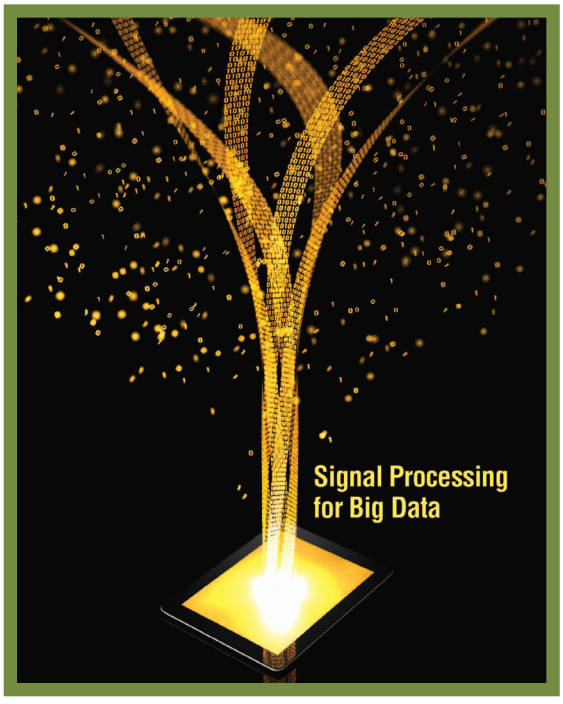
Konstantinos Slavakis, Georgios B. Giannakis, and Gonzalo Mateos

Modeling and Optimization for Big Data Analytics



(Statistical) learning tools for our era of data deluge

With pervasive sensors continuously collecting and storing massive amounts of information, there is no doubt this is an era of data deluge. Learning from these large volumes of data is expected to bring significant science and engineering advances along with improvements in quality of life. However, with such a big blessing come big challenges. Running analytics on voluminous data sets by central processors and storage units seems infeasible, and with the advent of streaming data sources, learning must often be performed in real time, typically without a chance to revisit past entries. "Workhorse" signal processing (SP) and statistical learning tools have to be re-examined in today's high-dimensional data regimes. This article contributes to the ongoing cross-disciplinary efforts in data science by putting forth encompassing models capturing a wide range of SP-relevant data analytic tasks, such as principal component analysis (PCA), dictionary learning (DL), compressive sampling (CS), and subspace clustering. It offers scalable architectures and optimization algorithms for decentralized and online learning problems, while revealing fundamental insights into the various analytic and implementation tradeoffs involved. Extensions of the encompassing models to timely data-sketching, tensor- and kernel-based learning tasks are also provided. Finally,

Digital Object Identifier 10.1109/MSP.2014.2327238
Date of publication: 19 August 2014

1053-5888/14©2014IEEE

the close connections of the presented framework with several big data tasks, such as network visualization, decentralized and dynamic estimation, prediction, and imputation of network link load traffic, as well as imputation in tensor-based medical imaging are highlighted.

INTRODUCTION

The information explosion propelled by the advent of online social media, Internet, and global-scale communications has rendered data-driven statistical learning increasingly important. At any time around the globe, large volumes of data are generated by today's ubiquitous communication, imaging, and mobile devices such as cell phones, surveillance cameras and drones, medical and e-commerce platforms, as well as social networking sites. The term *big data* is coined to describe this information deluge and, quoting a recent press article, "their effect is being felt everywhere, from business to science, and from government to the arts" [18]. Large economic growth and improvement in the quality of life hinge upon harnessing the potential benefits of analyzing massive data [18], [55]. Mining unprecedented volumes of data promises to limit the spread of epidemics and maximize the odds that online marketing campaigns go viral [35]; to identify trends in financial markets, visualize networks, understand the dynamics of emergent social-computational systems, as well as protect critical infrastructure including the Internet's backbone network [48], and the power grid [26].

BIG DATA CHALLENGES AND SP OPPORTUNITIES

While big data come with "big blessings," there are formidable challenges in dealing with large-scale data sets. First, the sheer volume and dimensionality of data make it often impossible to run analytics and traditional inferential methods using standalone processors, e.g., [8] and [31]. Decentralized learning with parallelized multicores is preferred [9], [22], while the data themselves are stored in the cloud or distributed file systems as in MapReduce/Hadoop [19]. Thus, there is an urgent need to explicitly account for the storage, query, and communication burden. In some cases, privacy concerns prevent disclosing the full data set, allowing only preprocessed data to be communicated through carefully designed interfaces. Due to their possibly disparate origins, big data sets are often incomplete and a sizable portion of them is missing. Large-scale data inevitably contain corrupted measurements, communication errors, and even suffer from cyberattacks as the acquisition and transportation cost per entry is driven to the minimum. Furthermore, as many of the data sources continuously generate data in real time, analytics must often be performed online subject to time constraints so that a high-quality answer obtained slowly can be less useful than a medium-quality answer that is obtained quickly [46], [48], [75].

Although past research on databases and information retrieval is viewed as having focused on storage, look-up, and search, the opportunity now is to comb through massive data sets, to discover new phenomena, and to "learn" [31]. Big data challenges offer ample opportunities for SP research [55], where data-driven statistical learning algorithms are envisioned to facilitate distributed and real-time analytics (cf. Figure 1). Both classical and modern SP techniques have already placed significant emphasis on time/data adaptivity, e.g., [69], robustness [32], as well as compression and dimensionality reduction [43]. Testament to this fact is the recent "rediscovery" of stochastic approximation and stochastic-gradient algorithms for scalable online convex optimization and learning [65], oftentimes neglecting Robbins–Monro and Widrow's seminal works that go back half a century [60], [69], [79]. While the principal role of computer science in big data research is undeniable, the nature and scope of the emerging data science field is certainly multidisciplinary and welcomes SP expertise and its recent advances. For example, Web-collected data are often replete with missing entries, which motivates innovative SP imputation techniques that leverage timely (low-rank) matrix decompositions [39], [52], or, suitable kernel-based interpolators [6]. Data matrices gathering traffic values observed in the backbone of large-scale networks can be modeled as the superposition of unknown "clean" traffic, which is usually low-rank due to temporal periodicities as well as network topology-induced correlations, and traffic volume anomalies that occur sporadically in time and space, rendering the associated matrix component sparse across rows and columns [38]. Both quantity and richness of high-dimensional data sets offer the potential to improve statistical learning performance, requiring however innovative models that exploit latent low-dimensional structure to effectively separate the data "wheat from the chaff." To learn these models however, there is a consequent need to advance online, scalable optimization algorithms for information processing over graphs (an abstraction of both networked sources of decentralized data, and multiprocessor, high-performance computing architectures); see, e.g., GraphLab [42] and the alternating direction method of multipliers (ADMM) [9], [10], [51] that enjoy growing popularity for distributed machine learning tasks.

ENCOMPASSING MODELS FOR SUCCINCT BIG DATA REPRESENTATIONS

This section introduces a versatile model to fit data matrices as a superposition of a low-rank matrix capturing correlations and periodic trends, plus a linearly compressed sparse matrix explaining data innovations parsimoniously through a set of (possibly latent) factors. The model is rich enough to subsume various statistical learning paradigms with well-documented merits for high-dimensional data analysis, including PCA [28], DL [56], compressive sampling CS [11], and principal components pursuit (PCP) [12], [14], [52], to name a few.

THE "BACKGROUND" PLUS "PATTERNS AND INNOVATIONS" MODEL FOR MATRIX DATA

Let $\mathbf{L} \in \mathbb{R}^{N\times T}$ denote a low-rank matrix ($\text{rank}(\mathbf{L}) \ll \min\{N, T\}$), and $\mathbf{S} \in \mathbb{R}^{M\times T}$ a sparse matrix with support size considerably smaller than MT. Consider also the large-scale data set $\mathbf{Y} \in \mathbb{R}^{N\times T}$ generically modeled as a superposition of 1) the low-rank matrix $\mathbf{L}$; the "data background or trend," e.g., nominal

[FIG1] **SP-relevant big data themes.**

load curves across the power grid or the background scene captured by a surveillance camera, plus, 2) the "data patterns, (co) clusters, innovations, or outliers" expressed by the product of a (possibly unknown) dictionary $\mathbf{D} \in \mathbb{R}^{N\times M}$ times the sparse matrix $\mathbf{S}$, and 3) a matrix $\mathbf{V} \in \mathbb{R}^{N\times T}$, which accounts for modeling and measurement errors; in short, $\mathbf{Y} = \mathbf{L} + \mathbf{DS} + \mathbf{V}$. Matrix $\mathbf{D}$ could be an overcomplete set of bases or a linear compression operator with $N \leq M$. The aforementioned model offers a parsimonious description of $\mathbf{Y}$, that is welcomed in big data analytics where data sets involve numerous features. Such parsimony facilitates interpretability, model identifiability, and it enhances the model's predictive performance by discarding "noisy" features that bear little relevance to the phenomenon of interest [49].

To explicitly account for missing data in $\mathbf{Y}$ introduce 1) the set $\Omega \subseteq \{1,\ldots,N\}\times\{1,\ldots,T\}$ of index pairs (n,t), and 2) the sampling operator $\mathcal{P}_\Omega(\cdot)$, which nulls entries of its matrix argument not in Ω, leaving the rest unchanged. This way, one can express incomplete and (possibly noise-)corrupted data as

$$\mathcal{P}_\Omega(\mathbf{Y}) = \mathcal{P}_\Omega(\mathbf{L} + \mathbf{DS} + \mathbf{V}). \tag{1}$$

Given $\mathcal{P}_\Omega(\mathbf{Y})$, the challenging goal is to estimate the matrix components $\mathbf{L}$ and $\mathbf{S}$ (and $\mathbf{D}$ if not given), which further entails denoising the observed entries and imputing the missing ones.

An estimator leveraging the low-rank property of $\mathbf{L}$ and the sparsity of $\mathbf{S}$ will be sought to fit the data $\mathcal{P}_\Omega(\mathbf{Y})$ in the least-squares (LS) error sense, as well as minimize the rank of $\mathbf{L}$, and the number of nonzero entries of $\mathbf{S} := [s_{m,t}]$ measured by its ℓ_0-(pseudo) norm. Unfortunately, albeit natural both rank and ℓ_0-norm criteria are in general NP-hard to optimize [53]. With $\sigma_k(\mathbf{L})$ denoting the kth singular value of $\mathbf{L}$, the nuclear norm $\|\mathbf{L}\|_* := \sum_k \sigma_k(\mathbf{L})$, and the ℓ_1-norm $\|\mathbf{S}\|_1 := \sum_{m,t}|s_{m,t}|$ are adopted as surrogates, as they are the closest convex approximants to $\mathbf{rank}(\mathbf{L})$ and $\|\mathbf{S}\|_0$, respectively, e.g., [14] and [48]. Accordingly, assuming known $\mathbf{D}$ for now, one solves

$$\min_{\{\mathbf{L},\mathbf{S}\}} \frac{1}{2}\|\mathcal{P}_\Omega(\mathbf{Y}-\mathbf{L}-\mathbf{DS})\|_{\mathrm{F}}^2 + \lambda_*\|\mathbf{L}\|_* + \lambda_1\|\mathbf{S}\|_1, \tag{P1}$$

where $\lambda_*, \lambda_1 \geq 0$ are rank- and sparsity-controlling parameters. Being convex, (P1) is computationally appealing as elaborated in the section "Algorithms," in addition to being widely applicable as it encompasses a gamut of known paradigms. Notice however that when $\mathbf{D}$ is unknown, one obtains a bilinear model that gives rise to nonconvex estimation criteria. The approaches highlighted next can in fact accommodate more general models than (P1), where data-fitting terms other than the Frobenius-norm one and different regularizers can be utilized to account for various types of a priori knowledge, e.g., structured sparsity or smoothness.

APPLICATION DOMAINS AND SUBSUMED PARADIGMS

Model (1) emerges in various applications, such as 1) network anomaly detection outlined in the section "Inference and Imputation," where $\mathbf{Y} \in \mathbb{R}^{N\times T}$ represents traffic volume over N links and T time slots; $\mathbf{L}$ captures the nominal link-level traffic (which

is low-rank due to temporal periodicities and topology-induced correlations on the underlying flows); $\mathbf{D}$ represents a link $\times$ flow binary routing matrix; and $\mathbf{S}$ sparse anomalous flows [47], [48]; 2) medical imaging, where dynamic magnetic resonance imaging separates the background $\mathbf{L}$ from the motion component (e.g., a heart beating) modeled via sparse dictionary representation $\mathbf{DS}$ [25] (see also the section "Inference and Imputation"); 3) face recognition in the presence of shadows and specularities [12]; and 4) acoustic SP for singing voice separation from its music accompaniment [71], to name a few.

