Stochastic Optimization for Multiview Representation Learning using Partial Least Squares

Raman Arora ARORA@CS.JHU.EDU
Poorya Mianjy MIANJY@JHU.EDU
Department of Computer Science, Johns Hopkins University, Baltimore, MD 21218

Teodor V. Marinov T.V.MARINOV@SMS.ED.AC.UK
School of Informatics, University of Edinburgh, Edinburgh UK, EH8 9AB

Abstract

Partial Least Squares (PLS) is a ubiquitous statistical technique for bilinear factor analysis. It is used in many data analysis, machine learning, and information retrieval applications to model the covariance structure between a pair of data matrices. In this paper, we consider PLS for representation learning in a multiview setting where we have more than one view in data at training time. Furthermore, instead of framing PLS as a problem about a fixed given data set, we argue that PLS should be studied as a stochastic optimization problem, especially in a "big data" setting, with the goal of optimizing a population objective based on sample. This view suggests using Stochastic Approximation (SA) approaches, such as Stochastic Gradient Descent (SGD) and enables a rigorous analysis of their benefits. In this paper, we develop SA approaches to PLS and provide iteration complexity bounds for the proposed algorithms.

1. Introduction

Learning useful representations of data is one of the most basic challenges in machine learning. Unsupervised representation learning techniques capitalize on unlabeled data which is often cheap and abundant and sometimes virtually unlimited. The goal of these ubiquitous techniques is to learn a representation that reveals intrinsic low-dimensional structure in data and dis-entangles underlying factors of variation. This paper focuses on *new theory and methods* for large-scale *multiview representation learning*.

Proceedings of the 33^{rd} International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48.

Representation learning is typically phrased as a question about a fixed data set. For instance, partial least squares (PLS), a ubiquitous procedure in data science, is often posed as the following problem: given a data set of n samples of two set of variates (or views), $\mathrm{x} \in \mathbb{R}^{d_x}$ and $\mathrm{y} \in \mathbb{R}^{d_y}$, respectively, what is the k-dimensional subspace that captures most of the covariance between the two views. It is well known that this subspace is given by the leading k components of the singular value decomposition of the cross-covariance matrix $\mathbb{E}[\mathrm{x}\mathrm{y}^\top]$. And so, the study of computational approaches for PLS has mostly focused on methods for finding the singular value decomposition (SVD), or leading components of the SVD, for a given $d_x \times d_y$ matrix.

We argue that if we aim to capitalize on massive amounts of unlabeled data, we must also develop appropriate computational approaches and study them in the "data laden" regime. Accordingly, in this paper, we take a stochastic optimization view of representation learning rather than thinking of them as dimensionality reduction techniques for a given finite data set. In this paper, we argue that in the data laden ("big data") regime, representation learning techniques, including PLS, and other related problems, are better studied as stochastic optimization problems, where the goal is to optimize a "population objective" based on i.i.d. draws from the population. That is, in the case of PLS, we consider a setting in which we have some unknown source ("population") distribution $\mathcal{D}$ over $\mathbb{R}^{d_x} \times \mathbb{R}^{d_y}$, and the goal is to find the k-dimensional subspace maximizing the (uncentered) covariance of $\mathcal{D}$ inside the subspace based on i.i.d. samples from $\mathcal{D}$. The main point here is that the true objective is not how well the subspace captures the *sample* (i.e. the "training error"), but rather how well the subspace captures the underlying source distribution (i.e. the "generalization error"). Furthermore, we are not concerned here with capturing some "true" subspace, and so do not measure the angle to it, but rather at finding a "good" subspace, that is almost as good as the optimal one.

This allows our analysis to be independent of any eigengap (though if an eigengap does exist, a "good" subspace is also necessarily close to the "correct" one).

Of course, finding the subspace that maximizes the sample covariance, that is, that minimizes the "training error", is a very reasonable approach to PLS on the population. This is essentially an Empirical Risk Minimization (ERM) or Sample Average Approximation (SAA) approach. However, when comparing it to alternative, perhaps cheaper, computational approaches, we argue that one should not compare the error on the sample, but rather the population objective. Such a view can justify and favor computational approaches that are far from optimal on the sample, but are essentially as good as the ERM/SAA approach *on the population*. This formalizes and quantifies the intuition that there is no point in spending much effort in being exact on the sample, if in any case it just estimates the population.

Such a population-based view of optimization has recently been advocated in machine learning, and has been used to argue for crude stochastic approximation approaches (online-type methods) over sophisticated deterministic optimization of the empirical (training) objective (i.e. "batch" methods) (Bottou & Bousquet, 2007; Shalev-Shwartz & Srebro, 2008). A similar argument was also made in the context of stochastic optimization, where (Nemirovski et al., 2009) argue for stochastic approximation (SA) approaches over SAA. Accordingly, SA approaches, mostly variants of Stochastic Gradient Descent (SGD), are often the methods of choice for many learning problems, especially when very large data sets are available (Shalev-Shwartz et al., 2007; Collins et al., 2008; Shalev-Shwartz & Tewari, 2009).

Most work on stochastic approximation approaches in learning so far has been in the context of supervised learning. Here, we would like to take carry the same view over to unsupervised learning, and develop stochastic approximation approaches for partial least squares.

In our analysis, we focus on the "data laden" regime (Shalev-Shwartz & Srebro, 2008), where we have access to as many samples as we would like. That is, we focus on the runtime required to achieve a good (population) objective value, given access to as many samples as we want. Our runtime analysis therefore doesn't depend on the "data set size" (and certainly runtime does not increase if we have access to more data). This is the relevant regime in many modern "big data" problems, where we have access to effectively infinite amounts of data (e.g. images, speech recordings, text, videos, etc), and is especially true for unsupervised problems that use *unlabeled* data.

Furthermore, we focus on multiview representation learning where multiple "views" of the data, possibly from different measurement modalities are readily available. For instance, in web-page classification, one view may be the text of the page and the other the hyperlink structure; in automatic speech recognition, the views may be the acoustics and articulatory measurements such as tracks of flesh points (Bharadwaj et al., 2012). In such *multiview learning* problems, a common representation of the two views is provided by the shared semantic space.

A common approach to extracting this space is through canonical correlation analysis (CCA), which finds pairs of maximally correlated projections of the data in the two views. CCA has been successfully applied to various tasks in speech (Arora & Livescu, 2012; Bharadwaj et al., 2012; Arora & Livescu, 2013; Wang et al., 2015b), natural language processing (Haghighi et al., 2008; Dhillon et al., 2011; Wang et al., 2015a; Benton et al., 2016), and computer vision (Blaschko & Lampert, 2008; Hardoon et al., 2004). CCA admits a non-standard stochastic optimization problem where not only the objective but the constraints are stochastic, or equivalently the objective is a ratio of two expectations rather than an expectation of a loss function (Arora et al., 2012; Wang et al., 2015c; 2016). Consequently, the CCA objective does not decompose over the sample and designing stochastic approximation algorithms for CCA remains a challenging open problem.

