Online Kernel Dictionary Learning

Seung-Jun Kim
Dept. of Computer Science & Electrical Engr.
University of Maryland, Baltimore County
Baltimore, MD 21250, USA
E-mail: sjkim@umbc.edu

***Abstract*—Efficient online algorithms are developed to perform dictionary learning (DL) for the features lifted to a high-dimensional space via nonlinear mapping. Inspired by recent works on batch kernelized DL with promising performance for real-world learning tasks, two kernel DL formulations are put forth, amenable to online processing. The first formulation aims at faithfully representing the high-dimensional features in an unsupervised manner, while the the second one focuses on discriminative DL, where the dictionary is optimized for a specific supervised learning task. Motivated by Big Data processing applications, our algorithms are based on computationally efficient stochastic gradient descent variants. Numerical tests were performed to verify the convergence of the algorithms and to compare classification performances.**

I. Introduction

The dictionary learning (DL) problem pursues a set of potentially overcomplete bases, so that the signals of interest can be represented via linear combinations of few basis vectors. Since the seminal work by Olshausen and Field, which investigated the DL model to understand human vision [1], the technique has been applied in a variety of signal processing tasks to offer state-of-the-art performances [2], [3], [4]. DL can be viewed as an unsupervised learning (or blind signal processing) task, where the signal matrix is decomposed into bi-factors.

Extending the basic DL formulation developed for representing a given signal faithfully, the discriminative DL approaches try to optimize the dictionary for specific machine learning tasks, such as classification, regression, and signal separation [5], [6], [7]. In a supervised learning framework, the labels are provided in addition to the feature vectors, and the classifier are often jointly trained with the dictionary. Once the discriminative dictionary is obtained in the training phase, the labels can be predicted in the operational phase, using the sparse codes as new features (possibly in addition to the original features).

A variety of DL algorithms have been proposed in the literature. They are based on the Lagrange dual method [8], a generalization of k-means algorithm [9], or stochastic approximation techniques [10]. In particular, the method in [10] allows *online* learning of dictionaries. That is, as the data arrive one by one in a sequential fashion, the dictionary can be updated accordingly in a computationally efficient manner, so that the dictionary is close to the one obtained from batch processing after seeing enough examples. With the advent of Big Data analytics, it is essential to devise algorithms that can process large-scale data efficiently. In particular, the merits of online and randomized variants are widely recognized, as they require little computational effort per step, and can quickly provide a rough solution, which can be further refined over time as needed [11], [12].

Kernel-based learning significantly broadens the applicability of the statistical learning tools by transforming the features to a high (possibly infinite) dimensional space through a nonlinear mapping, capturing nonlinear traits in the data [13]. Thanks to the so-called "kernel trick," the learning algorithms do not actually need to compute the high-dimensional features, but access the data only through inner products of the features, provided by well-defined kernel functions [14], [15]. The kernel-based approaches adopt a nonparametric framework where the number of parameters increases with the number of data points. Therefore, the computational complexity of the kernel-based learning may be high when processing large-volume data. Thus, online versions of the kernel-based learning techniques are well motivated [16], [17], [18].

Recently, the benefits of the kernel-based learning was recognized in the DL context [19], [20]. Based on the matrix factorization perspective, individual factors were kernelized in [20], capturing correlations among the factors, and thus facilitating imputation and prediction tasks. Geared more to discriminative tasks, the kernel DL technique proposed in [19] capitalized on the kernel trick to learn the dictionary for the transformed feature vectors. Supervised kernel DL was proposed in [21], where a kernel-PCA-like algorithm was devised with the dictionary optimized in a space where the dependency between features and labels was maximized. All these algorithms are based on the *batch* processing of data.

The present work derives online kernel DL algorithms based on the stochastic approximation method. Both reconstructive as well as discriminative DL formulations are put forth, which are shown to be amenable to online processing. Advocating low-complexity per-step updates, first-order stochastic gradient descent variants are proposed. The idea is to minimize at each iteration an appropriate surrogate function, which upper-bounds the desired per-step cost function [22]. Although a rigorous convergence analysis is relegated to a journal version of this paper, the numerical results show fast convergence.

The rest of the paper is organized as follows. In Section II, the relevant DL formulations are presented. The online solutions are derived in Section III. The results from numerical tests are presented in Section IV. Conclusions and future research directions are provided in Section V.

II. Problem Formulation

A. Dictionary learning

Given N data vectors $\{\mathbf{x}_n \in \mathbb{R}^p\}_{n=1}^N$, DL aims at obtaining a dictionary $\tilde{\mathbf{D}} \in \mathbb{R}^{p\times K}$, which serves as a (possibly overcomplete) basis to explain the data in a compact fashion. That is, $\{\mathbf{x}_n\}$ are closely approximated by a linear combination of few columns (called *atoms*) $\{\tilde{\mathbf{d}}_k\}_{k=1}^K$ in $\tilde{\mathbf{D}}$. One way to accomplish this is to leverage a sparsity-promoting ℓ_1-norm regularizer augmented to a fitting term as in