In the absence of $\mathbf{L}$ and missing data ($\mathbf{L}=\mathbf{0}$, $\Omega=\{1,\ldots,N\}\times\{1,\ldots,T\}$), model (1) describes an underdetermined sparse signal recovery problem typically encountered with CS [11]. If in addition $\mathbf{D}$ is unknown, (P1) boils down to DL [2], [46], [56], [67], or, to nonnegative matrix factorization (NNMF) if the entries of $\mathbf{D}$ and $\mathbf{S}$ are nonnegative [39]. For $\mathbf{L}=\mathbf{0}$, $\Omega=\{1,\ldots,N\}\times\{1,\ldots,T\}$, and if the columns of $\mathbf{Y}$ lie close to a union of a small number of unknown low-dimensional linear subspaces, then looking for a sparse $\mathbf{S}$ in (1) with $M\ll T$ amounts to subspace clustering [78]; see also [70] for outlier-robust variants with strong performance guarantees. Without $\mathbf{D}$ and with $\mathbf{V}=\mathbf{0}$, decomposing $\mathbf{Y}$ into $\mathbf{L}+\mathbf{S}$ corresponds to PCP, also referred to as *robust PCA* (*R-PCA*) [12], [14]. Even when $\mathbf{L}$ is nonzero, one could envision a variant where the measurements are corrupted with correlated (low-rank) noise [15]. Last but not least, when $\mathbf{S}=\mathbf{0}$ and $\mathbf{V}\neq\mathbf{0}$, recovery of $\mathbf{L}$ subject to a rank constraint is nothing else than PCA—arguably, the workhorse of high-dimensional big data analytics [28]. This same formulation is adopted for low-rank matrix completion—the basic task carried out by recommender systems—to impute the missing entries of a low-rank matrix observed in noise, i.e., $\mathcal{P}_\Omega(\mathbf{Y})=\mathcal{P}_\Omega(\mathbf{L}+\mathbf{V})$ [13]. Based on the maximum likelihood principle, an alternative approach for missing value imputation by expectation-maximization can be found in [73].

ALGORITHMS

As (P1) is jointly convex with respect to (w.r.t.) both $\mathbf{L}$ and $\mathbf{S}$, various iterative solvers are available, including interior point methods and centralized online schemes based on (sub)gradient-based recursions [65]. For big data however, off-the-shelf interior point methods are computationally prohibitive, and are not amenable to decentralized or parallel implementations. Subgradient-based methods are structurally simple but are often hindered by slow convergence due to restrictive step size selection rules. The desiderata for large-scale problems are low-complexity, real-time algorithms capable of processing massive data sets in a parallelizable and/or fully decentralized fashion. The few such algorithms available can be classified as decentralized or parallel schemes, splitting, sequential, and online or streaming.

DECENTRALIZED AND PARALLEL ALGORITHMS

In these divide-and-conquer schemes, multiple agents operate in parallel on disjoint or randomly subsampled subsets of the massive-scale data, and combine their outputs as iterations proceed to accomplish the original learning or inference task [34], [44]. Unfortunately, the nuclear-norm $\|\mathbf{L}\|_*$ in (P1) cannot be easily distributed across multiple learners, since the full singular value decomposition (SVD) of $\mathbf{L}$ has to be computed centrally, prior distributing its set of singular values to each node. In search of a nuclear-norm surrogate amenable to decentralized processing, it is useful to recall that minimizing $\|\mathbf{L}\|_*$ is tantamount to minimizing $(\|\mathbf{P}\|_{\mathrm{F}}^2+\|\mathbf{Q}\|_{\mathrm{F}}^2)/2$, where $\mathbf{L}=\mathbf{P}\mathbf{Q}^\top$, with $\mathbf{P}\in\mathbb{R}^{N\times\rho}$ and $\mathbf{Q}\in\mathbb{R}^{T\times\rho}$, for some $\rho\ll\min\{N,T\}$, is a bilinear decomposition of the low-rank component $\mathbf{L}$ [47], [72]. In other words, each column vector of $\mathbf{L}$ is assumed to lie in a low ρ-dimensional range space spanned by the columns of $\mathbf{P}$. This gives rise to the following problem:

$$\min_{\{\mathbf{P},\mathbf{Q},\mathbf{S}\}}\frac{1}{2}\|\mathcal{P}_\Omega(\mathbf{Y}-\mathbf{P}\mathbf{Q}^\top-\mathbf{D}\mathbf{S})\|_{\mathrm{F}}^2+\frac{\lambda_*}{2}(\|\mathbf{P}\|_{\mathrm{F}}^2+\|\mathbf{Q}\|_{\mathrm{F}}^2)+\lambda_1\|\mathbf{S}\|_1. \tag{P2}$$

Unlike (P1), the bilinear term $\mathbf{P}\mathbf{Q}^\top$ renders (P2) nonconvex, even if $\mathbf{D}$ is known. Interestingly, [47, Prop. 1] offers a certificate for stationary points of (P2), qualifying them as global optima of (P1).

Thanks to the decomposability of $\|\cdot\|_{\mathrm{F}}^2$ and $\|\cdot\|_1$ across rows, and ignoring for a moment the operator $\mathcal{P}_\Omega$, (P2) can be distributed over a number $\mathcal{V}$ of nodes or processing cores $\mathscr{V}$ with cardinality $|\mathscr{V}|=\mathcal{V}$, where each node $\nu\in\mathscr{V}$ learns from a subset of rows $\mathcal{R}_\nu\subset\{1,\ldots,N\}$. In other words, the N rows of $\mathbf{Y}$ are distributed over a partition of rows $\{\mathcal{R}_\nu\}_{\nu=1}^{\mathcal{V}}$, where by definition $\bigcup_{\nu=1}^{\mathcal{V}}\mathcal{R}_\nu=\{1,\ldots,N\}$, and $\mathcal{R}_{\nu_i}\cap\mathcal{R}_{\nu_j}=\emptyset$, if $i\neq j$. Naturally, (P2) is equivalent to this (modulo $\mathcal{P}_\Omega$) task:

$$\min_{\{\{\mathbf{P}_\nu\}_{\nu=1}^{\mathcal{V}},\mathbf{Q},\mathbf{S}\}}\frac{1}{2}\sum_{\nu=1}^{\mathcal{V}}\|\mathbf{Y}_\nu-\mathbf{P}_\nu\mathbf{Q}^\top-\mathbf{D}_\nu\mathbf{S}\|_{\mathrm{F}}^2+\frac{\lambda_*}{2}\sum_{\nu=1}^{\mathcal{V}}\|\mathbf{P}_\nu\|_{\mathrm{F}}^2+\frac{\lambda_*}{2}\|\mathbf{Q}\|_{\mathrm{F}}^2+\lambda_1\|\mathbf{S}\|_1, \tag{2}$$

where $\mathbf{Y}_\nu$, $\mathbf{P}_\nu$, and $\mathbf{D}_\nu$ are submatrices formed by keeping only the $\mathcal{R}_\nu$ rows of $\mathbf{Y}$, $\mathbf{P}$, and $\mathbf{D}$, respectively.

An obstacle in (2) is the coupling of the data-fitting term with the regularization terms via $\{\mathbf{P}_\nu,\mathbf{Q},\mathbf{S}\}$. Direct utilization of iterative subgradient-type methods, due to the nonsmooth loss function, are able to identify local minimizers of (2), at the cost of slow convergence and meticulous choice of step sizes. In the convex analysis setting, successful optimization approaches to surmount this obstacle include the ADMM [10] and the more general Douglas–Rachford (DR) algorithm [5] that split or decouple variables in the nuclear-, ℓ_1-, and Frobenius-norms. The crux of splitting methods, such as ADMM and DR, lies on computing efficiently the proximal mapping of regularizing functions, which for a (non)differentiable lower-semicontinuous convex function g and $\gamma>0$, is defined as $\mathrm{Prox}_{\gamma g}(\mathbf{A}):=\arg\min_{\mathbf{A}'}(1/2)\|\mathbf{A}-\mathbf{A}'\|_{\mathrm{F}}^2+\gamma g(\mathbf{A}')$, $\forall\mathbf{A}$ [5]. The computational cost incurred by $\mathrm{Prox}_{\gamma g}$ depends on g. For example, if g is the nuclear-norm, then $\mathrm{Prox}_{\gamma\|\cdot\|_*}(\mathbf{A})=\mathbf{U}\,\mathrm{Soft}_\gamma(\Sigma)\mathbf{V}^\top$, where $\mathbf{A}=\mathbf{U}\Sigma\mathbf{V}^\top$ is the computationally demanding SVD of $\mathbf{A}$, and $\mathrm{Soft}_\gamma(\Sigma)$ is the soft-thresholding operator whose (i,j)th

entry is $[\text{Soft}_\gamma(\Sigma)]_{ij}=\text{sgn}([\Sigma]_{ij})\max\{0,|[\Sigma]_{ij}|-\gamma\}$. On the contrary, if $g=\|\cdot\|_1$, then $\text{Prox}_{\gamma\|\cdot\|_1}(\mathbf{A})=\text{Soft}_\gamma(\mathbf{A})$, which is a computationally affordable, parallelizable operation.

Even if (2) is a nonconvex task, a splitting strategy mimicking ADMM and DR is promising also in the current context. If the network nodes or cores can also exchange messages, then (2) can be decentralized. This is possible if e.g., $\nu\in\mathscr{V}$ has a neighborhood $\mathscr{N}_\nu\subset\mathscr{V}$, where $\nu\in\mathscr{N}_\nu$ and all members of $\mathscr{N}_\nu$ exchange information. The decentralized rendition of (P2) becomes

$$\min_{\substack{\{\mathbf{P}_\nu,\mathbf{Q}_\nu,\mathbf{S}_\nu\}\\ \{\mathbf{P}'_\nu,\mathbf{Q}'_\nu,\mathbf{S}'_\nu\}}} \frac{1}{2}\|\mathscr{P}_{\Omega_\nu}(\mathbf{Y}_\nu-\mathbf{P}_\nu\mathbf{Q}_\nu^\top-\mathbf{D}_\nu\mathbf{S}_\nu)\|_{\text{F}}^2 + \frac{\lambda_*}{2}(\|\mathbf{P}'_\nu\|_{\text{F}}^2+\|\mathbf{Q}'_\nu\|_{\text{F}}^2)+\lambda_1\|\mathbf{S}'_\nu\|_1,\ \forall\nu\in\mathscr{V}$$
$$\text{s.to}\quad \forall\nu\in\mathscr{V}:\begin{cases}\forall\nu'\in\mathscr{N}_\nu:\begin{cases}\mathbf{Q}_\nu=\mathbf{Q}_{\nu'},\ \ \mathbf{S}_\nu=\mathbf{S}_{\nu'}\\ \mathbf{Q}'_\nu=\mathbf{Q}'_{\nu'},\ \ \mathbf{S}'_\nu=\mathbf{S}'_{\nu'},\end{cases}\\ \mathbf{P}_\nu=\mathbf{P}'_\nu\end{cases} \tag{P3}$$

where consensus constraints are enforced per neighborhood $\mathscr{N}_\nu$, and $\{\mathbf{P}'_\nu,\mathbf{Q}'_\nu,\mathbf{S}'_\nu\}$ are utilized to split the LS cost from the Frobenius- and ℓ_1-norms. Typically, (P3) is expressed in unconstrained form using the (augmented) Lagrangian framework. Decentralized inference algorithms over networks, implementing the previous splitting methodology, can been found in [22], [47], [51], and [62]. ADMM and DR are convergent for convex costs, but they offer no convergence guarantees for the nonconvex (P3). There is, however, ample experimental evidence in the literature that supports empirical convergence of ADMM, especially when the nonconvex problem at hand exhibits "favorable" structure [10], [47].

Methods offering convergence guarantees for (P3), after encapsulating consensus constraints into the loss function, are sequential schemes, such as the block coordinate descent methods (BCDMs) [59], [77]. BCDMs minimize the underlying objective sequentially over one block of variables per iteration, while keeping all other blocks fixed to their most up-to-date values. For example, a BCDM for solving the DL subtask of (2), that is when $\{\mathbf{P}_\nu,\mathbf{Q}\}$ are absent from the optimization problem, is the K-SVD algorithm [2]. Per iteration, K-SVD alternates between sparse coding of the columns of $\mathbf{Y}$ based on the current dictionary and updating the dictionary atoms to better fit the data. For a consensus-based decentralized implementation of K-SVD in the cloud, see [58].

It is worth stressing that (P3) is convex w.r.t. each block among $\{\mathbf{P}_\nu,\mathbf{Q}_\nu,\mathbf{S}_\nu,\mathbf{P}'_\nu,\mathbf{Q}'_\nu,\mathbf{S}'_\nu\}$, whenever the rest are held constant. Recent *parallel* schemes with convergence guarantees take advantage of this underlying structure to speed-up decentralized and parallel optimization algorithms [33], [64]. Additional BCDM examples will be given next in the context of online learning.