A related approach to multiview representation learning is based on PLS which finds pairs of maximally *covarying* projections of the data in the two views. PLS has been applied to a host of problems in various areas including chemometrics, bioinformatics, medicine, social sciences, physiology (Rosipal & Krämer, 2006). Furthermore, assuming that the data is whitened in each view (after preprocessing), CCA reduces to PLS. More importantly, since the covariance objective decomposes over samples, PLS is amenable to standard stochastic optimization. However, the optimization problem associated with PLS is non-convex and stochastic approximation approaches recently proposed for PLS do not enjoy any theoretical guarantees (Arora et al., 2012).

In this paper, we present a convex relaxation of the stochastic optimization problem for PLS and study two stochastic approximation algorithms, which may be viewed as instances of stochastic mirror descent for different choices of potential functions. We provide rigorous theoretical guarantees for both of these algorithms in terms of the number of iterations needed to guarantee an ϵ-suboptimal solution to PLS. Furthermore, we empirically compare and evaluate the proposed algorithms with other standard stochastic approximation approaches including the stochastic power method and the incremental singular value decomposition (SVD) algorithm for PLS (Arora et al., 2012).

2. Partial Least Squares (PLS)

Formally, we can phrase partial least squares as the following stochastic optimization problem. Consider a joint distribution $\mathcal{D}$ over pairs of vectors $\mathrm{x} \in \mathbb{R}^{d_x}$ and $\mathrm{y} \in \mathbb{R}^{d_y}$. We represent the k-dimensional subspace that captures the maximal covariance in the distribution $\mathcal{D}$ by a pair of basis matrices $\mathrm{U} \in \mathbb{R}^{d_x \times k}$ and $\mathrm{V} \in \mathbb{R}^{d_y \times k}$, where the corresponding columns of U and V represent corresponding covarying directions. The PLS problem can now be expressed as finding U, V that:

$$\begin{aligned} \underset{\mathrm{U}\in\mathbb{R}^{d_x\times k},\ \mathrm{V}\in\mathbb{R}^{d_y\times k}}{\text{maximize}} \quad & \mathbb{E}_{\mathrm{x},\mathrm{y}}\left[\mathrm{x}^\top \mathrm{U}\mathrm{V}^\top \mathrm{y}\right] \\ \text{subject to} \quad & \mathrm{U}^\top\mathrm{U} = \mathrm{I}_k, \mathrm{V}^\top\mathrm{V} = \mathrm{I}_k \end{aligned} \tag{1}$$

The columns of U and V are the singular vectors of the covariance matrix $\Sigma_{\mathrm{xy}} = \mathbb{E}[\mathrm{x}\mathrm{y}^\top]$. The sample average approximation (SAA) or empirical risk minimization (ERM) approach to PLS amounts to finding top-k singular vectors of the empirical covariance matrix $\hat{\Sigma}_{\mathrm{xy}} = \frac{1}{T}\sum_{t=1}^{T} \mathrm{x}_t\mathrm{y}_t^\top$.

Like most other learning problems, PLS is an optimization of an expectation subject to fixed constraints, and is therefore amenable to a stochastic approximation approach. Several SA methods have been proposed for PLS (Arora et al., 2012); While these algorithms work great in practice, we need a better theoretical understanding of these algorithms, in terms of the total runtime needed to find an ϵ-suboptimal solution to Problem 1, which is the focus of this paper. We next briefly describe SA algorithms for PLS proposed by Arora et al. (2012).

Stochastic Power Method for PLS: The PLS objective can be written in terms of the trace operator as $\mathrm{Tr}\left(\mathrm{U}^\top\Sigma_{\mathrm{xy}}\mathrm{V}\right)$, which has gradients $\Sigma_{\mathrm{xy}}\mathrm{V}$ with respect to U and $\Sigma_{\mathrm{xy}}^\top\mathrm{U}$ with respect to V. This motivates the following simple alternating minimization approach based on stochastic gradient descent (Arora et al., 2012):

$$\begin{aligned} \mathrm{U}_t &= \mathscr{P}_{\text{orth}}\left(\mathrm{U}_{t-1} + \eta_t \mathrm{x}_t\mathrm{y}_t^\top\mathrm{V}_{t-1}\right) \\ \mathrm{V}_t &= \mathscr{P}_{\text{orth}}\left(\mathrm{V}_{t-1} + \eta_t \mathrm{y}_t\mathrm{x}_t^\top\mathrm{U}_{t-1}\right), \end{aligned}$$

where the operator $\mathscr{P}_{\text{orth}}(\cdot)$ projects onto the set of orthogonal matrices. The stochastic power method for PLS requires minimal computational resources – the space complexity of the resulting algorithm is the sum of the sizes of U and V: $O\left(k\left(d_x + d_y\right)\right)$. Ignoring the cost of the projection[1], the computational cost is dominated by the matrix-vector multiplications of the update equation which costs $O\left(k\left(d_x + d_y\right)\right)$ operations per iteration. However, there are no theoretical guarantees on the convergence of the stochastic power method for PLS.

Incremental PLS: Perhaps the most straightforward approach to PLS in the stochastic setting is motivated by the follow the leader algorithm, where, at every step t we take $\widehat{\mathrm{C}}_{\mathrm{xy}}^{(t+1)} = \frac{1}{(t+1)}(t\widehat{\mathrm{C}}_{\mathrm{xy}}^{(t)} + \mathrm{x}_t\mathrm{y}_t^\top)$ to be the empirical cross-covariance matrix of all the samples seen so far, calculate its singular value decomposition (SVD), and compute the top-k left and right singular vectors. This essentially requires finding the ERM solution at each iteration, and is far from being practical – it requires $O(d_x d_y)$ operations just to update the covariance matrix. However, we can instead perform an approximate ERM at much lower computational cost by explicitly constraining the rank of the empirical cross-covariance matrix, and performing a rank-one update incrementally, as each new sample is observed. This results in the following update: $\widehat{\mathrm{C}}_{\mathrm{xy}}^{(t+1)} = \frac{1}{(t+1)}\mathscr{P}_{\text{rank-}k}(t\widehat{\mathrm{C}}_{\mathrm{xy}}^{(t)} + \mathrm{x}_t\mathrm{y}_t^\top)$. Furthermore, this update can be implemented efficiently by simply storing and manipulating the singular vectors and singular values of a rank-k matrix requiring space complexity of $O\left(k\left(d_x + d_y\right)\right)$ and computational cost of $O\left(k^2\left(d_x + d_y\right)\right)$ (Arora et al., 2012).

Online PLS: Finally, the online randomized PCA algorithm of Warmuth & Kuzmin (2008) can be adapted to PLS by reducing the PLS problem to an equivalent PCA-like problem (Arora et al., 2012), the goal of which is to find the top eigenvectors of a self-adjoint dilation of the matrix Σ_{xy} (Tropp, 2011). We formally study the online PLS algorithm as an instance of matrix exponentiated gradient (MEG) algorithm in Section 3.2.