$$\min_{\tilde{\mathbf{D}}\in\tilde{\mathcal{D}},\{\boldsymbol{\alpha}_n\in\mathbb{R}^K\}_{n=1}^N} \sum_{n=1}^N \left[\frac{1}{2}\|\mathbf{x}_n-\tilde{\mathbf{D}}\boldsymbol{\alpha}_n\|_2^2+\lambda\|\boldsymbol{\alpha}_n\|_1\right] \tag{1}$$

where $\tilde{\mathcal{D}} := \{\tilde{\mathbf{D}} := [\tilde{\mathbf{d}}_1, \tilde{\mathbf{d}}_2, \ldots, \tilde{\mathbf{d}}_K] \in \mathbb{R}^{p\times K} : \|\tilde{\mathbf{d}}_k\|_2 \leq 1\}$ is a compact set preventing the norms of the atoms from growing without bound, and λ is a weight for tuning the sparsity of $\{\boldsymbol{\alpha}_n\}$. Since (1) is a nonconvex optimization problem, globally optimal solutions are hard to come by. Greedy or alternating minimization approaches are often employed to obtain high-quality solutions [9], [10].

B. *Kernel DL*

There have been a few approaches for kernelizing DL [20], [21], [19]. Essentially, the idea is to perform DL after mapping the data to a high-dimensional feature space via a nonlinear mapping $\Phi : \mathbb{R}^p \to \mathbb{R}^P$. Upon defining

$$f(\mathbf{D}, \mathbf{x}) := \min_{\boldsymbol{\alpha}} \frac{1}{2}\|\Phi(\mathbf{x}) - \mathbf{D}\boldsymbol{\alpha}\|_F^2 + \lambda\|\boldsymbol{\alpha}\|_1 \tag{2}$$

the corresponding DL problem can be formulated as

$$\min_{\mathbf{D}\in\mathcal{D}, \{\boldsymbol{\alpha}_n \in \mathbb{R}^K\}} \sum_{n=1}^{N} f(\mathbf{D}, \mathbf{x}_n) \tag{3}$$

where $\mathbf{D} \in \mathcal{D} := \{\mathbf{D} := [\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_K] \subset \mathbb{R}^{P\times K} : \|\mathbf{d}_k\|_2 \leq 1\}$ is the dictionary in the high-dimensional feature space. One can show that confining the search of the dictionary in the form $\mathbf{D} = \mathbf{\Phi}(\mathbf{X}_N)\mathbf{\Omega}$, where $\Phi(\mathbf{X}_N) := [\Phi(\mathbf{x}_1), \Phi(\mathbf{x}_2), \ldots, \Phi(\mathbf{x}_N)]$ and $\mathbf{\Omega} \in \mathbb{R}^{N\times K}$ denotes the linear combination coefficients for K atoms, does not sacrifice optimality, in the spirit of the representer theorem [19]. Upon introducing a Mercer kernel $\kappa(\cdot,\cdot)$ with $\kappa(\mathbf{x}, \mathbf{y}) = \langle \Phi(\mathbf{x}), \Phi(\mathbf{y})\rangle$, and defining $\mathbf{X}_N := [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N] \in \mathbb{R}^{p\times N}$, $\boldsymbol{\kappa}(\mathbf{X}_N, \mathbf{x}_n) := [\kappa(\mathbf{x}_1, \mathbf{x}_n), \ldots, \kappa(\mathbf{x}_N, \mathbf{x}_n)]^T \in \mathbb{R}^N$ and $\boldsymbol{\mathcal{K}}(\mathbf{X}_N, \mathbf{X}_N) := [\boldsymbol{\kappa}(\mathbf{X}_N, \mathbf{x}_1), \ldots, \boldsymbol{\kappa}(\mathbf{X}_N, \mathbf{x}_N)] \in \mathbb{R}^{N\times N}$, it is readily verified that

$$\begin{aligned}\|\Phi(\mathbf{x}_n) - \Phi(\mathbf{X}_N)\mathbf{\Omega}\boldsymbol{\alpha}\|_2^2 &= \kappa(\mathbf{x}_n, \mathbf{x}_n) - 2\boldsymbol{\alpha}^T\mathbf{\Omega}^T\boldsymbol{\kappa}(\mathbf{X}_N, \mathbf{x}_n)\\ &\quad + \boldsymbol{\alpha}^T\mathbf{\Omega}^T\boldsymbol{\mathcal{K}}(\mathbf{X}_N, \mathbf{X}_N)\mathbf{\Omega}\boldsymbol{\alpha}\end{aligned} \tag{4}$$

which leads to a tractable objective for (3) even when P is large (and possibly infinite).

C. *Discriminative kernel DL*

Discriminative (or *supervised*, *task-driven*) DL is employed when the dictionary is used for prediction of labels y based on the features $\mathbf{x}$, rather than mere reconstruction of the data $\mathbf{x}$ [5]. Kernelized discriminative DL was studied in [21].

