI Grant No. AFOSR FA9550-10-1-0567.

978-0-9928626-3-3/15/$31.00 ©2015 IEEE

which the exact decomposition is possible is (by definition) the rank of $\underline{\mathbf{X}}$. PARAFAC is the model of choice when one is primarily interested in revealing latent structure. Considering the analysis of dynamic social networks for instance, each of the rank-one factors could correspond to communities that e.g., persist or form and dissolve periodically across time. Different from the matrix case, there is no straightforward algorithm to determine the rank of a given tensor, a problem that has been shown to be NP-hard [1, 3, 4].

With reference to low-rank decomposition, introduce the factor matrix $\mathbf{A} := [\mathbf{a}_1, \ldots, \mathbf{a}_R] \in \mathbb{R}^{M\times R}$, and likewise for $\mathbf{B} \in \mathbb{R}^{N\times R}$ and $\mathbf{C} \in \mathbb{R}^{T\times R}$. Let $\mathbf{X}_t,\ t \in [T]$ denote the t-th slice of $\underline{\mathbf{X}}$ along its third (time) dimension, such that $\mathbf{X}_t(m,n) = \underline{\mathbf{X}}(m,n,t)$. The following compact matrix form of the PARAFAC decomposition in terms of slice factorizations will be used in the sequel

$$\mathbf{X}_t = \mathbf{A}\mathrm{diag}(\boldsymbol{\gamma}_t)\mathbf{B}^\top = \sum_{r=1}^{R} \boldsymbol{\gamma}_{t,r}\mathbf{a}_r\mathbf{b}_r^\top, \quad t \in [T] \tag{1}$$

where $\boldsymbol{\gamma}_t$ denotes the t-th row of $\mathbf{C}$ (recall that $\mathbf{c}_r$ instead denotes the r-th column of $\mathbf{C}$). It is apparent that each slice $\mathbf{X}_t$ can be represented as a linear combination of R rank-one matrices $\{\mathbf{a}_r\mathbf{b}_r^\top\}_{r=1}^R$ forming the bases for the tensor subspace.

In practice, one may have only access to a small fraction of the possibly noisy entries of $\underline{\mathbf{X}}$, namely $y_t^{(i,j)} = x_t^{(i,j)} + v_t^{(i,j)},\ (i,j) \in \Omega_t,\ t \in [T]$, where the set $\Omega_t \in [M] \times [N]$ indexes the available features at time instant t. Given the data $\{y_t^{(i,j)},\ (i,j) \in \Omega_t\}_{t=1}^T$, tensor imputation aims to impute (or interpolate) the missing features, and possibly denoise the present ones. In essence, feasibility of the imputation task relies fundamentally on assuming a low-rank PARAFAC model for the data, to couple the available and missing entries as well as reduce the effective degrees of freedom in the problem. Generalizing the nuclear-norm regularization technique from low-rank matrices to tensor completion is not however straightforward if one also desires to unveil the latent structure in the data [4].

Interestingly, it was argued in [2] that the Frobenius-norm regularization of the tensor factors, namely

$$h(\mathbf{A},\mathbf{B},\mathbf{C}) := \|\mathbf{A}\|_F^2 + \|\mathbf{B}\|_F^2 + \|\mathbf{C}\|_F^2 \tag{2}$$

offers a viable option for *batch* low-rank tensor completion under the PARAFAC model, by solving

$$\text{(P1)} \quad \min_{\{\mathbf{X},\mathbf{A},\mathbf{B},\mathbf{C}\}} \frac{1}{2}\sum_{t=1}^{T}\sum_{(i,j)\in\Omega_t}\big(y_t^{(i,j)} - x_t^{(i,j)}\big)^2 + \frac{\lambda}{2}\, h(\mathbf{A},\mathbf{B},\mathbf{C})$$
$$\text{s. to } \mathbf{X}_t = \mathbf{A}\mathrm{diag}(\boldsymbol{\gamma}_t)\mathbf{B}^\top, \quad t \in [T].$$

The regularizer (2) encourages low-rank tensor decompositions, with controllable rank by tuning the parameter λ [2]. The present paper primarily focuses on adaptively designing an appropriate subsampling sets $\{\Omega_t\}$ in *real time*. Towards this objective, the first step devises online solvers of (P2) as detailed next.