3. Stochastic approximation for PLS

We present two algorithms for stochastic optimization of PLS, the first one consists of additive updates and we refer to it as Matrix Stochastic Gradient (MSG), the second one is based on multiplicative updates and is referred to as Matrix Exponential Gradient (MEG). These algorithms can be viewed as instances of stochastic mirror descent with different choices of potential function, Frobenius norm for MSG and von Neumann entropy for MEG. Both algorithms can be justified in an online setup where we process a single data point at each iteration as follows. Given an estimate of the parameter matrix M_{t-1} representing the maximally covarying subspace, computed on previously observed data, and a new data pair $(\mathrm{x}_t, \mathrm{y}_t)$ at time t, we update the parameter matrix by solving the following optimization problem: $\mathrm{M}_t = \arg\min_{\mathrm{M}} d(\mathrm{M}, \mathrm{M}_{t-1}) + \eta\ell(\mathrm{M}, \mathrm{x}_t, \mathrm{y}_t)$, where $d(\mathrm{M}, \mathrm{M}_{t-1})$ is a divergence function that encourages next iterate to stay close to the current iterate while $\ell(\mathrm{M}, \mathrm{x}_t, \mathrm{y}_t)$ is the loss on the current instance pair. The parameter η provides the trade-off between the instantaneous loss function and our summary of the past observations encoded as the parameter matrix. The divergence functions for MSG and MEG are generated by Frobenius norm and von Neumann entropy, respectively, and the loss function for both

[1] Performed for purely numerical reasons, and only very infrequently (Arora et al., 2012).

MSG and MEG is $-\langle \mathrm{M}, \mathrm{C}_t \rangle = -\mathrm{Tr}\left(\mathrm{M}^\top \mathrm{C}_t\right)$ where C_t corresponds to the instantaneous estimate of the covariance matrix for MSG and its dilation for MEG.

3.1. Matrix Stochastic Gradient

The PLS optimization problem in equation (1) is a non-convex optimization problem; both the objective and constraints are non-convex. In this section, we consider a convex relaxation of Problem 1 and analyze the SGD algorithm for the same. We first introduce a simple variable substitution, $\mathrm{M} = \mathrm{U}\mathrm{V}^\top$, which allows us to reformulate the problem as:

$$\begin{aligned} \underset{\mathrm{M}\in\mathbb{R}^{d_x\times d_y}}{\text{maximize}} \quad & \mathbb{E}_{\mathrm{x},\mathrm{y}}\left[\mathrm{x}^\top \mathrm{M}\mathrm{y}\right] \\ \text{subject to} \quad & \sigma_i(\mathrm{M}) \in \{0,1\},\ i \in [d'] \\ & \|\mathrm{M}\|_* = \sum_{i=1}^{d'} \sigma_i(\mathrm{M}) = k, \end{aligned} \tag{2}$$

where $d' = \min\{d_x, d_y\}$, $\sigma_i(\mathrm{M})$ are the singular values of matrix M with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{d'}$, and $\|\mathrm{M}\|_*$ denotes the nuclear norm of matrix M. We now have a convex (linear, in fact) objective, but the constraint set is still non-convex. We consider the following convex relaxation of the first set of constraints: $\sigma_i(\mathrm{M}) \in [0,1],\ i \in [d']$, or equivalently, $\sigma_1(\mathrm{M}) = \|\mathrm{M}\|_2 \leq 1$, where $\|\mathrm{M}\|_2$ denotes the spectral norm of matrix M. Furthermore, we relax the non-linear equality constraints to give us the following convex program:

$$\begin{aligned} \underset{\mathrm{M}\in\mathbb{R}^{d_x\times d_y}}{\text{maximize}} \quad & \mathbb{E}_{\mathrm{x},\mathrm{y}}\left[\mathrm{x}^\top \mathrm{M}\mathrm{y}\right] \\ \text{subject to} \quad & \|\mathrm{M}\|_2 \leq 1, \|\mathrm{M}\|_* \leq k. \end{aligned} \tag{3}$$

Since the objective $\mathbb{E}_{\mathrm{x},\mathrm{y}}\left[\mathrm{x}^\top \mathrm{M}\mathrm{y}\right]$ is linear in M, the maximum will always occur at the boundary of the convex body of the feasible region given by constraints in (3), so that at any optimum we have $\|\mathrm{M}\|_2 = 1$ and $\|\mathrm{M}\|_* = k$.

This problem can now be solved using projected stochastic gradient method with the following update rule:

$$\mathrm{M}_t = \mathscr{P}_F(\mathrm{M}_{t-1} + \eta_t \mathrm{x}_t \mathrm{y}_t^\top)$$

where $\mathscr{P}_F$ projects onto the convex feasible set of Problem (3) with respect to the Frobenius norm, i.e. $\mathscr{P}_F(\mathrm{M}')$ solves:

$$\begin{aligned} \underset{\mathrm{M}}{\text{minimize}} \quad & \left\|\mathrm{M} - \mathrm{M}'\right\|_F^2 \\ \text{subject to} \quad & \|\mathrm{M}\|_2 \leq 1, \|\mathrm{M}\|_* \leq k. \end{aligned} \tag{4}$$

If the solution to the relaxed Problem 3 is not rank-k, and hence not a feasible point of Problem 2, we can sample a rank-k solution from it, which gives the same objective in expectation. Algorithm 2 of Warmuth & Kuzmin (2008) describes an efficient procedure to express any solution of Problem 3 as a convex combination of at most d feasible solutions of Problem 2, from which we can sample a rank-k solution. We refer to this procedure as `rounding`. The pseudocode for MSG is given in Algorithm 1.

Algorithm 1 Matrix Stochastic Gradient

Input: $\mathrm{M}_0, \{(\mathrm{x}_t, \mathrm{y}_t)\}_{t=1}^T, \eta$
Output: $\bar{\mathrm{M}}$

1: **for** $t = 1, \cdots, T$ **do**
2: $\quad \widehat{\mathrm{M}}_t = \mathrm{M}_{t-1} + \eta \mathrm{x}_t \mathrm{y}_t^\top$
3: $\quad \mathrm{M}_t = \mathscr{P}_F(\widehat{\mathrm{M}}_t)$ $\quad$ given by lemma 3.1
4: **end for**
5: $\bar{\mathrm{M}} = \frac{1}{T}\sum_{t=1}^T \mathrm{M}_{t-1}$
6: $\tilde{\mathrm{M}} = \mathtt{rounding}(\bar{\mathrm{M}})$

Efficient Implementation and Projection: A naive implementation of the MSG update requires $O(d_x d_y)$ memory and $O\left(d_x^2 d_y\right)$ operations per iteration only for the update. We follow (Arora et al., 2012) to perform updates efficiently by maintaining an up-to-date SVD decomposition of $\mathrm{M}^{(t)}$, so that the rank-1 update at each iterate costs $O\left(k^2(d_x + d_y)\right)$. Furthermore, we show in Lemma 3.1, that the projection step can be performed efficiently, since the projection operates *only* on the singular values, leaving singular vectors intact. Following Algorithm 2 in (Arora et al., 2013), we can perform the projection step in $O(k \log(k))$ operations, using a shift-and-clip procedure, which involves finding the smallest ν (see Lemma 3.1) such that after decreasing the singular values by ν and clipping them to $[0,1]$, the result is feasible. The overall computational complexity of the proposed algorithm is linear in input dimension $(d_x + d_y)$.