Given a dictionary $\mathbf{D}$ and the feature $\mathbf{x}$, the corresponding sparse coding vector $\boldsymbol{\alpha}^*(\mathbf{D}, \mathbf{x})$ is computed by

$$\boldsymbol{\alpha}^*(\mathbf{D}, \mathbf{x}) = \arg\min_{\boldsymbol{\alpha}} \frac{1}{2}\|\Phi(\mathbf{x}) - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + \lambda_1\|\boldsymbol{\alpha}\|_1 + \frac{\lambda_2}{2}\|\boldsymbol{\alpha}\|_2^2. \tag{5}$$

Discriminative kernel DL takes into account the task of fitting the label y for $\mathbf{x}$ based on the sparse coding $\boldsymbol{\alpha}^*(\mathbf{D}, \mathbf{x})$. Denote the regression coefficient vector as $\mathbf{w}$, and the fitting cost as $c(\mathbf{w}, \boldsymbol{\alpha}, y)$. For linear regression,

$$c(\mathbf{w}, \boldsymbol{\alpha}, y) = \frac{1}{2}(y - \mathbf{w}^T\boldsymbol{\alpha})^2 \tag{6}$$

can be used. Then, a reasonable formulation for discriminative kernel DL is

$$\min_{\mathbf{D}\in\mathcal{D}, \mathbf{w}\in\mathbb{R}^K} \sum_{n=1}^{N} \left[c(\mathbf{w}, \boldsymbol{\alpha}^*(\mathbf{D}, \mathbf{x}_n), y_n) + \frac{\nu}{2}\|\mathbf{w}\|_2^2\right] \tag{7}$$

where $\mathbf{D}$ and $\mathbf{w}$ are jointly optimized, and ν is a positive tuning parameter capturing the trade-off between trusting in the data and the prior information.

III. Online Solution

A. *Stochastic approximation approach*

To solve (3) and (7) even for Big Data applications, it is imperative to obtain low-complexity algorithms, capable of providing rough solutions quickly. Here, a stochastic approximation approach is taken [23]. The main challenges are the bilinear nonconvex nature of the problem, as well as the use of kernels. Stochastic approximation approaches were seen to be effective for (bilinear) nonconvex problems recently [10], [22], [24]. Online implementation of kernel-based learning has been investigated as well [17], [25]. Our goal is to combine these ideas to provide an efficient online implementation for kernel DL.

Stochastic approximation aims at optimizing an objective expressed in terms of an expectation. In our setup, the relevant problem for nondiscriminative kernel DL is

$$\min_{\mathbf{D}} \mathbb{E}_{\mathbf{x}}\left[f(\mathbf{D}, \mathbf{x}) + \frac{\mu}{2}\|\mathbf{D}\|_F^2\right] \tag{8}$$

where $\mathbb{E}_{\mathbf{x}}[\cdot]$ denotes the expectation with respect to $\mathbf{x}$. Note that instead of explicitly requiring $\mathbf{D} \in \mathcal{D}$, an additional regularization term $\frac{\mu}{2}\|\mathbf{D}\|_F^2$ has been added in (8) with a positive weight μ, which can still prevent $\mathbf{D}$ from growing unbounded, while being amenable to online solution. Upon defining

$$g(\mathbf{D}, \mathbf{w}, \mathbf{x}, y) = c(\mathbf{w}, \boldsymbol{\alpha}^*(\mathbf{D}, \mathbf{x}), y) + \frac{\nu}{2}\|\mathbf{w}\|_2^2 \tag{9}$$

the discriminative counterpart is given as

$$\min_{\mathbf{D}, \mathbf{w}} \mathbb{E}_{\mathbf{x}, y}\left[g(\mathbf{D}, \mathbf{w}, \mathbf{x}, y) + \frac{\mu}{2}\|\mathbf{D}\|_F^2\right]. \tag{10}$$

B. *Online kernel DL*

First, consider the reconstructive kernel DL problem (3) and its stochastic counterpart (8). To derive a stochastic gradient descent-type algorithm, a majorizing surrogate of $f(\mathbf{D}, \mathbf{x}) + \frac{\mu}{2}\|\mathbf{D}\|_F^2$ involving the first-order approximation must be constructed [22]. With f defined as in (2), the following serves the purpose.

$$\begin{aligned}\hat{f}(\mathbf{D}, \bar{\mathbf{D}}, \mathbf{x}) := \left\langle \mathbf{D} - \bar{\mathbf{D}}, \nabla_{\mathbf{D}}\left(\frac{1}{2}\|\Phi(\mathbf{x}) - \mathbf{D}\bar{\boldsymbol{\alpha}}\|_F^2\right)\Bigg|_{\mathbf{D}=\bar{\mathbf{D}}} + \mu\bar{\mathbf{D}}\right\rangle\\ + f(\bar{\mathbf{D}}, \mathbf{x}) + \frac{\mu}{2}\|\bar{\mathbf{D}}\|_F^2 + \frac{\eta}{2}\|\mathbf{D} - \bar{\mathbf{D}}\|_F^2\end{aligned} \tag{11}$$

where

$$\bar{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} \frac{1}{2}\|\Phi(\mathbf{x}) - \bar{\mathbf{D}}\boldsymbol{\alpha}\|_F^2 + \lambda\|\boldsymbol{\alpha}\|_1. \tag{12}$$

It can be verified that $\hat{f}(\bar{\mathbf{D}}, \bar{\mathbf{D}}, \mathbf{x}) = f(\bar{\mathbf{D}}, x)$ and $\hat{f}(\mathbf{D}, \bar{\mathbf{D}}, \mathbf{x})$ majorizes $f(\mathbf{D}, \mathbf{x})$ in the sense that $\hat{f}(\mathbf{D}, \bar{\mathbf{D}}, \mathbf{x}) \geq f(\mathbf{D}, \mathbf{x})$ for all $\mathbf{D} \in \mathcal{D}$ for some η.