3. TENSOR SUBSPACE LEARNING

As argued earlier the underlying slice stream $\{\mathbf{X}_t\}$ lives in a low-dimensional subspace. With reference to tensor PARAFAC rank, this low-dimensional subspace is characterized by a small number of R rank-one matrices $\{\mathbf{a}_r\mathbf{b}_r^\top\}_{r=1}^R$, expressed in the compact matrix form $\{\mathbf{A},\mathbf{B}\}$. Learning these *time-invariant* factor matrices comprises an initial stage towards imputing the absent features. In the streaming scenario, at t-th acquisition time with $T = t$ (partial) data snapshots in hand, one is motivated to recast (P1) in the empirical form

$$\text{(P2)} \quad \min_{\{\mathbf{A},\mathbf{B},\mathbf{C}\}} \frac{1}{2t}\sum_{\tau=1}^{t} f_\tau\big(\mathbf{A},\mathbf{B};\boldsymbol{\gamma}_t\big).$$

where the instantaneous cost is defined as

$$f_\tau(\mathbf{A},\mathbf{B};\boldsymbol{\gamma}_\tau) := \sum_{(i,j)\in\Omega_\tau}\Big(y_\tau^{(i,j)} - \sum_{r=1}^{R}\gamma_{\tau,r}a_{i,r}b_{j,r}\rangle\Big)^2 + \frac{\lambda}{2}\|\boldsymbol{\gamma}_\tau\|^2 + \frac{\lambda}{2t}\Big(\|\mathbf{A}\|_F^2 + \|\mathbf{B}\|_F^2\Big)$$

Apparently, finding the optimal solution to the nonconvex program (P2) is a daunting task for infinite data streams ($t \to \infty$). Hence, one needs to devise valid approximations that can afford simple iterative updates while approaching the optimal solution.

3.1. Stochastic alternating minimization

Towards deriving a real-time, computationally efficient, and recursive solver of (P2), an alternating-minimization method is adopted, in which, iterations coincide with the time-scale t of data acquisition. In accordance with (P2), the iterative procedure adopted here consists of two major steps. The first step (S1) relies on recently updated subspace, namely $\{\mathbf{A}[t-1],\mathbf{B}[t-1]\}$ to solve the inner optimization which yields $\hat{\boldsymbol{\gamma}}_t = \arg\min\ f_t(\mathbf{A}[t-1],\mathbf{B}[t-1];\boldsymbol{\gamma}_t)$. The second step (S2) then adjusts the tensor subspace by moving $\{\mathbf{A},\mathbf{B}\}$ along the opposite direction of the gradient, namely $-\nabla f_t(\mathbf{A}[t-1],\mathbf{B}[t-1];\hat{\boldsymbol{\gamma}}_t)$.

Towards (S1), introduce $\boldsymbol{\Phi}_t := [\boldsymbol{\phi}_t^{(1)},\ldots,\boldsymbol{\phi}_t^{(|\Omega_t|)}]^\top \in \mathbb{R}^{|\Omega_t|\times R}$, where $[\boldsymbol{\phi}_t^{(i,j)}]_r := a_{i,r}[t-1]b_{j,r}[t-1]$. The projection of $\mathbf{y}_t$ onto the tensor subspace estimate (so-termed principle components) then forms the ridge regressors

$$\hat{\boldsymbol{\gamma}}_t = \arg\min_{\boldsymbol{\gamma}\in\mathbb{R}^R} \frac{1}{2}\|\mathbf{y}_t - \boldsymbol{\Phi}_t\boldsymbol{\gamma}\|^2 + \frac{\lambda}{2}\|\boldsymbol{\gamma}\|^2 \tag{3}$$

which admits the closed-form $\hat{\boldsymbol{\gamma}}_t = (\boldsymbol{\Phi}_t^\top\boldsymbol{\Phi}_t + \lambda\mathbf{I}_R)^{-1}\boldsymbol{\Phi}_t^\top\mathbf{y}_t$. To avoid $R\times R$ inversion, consider the SVD $\boldsymbol{\Phi}_t = \mathbf{U}_t\boldsymbol{\Sigma}_t\mathbf{V}_t^\top$ to end up with $\hat{\boldsymbol{\gamma}}_t = \mathbf{V}_t\boldsymbol{\Sigma}_t^{-1}\mathbf{D}_t\mathbf{U}_t^\top\mathbf{y}_t$, with the diagonal matrix $\mathbf{D}_t \in \mathbb{R}^{R\times R}$ and $[\mathbf{D}_t]_{i,i} = \sigma_i^2/(\sigma_i^2+\lambda)$.