Lemma 3.1. *Let $\mathrm{M}' \in \mathbb{R}^{d_x \times d_y}$, $d_x \leq d_y$, be a real matrix, with singular value decomposition $\{\sigma'_i, \mathrm{u}_i, \mathrm{v}_i\}_{i=1}^{d_x}$ where singular values are sorted in descending order, and u_i and v_i are the corresponding left and right singular vectors. Let $\mathrm{M} = \mathscr{P}_F(\mathrm{M}')$ be a projection of M' onto the feasible region of Problem (3) with respect to the Frobenius norm. Then, M is the unique feasible matrix which has the same set of singular vectors as M', with the associated singular values $\sigma_1, \ldots, \sigma_{d_x}$ satisfying:*

$$\sigma_i = \max(\min(\sigma'_i - \nu, 1), 0)$$

with $\nu \in \mathbb{R}_{\geq 0}$ the smallest shift such that $\sum_{i=1}^{d_x} \sigma_i \leq k$.

Proof. The objective in Problem (4) is strongly convex, and the feasible set is convex, so the problem has a unique solution. First, we show that the singular vectors of M are the same as that of M'. Note that

$$\left\|\mathrm{M} - \mathrm{M}'\right\|_F^2 = \|\mathrm{M}\|_F^2 + \left\|\mathrm{M}'\right\|_F^2 - 2\,\mathrm{Tr}\left(\mathrm{M}^\top \mathrm{M}'\right).$$

Let $\mathrm{M} = \mathrm{U}\Sigma\mathrm{V}^\top$. By a change of variable, we can state Problem 4 in terms of $\mathrm{U}, \Sigma, \mathrm{V}$:

$$\begin{aligned} \underset{\Sigma}{\text{minimize}} \quad & \|\Sigma\|_F^2 - \sup_{\mathrm{U},\mathrm{V}} 2\,\mathrm{Tr}\big(\mathrm{M}'\mathrm{U}\Sigma\mathrm{V}^\top\big) \\ \text{subject to} \quad & \|\Sigma\|_2 \le 1, \|\Sigma\|_* \le k. \end{aligned} \tag{5}$$

Now, for any square real matrix $\mathrm{A} \in \mathbb{R}^{d_x \times d_x}$, and for any $i \in [d_x]$, we have that ((Horn & Johnson, 1991), pp. 151)

$$\sigma_k(\mathrm{A}) \ge \lambda_k\left(\frac{\mathrm{A}+\mathrm{A}^\top}{2}\right),$$

where $\sigma_k(\cdot)$ and $\lambda_k(\cdot)$ denote the k^{th} largest singular and eigenvalue respectively. Therefore,

$$\mathrm{Tr}(\mathrm{A}) = \mathrm{Tr}\left(\frac{\mathrm{A}+\mathrm{A}^\top}{2}\right) = \sum_{i=1}^{d_x} \lambda_i\left(\frac{\mathrm{A}+\mathrm{A}^\top}{2}\right) \le \sum_{i=1}^{d_x} \sigma_i(\mathrm{A}).$$

This gives $\mathrm{Tr}(\mathrm{M}^\top\mathrm{M}') \le \sum_{i=1}^{d_x} \sigma_i(\mathrm{M}^\top\mathrm{M}')$. We next use the following inequality on the sum of the singular values of product of matrices ((Horn & Johnson, 1991), pp. 176-177):

$$\sum_{i=1}^{k} \sigma_i(\mathrm{M}^\top\mathrm{M}') \le \sum_{i=1}^{k} \sigma_i(\mathrm{M})\sigma_i(\mathrm{M}'),\ k \in [d_x], \tag{6}$$

where the equality holds when M and M′ have the same set of singular vectors. Therefore, we can see that $\left\|\mathrm{M}-\mathrm{M}'\right\|_F^2$ is minimized when M and M′ have the same set of singular vectors and Problem 4 can be reduced to solving the following problem:

$$\begin{aligned} \underset{\boldsymbol{\sigma}}{\text{minimize}} \quad & \frac{1}{2}\left\|\boldsymbol{\sigma}-\boldsymbol{\sigma}'\right\|^2 \\ \text{subject to} \quad & 0 \le \sigma_i \le 1,\ i \in [d_x], \sum_{i=1}^{d_x} \sigma_i \le k. \end{aligned} \tag{7}$$

To solve this problem, we first form the Lagrangian:

$$\begin{aligned} \mathcal{L}(\boldsymbol{\sigma}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \nu) = \frac{1}{2}\left\|\boldsymbol{\sigma}-\boldsymbol{\sigma}'\right\|^2 + \boldsymbol{\alpha}^\top(\boldsymbol{\sigma}-\mathbf{1}) \\ - \boldsymbol{\beta}^\top\boldsymbol{\sigma} + \nu\left(\sum_{i=1}^{d_x}\sigma_i - k\right), \end{aligned}$$

where $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\nu}$ are (non-negative) dual variables. KKT conditions require that the derivative of the Lagrangian with respect to $\boldsymbol{\sigma}$ vanishes at the optimum. Hence,

$$\sigma_i - \sigma_i' + \alpha_i - \beta_i + \nu = 0,\ i \in [d_x]. \tag{8}$$

Further, by complementary slackness, we have

$$\nu\left(\sum_{i=1}^{d_x}\sigma_i - k\right) = \alpha_i(\sigma_i - 1) = \beta_i\sigma_i = 0.$$

Complementary slackness together with equation (8) implies that if $0 \le \sigma_i' - \nu \le 1$, then $\alpha_i = \beta_i = 0$ and $\sigma_i = \sigma_i' - \nu$, otherwise, α_i and β_i will "clip" σ_i to the active constraint:

$$\sigma_i = \max(\min(\sigma_i' - \nu, 1), 0).$$

Primal feasibility with respect to the constraint $\sum_{i=1}^{d_x}\sigma_i \le k$ together with the complementary slackness $\nu(\sum_{i=1}^{d_x}\sigma_i - k) = 0$ gives that ν is non-zero (positive) only if the constraint is violated, where ν "shifts" the solution towards the active constraint $\sum_{i=1}^{d_x}\sigma_i = k$, completing the proof. □

Convergence rate: Our first main result gives a bound on the ϵ-suboptimality of the MSG on the PLS objective in terms of the number of iterations of the algorithm. We assume without loss of generality that the data are scaled in such a way that $\mathbb{E}[\|\mathrm{x}\|^2] \le 1$ and $\mathbb{E}[\|\mathrm{y}\|^2] \le 1$.