The online algorithm updates the dictionary iteratively as each new datum arrives. Thus, at iteration n, dictionary $\mathbf{D}_n$ must depend only on $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$. Given dictionary $\mathbf{D}_{n-1}$ from the previous iteration $n-1$, and the new datum $\mathbf{x}_n$, the sparse coding step is performed as

$$\boldsymbol{\alpha}_n = \arg\min_{\boldsymbol{\alpha}} \frac{1}{2}\|\Phi(\mathbf{x}_n) - \mathbf{D}_{n-1}\boldsymbol{\alpha}\|_F^2 + \lambda\|\boldsymbol{\alpha}\|_1. \tag{13}$$

The update for $\mathbf{D}$ is carried out with the step size $\rho_n > 0$ as

$$\mathbf{D}_n = (1 - \mu\rho_n)\mathbf{D}_{n-1} + \rho_n(\Phi(\mathbf{x}_n) - \mathbf{D}_{n-1}\boldsymbol{\alpha}_n)\boldsymbol{\alpha}_n^T \tag{14}$$

```
Input: $\{\mathbf{x}_n\}_{n=1}^N$, $\lambda$, $\{\rho_n\}$
Output: $\boldsymbol{\Omega}_N$
Set $\boldsymbol{\Omega}_0 = \mathbf{0}$
For $n = 1, 2, \ldots, N$
    Perform sparse coding
        $\boldsymbol{\alpha}_n = \arg\min_{\boldsymbol{\alpha}} \frac{1}{2}\left[\kappa(\mathbf{x}_n, \mathbf{x}_n) - 2\boldsymbol{\alpha}^T\boldsymbol{\Omega}_{n-1}^T\boldsymbol{\kappa}(\mathbf{X}_{n-1}, \mathbf{x}_n)\right.$
        $\left.+\boldsymbol{\alpha}^T\boldsymbol{\Omega}_{n-1}^T\boldsymbol{\mathcal{K}}(\mathbf{X}_{n-1}, \mathbf{X}_{n-1})\boldsymbol{\Omega}_{n-1}\boldsymbol{\alpha}\right] + \lambda\|\boldsymbol{\alpha}\|_1$
    Perform dictionary update in (16)–(18).
Next $n$
```

TABLE I
ONLINE KERNEL DL ALGORITHM.

```
Input: $\{\mathbf{x}_n, y_n\}_{n=1}^N$, $\lambda_1$, $\lambda_2$, $\{\rho_n\}$
Output: $\boldsymbol{\Omega}_N$, $\mathbf{w}_N$
Set $\boldsymbol{\Omega}_0 = \mathbf{0}$ (or, initialize it based on reconstructive kernel DL)
For $n = 1, 2, \ldots, N$
    Perform sparse coding
        $\boldsymbol{\alpha}_n^* = \arg\min_{\boldsymbol{\alpha}} \frac{1}{2}\left[\kappa(\mathbf{x}_n, \mathbf{x}_n) - 2\boldsymbol{\alpha}^T\boldsymbol{\Omega}_{n-1}^T\boldsymbol{\kappa}(\mathbf{X}_{n-1}, \mathbf{x}_n)\right.$
        $\left.+\boldsymbol{\alpha}^T\boldsymbol{\Omega}_{n-1}^T\boldsymbol{\mathcal{K}}(\mathbf{X}_{n-1}, \mathbf{X}_{n-1})\boldsymbol{\Omega}_{n-1}\boldsymbol{\alpha}\right] + \lambda_1\|\boldsymbol{\alpha}\|_1 + \frac{\lambda_2}{2}\|\boldsymbol{\alpha}\|_2^2$
    Set $\Lambda = \{j \in \{1, 2, \ldots, K\} : \boldsymbol{\alpha}_n^*[j] \neq 0\}$
    Compute $\boldsymbol{\beta}_n^*$ by (33)–(34)
    Perform dictionary update in (30)–(32)
    Update $\mathbf{w}_n$ by (27)
Next $n$
```

TABLE II
ONLINE DISCRIMINATIVE KERNEL DL ALGORITHM.

which is essentially the stochastic gradient descent applied to $\hat{f}(\mathbf{D}, \mathbf{D}_{n-1}, \mathbf{x}_n)$. Now, substituting $\mathbf{D}_n = \boldsymbol{\Phi}(\mathbf{X}_N)\boldsymbol{\Omega}_n$ to (14), the update becomes

$$\begin{aligned}\Phi(\mathbf{X}_N)\boldsymbol{\Omega}_n = \Phi(\mathbf{X}_N)\boldsymbol{\Omega}_{n-1}[(1-\mu\rho_n)\mathbf{I} - \rho_n\boldsymbol{\alpha}_n\boldsymbol{\alpha}_n^T]&\\ + \rho_n\Phi(\mathbf{x}_n)\boldsymbol{\alpha}_n^T.&\end{aligned} \tag{15}$$