The second step (S2) primarily aims to refine the subspace factor matrices based on the principal components $\{\hat{\boldsymbol{\gamma}}_\tau\}_{\tau=1}^t$ via $\{\mathbf{A}[t],\mathbf{B}[t]\}=\arg\min_{\{\mathbf{A},\mathbf{B}\}}\ (1/t)\sum_{\tau=1}^t f_t(\mathbf{A},\mathbf{B};\hat{\boldsymbol{\gamma}}_\tau)$. This is a nonconvex program due to the bilinear terms in the LS cost, and thus challenging to solve optimally. To mitigate this hurdle, along the lines of our precursor [7], a proper quadratic approximant of f_t is adopted that is easy to solve optimally. Bypassing the technical details (refer to [7]), the solution admits a close form, that amounts to a stochastic gradient-descent recursion. Interestingly, the gradient of the tensor factors is separable across columns of $\mathbf{A}$ and $\mathbf{B}$, where w.r.t. $\mathbf{a}_r$ it admits

$$\nabla_{\mathbf{a}_r} f_t(\mathbf{A},\mathbf{B};\hat{\boldsymbol{\gamma}})=(\lambda/t)\mathbf{a}_r - \sum_{(i,j)\in\Omega_t}\hat{\gamma}_{t,r}\Big(y_t^{(i,j)}-\sum_{r=1}^R\hat{\gamma}_{t,r}a_{i,r}b_{j,r}\rangle\Big)b_{j,r}\mathbf{e}_i, \quad (4)$$

and likewise for $\nabla_{\mathbf{b}_r} f_t(\mathbf{A},\mathbf{B})$.

All in all, the gradient iterations for learning the tensor subspace proceed in parallel as follows ($r\in[R]$)

$$\mathbf{a}_r[t]=\mathbf{a}_r[t-1]-\mu_t\nabla_{\mathbf{a}_r}f_t(\mathbf{A}[t-1],\mathbf{B}[t-1];\hat{\boldsymbol{\gamma}}_t), \quad (5)$$

where μ_t stands for the step size; likewise for $\nabla_{\mathbf{b}_r} f_t(\mathbf{A},\mathbf{B};\hat{\boldsymbol{\gamma}}_t)$.

4. SUBSPACE LEARNING VIA RANDOMIZED SKETCHING

While the prevailing missing data paradigm pertains to lack of (valid) measurements, one can purposely skip data to either facilitate the acquisition process, or to lower the computational burden of data processing algorithms. The former is well motivated by recent efforts towards accelerating the long MRI scans, creating a lot of artifacts especially for imaging of moving objects. Imagine for instance the MR scanner knowing a priori the best minimal subset of k-space data to collect in each cardiac snapshot. It has then sufficient time to acquire important samples before the heart moves to a new state.

In essence, the optimal sampling trajectory demands knowledge of an unseen physical phenomenon, that is practically infeasible. Typical sampling strategies, e.g., in the context of randomized linear algebra, assume data are *fully acquired* to score and subsequently select a subset of data; see e.g., [8]. However, in certain applications such as MRI, *data acquisition* is the main challenge [10], and data are streaming. All in all, given the subspace estimates offered by the online scheme until time $t-1$, our goal is to adaptively design/predict the subsampling set Ω_t for the next time instant t, with a small size $|\Omega_t|$ while attaining a reasonable estimation accuracy.

4.1. Importance Scores

Albeit streaming data poses an extra challenge to sketching (since future data are not given), online learning offers intermediate estimates of the latent tensor subspace, namely $\{\mathbf{A}[t-1],\mathbf{B}[t-1]\}$, that can be leveraged to devise adaptive sampling strategies. In order to predict Ω_t, our basic idea is to rank the features according to their level of importance measured by a certain score along the lines of [8, 9]. Note again [8, 9] deal with *batch* processing of data.

To understand the idea, it is instructive to first focus on the batch scenario, with the entire tensor data $\underline{\mathbf{Y}}$ at hand, which can be decomposed to end up with the factor matrices $\mathbf{A}$ and $\mathbf{B}$. The (m,n)-th entry of t-th tensor entry can then be expressed as

$$[\mathbf{Y}_t]_{m,n}\approx\sum_{r=1}^R\gamma_{t,r}a_{m,r}[t]b_{n,r}[t],$$

where the principal components are shared by all features in slice t. The distinction between features are due to the weights $\{a_{m,r}\}_{r=1}^R$ and $\{b_{n,r}\}_{r=1}^R$, corresponding two m- and t-th rows of $\mathbf{A}$ and $\mathbf{B}$, respectively. Broadening the scope of [8], a reasonable metric to score the (m,n)-th feature is based on the energy of the corresponding rows in the subspace matrices $\mathbf{A}$ and $\mathbf{B}$.