ONLINE ALGORITHMS FOR STREAMING ANALYTICS

So far, $\mathbf{Y}$ has been decomposed across its rows corresponding to network agents or processors; in what follows, $\mathbf{Y}$ will be split across its columns. Aiming at online solvers of (P2), with t indexing the columns of $\mathbf{Y}:=[\mathbf{y}_1,\ldots,\mathbf{y}_t]$, and $\{\Omega_\tau\}_{\tau=1}^t$ indicating the locations of known data values across time, consider the analytics engine acquiring a stream of vectors $\mathscr{P}_{\Omega_t}(\mathbf{y}_t)$, $\forall t$. An online counterpart of (P2) is the following exponentially weighted LS estimate [48]

$$\min_{\substack{\mathbf{P}\\ \{\mathbf{q}_\tau,\mathbf{s}_\tau\}_{\tau=1}^t}}\ \sum_{\tau=1}^t\delta^{t-\tau}\Big[\frac{1}{2}\|\mathscr{P}_{\Omega_\tau}(\mathbf{y}_\tau-\mathbf{P}\mathbf{q}_\tau-\mathbf{D}_\tau\mathbf{s}_\tau)\|^2 + \frac{\lambda_*}{2\sum_{\tau'=1}^t\delta^{t-\tau'}}\|\mathbf{P}\|_{\text{F}}^2+\frac{\lambda_*}{2}\|\mathbf{q}_\tau\|^2+\lambda_1\|\mathbf{s}_\tau\|_1\Big], \tag{P4}$$

where $\mathbf{P}\in\mathbb{R}^{N\times\rho}$, $\{\mathbf{q}_\tau\}_{\tau=1}^t\subset\mathbb{R}^\rho$, $\{\mathbf{s}_\tau\}\subset\mathbb{R}^M$, and $\delta\in(0,1]$ denotes the so-termed forgetting factor. With $\delta<1$, past data are exponentially discarded to track nonstationary features. Clearly, $\mathscr{P}_{\Omega_t}$ can be represented by a matrix $\mathbf{\Omega}_t$, whose rows are a subset of the rows of the N-dimensional identity matrix.

A provably convergent BCDM approach to efficiently solve a simplified version of (P4) was put forth in [48]. Each time t a new datum is acquired, only $\mathbf{q}_t$ and $\mathbf{s}_t$ are jointly updated via Lasso for fixed $\mathbf{P}=\mathbf{P}_{t-1}$, and then (P4) is solved w.r.t. $\mathbf{P}$ to update $\mathbf{P}_{t-1}$ using recursive LS (RLS). The latter step can be efficiently split across rows $\mathbf{p}_{n,t}=\arg\min_{\mathbf{p}}\sum_{\tau=1}^t\delta^{t-\tau}\omega_{n,\tau}(y_{n,\tau}-\mathbf{p}^\top\mathbf{q}_\tau-\mathbf{d}_{n,\tau}^\top\mathbf{s}_\tau)^2+(\lambda_*/2)\|\mathbf{p}\|^2$—an attractive feature facilitating parallel processing, which nevertheless entails a matrix inversion when $\delta<1$. Since first introduced in [48], the idea of performing online rank-minimization leveraging the separable nuclear-norm regularization in (P4) has gained popularity in real-time NNMF for audio SP [71], and online robust PCA [21], to name a few examples. In the case where $\mathbf{P},\{\mathbf{q}_\tau\}_{\tau=1}^t$ are absent from (P4), an online DL method of the same spirit as in [48] can be found in [46], [67].

Algorithms in [48] are closely related to timely robust subspace trackers, which aim at estimating a low-rank subspace $\mathbf{P}$ from grossly corrupted and possibly incomplete data, namely $\mathscr{P}_{\Omega_t}(\mathbf{y}_t)=\mathscr{P}_{\Omega_t}(\mathbf{P}\mathbf{q}_t+\mathbf{s}_t+\mathbf{v}_t)$, $t=1,2,\ldots$. In the absence of sparse outliers $\{\mathbf{s}_t\}_{t=1}^\infty$, an online algorithm based on incremental gradient descent on the Grassmannian manifold of subspaces was put forth in [4]. The second-order RLS-type algorithm in [16] extends the seminal projection approximation subspace tracking (PAST) algorithm to handle missing data; see also [50]. When outliers are present, robust counterparts can be found in [15] and [29]. Relative to all aforementioned works, the estimation problem (P4) is more challenging due to the presence of the (compression) dictionary $\mathbf{D}_t$.

Reflecting on (P1)–(P4), all objective functions share a common structure: they are convex w.r.t. each of their variable blocks, provided the rest are held fixed. Naturally, this calls for BCDMs for minimization, as in the previous discussion. However, matrix inversions and solving a batch Lasso per slot t may prove prohibitive for large-scale optimization tasks. Projected or proximal stochastic (sub)gradient methods are attractive low-complexity online alternatives to BCDMs mainly for optimizing convex objectives [65]. Unfortunately, due to their diminishing step-sizes, such first-order solutions exhibit slow convergence even for convex problems. On the other hand, accelerated variants for convex problems offer quadratic convergence of the

objective function values, meaning they are optimally "fast" among first-order methods [54], [80]. Although quadratic convergence issues for nonconvex and time-varying costs as in (P4) are largely unexplored, the online, accelerated, first-order method outlined in Figure 2 offers a promising alternative for generally nonsmooth and nonconvex minimization tasks [68].

Let $\mathbf{x}^{(i)}$ be a block of variables, which in (P4) can be $\mathbf{P}$, or $\{\mathbf{q}_\tau\}_{\tau=1}^t$, or $\{\mathbf{s}_\tau\}_{\tau=1}^t$; that is, $i \in \{1,2,3\}$; and let $\mathbf{x}^{(-i)}$ denote all blocks in $\mathbf{x} := (\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(I)})$ except for $\mathbf{x}^{(i)}$. Consider the sequence of loss functions $F_t(\mathbf{x}) := f_t(\mathbf{x}) + \sum_{i=1}^I g_i(\mathbf{x}^{(i)})$, where f_t is nonconvex, and Lipschitz continuously differentiable but convex w.r.t. each $\mathbf{x}^{(i)}$, whenever $\{\mathbf{x}^{(j)}\}_{j\neq i}$ are held fixed; $\{g_i\}_{i=1}^I$ are convex and possibly nondifferentiable; hence, F_t is nonsmooth. Clearly, the data fit term in (P4) corresponds to f_t, $g_1(\mathbf{x}^{(1)}) := (\lambda_*/2)\|\mathbf{P}\|_{\mathrm{F}}^2$, while g_2 and g_3 describe the other two regularization terms.

The acceleration module Accel of [80], developed originally for offline convex analytic tasks, is applied to F_t in a sequential, per-block (Gauss–Seidel) fashion. Having $\mathbf{x}^{(-i)}$ fixed, unless $\min_{\mathbf{x}^{(i)}\in\mathscr{X}_i} f_t(\mathbf{x}^{(i)} \mid \mathbf{x}_t^{(-i)}) + g_i(\mathbf{x}^{(i)})$ is easily solvable, Accel is employed for $R_i \geq 1$ times to update $\mathbf{x}^{(i)}$. The same procedure is carried over to the next block $\mathbf{x}^{(i+1)}$, until all blocks are updated, and subsequently to the next time instant $t+1$ (Figure 2). Unlike ADMM, this first-order algorithm requires no matrix inversions, and can afford inexact solutions of minimization subtasks. Under several conditions, including (statistical) stationarity of $\{F_t\}_{t=1}^\infty$, it also guarantees quadratic-rate convergence to a stationary point of $\mathbb{E}\{F_t\}$, where $\mathbb{E}\{\cdot\}$ denotes expectation over noise and input data distributions [68]. An application of this method to the dictionary-learning context can be found in the "Inference and Imputation" section.

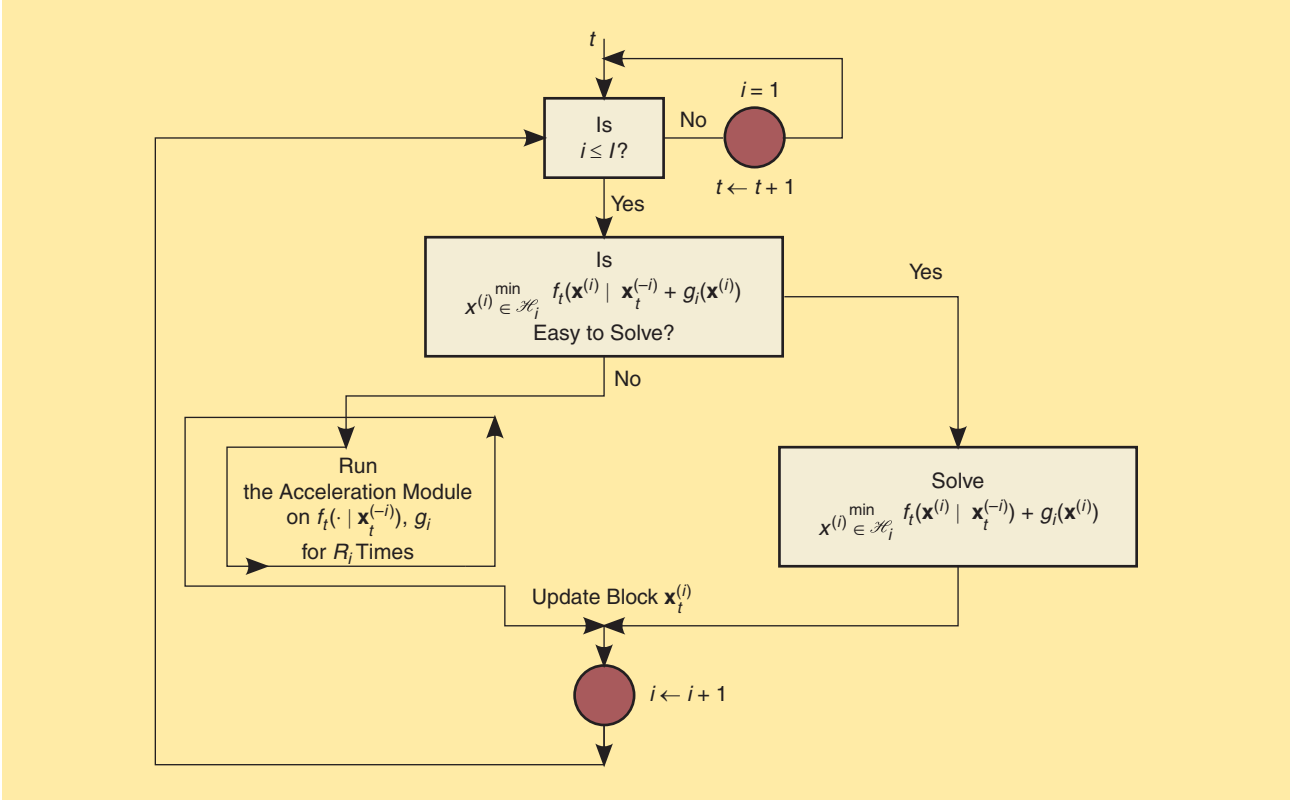
DATA SKETCHING, TENSORS, AND KERNELS

The scope of the "Algorithms" section can be broadened to include random subsampling schemes on $\mathbf{Y}$ (also known as data sketching), as well as multiway data arrays (tensors) and nonlinear modeling via kernel functions.

DATA SKETCHING

Catering to decentralized or parallel solvers, all variables in (P3) should be updated in parallel across learners of individual network nodes. However, there are cases where solving all learning subtasks simultaneously may be prohibitive or inefficient for two main reasons. First, the data size might be so large that computing function values or first-order information over all variables is impossible. Second, the nature and structure of data may prevent a fully parallel operation; e.g., when data are not available in their entirety, but are acquired either in batches over time or where not all of the network nodes are equally responsive or functional.

A recent line of research aiming at obtaining informative subsets of measurements for asynchronous and reduced-dimensionality processing of big data sets is based on (random) subsampling or data



[FIG2] The online, accelerated, sequential (Gauss–Seidel) optimization scheme for asymptotically minimizing the sequence $(F_t)_{t\in\mathbb{N}}$ of nonconvex functions.

sketching (via $\mathcal{P}_\Omega$) of the massive $\mathbf{Y}$ [45]. The basic principles of data sketching will be demonstrated here for the overdetermined $(N \gg \rho)$ LS $\mathbf{q}_* := \mathbf{P}^\dagger \mathbf{y} \in \arg\min_{\mathbf{q}\in\mathbb{R}^\rho} \|\mathbf{y} - \mathbf{P}\mathbf{q}\|^2$ [a task subsumed by (P2) as well], where $\dagger$ denotes pseudo-inverse, and $\mathbf{P}^\dagger = (\mathbf{P}^\top\mathbf{P})^{-1}\mathbf{P}^\top$, for $\mathbf{P}$ full column-rank. Popular strategies to obtain $\mathbf{q}_*$ include the expensive SVD; the Cholesky decomposition if $\mathbf{P}$ is full column-rank and well conditioned; and the slower but more stable QR decomposition [45].