Suppose that $\boldsymbol{\Omega}_0 = \mathbf{0}$, and denote the rows $k, k+1, \ldots, m$ of $\boldsymbol{\Omega}_n$ as $\boldsymbol{\Omega}_n^{k:m}$. Then, the update (15) can be equivalently written as

$$\boldsymbol{\Omega}_n^{1:n-1} = \boldsymbol{\Omega}_{n-1}^{1:n-1}[(1-\mu\rho_n)\mathbf{I} - \rho_n\boldsymbol{\alpha}_n\boldsymbol{\alpha}_n^T] \tag{16}$$

$$\boldsymbol{\Omega}_n^{n:n} = \rho_n\boldsymbol{\alpha}_n^T \tag{17}$$

$$\boldsymbol{\Omega}_n^{n+1:N} = \mathbf{0}. \tag{18}$$

Note that as is clear through (18), the last $(N-n)$ rows of matrix $\boldsymbol{\Omega}_n$ do not need to be stored, nor the actual value of N need to be known a priori. Furthermore, the first term in the objective of (13) can be rewritten as

$$\begin{aligned}\|\Phi(\mathbf{x}_n) - \mathbf{D}_{n-1}\boldsymbol{\alpha}\|_F^2 = \kappa(\mathbf{x}_n, \mathbf{x}_n) - 2\boldsymbol{\alpha}^T\boldsymbol{\Omega}_{n-1}^T\boldsymbol{\kappa}(\mathbf{X}_{n-1}, \mathbf{x}_n)&\\ + \boldsymbol{\alpha}^T\boldsymbol{\Omega}_{n-1}^T\boldsymbol{\mathcal{K}}(\mathbf{X}_{n-1}, \mathbf{X}_{n-1})\boldsymbol{\Omega}_{n-1}\boldsymbol{\alpha}.&\end{aligned} \tag{19}$$

It is noted that even though update (15) for $\mathbf{D}_n$ depends on the entire dataset $\mathbf{X}_N$, the actual updates performed in practice for $\boldsymbol{\Omega}_n$ involve only the past and the present data $\mathbf{x}_1, \ldots, \mathbf{x}_n$ as desired [17]. The overall algorithm for online kernel DL is listed in Table I.

C. Online discriminative kernel DL

For the discriminative case, a first-order online algorithm was derived for the non-kernelized case in [5]. To extend this to kernel DL, an appropriate surrogate for $g + \frac{\mu}{2}\|\mathbf{D}\|_F^2$ can be constructed as

$$\begin{aligned}&\hat{g}(\mathbf{D}, \mathbf{w}, \bar{\mathbf{D}}, \bar{\mathbf{w}}, \mathbf{x}, y) := \langle \mathbf{D} - \bar{\mathbf{D}}, \nabla_{\mathbf{D}}g(\bar{\mathbf{D}}, \bar{\mathbf{w}}, \mathbf{x}, y) + \mu\bar{\mathbf{D}}\rangle\\ &+ \langle \mathbf{w} - \bar{\mathbf{w}}, \nabla_{\mathbf{w}}g(\bar{\mathbf{D}}, \bar{\mathbf{w}}, \mathbf{x}, y)\rangle + g(\bar{\mathbf{D}}, \bar{\mathbf{w}}, \mathbf{x}, y)\\ &+ \frac{\mu}{2}\|\bar{\mathbf{D}}\|_F^2 + \frac{\eta}{2}\|\mathbf{D} - \bar{\mathbf{D}}\|_F^2 + \frac{\eta}{2}\|\mathbf{w} - \bar{\mathbf{w}}\|_2^2\end{aligned} \tag{20}$$

where the partial gradients can be shown to be [5]

$$\nabla_{\mathbf{w}}g(\mathbf{D}, \mathbf{w}, \mathbf{x}, y) = \nabla_{\mathbf{w}}c(\mathbf{w}, \boldsymbol{\alpha}^*, y) + \nu\mathbf{w} \tag{21}$$

$$\nabla_{\mathbf{D}}g(\mathbf{D}, \mathbf{w}, \mathbf{x}, y) = -\mathbf{D}\boldsymbol{\beta}^*\boldsymbol{\alpha}^{*T} + (\Phi(\mathbf{x}) - \mathbf{D}\boldsymbol{\alpha}^*)\boldsymbol{\beta}^{*T} \tag{22}$$

where $\boldsymbol{\alpha}^*$ is the shorthand for $\boldsymbol{\alpha}^*(\mathbf{D}, \mathbf{x})$, and $\boldsymbol{\beta}^* \in \mathbb{R}^K$ is a vector defined as follows. Let us denote the entries of $\boldsymbol{\beta}^*$ whose indices are in set Λ as $\boldsymbol{\beta}_\Lambda^*$. Similarly, the columns of $\mathbf{D}$ with indices in Λ is represented as $\mathbf{D}_\Lambda$. Let Λ be the set of indices of nonzero entries in $\boldsymbol{\alpha}^*$, i.e.,

$$\Lambda := \{j \in \{1, 2, \ldots, K\} : \boldsymbol{\alpha}^*[j] \neq 0\}. \tag{23}$$