From the viewpoint of low-rank matrix completion [9], important features are the ones that if dropped, they cannot be reconstructed using the present ones. In essence, the column vectors $\{\mathbf{a}_r\}_{r=1}^R$ (resp. $\{\mathbf{b}_r\}_{r=1}^R$) of $\mathbf{A}$ (resp. $\mathbf{B}$) span the column- (resp. row-) space of the tensor slices. Albeit not necessarily orthonormal, the factor matrices $\mathbf{A}$ and $\mathbf{B}$ play similar role as the orthonormal factors $\mathbf{U}$ and $\mathbf{V}$ forming the SVD $\mathbf{Y}_t=\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top$. For row and column spaces, introduce the so termed *incoherence* measures $\mu_i:=\|\mathbf{U}^\top\mathbf{e}_i\|^2$ and $\nu_j:=\|\mathbf{V}^\top\mathbf{e}_j\|^2$, denoting projection of the canonical basis $\mathbf{e}_i$ onto the column and row spaces, respectively. It is well understood from the matrix completion context [9] that (i,j)-th entry with high degree of coherence (μ_i,ν_j) are more susceptible to misidentification if absent. This is simply because the corresponding row and column are more *aligned* (larger projection) with the column and row space of the underlying matrix, and hence they can be wrongly estimated while achieving a small rank.

Along this line of thought, consider now the online setup where at time instant t one has access to the subspace estimate $(\mathbf{A}[t-1],\mathbf{B}[t-1])$, and aims to acquire a few features of the next slice $\mathbf{Y}_t$ indexed by Ω_t. Suppose that the slices $\{\mathbf{Y}_t\}$ change slowly over time; this is the case for instance in dynamic cardiac MRI where different slices correspond to different snapshots of a patient's beating heart. Assume also the subspace matrices $(\mathbf{A}[t-1],\mathbf{B}[t-1])$ updated at time $t-1$ provide a *decent* estimate of the underlying tensor subspace e.g., as a result of a warm initialization. It is then reasonable to adopt the metric $\|\mathbf{A}^\top[t-1]\mathbf{e}_m\|_F^2+\|\mathbf{B}^\top[t-1]\mathbf{e}_n\|_F^2$ as the importance score for the (m,n)-th feature at time t.

Apparently if $\mathbf{Y}_t$ contains innovations, not captured by the subspace $(\mathbf{A}[t-1],\mathbf{B}[t-1])$ (learned from past data $\{\mathbf{Y}_\tau\}_{\tau=1}^{t-1}$), the important features may be misidentified. To

cope with this issue, and to avoid any sampling bias due to the initialization, the sketching is randomized as discussed next.

4.2. Randomized sketching

Normalize columns of the factor matrices $\mathbf{A}[t-1]$ and $\mathbf{B}[t-1]$ to end up with $\tilde{\mathbf{A}}[t-1]$ and $\tilde{\mathbf{B}}[t-1]$, respectively. For notational brevity also introduce $\tilde{\mathbf{A}}[t-1] := [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_M]^\top$, and $\tilde{\mathbf{B}}[t-1] := [\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_N]^\top$. Then, for the feature $(m,n) \in [M] \times [N]$ at time t, associate the score

$$\rho_t(m,n) := \frac{1}{R(M+N)} \left(\|\boldsymbol{\alpha}_m\|^2 + \|\boldsymbol{\beta}_n\|^2 \right) \tag{6}$$

The scores $\{\rho_t(m,n)\}$ are positive-valued and sum up to unity, and thus one can interpret them as a probability distribution over the entries. For a prescribed maximum sample count K, one can then draw K random trials from the distribution ρ_t to collect the important features in the set Ω_t. It is more natural to consider random trials drawn *with* replacement so as the resulting sample count $|\Omega_t| \leq K$. This is particularly effective when features are not equally important since it skips naive features with tiny scores.