Theorem 3.2. *After T iterations of Algorithm 1 with step size $\eta = \sqrt{\frac{k}{T}}$, and starting at $\mathrm{M}^{(0)} = 0$,*

$$\mathbb{E}\left[\mathbb{E}_{\mathrm{x},\mathrm{y}}\left[\mathrm{x}^\top\tilde{\mathrm{M}}\mathrm{y}\right]\right] \ge \mathbb{E}_{\mathrm{x},\mathrm{y}}\left[\mathrm{x}^\top\mathrm{M}^*\mathrm{y}\right] - \frac{1}{2}\sqrt{\frac{k}{T}}, \tag{9}$$

where the expectation is w.r.t. the i.i.d. samples $\{(\mathrm{x}_t, \mathrm{y}_t)\}_{t=1}^T \sim \mathcal{D}$ and the `rounding`*, and M^* is the optimum of (2).*

Proof. Standard SGD analysis of (Nemirovsky & Yudin, 1983) yields that

$$\mathbb{E}\left[\mathrm{x}^\top\mathrm{M}^*\mathrm{y} - \mathrm{x}^\top\bar{\mathrm{M}}\mathrm{y}\right] \le \frac{\eta}{2}\mathbb{E}_{\mathrm{x},\mathrm{y}}\left[\|g\|_F^2\right] + \frac{\left\|\mathrm{M}^* - \mathrm{M}^{(0)}\right\|_F^2}{2\eta T},$$

where $g = \mathrm{x}\mathrm{y}^\top$ is the gradient of the PLS objective. Now, $\mathbb{E}_{\mathrm{x},\mathrm{y}}\left[\|g\|_F^2\right] \le \mathbb{E}_{\mathrm{x},\mathrm{y}}\left[\|\mathrm{x}\|^2\|\mathrm{y}\|^2\right] \le 1$ and $\left\|\mathrm{M}^* - \mathrm{M}^{(0)}\right\|_F^2 = \|\mathrm{M}^*\|_F^2 = k$. In the last equality, we used the fact that M^* has k singular values of value 1 each, and hence $\|\mathrm{M}^*\|_F = \sqrt{k}$. □

Equivalently, the number of iterations of MSG to achieve an ϵ-suboptimal solution to the PLS objective is $O\left(\frac{k}{\epsilon^2}\right)$.

3.2. Matrix Exponentiated Gradient

In this section, we consider an alternate formulation for the PLS problem based on a self-adjoint dilation and symmetrization of the cross-covariance matrix. This formulation leads to a multiplicative update which we refer to as matrix exponentiated gradient or MEG. MEG is essentially the same as the online PLS algorithm presented in (Arora et al., 2012). Here, we motivate MEG as an instance of SGD for a convex relaxation of an alternate formulation of PLS, and give a convergence analysis.

Consider a self-adjoint dilation (Tropp, 2011) of the empirical covariance matrix $\mathrm{x}_t\mathrm{y}_t^\top$ based on a single sample,

$$\mathrm{C}_t := \begin{pmatrix} 0 & \mathrm{x}_t\mathrm{y}_t^\top \\ \mathrm{y}_t\mathrm{x}_t^\top & 0 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} \mathrm{x}_t \\ \mathrm{y}_t \end{pmatrix}\begin{pmatrix} \mathrm{x}_t \\ \mathrm{y}_t \end{pmatrix}^\top - \frac{1}{2}\begin{pmatrix} \mathrm{x}_t \\ -\mathrm{y}_t \end{pmatrix}\begin{pmatrix} \mathrm{x}_t \\ -\mathrm{y}_t \end{pmatrix}^\top$$

Then $\mathrm{C} := \mathbb{E}_{\mathrm{x}_t,\mathrm{y}_t}[\mathrm{C}_t]$ is the symmetrization of the population cross-covariance matrix $\Sigma_{\mathrm{xy}} = \mathbb{E}_{\mathrm{x},\mathrm{y}}\left[\mathrm{x}\mathrm{y}^T\right]$. Let

$W\Lambda W^\top$ be the rank-k eigendecomposition of matrix C and let $USV^\top$ be the truncated rank-k SVD of Σ_{xy}. Then, assuming that Σ_{xy} does not have repeated nonzero singular values, the positive eigenvalues of C are precisely the singular values of Σ_{xy} and the optimal value of the objective in Problem (1) is obtained by taking the sum of the top-k eigenvalues of C. Furthermore, the solution to the stochastic PLS Problem (1), i.e. the top-k left and right singular vectors of Σ_{xy}, are embedded in the top-k eigenvectors of matrix C as follows:

$$W := \frac{1}{\sqrt{2}} \begin{pmatrix} U & U \\ V & -V \end{pmatrix}. \tag{10}$$

In other words, if we take the first k columns of W, i.e. the top-k eigenvectors of C, then the first d_x rows correspond to the left singular vectors of Σ_{xy} and the next d_y rows correspond to the right singular vectors of Σ_{xy}. Defining $d = d_x + d_y$, PLS can be formulated as

$$\begin{array}{ll} \underset{W \in \mathbb{R}^{d\times k}}{\text{maximize}} & \text{Tr}(WW^\top C) \\ \text{subject to} & W^\top W = I \end{array}, \tag{11}$$

While the problem above is not convex, it admits a simple convex relaxation. Substitute $M = \frac{1}{d-k}(I - WW^\top)$, to get

$$\begin{array}{ll} \underset{M \in \mathbb{R}^{d\times d}}{\text{minimize}} & \mathbb{E}_{x_t, y_t}\left[\text{Tr}(MC_t)\right] \\ \text{subject to} & M \succeq 0, \|M\|_2 \leq \frac{1}{d-k}, \text{Tr}(M) = 1 \end{array} \tag{12}$$

Following Warmuth & Kuzmin (2008) we consider a stochastic mirror descent algorithm for Problem (12) with quantum relative entropy as the Bregman divergence. This results in the following multiplicative matrix exponentiated gradient updates:

$$\widehat{M}_t = \frac{e^{\log(M_{t-1}) - \eta C_t}}{\text{Tr}\left(e^{\log(M_{t-1}) - \eta C_t}\right)}, \; M_t = \mathscr{P}_{RE}\left(\widehat{M}_t\right) \tag{13}$$

where e^X and $\log(X)$ denote matrix exponential and matrix logarithm, respectively, and $\mathscr{P}_{RE}(\cdot)$ is the Bregman projection onto the convex set of constraints w.r.t. the quantum relative entropy. Algorithm 4 of (Warmuth & Kuzmin, 2008) gives an efficient procedure for this projection. Pseudocode for MEG is given in Algorithm 2.