Then, $\boldsymbol{\beta}^*$ is defined as

$$\boldsymbol{\beta}_{\Lambda^c}^* = \mathbf{0} \tag{24}$$

$$\boldsymbol{\beta}_\Lambda^* = (\mathbf{D}_\Lambda^T\mathbf{D}_\Lambda + \lambda_2\mathbf{I})^{-1}\nabla_{\boldsymbol{\alpha}_\Lambda}c(\mathbf{w}, \boldsymbol{\alpha}^*, y). \tag{25}$$

Based on the surrogate function $\hat{g}$, one can derive an online algorithm to solve (10). Given $\mathbf{D}_{n-1}$ at iteration n, and an incoming datum $(\mathbf{x}_n, y_n)$, the sparse coding is done as [cf. (5)]

$$\boldsymbol{\alpha}_n^* = \boldsymbol{\alpha}^*(\mathbf{D}_{n-1}, \mathbf{x}_n). \tag{26}$$

Then, the updates for $\mathbf{w}_n$ and $\mathbf{D}_n$ are the stochastic gradient descent updates given by

$$\mathbf{w}_n = \mathbf{w}_{n-1} - \rho_n\nabla_{\mathbf{w}}g(\mathbf{D}_{n-1}, \mathbf{w}_{n-1}, \mathbf{x}_n, y_n) \tag{27}$$

$$\mathbf{D}_n = (1-\mu\rho_n)\mathbf{D}_{n-1} - \rho_n\nabla_{\mathbf{D}}g(\mathbf{D}_{n-1}, \mathbf{w}_{n-1}, \mathbf{x}_n, y_n). \tag{28}$$

For the quadratic cost in (6), (27) becomes

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \rho_n(y_n - \mathbf{w}_{n-1}^T\boldsymbol{\alpha}_n^*)\boldsymbol{\alpha}_n^* - \rho_n\nu\mathbf{w}_{n-1} \tag{29}$$

which coincides with the leaky least mean-square (LMS) update rule [26, Ch. 9]. To perform the dictionary update in (28), one again assumes $\boldsymbol{\Omega}_0 = \mathbf{0}$ and replaces $\mathbf{D}_n$ by $\Phi(\mathbf{X}_N)\boldsymbol{\Omega}_n$ to obtain

$$\boldsymbol{\Omega}_n^{1:n-1} = \boldsymbol{\Omega}_{n-1}^{1:n-1}[(1-\mu\rho_n)\mathbf{I} + \rho_n\boldsymbol{\beta}_n^*\boldsymbol{\alpha}_n^{*T} + \rho_n\boldsymbol{\alpha}_n^*\boldsymbol{\beta}_n^{*T}] \tag{30}$$

$$\boldsymbol{\Omega}_n^{n:n} = -\rho_n\boldsymbol{\beta}_n^{*T} \tag{31}$$

$$\boldsymbol{\Omega}_n^{n+1:N} = \mathbf{0} \tag{32}$$

where $\boldsymbol{\beta}_n^*$ is given from (24)–(25) as

$$(\boldsymbol{\beta}_n^*)_{\Lambda^c} = \mathbf{0} \tag{33}$$

$$\begin{aligned}(\boldsymbol{\beta}_n^*)_\Lambda = [(\boldsymbol{\Omega}_{n-1})_\Lambda^T\boldsymbol{\mathcal{K}}(\mathbf{X}_{n-1}, \mathbf{X}_{n-1})(\boldsymbol{\Omega}_{n-1})_\Lambda + \lambda_2\mathbf{I}]^{-1}&\\ \nabla_{\boldsymbol{\alpha}_\Lambda}c(\mathbf{w}_{n-1}, \boldsymbol{\alpha}_n^*, y_n).&\end{aligned} \tag{34}$$

The matrix inversion in (34) involves $O(K^3)$ complexity, which may be manageable for $K \ll N$. A lower complexity implementation of this step is left as a future work. The overall algorithm is listed in Table II.

IV. NUMERICAL TESTS

Numerical tests were performed to verify the performance of the proposed algorithms. Recognition of handwritten digits was considered using the USPS digit dataset [27]. The dataset contains 16-by-16-pixel images of handwritten digits in gray scale. The recognition task is a 10-class classification problem of recognizing digits from 0 to 9. Both reconstructive and discriminative DL formulations can be used for this as follows. For the reconstructive formulation in Section II-B, 10 separate dictionaries are trained using only the data belonging to the respective classes. For prediction, the index of the dictionary that yields the least fitting cost in (2) is returned. For the discriminative formulation in Section II-C, the quadratic fitting cost with linear classifier in (6) was adopted. Then the one-versus-all-others strategy was employed for the multi-class classification. That is, for the dictionary $\mathbf{D}_i$ and the classifier $\mathbf{w}_i$ for the i-th class, the

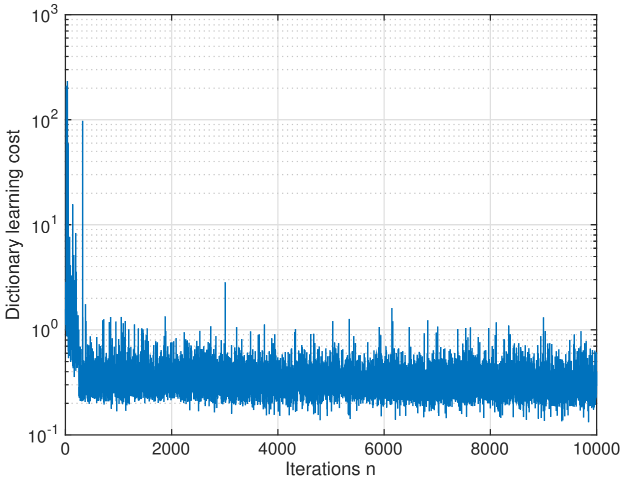

Fig. 1. Evolution of the reconstructive kernel DL cost

labels $y_n \in \{-1, 1\}$ for training were provided as $y_n = 1$ for the images in class i, and $y_n = -1$ for all other classes. The classification is then performed by choosing the index i with the prediction closest to 1. A 4-th order polynomial kernel was adopted.