The basic premise of the subsampling or data sketching techniques is to largely reduce the number of rows of $\mathbf{Y}$ prior to solving the LS task [45]. A data-driven methodology of keeping only the "most" informative rows relies on the so-termed (statistical) leverage scores and is outlined next as a three-step procedure. Given the (thin) SVD $\mathbf{P} = \mathbf{U}\Sigma\mathbf{V}^\top$: (S1) find the normalized leverage scores $\{l_n\}_{n=1}^N$, where $l_n := \rho^{-1}\mathbf{e}_n^\top\mathbf{U}\mathbf{U}^\top\mathbf{e}_n = \rho^{-1}\mathbf{e}_n^\top\mathbf{P}\mathbf{P}^\dagger\mathbf{e}_n$, with $\mathbf{e}_n \in \mathbb{R}^N$ being the nth canonical vector. Clearly, l_n equals the (normalized) nth diagonal element of $\mathbf{P}\mathbf{P}^\dagger$, and since $\mathbf{P}\mathbf{P}^\dagger = \mathbf{U}\mathbf{U}^\top$ is the orthogonal projector onto the linear subspace spanned by the columns of $\mathbf{P}$, it follows that $\mathbf{P}\mathbf{P}^\dagger\mathbf{y}$ offers the best approximation to $\mathbf{y}$ within this subspace. Then, (S2) for an arbitrarily small $\epsilon > 0$, and by using $\{l_n\}_{n=1}^N$ as an importance sampling distribution, randomly sample and rescale by $(rl_n)^{-1}$ a number of $r = \mathcal{O}(\epsilon^{-2}\rho\log\rho)$ rows of $\mathbf{P}$, together with the corresponding entries of $\mathbf{y}$. Such a sampling and rescaling operation can be expressed by a matrix $\mathbf{\Psi} \in \mathbb{R}^{r\times N}$. Finally, (S3) solve the reduced-size LS problem $\tilde{\mathbf{q}}_* \in \arg\min_{\mathbf{q}\in\mathbb{R}^\rho} \|\mathbf{\Psi}(\mathbf{y} - \mathbf{P}\mathbf{q})\|^2$. With $\kappa(\cdot)$ denoting condition number and $\gamma := \|\mathbf{y}\|^{-1}\|\mathbf{U}\mathbf{U}^\top\mathbf{y}\|$, it holds that [45]

$$\|\mathbf{y} - \mathbf{P}\tilde{\mathbf{q}}_*\| \le (1+\epsilon)\|\mathbf{y} - \mathbf{P}\mathbf{q}_*\| \tag{3a}$$

$$\|\mathbf{q}_* - \tilde{\mathbf{q}}_*\| \le \sqrt{\epsilon}\,\kappa(\mathbf{P})\sqrt{\gamma^{-2}-1}\,\|\mathbf{q}_*\| \tag{3b}$$

so that performance degrades gracefully after reducing the number of equations.

Similar to the nuclear-norm, a major difficulty is that leverage scores are not amenable to decentralized computation [cf. discussion prior (P2)], since the SVD of $\mathbf{P}$ is necessary prior to decentralizing the original learning task. To avoid computing the statistical leverage scores, the following data-agnostic strategy has been advocated [45]: 1) Premultiply $\mathbf{P}$ and $\mathbf{y}$ with the $N\times N$ random Hadamard transform $\mathbf{H}_N\mathbf{\Delta}$, where $\mathbf{H}_N$ is defined inductively as

$$\mathbf{H}_N = \frac{1}{\sqrt{N}}\begin{bmatrix}\mathbf{H}_{N/2} & \mathbf{H}_{N/2}\\ \mathbf{H}_{N/2} & -\mathbf{H}_{N/2}\end{bmatrix}, \mathbf{H}_2 := \frac{1}{\sqrt{2}}\begin{bmatrix}+1 & +1\\ +1 & -1\end{bmatrix},$$

and $\mathbf{\Delta}$ is a diagonal matrix whose nonzero entries are drawn independently and uniformly from $\{-1, +1\}$, 2) uniformly sample and rescale a number of $r = \mathcal{O}(\rho\log\rho\cdot\log N + \epsilon^{-1}N\log\rho)$ rows from $\mathbf{H}_N\mathbf{\Delta}\mathbf{P}$ together with the corresponding components from $\mathbf{H}_N\mathbf{\Delta}\mathbf{y}$, and 3) find $\tilde{\mathbf{q}}_* \in \arg\min_{\mathbf{q}\in\mathbb{R}^\rho} \|\mathbf{\Psi}\mathbf{H}_N\mathbf{\Delta}(\mathbf{y} - \mathbf{P}\mathbf{q})\|^2$, where $\mathbf{\Psi}$ stands again for the sampling and rescaling operation. Error bounds similar to those in (1) can be also derived for this preconditioning strategy [45]. Key to deriving such performance bounds is the Johnson–Lindenstrauss lemma, which loosely asserts that for any $\epsilon \in (0,1)$, any set of ρ points in N dimensions can be (linearly) embedded into $r \ge 4(2^{-1}\epsilon^2 - 3^{-1}\epsilon^3)^{-1}\ln\rho$ dimensions, while preserving the pairwise Euclidean distances of the original points up to a multiplicative factor of $(1 \pm \epsilon)$.

Besides the previous overdetermined LS task, data sketching has been employed to ease the computational burden of several large-scale tasks ranging from generic matrix multiplication, SVD computation, to k-means clustering and tensor approximation [20], [45]. In the spirit of $\mathbf{H}_N\mathbf{\Delta}$, methods utilizing sparse embedding matrices have been also developed for overconstrained LS and ℓ_p-norm regression, low-rank and leverage scores approximation [17]; in particular, they exhibit complexity $\mathcal{O}(|\operatorname{supp}(\mathbf{P})|) + \mathcal{O}(\rho^3\epsilon^{-2}\log^l(\rho^3\epsilon^{-2}))$ for solving the LS task satisfying (3a), where $|\operatorname{supp}(\mathbf{P})|$ stands for the cardinality of the support of $\mathbf{P}$, and $l \in \mathbb{N}_*$. Viewing the sampling and rescaling operator $\mathbf{\Psi}$ as a special case of a (weighted) $\mathcal{P}_\Omega$ allows carrying over the algorithms outlined in the "Encompassing Models for Succinct Big Data Representations" and "Algorithms" sections to the data sketching setup as well.

BIG DATA TENSORS

Although the matrix model in (1) is quite versatile and can subsume a variety of important frameworks as special cases, the particular planar arrangement of data poses limitations in capturing available structures that can be crucial for effective interpolation. In the example of movie recommender systems, matrix models can readily handle two-dimensional structures of people $\times$ movie ratings. However, movies are classified in various genres and one could explicitly account for this information by arranging ratings in a sparse person $\times$ genre $\times$ title three-way array or tensor. In general, various tensor data analytic tasks for network traffic, social networking, or medical data analysis aim at capturing an underlying latent structure, which calls for high-order factorizations even in the presence of missing data [1], [50].

A rank-one three-way array $\underline{\mathbf{Y}} = [y_{i_a i_b i_c}] \in \mathbb{R}^{I_a\times I_b\times I_c}$, where the underline denotes tensors, is the outer product $\mathbf{a}\circ\mathbf{b}\circ\mathbf{c}$ of three vectors $\mathbf{a}\in\mathbb{R}^{I_a}, \mathbf{b}\in\mathbb{R}^{I_b}, \mathbf{c}\in\mathbb{R}^{I_c}$: $y_{i_a i_b i_c} = a_{i_a} b_{i_b} c_{i_c}$. One can interpret a_{i_a}, b_{i_b}, and c_{i_c} as corresponding to the people, genre, and title components, respectively, in the previous example. The rank of a tensor is the smallest number of rank-one tensors that sum up to generate the given tensor. These notions readily generalize to higher-way tensors, depending on the application. Notwithstanding, this is not an incremental extension from low-rank matrices to low-rank tensors, since even computing the tensor rank is an NP-hard problem in itself [36]. Defining a convex surrogate for the rank penalty such as the nuclear norm for matrices is not obvious either, since singular values when applicable, e.g., in the Tucker model, are not related to the rank [74]. Although a three-way array can be "unfolded" to obtain a matrix exhibiting latent Kronecker product structure, such an unfolding typically destroys the structure that one looks for.

These considerations, motivate forming a low-rank approximation of tensor $\underline{\mathbf{Y}}$ as

$$\underline{\mathbf{Y}} \approx \sum_{r=1}^{\rho} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r. \tag{4}$$

Low-rank tensor approximation is a relatively mature topic in multilinear algebra and factor analysis, and when exact, the decomposition (4) is called parallel factor analysis (PARAFAC) or canonical decomposition (CANDECOMP) [36]. PARAFAC is the model of choice when one is primarily interested in revealing latent structure. Unlike the matrix case, low-rank tensor decomposition can be unique. There is deep theory behind this result, and algorithms recovering the rank-one factors [37]. However, various computational and big data-related challenges remain. Missing data have been handled in rather ad hoc ways [76]. Parallel and decentralized implementations have not been thoroughly addressed; see, e.g., ParCube and GigaTensor algorithms for recent scalable approaches [57].

With reference to (4), introduce the factor matrix $\mathbf{A} := [\mathbf{a}_1, \ldots, \mathbf{a}_\rho] \in \mathbb{R}^{I_a \times \rho}$, and likewise for $\mathbf{B} \in \mathbb{R}^{I_b \times \rho}$ and $\mathbf{C} \in \mathbb{R}^{I_c \times \rho}$. Let $\mathbf{Y}_{i_c}, i_c = 1, \ldots, I_c$ denote the i_cth slice of $\underline{\mathbf{Y}}$ along its third (tube) dimension, such that $\mathbf{Y}_{i_c}(i_a, i_b) = y_{i_a i_b i_c}$. It follows that (4) can be compactly represented in matrix form, in terms of slice factorizations $\mathbf{Y}_{i_c} = \mathbf{A}\,\mathrm{diag}\,(\mathbf{e}_{i_c}^\top \mathbf{C})\,\mathbf{B}^\top, \forall i_c$. Capitalizing on the Frobenius-norm regularization (P2), decentralized algorithms for low-rank tensor completion under the PARAFAC model can be based on the optimization task:

$$\min_{\{\mathbf{A},\mathbf{B},\mathbf{C}\}} \sum_{i_c=1}^{I_c} \| \mathcal{P}_{\Omega_{i_c}}(\mathbf{Y}_{i_c} - \mathbf{A}\,\mathrm{diag}\,(\mathbf{e}_{i_c}^\top \mathbf{C})\,\mathbf{B}^\top) \|_{\mathrm{F}}^2 + \lambda_* [\| \mathbf{A} \|_{\mathrm{F}}^2 + \| \mathbf{B} \|_{\mathrm{F}}^2 + \| \mathbf{C} \|_{\mathrm{F}}^2]. \tag{5}$$

Different from the matrix case, it is unclear whether the regularization in (5) bears any relation with the tensor rank. Interestingly, [7] asserts that (5) provably yields a low-rank $\hat{\underline{\mathbf{Y}}}$ for sufficiently large λ_*, while the potential for scalable BCDM-based interpolation algorithms is apparent. For an online algorithm, see also (9) in the section "Big Data Tasks" and [50] for further details.

KERNEL-BASED LEARNING

In imputing random missing entries, prediction of multiway data can be viewed as a tensor completion problem, where an entire slice (say, the one orthogonal to the tube direction representing time) is missing. Notice that since (5) does not specify a correlation structure, it cannot perform this extrapolation task. Kernel functions provide the nonlinear means to infuse correlations or side information (e.g., user age range and educational background for movie recommendation systems) in various big data tasks spanning disciplines such as 1) statistics, for inference and prediction [28], 2) machine learning, for classification, regression, clustering, and dimensionality reduction [63], and 3) SP, as well as (non)linear system identification, sampling, interpolation, noise removal, and imputation; see, e.g., [6] and [75].