Efficient Implementation and Projection: The capping step in Algorithm 2 takes $O(d)$ time (Warmuth & Kuzmin, 2008) and the rank-2 update, assuming we keep an up-to-date SVD, takes $O\left(k^2 d\right)$ time (Arora et al., 2012). We analyze the MEG algorithm for average of the iterates $\tilde{M} = \texttt{rounding}(\frac{1}{T}\sum_{t=1}^T M_t)$. We get same rates if we sample uniformly one of the iterates. In practice, we use the parameter matrix from the final iterate.

Convergence rate: The standard analysis of Algorithm 2 would require stochastic gradients C_t to be positive semi-definite, which does not hold in general. However, it is easy to check that the update in (13) is invariant to adding a scaled identity matrix to C_t, i.e.

$$\frac{e^{\log(M_{t-1}) - \eta C_t + \alpha I}}{\text{Tr}\left(e^{\log(M_{t-1}) - \eta C_t + \alpha I}\right)} = \frac{e^{\log(M_{t-1}) - \eta C_t}}{\text{Tr}\left(e^{\log(M_{t-1}) - \eta C_t}\right)}.$$

In other words, we can always replace C_t by its "spectrum-shifted" version $\tilde{C}_t = C_t - \lambda_{min}(C_t)I$ such that $0 \preceq \tilde{C}_t \preceq rI$. We next present our analysis of the MEG algorithm.

Algorithm 2 Matrix Exponentiated Gradient

Input: $M_0, \{(x_t, y_t)\}_{t=1}^T, \eta$
Output: $\tilde{M}$

1: **for** $t = 1, \cdots, T$ **do**
2: $\quad C_t = \begin{pmatrix} 0 & x_t y_t^\top \\ y_t x_t^\top & 0 \end{pmatrix}$
3: $\quad \widehat{M}_t = \frac{e^{\log(M_{t-1}) - \eta C_t}}{\text{Tr}\left(e^{\log(M_{t-1}) - \eta C_t}\right)}$
4: $\quad M_t = \mathscr{P}_{RE}(\widehat{M}_t)$ given by Algorithm 4 (Warmuth & Kuzmin, 2008)
5: **end for**
6: $\bar{M} = \frac{1}{T}\sum_{t=1}^T M_{t-1}$
7: $\tilde{M} = \texttt{rounding}(\bar{M})$

Theorem 3.3. *After T iterations of Algorithm 2 starting from $M_0 = \frac{1}{d}I$ with a step size of $\eta = \frac{1}{r}\log\left(1 + \sqrt{\frac{2r\log(d)}{LT}}\right)$, we have that*

$$\mathbb{E}\left[\text{Tr}\left(\tilde{M}C\right)\right] - \text{Tr}\left(M^* C\right) \leq \sqrt{\frac{2rL\log(d)}{T}} + \frac{r\log(d)}{T}$$

where the expectation is w.r.t. the i.i.d. samples $\{(x_t, y_t)\}_{t=1}^T \sim \mathscr{D}$ and the `rounding`*, M^* is the optimum of (12), and $Tr(M^* C_t) \leq L$ for all $t \in [T]$, and r is chosen such that $0 \preceq \tilde{C}_t \preceq rI$.*

Proof. We will need the following lemma which follows from Theorem 2 of Warmuth & Kuzmin (2006); For completeness sake, we include a proof in the supplement.

Lemma 3.4. *With the same assumptions as in Theorem 3.3, the following regret bound holds*

$$\sum_{t=1}^T \text{Tr}\left(M_{t-1} C_t\right) - \sum_{t=1}^T \text{Tr}\left(M^* C_t\right) \leq \sqrt{2rLT\log(d)} + r\log(d) \tag{14}$$

We now take the expectation on both sides of (14) with respect to $\{x_t, y_t\}_{t=1}^T$. For a more compact notation, let $\mathbb{E}_{(\tau)}[\cdot]$ and $\mathbb{E}_\tau[\cdot]$ denote the expected value with respect to $\{x_t, y_t\}_{t=1}^\tau$ and $\{x_\tau, y_\tau\}$, respectively. Then, for the second term on the left hand side, we have

$$\mathbb{E}\left[\sum_{t=1}^T \text{Tr}\left(M^* C_t\right)\right] = T \cdot \text{Tr}\left(M^* C\right) \tag{15}$$

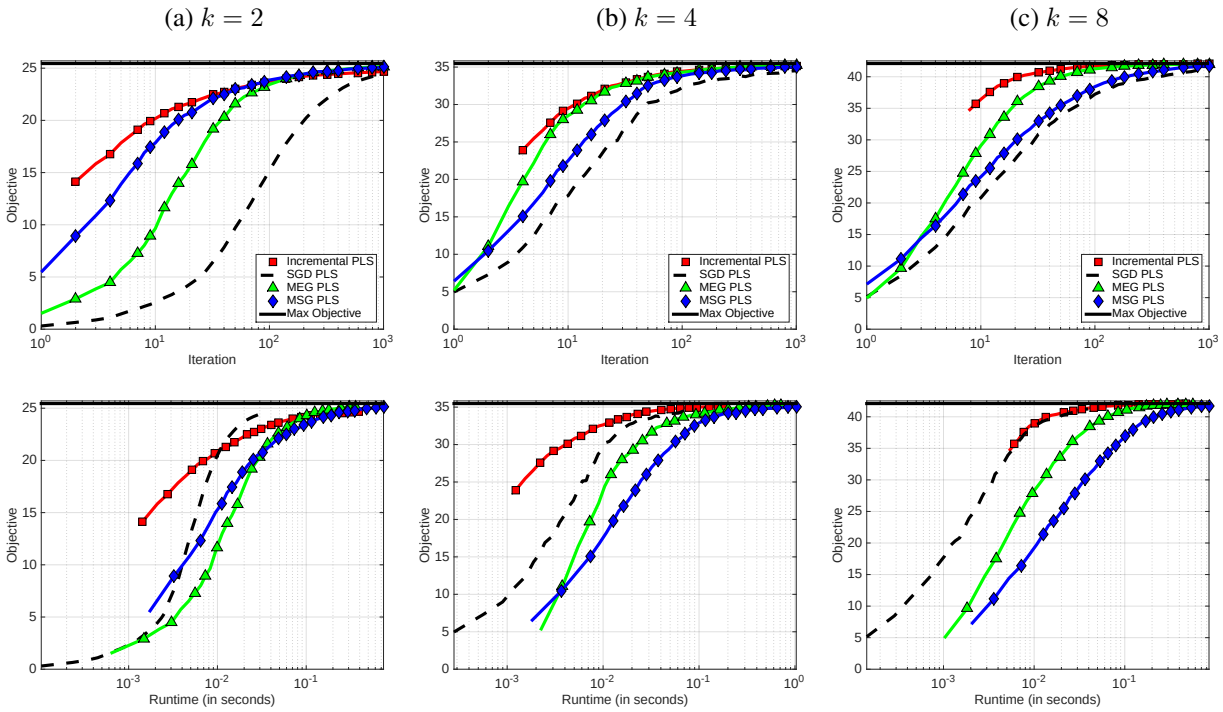

Figure 1. Comparisons of the incremental, SGD with proposed algorithms, MSG and MEG for stochastic PLS optimization on a synthetic dataset, in terms of the objective value as a function of iteration (top) and as a function of CPU runtime (bottom).