Fig. 1 shows the evolution of the cost $f(\mathbf{D}_n, \mathbf{x}_n) + \frac{\mu}{2}\|\mathbf{D}_n\|_F^2$ for reconstructive DL versus the iteration count n for training of the dictionary for the images for digit 0. A total of 250 images were used per class, which were first passed sequentially and subsequently randomly sampled for $10,000$ iterations. The step size ρ_n was set to $\rho_n = 0.1/(1 + n/10^3)$ for $1 \leq n < 2,000$, and and then fixed at ρ_{2000} for $n \geq 2,000$. The value of λ was set to 0.01. It can be seen that the convergence is quite fast.

Fig. 2(a) depicts the cost $g(\mathbf{D}_n, \mathbf{w}_n, \mathbf{x}_n, y_n) + \frac{\mu}{2}\|\mathbf{D}_n\|_F^2$ for the discriminative kernel DL, again for digit 0. The evolution of the dictionary learning cost, which is the objective in (5), is plotted in Fig. 2(b). A total of $2,500$ data points were used, for a fair comparison with the reconstructive DL case, and the step size was set as $\rho_n = 0.1/(1 + n/10^3)$ for all n. The values of λ_1, λ_2, and ν were set to 0.01, 10^{-3}, and 0.01 respectively. It is seen that from Fig. 2(a) that the regression cost is converging, although the convergence is considerably slower than the reconstructive kernel DL counterpart. It is also observed from Fig. 2(b) that the reconstruction error of the data is gradually increasing while the regression cost is decreasing. That is, the dictionary is adapted for the discriminative task at hand at the sacrifice of the reconstruction ability.

A test dataset consisting of $4,649$ unseen images was used to assess the classification performance. Using the reconstructive kernel dictionaries, about 3.20% of the images were misclassified. When the discriminative kernel dictionaries were used, the misclassification rate was reduced to 2.86%. Although the difference is rather small, further performance improvement is expected by fine-tuning such parameters as λ, λ_1, λ_2, and ν.

V. Conclusions and Future Work

Online update algorithms for kernelized DL have been derived based on stochastic approximation, motivated by Big Data analytics. The developed stochastic gradient descent-type updates have low per-step computational complexity. Both basic reconstructive formulation as well as the discriminative formulations were proposed, which are amenable to online algorithms. Numerical tests involving the handwritten digit recognition task showed convergence of the proposed algorithms, and some performance advantage of the discriminative alternative. In addition to the rigorous convergence analysis, the future work will include incorporating accelerated gradient updates and pruning the data points participating in the dictionary for lower memory requirement. Also, extensive numerical tests will be performed for various applications.

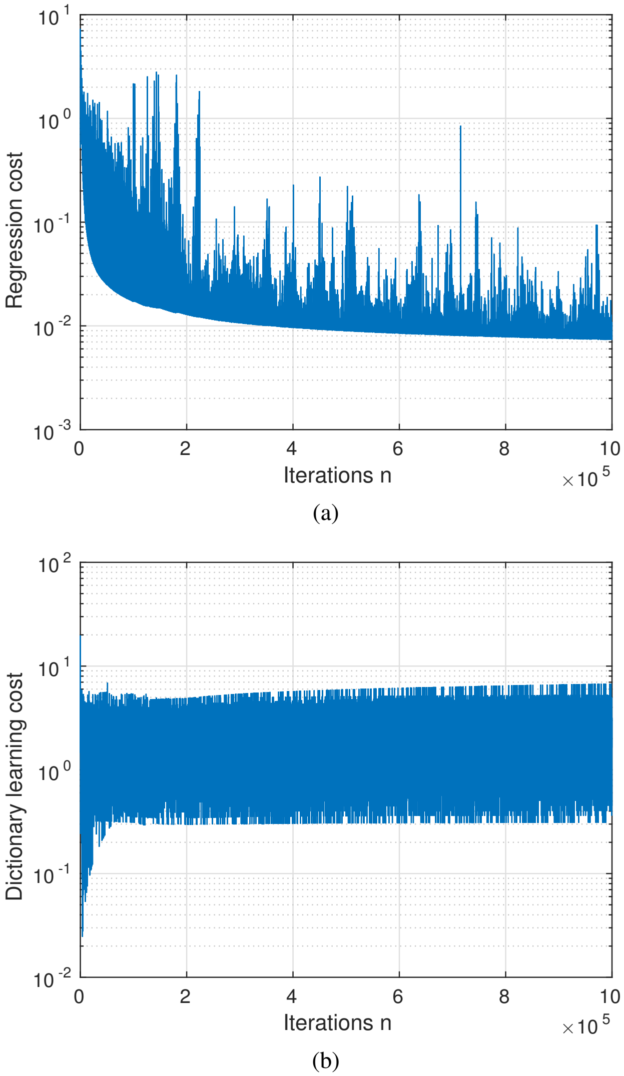

Fig. 2. Evolution of costs. (a) Regression cost. (b) Discriminative kernel DL cost.