Adopting randomized sketching along with the online iterates for subspace learning in (5) and (3), the resulting procedure is listed under Algorithm 2. The algorithm begins with a warm initialization, obtained for instance after first running the algorithm over a small training dataset. The resultant algorithm at iteration (time instant) t comprises three major steps, where the first step (S1) *probabilistically* decides on the sampling set Ω_t, which is subsequently used to acquire the corresponding features in $\mathbf{Y}_t$. Based on the partial features in Ω_t, the second step finds the principal components of t-th frame across the subspace bases $\{\mathbf{a}_r[t-1]\mathbf{b}_r^\top[t-1]\}_{r=1}^R$. The innovation of the new (imputed) datum captured through the error term $\{e_t^{(i,j)}\}_{(i,j)\in\Omega_t}$, in the next step (S3) then refines the subspace bases.

An important question in this context pertains to the (average) number of samples acquired per slice by the randomized sketcher. This essentially depends on the nonuniformity of the features and initialization of the algorithm. In an extreme case with equally important features, exactly K samples are acquired per time, which can be significantly lowered for nonuniform features. To see this, introduce a random variable $X_{m,n}$ denoting the frequency of choosing (m,n)-th entry after K trials. The random sample count is then $|\Omega_t| = \sum_{m,n} I_{\{X_{m,n}\}}$, where the indicator $I_{\{x\}}$ takes value of one if $x > 0$, and zero otherwise. In general, the random variables $I_{\{X_{m,n}\}}$ are dependent, which renders the pdf analysis for $|\Omega_t|$ formidable. The expected sample count per each slice however can be expressed as

$$\mathbb{E}[|\Omega_t|] = \sum_{m=1}^{M} \sum_{n=1}^{N} \left(1 - [1 - \rho_t(m,n)]^K \right)$$

Algorithm 1 Random subsampling

input $\mathbf{A}[t-1]$, $\mathbf{B}[t-1]$, K, and R

$\tilde{\mathbf{A}}[t-1] := \mathbf{A}[t-1]\text{diag}\big(\|\mathbf{a}_1[t-1]\|, \ldots, \|\mathbf{a}_R[t-1]\|\big)^{-1}$

$\tilde{\mathbf{B}}[t-1] := \mathbf{B}[t-1]\text{diag}\big(\|\mathbf{b}_1[t-1]\|, \ldots, \|\mathbf{b}_R[t-1]\|\big)^{-1}$

$\tilde{\mathbf{A}}[t-1] := [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_M]^\top$, $\tilde{\mathbf{B}}[t-1] := [\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_N]^\top$

$\rho_t(m,n) := \frac{1}{R(M+N)} \left(\|\boldsymbol{\alpha}_m\|^2 + \|\boldsymbol{\beta}_n\|^2 \right)$

Draw K random trials from $[M] \times [N]$ based on ρ_t to form Ω_t

output Ω_t

Algorithm 2 Randomized tensor subspace learning

input $\{\mu_t\}_{t=1}^\infty, K, R, \lambda$

initialize $\{\mathbf{A}[0], \mathbf{B}[0]\}$ with a warm startup

for $t = 1, \ldots$ **do**

(S1) Random subsampling

Acquire $\{y_t^{(m,n)}\}_{(m,n)\in\Omega_t}$ based on Algorithm 1

(S2) Principal-components update

$[\boldsymbol{\Phi}_t]_{(m,n),r} = a_{m,r}[t-1]b_{n,r}[t-1]$, $\boldsymbol{\Phi}_t = \mathbf{U}_t\boldsymbol{\Sigma}_t\mathbf{V}_t^\top$

$\mathbf{D}_t = \text{diag}\big[\sigma_1(\sigma_1^2+\lambda)^{-1}, \ldots, \sigma_R(\sigma_R^2+\lambda)^{-1}\big]$,

$\hat{\boldsymbol{\gamma}}_t = \mathbf{V}_t\boldsymbol{\Sigma}_t^{-1}\mathbf{D}_t\mathbf{U}_t^\top\mathbf{y}_t$

$e_t^{(m,n)} := y_t^{(m,n)} - \langle \boldsymbol{\phi}_t^{(m,n)}, \boldsymbol{\gamma}_t \rangle$

(S3) Parallel subspace update $[r \in [R]]$

$\mathbf{a}_r[t] = (1-\mu_t\lambda/t)\mathbf{a}_r[t-1]$
$\quad + \mu_t\hat{\gamma}_{t,r}\sum_{(m,n)\in\Omega_t} e_t^{(m,n)} b_{n,r}[t-1]\mathbf{e}_m$