In kernel-based learning, processing is performed in a high-, possibly infinite-dimensional reproducing kernel Hilbert space (RKHS) $\mathscr{H}$, where function $f \in \mathscr{H}$ to be learned is expressed as a superposition of kernels; i.e., $f(\mathbf{x}) := \sum_{i=1}^{\infty} \varphi_i \kappa(\mathbf{x}, \mathbf{x}_i)$, where $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is the kernel associated with $\mathscr{H}$, $\{\varphi_i\}_{i=1}^{\infty}$ denote the expansion coefficients, and $\mathbf{x}, \mathbf{x}_i \in \mathcal{X}$, $\forall i$ [63]. Broadening the scope of (5), a kernel-based tensor completion problem is posed as follows. With index sets $\mathcal{X}_a := \{1, \ldots, I_a\}$, $\mathcal{X}_b := \{1, \ldots, I_b\}$, and $\mathcal{X}_c := \{1, \ldots, I_c\}$, and associated kernels $\kappa_{\mathcal{X}_a}(i_a, i_a')$, $\kappa_{\mathcal{X}_b}(i_b, i_b')$ and $\kappa_{\mathcal{X}_c}(i_c, i_c')$, tensor entry $y_{i_a i_b i_c}$ is approximated using functions from the set $\mathcal{F} := \{f(i_a, i_b, i_c) = \sum_{r=1}^{\rho} a_r(i_a) b_r(i_b) c_r(i_c) | a_r \in \mathscr{H}_{\mathcal{X}_a}, b_r \in \mathscr{H}_{\mathcal{X}_b}, c_r \in \mathscr{H}_{\mathcal{X}_c}\}$, where ρ is an upper bound on the rank. Specifically, with binary weights $\{\omega_{i_a i_b i_c}\}$ taking value 0 if $y_{i_a i_b i_c}$ is missing (and 1 otherwise), fitting low-rank tensors is possible using

$$\hat{f} = \arg\min_{f \in \mathcal{F}} \sum_{i_a, i_b, i_c} \omega_{i_a i_b i_c} [y_{i_a i_b i_c} - f(i_a, i_b, i_c)]^2 + \lambda_* \sum_{r=1}^{\rho} \left[\| a_r \|_{\mathscr{H}_{\mathcal{X}_a}}^2 + \| b_r \|_{\mathscr{H}_{\mathcal{X}_b}}^2 + \| c_r \|_{\mathscr{H}_{\mathcal{X}_c}}^2 \right]. \tag{6}$$

If all kernels are selected as Kronecker deltas, (6) reverts back to (5). The separable structure of the regularization in (6) allows application of Representer's theorem [63], which implies that a_r, b_r, and c_r admit finite dimensional representations given by $a_r(i_a) = \sum_{i_a''=1}^{I_a} \alpha_{r i_a''} \kappa_{\mathcal{X}_a}(i_a, i_a'')$, $b_r(i_b) = \sum_{i_b''=1}^{I_b} \beta_{r i_b''} \kappa_{\mathcal{X}_b}(i_b, i_b'')$, and $c_r(i_c) = \sum_{i_c''=1}^{I_c} \gamma_{r i_c''} \kappa_{\mathcal{X}_c}(i_c, i_c'')$, respectively. Coefficients $\hat{\mathbf{A}} := [\hat{\alpha}_{r i_a''}]$, $\hat{\mathbf{B}} := [\hat{\beta}_{r i_b''}]$, and $\hat{\mathbf{C}} := [\hat{\gamma}_{r i_c''}]$ turn out to be solutions of [cf. (5)]

$$(\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{C}}) := \arg\min_{\{\mathbf{A},\mathbf{B},\mathbf{C}\}} \sum_{i_c=1}^{I_c} \| \mathcal{P}_{\Omega_{i_c}}(\mathbf{Y}_{i_c} - \mathbf{K}_{\mathcal{X}_a} \mathbf{A}\,\mathrm{diag}\,(\mathbf{e}_{i_c}^\top \mathbf{K}_{\mathcal{X}_c} \mathbf{C})\,\mathbf{B}^\top \mathbf{K}_{\mathcal{X}_b}) \|_{\mathrm{F}}^2 + \lambda_* \,\mathrm{trace}\,[\mathbf{A}^\top \mathbf{K}_{\mathcal{X}_a} \mathbf{A} + \mathbf{B}^\top \mathbf{K}_{\mathcal{X}_b} \mathbf{B} + \mathbf{C}^\top \mathbf{K}_{\mathcal{X}_c} \mathbf{C}], \tag{P5}$$

where $\mathbf{K}_{\mathcal{X}_a} := [\kappa_{\mathcal{X}_a}(i_a, i_a')]$, and likewise for $\mathbf{K}_{\mathcal{X}_b}$ and $\mathbf{K}_{\mathcal{X}_c}$, stand for kernel matrices formed using (cross-)correlations estimated from historical data as detailed in, e.g., [7]. Remarkably, the cost in (P5) is convex w.r.t. any of $\{\mathbf{A}, \mathbf{B}, \mathbf{C}\}$, whenever the rest of them are held fixed. As such, the low-complexity online accelerated algorithms of the "Algorithms" section carry over to tensors too. Having $\hat{\mathbf{A}}$ available, the estimate $\hat{\alpha}_{r i_a''}$ is obtained, and likewise for $\hat{\beta}_{r i_b''}$ and $\hat{\gamma}_{r i_c''}$. The latter yield the desired predicted values as $\hat{y}_{i_a i_b i_c} := \sum_{r=1}^{\rho} \hat{a}_r(i_a) \hat{b}_r(i_b) \hat{c}_r(i_c) \approx y_{i_a i_b i_c}$.

BIG DATA TASKS

The tools and themes outlined so far will be applied in this section to a sample of big data SP-relevant tasks.

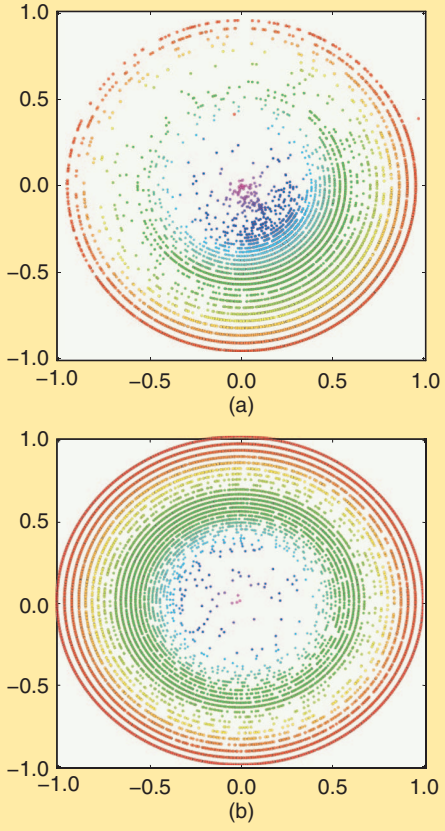
DIMENSIONALITY REDUCTION

NETWORK VISUALIZATION

The rising complexity and volume of networked (graph-valued) data presents new opportunities and challenges for visualization tools that capture global patterns and structural information such as hierarchy, similarity, and communities [3], [27]. Most visualization algorithms tradeoff the clarity of structural characteristics of the underlying data for aesthetic requirements

such as minimal edge crossing and fixed internode distance. Although efficient for relatively small networks or graphs (hundreds of nodes), embeddings for larger graphs using these techniques are seldom structurally informative. The growing interest in analysis of big data networks has prioritized the need for effectively capturing structure over aesthetics in visualization. For instance, layouts of metro-transit networks that show hierarchically the bulk of traffic convey a lucid picture about the most critical nodes in the event of a terrorist attack. To this end, [3] captures hierarchy in networks or graphs through well-defined measures of node importance, collectively known as *centrality* in the network science community. Examples are the betweenness centrality, which describes the extent to which information is routed through a specific node by measuring the fraction of all shortest paths traversing it, as well as closeness, eigenvalue, and Markov centrality [3].

THE RISING COMPLEXITY AND VOLUME OF NETWORKED (GRAPH-VALUED) DATA PRESENTS NEW OPPORTUNITIES AND CHALLENGES FOR VISUALIZATION TOOLS THAT CAPTURE GLOBAL PATTERNS AND STRUCTURAL INFORMATION SUCH AS HIERARCHY, SIMILARITY, AND COMMUNITIES.



[FIG3] The visualization of two snapshots of the large-scale network Gnutella [40] by means of the CC-LLE method. The centrality metric is defined by the node degree. Hence, nodes with low degree are placed far from the center of the embedding. (a) Gnutella-04 (08/04/2012). (b) Gnutella-24 (08/24/2012).

Consider an undirected graph $\mathcal{G}(\mathcal{V},\mathcal{E})$, where $\mathcal{V}$ denotes the set of vertices (nodes, agents, or processing cores) with cardinality $|\mathcal{V}|=V$, and $\mathcal{E}$ stands for edges (links) that represent pairs of nodes that can communicate. Following (P3), node $v\in\mathcal{V}$ communicates with its single- or multihop neighboring peers in $\mathcal{N}_v\subset\mathcal{V}$. Given a set of observed feature vectors $\{\mathbf{y}_v\}_{v\in\mathcal{V}}\subset\mathbb{R}^P$, and a prescribed embedding dimension $p\ll P$ (typically $p\in\{2,3\}$ for visualization), the graph embedding amounts to finding a set of $\{\mathbf{z}_v\}_{v\in\mathcal{V}}\subset\mathbb{R}^p$ vectors that preserve in the very low-dimensional $\mathbb{R}^p$ the network structure observed via $\{\mathbf{y}_v\}_{v\in\mathcal{V}}$. The dimensionality reduction module of [3] is based on local linear embedding (LLE) principles [61], which assume that the observed $\{\mathbf{y}_v\}_{v\in\mathcal{V}}$ live on a low-dimensional, smooth, but unknown manifold, with the objective of seeking an embedding that preserves the local structure of the manifold in the lower dimensional $\mathbb{R}^p$. In particular, LLE accomplishes this by approximating each data point via an affine combination (real weights summing up to 1) of its neighbors, followed by construction of a lower-dimensional embedding that best preserves the weights. If $\mathbf{Y}_v:=[\mathbf{y}_{v_1'},\ldots,\mathbf{y}_{v_{|\mathcal{N}_v|}'}]\in\mathbb{R}^{P\times|\mathcal{N}_v|}$ gathers all the observed data within the neighborhood of node v, and along the lines of LLE, the centrality constrained (CC-)LLE method comprises the following two steps:

$$\begin{aligned}
&\text{S1: } \forall v\in\mathcal{V},\ \mathbf{s}_v\in\arg\min_{\mathbf{s}}\|\mathbf{y}_v-\mathbf{Y}_v\mathbf{s}\|^2\\
&\qquad\qquad \text{s. to}\begin{cases}\|\mathbf{Y}_v\mathbf{s}\|^2=h^2(c_v)\\ \mathbf{s}^\top\mathbf{1}=1\end{cases}\\
&\text{S2: } \min_{\{\mathbf{z}_v\}_{v\in\mathcal{V}}}\sum_{v\in\mathcal{V}}\|\mathbf{z}_v-\sum_{v'\in\mathcal{V}}s_{vv'}\mathbf{z}_{v'}\|^2\\
&\qquad\qquad \text{s.to}\,\|\mathbf{z}_v\|^2=h^2(c_v),\ \forall v\in\mathcal{V},
\end{aligned}\tag{7}$$

where $\{c_v\}_{v\in\mathcal{V}}\subset\mathbb{R}$ are centrality metrics, $h(\cdot)$ is a monotone decreasing function that quantifies the centrality hierarchy, e.g., $h(c_v)=\exp(-c_v)$, and $\mathbf{s}^\top\mathbf{1}=1$ enforces the local affine approximation of $\mathbf{y}_v$ by $\{\mathbf{y}_{v'}\}_{v'\in\mathcal{N}_v}$. In other words, and in the spirit of (P3), $\mathbf{y}_v$ is affinely approximated by the "local" dictionary $\mathbf{D}_v:=\mathbf{Y}_v$. It is worth stressing that both objective and constraints in step 1 of (7) can be computed solely by means of the inner-products or correlations $\{\mathbf{Y}_v^\top\mathbf{y}_v,\mathbf{Y}_v^\top\mathbf{Y}_v\}_{v\in\mathcal{V}}$. Hence, knowledge of $\{\mathbf{y}_v\}_{v\in\mathcal{V}}$ is not needed in CC-LLE, and only a given set of dissimilarity measures $\{\delta_{vv'}\}_{(v,v')\in\mathcal{V}^2}$ suffices to formulate (7), where $\delta_{vv'}\in\mathbb{R}_{\geq 0}$, $\delta_{vv'}=\delta_{v'v}$, and $\delta_{vv}=0$, $\forall(v,v')\in\mathcal{V}^2$; e.g., $\delta_{vv'}:=1-|\mathbf{y}_v^\top\mathbf{y}_{v'}|\,\|\mathbf{y}_v\|^{-1}\|\mathbf{y}_{v'}\|^{-1}$ in (7).

After relaxing the nonconvex constraint $\|\mathbf{Y}_v\mathbf{s}\|^2=h^2(c_v)$ to the convex $\|\mathbf{Y}_v\mathbf{s}\|^2\leq h^2(c_v)$ one, a BCDM approach is followed to solve (7) efficiently, with computational complexity that scales linearly with the network size [3]. Figure 3 depicts the validation of CC-LLE on large-scale degree visualizations of snapshots of the Gnutella peer-to-peer file-sharing network ($|\mathcal{V}|=26{,}518$,

$|\mathcal{E}| = 65{,}369$) [40]. Snapshots of this directed network were captured on 4 and 24 August 2002, respectively, with nodes representing hosts. For convenience, undirected renditions of the two networks were obtained by symmetrization of their adjacency matrices. Notice here that the method can generalize to the directed case too, at the price of increased computational complexity. The centrality metric of interest was the node degree, and dissimilarities were computed based on the number of shared neighbors between any pair of hosts. It is clear from Figure 3 that despite the dramatic growth of the network over a span of 20 days, most new nodes had low degree, located thus far from the center of the embedding. The CC-LLE efficiency is manifested by the low running times for obtaining embeddings in Figure 3; 1,684 s for Gnutella-04, and 5,639 s for Gnutella-24 [3].