and for the first term we get:

$$\mathbb{E}_{(T)}\left[\sum_{t=1}^{T}\mathrm{Tr}\left(\mathrm{M}_{t-1}\mathrm{C}_t\right)\right] = \sum_{t=1}^{T}\mathrm{Tr}\left(\mathbb{E}_{(t)}\left[\mathrm{M}_{t-1}\mathrm{C}_t\right]\right)$$
$$= \sum_{t=1}^{T}\mathrm{Tr}\left(\mathbb{E}_{(t-1)}\mathbb{E}_{(t)}\left[\mathrm{M}_{t-1}\mathrm{C}_t|\{\mathrm{x}_i,\mathrm{y}_i\}_{i=1}^{t-1}\right]\right)$$
$$= \sum_{t=1}^{T}\mathrm{Tr}\left(\mathbb{E}_{(t-1)}\left[\mathrm{M}_{t-1}\right]\mathbb{E}_t\left[\mathrm{C}_t\right]\right)$$
$$= \sum_{t=1}^{T}\mathrm{Tr}\left(\mathbb{E}_{(t-1)}\left[\mathrm{M}_{t-1}\right]\mathrm{C}\right) = \mathbb{E}_{(T)}\left[\sum_{t=1}^{T}\mathrm{Tr}\left(\mathrm{M}_{t-1}\mathrm{C}\right)\right] \quad (16)$$

where the second equality above follows from the law of total expectation, and the third equality holds because M_{t-1} depends only on $\{\mathrm{x}_i, \mathrm{y}_i\}_{i=1}^{t-1}$ while C_t depends only on $\{\mathrm{x}_t, \mathrm{y}_t\}$. Finally, we use (15) and (16) in (14) after taking expectation and divide both sides by T. □

4. Experimental Results

In this section, we evaluate the performance of our methods against other stochastic baselines discussed in Section 2, in terms of the progress made on the objective as a function of the number of iterations as well as the CPU runtime, on both synthetic and real-world datasets.

4.1. Synthetic dataset

For synthetic experiments, we generate data from a pair of elliptical normal distributions with exponentially decaying variances. In particular, we draw a $d \times 3n$ data matrix X from the standard normal distribution $\mathcal{N}(0, \mathrm{I})$, compute its SVD, $\mathrm{X} = \mathrm{USV}^\top$, specify the spectrum by setting $\tilde{\mathrm{S}}_{ii} = (1.2)^{-i}$ and normalizing it to sum to one, and reconstructing the data matrix as $\tilde{\mathrm{X}} = \mathrm{U}\tilde{\mathrm{S}}\mathrm{V}^\top$. Next, we sample two random orthogonal matrices $\mathrm{U}_1 \in \mathbb{R}^{d\times d_x}$ and $\mathrm{U}_2 \in \mathbb{R}^{d\times d_y}$. We project the data points onto the subspaces spanned by $\mathrm{U}_1, \mathrm{U}_2$ and perform random rotations in those subspaces giving us the two views $\mathrm{X}_1 = \mathrm{R}_1\mathrm{U}_1^\top\tilde{\mathrm{X}}$ and $\mathrm{X}_2 = \mathrm{R}_2\mathrm{U}_2^\top\tilde{\mathrm{X}}$, for some random rotation matrices $\mathrm{R}_1 \in \mathbb{R}^{d_x\times d_x}$ and $\mathrm{R}_2 \in \mathbb{R}^{d_y\times d_y}$. Finally, we add i.i.d. zero-mean isotropic Gaussian noise. Each view is split into training, tuning and testing sets, each of size n. We set $(d, n) = (50, 1000)$, $d_x = d_y = 10$.

Our experiments compare performance in terms of the objective function value. Figure 1 shows the PLS objective as a function of the number of iterations (samples processed) as well as CPU runtime, for target dimensionality $k \in \{2, 4, 8\}$. We tune the initial learning rate parameter η_0 for each algorithm over the set $\{0.001, 0.01, 0.1, 1, 10\}$. All algorithms were run for only one "pass" over the training data. All results are averaged over 50 random train/test splits. It is evident from the plots, that incremental PLS

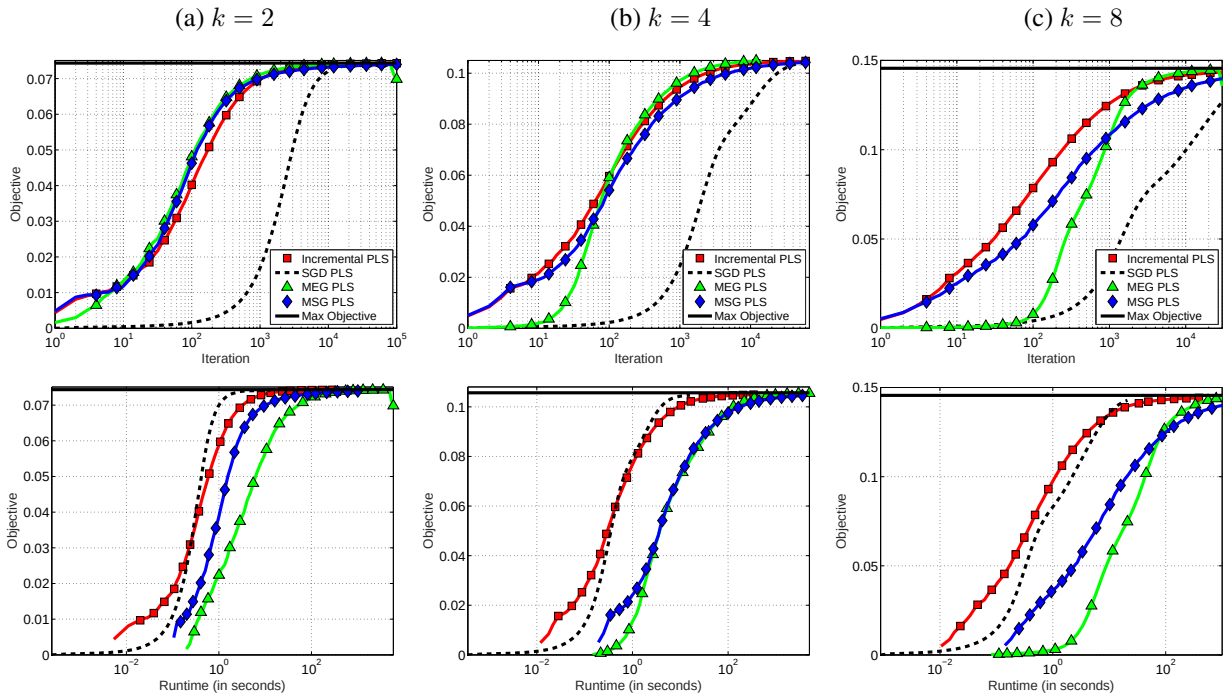

Figure 2. Comparisons of the incremental, SGD with proposed algorithms, MSG and MEG for stochastic PLS optimization on the XRMB dataset, in terms of the objective value as a function of iteration (top) and as a function of CPU runtime (bottom).