Acknowledgement

The authors would like to thank the anonymous reviewers for their helpful comments.

References

[1] B. A. Olshausen and D. J. Field, “Sparse coding with an covercomplete basis set: A strategy employed by V1?” *Vision Research*, vol. 37, no. 23, pp. 3311–3325, Dec. 1997.

[2] M. Elad and M. Aharon, “Image denoising via sparse and redundant representations over learned dictionaries,” *IEEE Trans. Image Proc.*, vol. 15, no. 12, pp. 3736–3745, Dec. 2006.

[3] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Apr. 2009.

[4] I. Ramirez, P. Sprechmann, and G. Sapiro, "Classification and clustering via dictionary learning with structured incoherence and shared features," in *Proc. of the IEEE Conf. Comput. Vis. Pattern Recogn.*, San Francisco, CA, Jun. 2010, pp. 3501–3508.

[5] J. Mairal, F. Bach, and J. Ponce, "Task-driven dictionary learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 791–804, Apr. 2012.

[6] Q. Zhang and B. Li, "Discriminative K-SVD for dictionary learning in face recognition," in *Proc. of the IEEE Conf. Comput. Vis. Pattern Recogn.*, San Francisco, CA, Jun. 2010, pp. 2691–2698.

[7] J. Z. Kolter, S. Batra, and A. Y. Ng, "Energy disaggregation via discriminative sparse coding," in *Advances in Neural Information Processing Systems 23*, J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, Eds., 2010, pp. 1153–1161.

[8] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *Advances in Neural Information Processing Systems 19*, B. Schölkopf, J. Platt, and T. Hoffman, Eds. Cambridge, MA: MIT Press, 2007, pp. 801–808.

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

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

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

[12] S. Shalev-Shwartz, "Online learning and online convex optimization," *Foundations and Trends in Machine Learning*, vol. 4, no. 2, pp. 107–194, Mar. 2012.

[13] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. New York, NY: Cambridge University Press, 2004.

[14] B. Schölkopf, A. Smola, and K.-R. Müller, "Kernel principal component analysis," in *Proc. of the 7th Intl. Conf. Artificial Neural Net.*, Lausanne, Switzerland, Oct. 1997, pp. 583–588.

[15] I. S. Dhillon, Y. Guan, and B. Kulis, "Kernel k-means: spectral clustering and normalized cuts," in *Proc. of the 10th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*, Seatttle, WA, Aug. 2004, pp. 551–556.

[16] Y. Engel, S. Mannor, and R. Meir, "The kernel recursive least-squares algorithm," *IEEE Trans. Sig. Proc.*, vol. 52, no. 8, pp. 2275–2285, Aug. 2004.

[17] J. Kivinen, A. J. Smola, and R. C. Williamson, "Online learning with kernels," *IEEE Trans. Sig. Proc.*, vol. 52, no. 8, pp. 2165–2176, Aug. 2004.

[18] K. Slavakis, S. Theodoridis, and I. Yamada, "Online kernel-based classification using adaptive projection algorithms," *IEEE Trans. Sig. Proc.*, vol. 56, no. 7, pp. 2781–2796, Jun. 2008.

[19] H. V. Nguyen, V. M. Patel, N. M. Nasrabadi, and R. Chellappa, "Design of non-linear kernel dictionaries for object recognition," *IEEE Trans. Image Proc.*, vol. 22, no. 12, pp. 5123–5135, Dec. 2013.

[20] J. A. Bazerque and G. B. Giannakis, "Nonparametric basis pursuit via sparse kernel-based learning: A unifying view with advances in blind methods," *IEEE Sig. Proc. Mag.*, vol. 30, no. 4, pp. 112–125, Jul. 2013.

[21] M. J. Gangeh, A. Ghodsi, and M. S. Kamel, "Kernelized supervised dictionary learning," *IEEE Trans. Sig. Proc.*, vol. 61, no. 19, pp. 4753–4767, Oct. 2013.

[22] M. Razaviyayn, M. Sanjabi, and Z.-Q. Luo, "A stochastic successive minimization method for nonsmooth nonconvex optimization with applications to transceiver design in wireless communication networks," arXiv:1307.4457v2, Jul. 2013.

[23] H. J. Kushner and G. G. Yin, *Stochastic Approximation Algorithms and Applications*. New York, NY: Springer, 1997.

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

[25] T. Diethe and M. Girolami, "Online learning with (multiple) kernels: A review," *Neural Comput.*, vol. 25, no. 3, pp. 567–625, Mar. 2013.

[26] S. Haykin, *Adaptive Filter Theory*, 3rd ed. Upper Saddle River, NJ: Prentice-Hall, 1996.

[27] "USPS handwritten digit data." [Online]. Available: http://www.gaussianprocess.org/gpml/data/