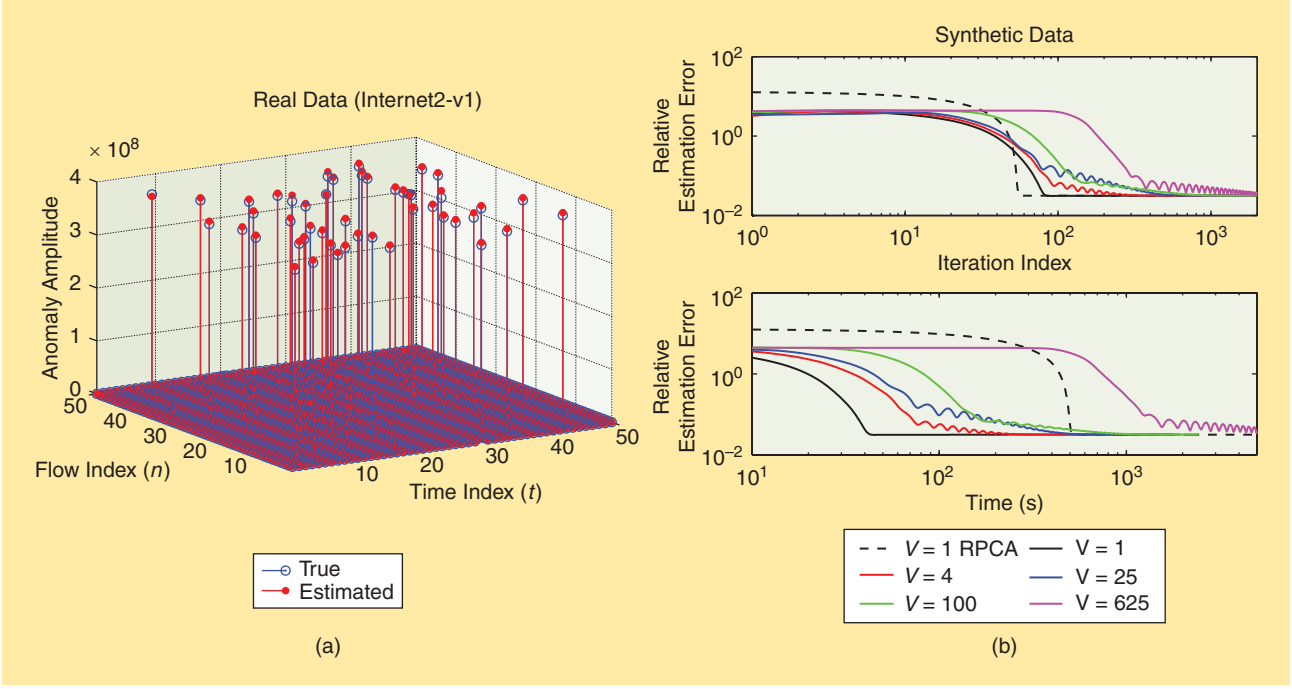
INFERENCE AND IMPUTATION

DECENTRALIZED ESTIMATION OF ANOMALOUS NETWORK TRAFFIC

In the backbone of large-scale networks, origin-to-destination (OD) traffic flows experience abrupt changes that can result in congestion and limit the quality of service provisioning of the end users. These traffic "anomalies" could be due to external sources such as network failures, denial of service attacks, or intruders [38]. Unveiling them is a crucial task in engineering network traffic. This is challenging however, since the available data are high-dimensional noisy link-load measurements, which comprise the superposition of "clean" and anomalous traffic.

Consider as in the section "Dimensionality Reduction" an undirected, connected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$. The traffic $\mathbf{Y} \in \mathbb{R}^{N \times T}$, carried over the edges or links $\mathcal{E}$ ($|\mathcal{E}| = N$) and measured at time instants $t \in \{1, \ldots, T\}$ is modeled as the superposition of unknown "clean" traffic flows $\mathbf{L}_*$, over the time horizon of interest, and the traffic volume anomalies $\mathbf{S}_*$ plus noise $\mathbf{V}$; $\mathbf{Y} = \mathbf{L}_* + \mathbf{S}_* + \mathbf{V}$. Common temporal patterns among the traffic flows in addition to their periodic behavior render most rows (respectively columns) of $\mathbf{L}_*$ linearly dependent, and thus $\mathbf{L}_*$ typically has low rank [38]. Anomalies are expected to occur sporadically over time, and only last for short periods relative to the (possibly long) measurement interval. In addition, only a small fraction of the flows is anomalous at any time slot. This renders matrix $\mathbf{S}_*$ sparse across rows and columns [48].

In the present context, real data including OD flow traffic levels and end-to-end latencies are collected from the operation of the Internet2 network (Internet backbone network across the United States) [30]. OD flow traffic levels were recorded for a three-week operation (sampled per 5 min) of Internet2-v1 during 8–28 December 2003 [38]. To better assess performance, large spikes of amplitude equal to the largest recorded traffic across all flows and time instants were injected into 1% randomly selected entries of the ground-truth matrix $\mathbf{L}_*$. Along the lines of (P3), where the number of links $N = 121$, and $T = 504$, the rows of the data matrix $\mathbf{Y}$ were distributed uniformly over a number of $\mathcal{V} = 11$ nodes. (P3) is solved using ADMM, and a small portion (50×50) of the estimated anomaly matrix $\hat{\mathbf{S}}$ is depicted in Figure 4(a).



[FIG4] Decentralized estimation of network traffic anomalies measured in byte units over 5 min time intervals: (a) only a small portion (50×50) of the sparse matrices $\mathbf{S}_*$ and $\hat{\mathbf{S}}$ entries are shown; (b) relative estimation error versus ADMM iteration index and central processing unit (CPU) time over networks with $\mathcal{V}$ number of nodes. The curve obtained by the centralized R-PCA method [12] is also depicted.

As a means of offering additional design insights, further validation is provided here to reveal the tradeoffs that become relevant as the network size increases. Specifically, comparisons in terms of running time are carried out w.r.t. its centralized counterpart. Throughout, a network modeled as a square grid (uniform lattice) with agents per row/column is adopted. To gauge running times as the network grows, consider a fixed size data matrix $\mathbf{Y} \in \mathbb{R}^{2,500 \times 2,500}$. The data are synthesized according to the previous model of $\mathbf{Y} = \mathbf{L}_* + \mathbf{S}_* + \mathbf{V}$, details for which can be found in [47, Sec. V]. Rows of $\mathbf{Y}$ are uniformly split among the network nodes. Figure 4(b) illustrates the relative estimation error $\|\hat{\mathbf{S}} - \mathbf{S}_*\|_F / \|\mathbf{S}_*\|_F$ ($\hat{\mathbf{S}}$ stands for the estimate of $\mathbf{S}_*$) versus both iteration index of the ADMM and CPU time over various network sizes.

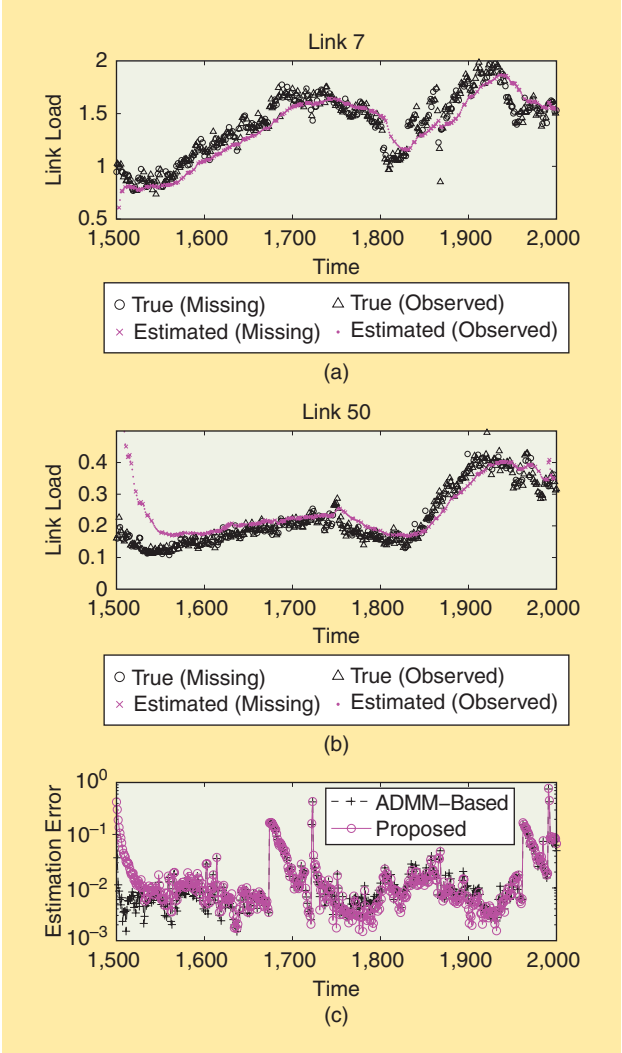
DYNAMIC LINK LOAD TRAFFIC PREDICTION AND IMPUTATION

Consider again the previous undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$. Connectivity and edge strengths of $\mathcal{G}$ are described by the adjacency matrix $\mathbf{W} \in \mathbb{R}^{V \times V}$, where $[\mathbf{W}]_{v v'} > 0$ if nodes v and v' are connected, while $[\mathbf{W}]_{v v'} = 0$ otherwise. At every $t \in \mathbb{N}_{>0}$, a variable $\chi_{tv} \in \mathbb{R}$, which describes a network-wide dynamical process of interest, corresponds to a node $v \in \mathcal{V}$. All node variables are collected in $\boldsymbol{\chi}_t := [\chi_{t1}, \ldots, \chi_{tV}]^\top \in \mathbb{R}^V$. A sparse representation of the process over $\mathcal{G}$ models $\boldsymbol{\chi}_t$ as a linear combination of "few" atoms in an $N \times M$ dictionary $\mathbf{D}$, with $M \geq N$; and $\boldsymbol{\chi}_t = \mathbf{D}\mathbf{s}_t$, where $\mathbf{s}_t \in \mathbb{R}^M$ is sparse. Further, only a portion of $\boldsymbol{\chi}_t$ is observed per time slot t. Let now $\boldsymbol{\Omega}_t \in \mathbb{R}^{N' \times N}$, $N' \leq N$, denote a binary measurement matrix, with each row of $\boldsymbol{\Omega}_t$ corresponding to the canonical basis vector for $\mathbb{R}^N$, selecting the measured components of $\mathbf{y}_t \in \mathbb{R}^{N'}$. In other words, the observed data per slot t are $\mathbf{y}_t = \boldsymbol{\Omega}_t \boldsymbol{\chi}_t + \mathbf{v}_t$, where $\mathbf{v}_t$ denotes noise. To impute missing entries of $\boldsymbol{\chi}_t$ in $\mathbf{y}_t$, the topology of $\mathcal{G}$ will be utilized. The spatial correlation of the process is captured by the (unnormalized) graph Laplacian matrix $\boldsymbol{\Lambda} := \operatorname{diag}(\mathbf{W}\mathbf{1}_N) - \mathbf{W}$, where $\mathbf{1}_N \in \mathbb{R}^N$ is the all-ones vector. Following Figure 2 and given a "forgetting factor" $\delta \in (0, 1]$, to gradually diminish the effect of past data (and thus account for nonstationarity), define

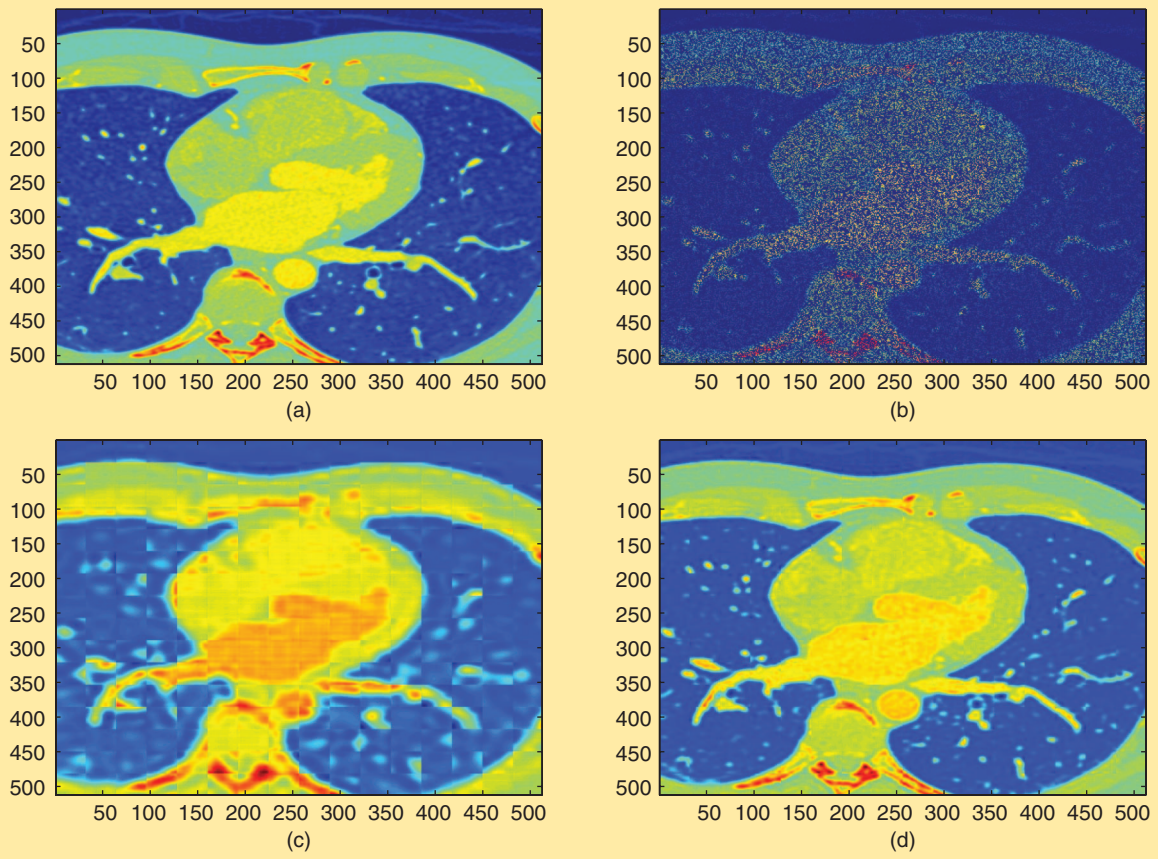
$$F_t(\mathbf{s}, \mathbf{D}) := \overbrace{\frac{1}{2\Delta_t} \sum_{\tau=1}^{t} \delta^{t-\tau} \|\mathbf{y}_\tau - \boldsymbol{\Omega}_\tau \mathbf{D}\mathbf{s}\|^2 + \frac{\lambda_\Lambda}{2} \mathbf{s}^\top \mathbf{D}^\top \boldsymbol{\Lambda} \mathbf{D}\mathbf{s}}^{f_t(\mathbf{s}, \mathbf{D})} + \overbrace{\lambda_1 \|\mathbf{s}\|_1}^{g_1(\mathbf{s})} + \overbrace{\iota_{\mathcal{D}}(\mathbf{D})}^{g_2(\mathbf{D})}, \tag{8}$$