$\mathbf{b}_r[t] = (1-\mu_t\lambda/t)\mathbf{b}_r[t-1]$
$\quad + \mu_t\hat{\gamma}_{t,r}\sum_{(m,n)\in\Omega_t} e_t^{(m,n)} a_{m,r}[t-1]\mathbf{e}_n$

end for

output $\{\mathbf{A}[t], \mathbf{B}[t]\}$

Finally, the average sample count across both time and entries is given as $\bar{N}_t := (1/t)\sum_{\tau=1}^t \mathbb{E}[|\Omega_\tau|]$

Remark (Convergence). Regarding the iterates in Algorithm 2, our precursor [7] establishes asymptotic convergence when the sampling sets $\{\Omega_t\}$ are independent of the data $\{\mathbf{Y}_t\}$, and form an i.i.d. sequence over time. The considered adaptive acquisition strategy however is data driven and temporally correlated, rendering the convergence analysis intricate. In this context, an intriguing question pertains to role of sampling trajectory towards convergence of the subspace sequence $\{(\mathbf{A}[t], \mathbf{B}[t])\}$. One may also ponder about the limit point of the probability sequence ρ_t. These studies go beyond the scope of the present paper, and will be reported in the journal version of this work.

5. SIMULATED TESTS AND CONCLUSIONS

Convergence and effectiveness of the novel adaptive sketching scheme is assessed in this section via computer simulations. Tensor slices are generated according to trilinear model

$$\mathbf{Y}_t = \mathbf{A}\text{diag}(\boldsymbol{\gamma}_t)\mathbf{B} + \mathbf{V}_t, \;\; t = 1, 2, \ldots$$

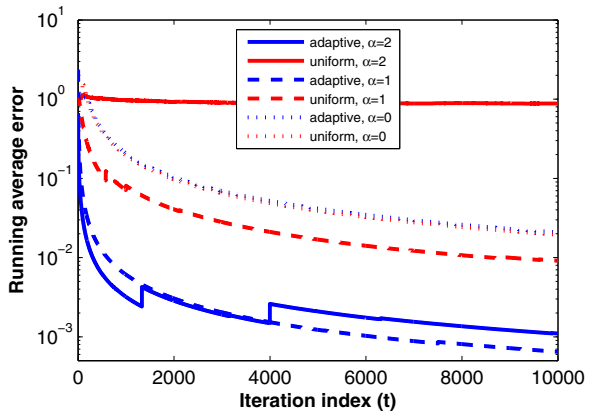

Fig. 1. Running average error under various cohernece levels α for adaptive and uniform subsampling. The average sampling rate for $\alpha = 0, 1, 2$ is $\pi = 0.08, 0.13, 0.16$, respectively.

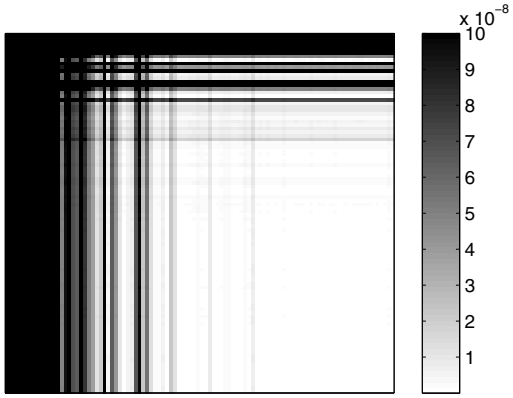

Fig. 2. Converged probability distribution ρ_t for $\alpha = 2$ with the average sampling probability $\pi = 0.08$.