where $\Delta_t := \sum_{\tau=1}^{t} \delta^{t-\tau}$, and $\iota_{\mathcal{D}}$ stands for the indicator function of $\mathcal{D} := \{\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_M] \in \mathbb{R}^{N \times M} \,|\, \|\mathbf{d}_m\| \leq 1, m \in \{1, \ldots, M\}\}$, i.e., $\iota_{\mathcal{D}}(\mathbf{D}) = 0$ if $\mathbf{D} \in \mathcal{D}$, and $\iota_{\mathcal{D}}(\mathbf{D}) = +\infty$ if $\mathbf{D} \notin \mathcal{D}$ (note that $\forall \gamma > 0$, $\operatorname{Prox}_{\gamma \iota_{\mathcal{D}}}$ is the metric projection onto the closed convex $\mathcal{D}$ [5]). The term including the known $\boldsymbol{\Lambda}$ quantifies the a priori information on the topology of $\mathcal{G}$, and promotes "smooth" solutions over strongly connected nodes of $\mathcal{G}$ [23]. This term is also instrumental for accommodating missing entries in $(\boldsymbol{\chi}_t)_{t \in \mathbb{N}_{>0}}$.

The algorithm of Figure 2 was validated on estimating and tracking network-wide link loads taken from the Internet2 measurement archive [30]. The network consists of $N = 54$ links and nine nodes. Using the network topology and routing information, network-wide link loads $(\boldsymbol{\chi}_t)_{t \in \mathbb{N}_{>0}} \subset \mathbb{R}^N$ become available (in gigabits per second). Per time slot t, only $N' = 30$ of the $\boldsymbol{\chi}_t$ components, chosen randomly via $\boldsymbol{\Omega}_t$, are observed in $\mathbf{y}_t \in \mathbb{R}^{N'}$. Cardinality of the time-varying dictionaries is set to $M = 80$, $\forall t$. To cope with pronounced temporal variations of the Internet2 link loads, the forgetting factor δ in (8) was set equal to 0.5. Figure 5 depicts estimated values of both observed (dots) and missing (crosses) link loads, for a randomly chosen link of the network. The normalized squared estimation error between the true $\boldsymbol{\chi}_t$ and the inferred $\hat{\boldsymbol{\chi}}_t$, specifically $\|\boldsymbol{\chi}_t - \hat{\boldsymbol{\chi}}_t\|^2 \|\boldsymbol{\chi}_t\|^{-2}$, is also plotted in Figure 5 versus time t. The accelerated algorithm was compared with the state-of-the-art scheme in [23] that relies on ADMM, to minimize a cost closely related to (8) w.r.t. $\mathbf{s}$, and uses BCD iterations requiring matrix inversion to optimize (8) w.r.t. $\mathbf{D}$. On the other hand, $R_1 = 1$ and $R_2 = 10$ in the algorithm of Figure 2. It is worth noticing here that ADMM in [23] requires multiple iterations to achieve a prescribed estimation accuracy, and that no matrix inversion



[FIG5] Link load tracking (dots and triangles) and imputation (crosses and circles) on Internet2 [30]. The proposed method is validated versus the ADMM-based approach of [23].



[FIG6] The imputation of missing functional MRI cardiac images by using the PARAFAC tensor model and the online framework of (9). The images were artificially colored to highlight the differences between the obtained recovery results. (a) The original image. (b) The degraded image (75% missing values). (c) The recovered image ($\rho = 10$) with relative estimation error 0.14. (d) The recovered image ($\rho = 50$) with relative estimation error 0.046.

was incorporated in the realization of the proposed scheme. Even if the accelerated first-order method operates under lower computational complexity than the ADMM approach, estimation error performance both on observed and missing values is almost identical.

CARDIAC MRI

Cardiac magnetic resonance imaging (MRI) is a major imaging tool for noninvasive diagnosis of heart diseases in clinical practice. However, time limitations posed by the patient's breath-holding time, and thus the need for fast data acquisition degrade the quality of MRI images, resulting often in missing pixel values. In the present context, imputation of the missing pixels utilizes the fact that cardiac MRI images intrinsically contain low-dimensional components.

The FOURDIX data set is considered, which contains 263 cardiac scans with ten steps of the entire cardiac cycle [24]. Each scan is an image of size 512×512 pixels, which is divided into 64 (32×32)-dimensional patches. Placing one after the other, patches form a sequence of slices of a tensor $\underline{\mathbf{Y}} \in \mathbb{R}^{32 \times 32 \times 67,328}$. Randomly chosen 75% of the $\underline{\mathbf{Y}}$ entries are dropped to simulate missing data. Operating on such a tensor via batch algorithms is computationally demanding, due to the tensor's size and the computer's memory limitations. Motivated by the batch formulation in (5), a weighted LS online counterpart is [50]

$$\min_{\{\mathbf{A},\mathbf{B},\mathbf{C}\}} \sum_{\tau=1}^{t} \delta^{t-\tau} \Big[\| \mathcal{P}_{\Omega}(\mathbf{Y}_{\tau} - \mathbf{A}\,\mathrm{diag}(\mathbf{e}_{\tau}^{\top}\mathbf{C})\,\mathbf{B}^{\top}) \|_{\mathrm{F}}^{2} + \frac{\lambda_*}{\sum_{\tau=1}^{t} \delta^{t-\tau}} (\| \mathbf{A} \|_{\mathrm{F}}^{2} + \| \mathbf{B} \|_{\mathrm{F}}^{2}) + \lambda_* \| \mathbf{e}_{\tau}^{\top}\mathbf{C} \|^{2} \Big], \quad (9)$$

where $\delta > 0$ is a forgetting factor, and $\mathbf{e}_{\tau}$ is the τth t-dimensional canonical vector. The third dimension t of $\underline{\mathbf{Y}}$ in (9) indicates the slice number. To solve (9), the variables $\{\mathbf{A}, \mathbf{B}, \mathbf{C}\}$ are sequentially processed; fixing $\{\mathbf{A}, \mathbf{B}\}$, (9) is minimized w.r.t. $\mathbf{C}$, while gradient steepest descent steps are taken w.r.t. each one of $\mathbf{A}$ and $\mathbf{B}$, having the other variables held constant. The resultant online learning algorithm is computationally light, with $256\rho^2$ operations (on average) per t. The results of its application to a randomly chosen scan image, for different choices of the rank ρ, are depicted in Figure 6 with relative estimation errors, $\| \mathbf{Y}_{\tau} - \hat{\mathbf{Y}}_{\tau} \|_{\mathrm{F}} / \| \mathbf{Y}_{\tau} \|_{\mathrm{F}}$, equal to 0.14 and 0.046 for $\rho = 10$ and 50, respectively.

Additional approaches for batch tensor completion of both visual and spectral data can be found in [41] and [66], whereas the algorithms in [1] and [7] carry out low-rank tensor decompositions from incomplete data and perform imputation as a by-product.

ACKNOWLEDGMENTS

Work in this article was supported by the National Science Foundation grants ECCS 1343248 and Eager 1343860. Moreover, it has been cofinanced by the European Union (European Social Fund and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework-Research Funding Program: Thalis–UoA–Secure Wireless Nonlinear Communications at the Physical Layer. We wish to thank Morteza Mardani and Brian Baingana, from the University of Minnesota, for the fruitful discussions and the numerical tests they provided.

AUTHORS

Konstantinos Slavakis (kslavaki@umn.edu) received his Ph.D. degree from the Tokyo Institute of Technology (TokyoTech), Japan, in 2002. He was a postdoctoral fellow with TokyoTech (2004–2006) and the Department of Informatics and Telecommunications, University of Athens, Greece (2006–2007). He was an assistant professor in the Department of Telecommunications and Informatics, University of Peloponnese, Tripolis, Greece (2007–2012). He is currently a research associate professor with the Department of Electrical and Computer Engineering and Digital Technology Center, University of Minnesota, United States. His current research interests include signal processing, machine learning, and big data analytics problems.

Georgios B. Giannakis (georgios@umn.edu) received his Ph.D. degree from the University of Southern California in 1986. Since 1999, he has been with the University of Minnesota, where he holds the ADC chair in wireless telecommunications in the Department of Electrical and Computer Engineering and serves as director of the Digital Technology Center. His interests are in the areas of communications, networking, and statistical signal processing—subjects on which he has published more than 360 journal and 620 conference papers, 21 book chapters, two edited books, and two research monographs (h-index 108). His current research focuses on sparsity and big data analytics, cognitive networks, renewables, power grid, and social networks. He is the (co) inventor of 22 patents and the (co)recipient of eight best paper awards from the IEEE Communications and Signal Processing Societies. He is a Fellow of the IEEE and EURASIP and has also received technical achievement awards from the IEEE Signal Processing Society and EURASIP.

Gonzalo Mateos (mate0058@umn.edu) received his B.Sc. degree in electrical engineering from Universidad de la Republica, Uruguay, in 2005 and the M.Sc. and Ph.D. degrees in electrical engineering from the University of Minnesota, in 2009 and 2012, respectively. Since 2014, he has been an assistant professor with the Department of Electrical and Computer Engineering, University of Rochester. During 2013, he was a visiting scholar with the Computer Science Department, Carnegie Mellon University. From 2003 to 2006, he worked as a systems engineer at ABB, Uruguay. His research interests lie in the areas of statistical learning from big data, network science, wireless communications, and signal processing. His current research focuses on algorithms, analysis, and application of statistical signal processing tools to dynamic network health monitoring, social, power grid, and big data analytics.

REFERENCES

[1] E. Acar, D. M. Dunlavy, T. G. Kolda, and M. Mørup, "Scalable tensor factorizations for incomplete data," *Chemomet. Intell. Lab. Syst.*, vol. 106, no. 1, pp. 41–56, 2011.

[2] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Processing*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.

[3] B. Baingana and G. B. Giannakis, "Embedding graphs under centrality constraints for network visualization," submitted for publication. arXiv:1401.4408.

[4] L. Balzano, R. Nowak, and B. Recht, "Online identification and tracking of subspaces from highly incomplete information," in *Proc. Allerton Conf. Communication, Control, and Computing*, Monticello, IL, 2010, pp. 704–711.

[5] H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. New York: Springer, 2011.

[6] J. A. Bazerque and G. B. Giannakis, "Nonparametric basis pursuit via sparse kernel-based learning," *IEEE Signal Process. Mag.*, vol. 30, no. 4, pp. 112–125, July 2013.

[7] J. A. Bazerque, G. Mateos, and G. B. Giannakis, "Rank regularization in Bayesian inference for tensor completion and extrapolation," *IEEE Trans. Signal Processing*, vol. 61, no. 22, pp. 5689–5703, Nov. 2013.

[8] T. Bengtsson, P. Bickel, and B. Li, "Curse-of-dimensionality revisited: Collapse of the particle filter in very large scale systems," in *Probability and Statistics: Essays in Honor of David A. Freedman*. Beachwood, OH: IMS, 2008, vol. 2, pp. 316–334.

[9] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Belmont, MA: Athena Scientific, 1999.

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

[11] E. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 21–30, 2008.

[12] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?," *J. ACM*, vol. 58, no. 1, pp. 1–37, 2011.

[13] E. J. Candes and Y. Plan, "Matrix completion with noise," *Proc. IEEE*, vol. 98, no. 6, pp. 925–936, June 2009.

[14] V. Chandrasekaran, S. Sanghavi, P. R. Parrilo, and A. S. Willsky, "Rank-sparsity incoherence for matrix decomposition," *SIAM J. Optim.*, vol. 21, no. 2, pp. 572–596, 2011.

[15] Q. Chenlu and N. Vaswani, "Recursive sparse recovery in large but correlated noise," in *Proc. Allerton Conf. Communication, Control, and Computing*, Sept. 2011, pp. 752–759.

[16] Y. Chi, Y. C. Eldar, and R. Calderbank, "PETRELS: Parallel subspace estimation and tracking using recursive least squares from partial observations," *IEEE Trans. Signal Processing*, vol. 61, no. 23, pp. 5947–5959, 2013.

[17] K. L. Clarkson and D. P. Woodruff, "Low rank approximation and regression in input sparsity time," in *Proc. Symp. Theory Computing*, June 1–4, 2013, pp. 81–90. arXiv:1207.6365v4.

[18] K. Cukier. (2010). Data, data everywhere. *The Economist*. [Online]. Available: http://www.economist.com/node/15557443

[19] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," in *Proc. Symp. Operating System Design and Implementation*, San Francisco, CA, 2004, vol. 6, p. 10.

[20] P. Drineas and M. W. Mahoney, "A randomized algorithm for a tensor-based generalization of the SVD," *Linear Algeb. Appl.*, vol. 420, no. 2–3, pp. 553–571, 2007.

[21] J. Feng, H. Xu, and S. Yan, "Online robust PCA via stochastic optimization," in *Proc. Advances in Neural Information Processing Systems*, Lake Tahoe, NV, Dec. 2013, pp. 404–412.

[22] P. Forero, A. Cano, and G. B. Giannakis, "Consensus-based distributed support vector machines," *J. Mach. Learn. Res.*, vol. 11, pp. 1663–1707, May 2010.

[23] P. Forero, K. Rajawat, and G. B. Giannakis, "Prediction of partially observed dynamical processes over networks via dictionary learning," *IEEE Trans. Signal Processing*, to be published.

[24] [Online]. Available: http://www.osirix-viewer.com/datasets/

[25] H. Gao, J. Cai, Z. Shen, and H. Zhao, "Robust principal component analysis-based four-dimensional computed tomography," *Phys. Med. Biol.*, vol. 56, no. 1, pp. 3181–3198, 2011.

[26] G. B. Giannakis, V. Kekatos, N. Gatsis, S. J. Kim, H. Zhu, and B. Wollenberg, "Monitoring and optimization for power grids: A signal processing perspective," *IEEE Signal Process. Mag.*, vol. 30, no. 5, pp. 107–128, Sept. 2013.

[27] L. Harrison and A. Lu, "The future of security visualization: Lessons from network visualization," *IEEE Netw.*, vol. 26, pp. 6–11, Dec. 2012.

[28] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. New York: Springer, 2009.

[29] J. He, L. Balzano, and A. Szlam, "Incremental gradient on the Grassmannian for online foreground and background separation in subsampled video," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Providence, RI, June 2012, pp. 1568–1575.

[30] [Online]. Available: http://www.internet2.edu/observatory/

[31] M. I. Jordan, "On statistics, computation and scalability," *Bernoulli*, vol. 19, no. 4, pp. 1378–1390, 2013.

[32] S. A. Kassam and H. V. Poor, "Robust techniques for signal processing: A survey," *Proc. IEEE*, vol. 73, no. 3, pp. 433–481, Mar. 1985.

[33] S.-J. Kim and G. B. Giannakis, "Optimal resource allocation for MIMO ad hoc cognitive radio networks," *IEEE Trans. Info. Theory*, vol. 57, no. 5, pp. 3117–3131, May 2011.

[34] A. Kleiner, A. Talwalkar, P. Sarkar, and M. I. Jordan, "A scalable bootstrap for massive data," *J. Royal Statist. Soc.: Ser. B,* to be published. [Online]. Available: http://dx.doi.org/10.1111/rssb.12050

[35] E. D. Kolaczyk, *Statistical Analysis of Network Data: Methods and Models*. New York: Springer, 2009.

[36] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, 2009.

[37] J. B. Kruskal, "Three-way arrays: Rank and uniqueness of trilinear decompositions, with applications to arithmetic complexity and statistics," *Linear Algeb. Appl.*, vol. 18, no. 2, pp. 95–138, 1977.

[38] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing network-wide traffic anomalies," in *Proc. SIGCOMM*, Aug. 2004, pp. 201–206.

[39] D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, pp. 788–791, Oct. 1999.

[40] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graph evolution: Densification and shrinking diameters," *ACM Trans. Knowl. Discov. Data*, vol. 1, no. 1, Mar. 2007.

[41] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, pp. 208–220, Jan. 2013.

[42] Y. Low, J. Gonzalez, A. Kyrola, D. Bickson, C. Guestrin, and J. Hellerstein, "GraphLab: A new framework for parallel machine learning," in *Proc. 26th Conf. Uncertainty in Artificial Intelligence*, Catalina Island: CA, 2010.

[43] Y. Ma, P. Niyogi, G. Sapiro, and R. Vidal, "Dimensionality reduction via subspace and submanifold learning [From the Guest Editors]," *IEEE Signal Process. Mag.*, vol. 28, no. 2, pp. 14–126, Mar. 2011.

[44] L. Mackey, A. Talwalkar, and M. I. Jordan, "Distributed matrix completion and robust factorization," submitted for publication. arXiv:1107.0789v7.

[45] M. W. Mahoney, "Randomized algorithms for matrices and data," *Found. Trends Machine Learn.*, vol. 3, no. 2, pp. 123–224, 2011.

[46] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *J. Machine Learn. Res.*, vol. 11, pp. 19–60, Mar. 2010.

[47] M. Mardani, G. Mateos, and G. B. Giannakis, "Decentralized sparsity-regularized rank minimization: Algorithms and applications," *IEEE Trans. Signal Processing*, vol. 61, no. 11, pp. 5374–5388, Nov. 2013.

[48] M. Mardani, G. Mateos, and G. B. Giannakis, "Dynamic anomalography: Tracking network anomalies via sparsity and low rank," *IEEE J. Sel. Topics Signal Process.*, vol. 8, pp. 50–66, Feb. 2013.

[49] M. Mardani, G. Mateos, and G. B. Giannakis, "Recovery of low-rank plus compressed sparse matrices with application to unveiling traffic anomalies," *IEEE Trans. Info. Theory*, vol. 59, no. 8, pp. 5186–5205, Aug. 2013.

[50] M. Mardani, G. Mateos, and G. B. Giannakis, "Subspace learning and imputation for streaming big data matrices and tensors," *IEEE Trans. Signal Processing*, submitted for publication.

[51] G. Mateos, J. A. Bazerque, and G. B. Giannakis, "Distributed sparse linear regression," *IEEE Trans. Signal Processing*, vol. 58, no. 10, pp. 5262–5276, Oct. 2010.

[52] G. Mateos and G. B. Giannakis, "Robust PCA as bilinear decomposition with outlier-sparsity regularization," *IEEE Trans. Signal Processing*, vol. 60, no. 10, pp. 5176–5190, Oct. 2012.

[53] B. K. Natarajan, "Sparse approximate solutions to linear systems," *SIAM J. Comput.*, vol. 24, no. 2, pp. 227–234, Apr. 1995.

[54] Y. Nesterov, "A method for solving the convex programming problem with convergence rate $O(1/k^2)$, *Dokl. Akad. Nauk SSSR*, vol. 269, no. 3, pp. 543–547, 1983.

[55] Office of Science and Technology Policy. (2012). Big data research and development initiative. *Executive Office of the President*. [Online]. Available: http://www.whitehouse.gov/sites/default/files/microsites/ostp/big_data_press_release_final_2.pdf

[56] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by V1?" *Vision Res.*, vol. 37, no. 23, pp. 3311–3325, 1997.

[57] E. E. Papalexakis, U. Kang, C. Faloutsos, N. D. Sidiropoulos, and A. Harpale, "Large scale tensor decompositions: Algorithmic developments and applications," *IEEE Data Eng. Bull.*, vol. 36, no. 3, pp. 59–66, Sept. 2013.

[58] H. Raja and W. U. Bajwa, "Cloud K-SVD: Computing data-adaptive representations in the cloud," in *Proc. Allerton Conf. Communication, Control, and Computing*, Oct. 2013, pp. 1474–1481.

[59] M. Razaviyayn, M. Hong, and Z.-Q. Luo, "A unified convergence analysis of block successive minimization methods for nonsmooth optimization," *SIAM J. Optim.*, vol. 23, no. 2, pp. 1126–1153, 2013.

[60] H. Robbins and S. Monro, "A stochastic approximation method," *Ann. Math. Statist.*, vol. 22, pp. 327–495, Sept. 1951.

[61] L. K. Saul and S. T. Roweis, "Think globally, fit locally: Unsupervised learning of low dimensional manifolds," *J. Mach. Learn. Res.*, vol. 4, pp. 119–155, Dec. 2003.

[62] I. D. Schizas, A. Ribeiro, and G. B. Giannakis, "Consensus in ad hoc WSNs with noisy links—Part I: Distributed estimation of deterministic signals," *IEEE Trans. Signal Processing*, vol. 56, no. 1, pp. 350–364, Jan. 2008.

[63] B. Schölkopf and A. J. Smola, *Learning with Kernels*. Cambridge, MA: MIT Press, 2001.

[64] G. Scutari, F. Facchinei, P. Song, D. P. Palomar, and J.-S. Pang, "Decomposition by partial linearization: Parallel optimization of multi-agent systems," *IEEE Trans. Signal Processing*, vol. 62, no. 3, pp. 641–656.

[65] S. Shalev-Shwartz, "Online learning and online convex optimization," *Found. Trends Mach. Learn.*, vol. 4, no. 2, pp. 107–194, 2012.

[66] M. Signoretto, R. V. Plas, B. D. Moor, and J. A. K. Suykens, "Tensor versus matrix completion: A comparison with application to spectral data," *IEEE Signal Process. Lett.*, vol. 18, pp. 403–406, July 2011.

[67] K. Skretting and K. Engan, "Recursive least squares dictionary learning algorithm," *IEEE Trans. Signal Processing*, vol. 58, no. 4, pp. 2121–2130, Apr. 2010.

[68] K. Slavakis and G. B. Giannakis, "Online dictionary learning from big data using accelerated stochastic approximation algorithms," in *Proc. ICASSP*, Florence, Italy, 2014, pp. 16–20.

[69] V. Solo and X. Kong, *Adaptive Signal Processing Algorithms: Stability and Performance*. Englewood Cliffs, NJ: Prentice Hall, 1995.

[70] M. Soltanolkotabi and E. J. Candès, "A geometric analysis of subspace clustering with outliers," *Ann. Statist.*, vol. 40, no. 4, pp. 2195–2238, Dec. 2012.

[71] P. Sprechmann, A. M. Bronstein, and G. Sapiro, "Real-time online singing voice separation from monaural recordings using robust low-rank modeling," in *Proc. Conf. Int. Society for Music Information Retrieval*, Oct. 2012, pp. 67–72.

[72] N. Srebro and A. Shraibman, "Rank, trace-norm and max-norm," in *Learning Theory*. Berlin/Heidelberg: Germany: Springer, 2005, pp. 545–560.

[73] N. Städler, D. J. Stekhoven, and P. Bühlmann, "Pattern alternating maximization algorithm for missing data in large p small n problems," *J. Mach. Learn. Res.*, to be published. arXiv:1005.0366v3.

[74] J. M. F. ten Berge and N. D. Sidiropoulos, "On uniqueness in CANDECOMP/PARAFAC," *Psychometrika*, vol. 67, no. 3, pp. 399–409, 2002.

[75] S. Theodoridis, K. Slavakis, and I. Yamada, "Adaptive learning in a world of projections: A unifying framework for linear and nonlinear classification and regression tasks," *IEEE Signal Process. Mag.*, vol. 28, no. 1, pp. 97–123, Jan. 2011.

[76] G. Tomasi and R. Bro, "PARAFAC and missing values," *Chemom. Intell. Lab. Syst.*, vol. 75, no. 2, pp. 163–180, 2005.

[77] P. Tseng, "Convergence of block coordinate decent method for nondifferentiable minimization," *J. Optim. Theory Appl.*, vol. 109, pp. 475–494, June 2001.

[78] R. Vidal, "Subspace clustering," *IEEE Signal Process. Mag.*, vol. 28, no. 2, pp. 52–68, Mar. 2011.

[79] B. Widrow and J. M. E. Hoff, "Adaptive switching circuits," *IRE WESCON Conv. Rec.*, vol. 4, pp. 96–104, Aug. 1960.

[80] M. Yamagishi and I. Yamada, "Over-relaxation of the fast iterative shrinkage-thresholding algorithm with variable stepsize," *Inverse Probl.*, vol. 27, no. 10, p. 105008, 2011.

[SP]