with Gaussian factors $\mathbf{A} = \mathbf{\Lambda}_{\alpha}^{M}\mathbf{G}\mathbf{\Lambda}_{-\alpha}^{R}$ and $\mathbf{B} = \mathbf{\Lambda}_{\alpha}^{N}\mathbf{G}\mathbf{\Lambda}_{-\alpha}^{R}$, constructed by the standard Gaussian matrix $[\mathbf{G}]_{i,j} \sim \mathcal{N}(0,1)$. The diagonal matrices $\mathbf{\Lambda}_{\alpha}^{M} := \text{diag}(1, 2^{-\alpha}, \ldots, M^{-\alpha})$ (likewise $\mathbf{\Lambda}_{\alpha}^{N}, \mathbf{\Lambda}_{-\alpha}^{R}$) are introduced to make the tensor entries statistically nonuniform. Note $\alpha \geq 0$ controls the incoherence for the column and row subspaces, where small value α induces an incoherent subspace with uniform features. Similarly, the principal components obey $\gamma_{t,r} \sim \mathcal{N}(0,1)$, and the noise term $v_t^{(m,n)} \sim \mathcal{N}(0, \sigma^2)$. As a warm startup, we initialize Algorithm 2 with $\mathbf{A}[0] = \mathbf{\Lambda}_{\alpha}^{M}\mathbf{G}\mathbf{\Lambda}_{-\alpha}^{R}$ and $\mathbf{B}[0] = \mathbf{\Lambda}_{\alpha}^{N}\mathbf{G}\mathbf{\Lambda}_{-\alpha}^{R}$, for some independent realization $\mathbf{G}$. Throughout the tests we adopt $M = N = 10^2$, $\rho = R = 5$, $\sigma = 10^{-4}$, and $\lambda = 10^{-2}$. To simulate the uniform sampling in the sequel, an entry (m, n) is assigned to the support set Ω_t with probability (w.p.) π, and discarded w.p. $1 - \pi$.

Fig. 1 plots the evolution of the running average error $e_t := \frac{1}{t}\sum_{\tau=1}^{t} \|\mathbf{X}_\tau - \hat{\mathbf{X}}_\tau\|_F^2 / \|\mathbf{X}_\tau\|_F^2$, with $\hat{\mathbf{X}}_\tau = \mathbf{A}[\tau]\text{diag}(\boldsymbol{\gamma}_\tau)\mathbf{B}[\tau]$ denoting the τ-th interpolated slice. The accuracy of our adaptive sketching scheme is compared against uniform sampling strategy under various subspace incoherence levels α. For $\alpha = 0$, both uniform and adaptive sampling strategies perform somewhat equally well. As α grows, adaptive sampling becomes crucial and one observes that for $\alpha = 2$ uniform sampling learns almost nothing over time ($e_t \approx 1$), while adaptive sampling performs quite well ($e_t \approx 10^{-3}$). The resulting average sampling rates for $\alpha = 0, 1, 2$ are respectively $\pi = 0.16, 0.13, 0.08$. The converged sampling probability distribution of features, namely ρ_t is depicted in Fig. 2 when $\alpha = 2$. The sparse probability map of Fig. 2 emanates form the fact that only a small fraction (8%) of features, associated with the upper left corner, are important, and the rest are almost naive features.

REFERENCES

[1] E. Acar, D. M. Dunlavy, T. G. Kolda, and M. Mrup, "Scalable tensor factorizations for incomplete data," *Chemo. Intelligence Lab. Sys.,* vol. 106, pp. 41-56, 2011.

[2] J. A. Bazerque, G. Mateos, and G. B. Giannakis, "Rank regularization and Bayesian inference for tensor completion and extrapolation," *IEEE Trans. Signal Process.*, vol. 61, pp. 5689–5703, Nov. 2013.

[3] S. Gandy, B. Recht, and I. Yamada, "Tensor completion and low-*n*-rank tensor recovery via convex optimization," *Inverse Problems,* vol. 27, pp. 1-19, 2011.

[4] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455-500, Sep. 2009.

[5] J. Kruskal, "Three-way arrays: Rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics," *Lin. Alg. Applicat.*, vol. 18, no. 2, pp. 95-138, 1977.

[6] M. Mardani, G. Mateos, and G. B. Giannakis, "Dynamic anomalography: Tracking network anomalies via sparsity and low-rank," *IEEE J. Sel. Tpoics in Signal Process.*, vol. 7, pp. 50-66, Feb. 2013.

[7] M. Mardani, G. Mateos, and G. B. Giannakis, "Streaming algorithms for imputation of Big Data matrices and tensors," *IEEE Trans. Signal Process.*, 2015.

[8] M. W. Mahoney, "Randomized algorithms for matrices and data," *Foundations and Trends® in Machine Learning*, vol. 3, no. 2, pp. 123–224, 2011.

[9] Y. Chen, S. Bhojanapalli, S. Sanghavi, and R. Ward, "Coherent matrix completion," in *Proc. Intl. Conf. on Machine Learning*, Beijing, China, Jun. 2014.

[10] Lauterbur, "Image formation by induced local interactions: Examples employing nuclear magnetic resonance," *Nature*, vol. 242, pp. 190–191, Mar. 1973.