is the fastest algorithm both in terms of the runtime and progress-per-iteration. MSG and MEG offer, in some sense, best of the both worlds, since they enjoy good theoretical guarantees, but also compare well with incremental PLS on the empirical performance. We emphasize that a better theoretical understanding of incremental PLS was what led to the development of MSG and MEG in the first place.

4.2. Real world dataset

In this section, we discuss experiments on the University of Wisconsin X-ray Microbeam (XRMB) Database (Westbury, 1994). XRMB contains simultaneously recorded acoustic and articulatory measurements. We use roughly $225,000$ examples from four different speakers. The dimensionality of the acoustic and the articulatory views are $d_x = 1638$ and $d_y = 1008$, respectively. All experiments include pre-normalization, consisting of mean-centering the feature vectors and then dividing each coordinate by its standard deviation times the square root of the length of the feature vector.

In order to ensure a fair comparison with the parameter-free incremental PLS algorithm, we deliberately set all initial learning rates $\eta_0 = 1$, choosing $\eta_t = 1/\sqrt{t}$ uniformly for all experiments. All algorithms were run for only one "pass" over the training data. Our experiments compare performance in terms of the objective function value. Because we cannot evaluate the true population objective for Problem 1, we instead approximate them by evaluating on a held-out testing sample (half of the dataset, with the other half being used for training). All results are averaged over 50 random train/test splits.

Figure 2 shows the PLS objective, as a function of the number of samples processed (iterations) as well as CPU runtime, for ranks $k \in \{2, 4, 8\}$. As expected, SGD is the fastest, but also makes the least progress, per iteration. Both MEG and MSG make better progress than SGD per iteration, in a comparable runtime. Amongst the stochastic algorithms, the incremental algorithm is consistently the best in terms of both runtime and progress-per-iteration, and generally attains an objective close to the optimum faster than the batch algorithm.

5. Conclusion

We study PLS as a stochastic optimization problem, with the goal of optimizing the population objective. This view motivates computationally cheaper algorithms that are variants of SGD in a big data setting. We give theoretical guarantees for online PLS and for a new algorithm called MSG. Our study is motivated by the desire to better understand incremental PLS which enjoys excellent empirical performance but can get stuck at a sub-optimum. MSG and MEG promise best of the both worlds, marrying theoretical guarantees with empirical performance similar to incremental PLS as confirmed by our experiments on both real and synthetic datasets.

Acknowledgements

The authors would like to thank Tuo Zhao for several helpful discussions. This research was supported in part by NSF BIGDATA grant IIS-1546482.

References

Arora, Raman and Livescu, Karen. Kernel CCA for multi-view learning of acoustic features using articulatory measurements. In MLSLP, 2012.

Arora, Raman and Livescu, Karen. Multi-view CCA-based acoustic features for phonetic recognition across speakers and domains. In ICASSP, 2013.

Arora, Raman, Cotter, Andrew, Livescu, Karen, and Srebro, Nathan. Stochastic optimization for PCA and PLS. In Allerton Conference, pp. 861–868, 2012.

Arora, Raman, Cotter, Andy, and Srebro, Nati. Stochastic optimization of PCA with capped MSG. In NIPS, 2013.

Benton, Adrian, Arora, Raman, and Dredze, Mark. Learning multiview embeddings of twitter users. In ACL, 2016.

Bharadwaj, Sujeeth, Arora, Raman, Livescu, Karen, and Hasegawa-Johnson, Mark. Multiview acoustic feature learning using articulatory measurements. In Intl. Workshop Stat. Mach. Learning Speech Recog., 2012.

Blaschko, Matthew B and Lampert, Christoph H. Correlational spectral clustering. In CVPR, 2008.

Bottou, Leon and Bousquet, Olivier. The tradeoffs of large scale learning. In NIPS'07, pp. 161–168, 2007.

Collins, Michael, Globerson, Amir, Koo, Terry, Carreras, Xavier, and Bartlett, Peter L. Exponentiated gradient algorithms for conditional random fields and max-margin markov networks. JMLR, 2008.

Dhillon, Paramveer, Foster, Dean P, and Ungar, Lyle H. Multi-view learning of word embeddings via CCA. In NIPS, pp. 199–207, 2011.

Haghighi, Aria, Liang, Percy, Berg-Kirkpatrick, Taylor, and Klein, Dan. Learning bilingual lexicons from monolingual corpora. In ACL, 2008.

Hardoon, D. R., Szedmak, S., and Shawe-Taylor, J. Canonical correlation analysis: An overview with application to learning methods. Neural Computation, 16(12), 2004.

Horn, Roger A and Johnson, Charles R. Topics in matrix analysis, 1991. Cambridge University Presss, Cambridge, 37:39, 1991.

Nemirovski, Arkadi, Juditsky, Anatoli, Lan, Guanghui, and Shapiro, Alexander. Robust stochastic approximation approach to stochastic programming. SIAM Journal on Optimization, 19(4):1574–1609, January 2009.

Nemirovsky, Arkadi Semenovich and Yudin, David Borisovich. Problem complexity and method efficiency in optimization. Wiley Interscience, 1983.

Rosipal, Roman and Krämer, Nicole. Overview and recent advances in partial least squares. In Subspace, latent structure and feature selection. Springer, 2006.

Shalev-Shwartz, Shai and Srebro, Nathan. SVM optimization: Inverse dependence on training set size. In ICML'08, pp. 928–935, 2008.

Shalev-Shwartz, Shai and Tewari, Ambuj. Stochastic methods for l_1 regularized loss minimization. In ICML, 2009.

Shalev-Shwartz, Shai, Singer, Yoram, and Srebro, Nathan. Pegasos: Primal Estimated sub-GrAdient SOlver for SVM. In ICML'07, pp. 807–814, 2007.

Tropp, Joel A. User-friendly tail bounds for sums of random matrices. Foundations of Computational Math., Aug 2011.

Wang, Weiran, Arora, Raman, Livescu, Karen, and Bilmes, Jeff. On deep multi-view representation learning. In ICML, 2015a.

Wang, Weiran, Arora, Raman, Livescu, Karen, and Bilmes, Jeff A. Unsupervised learning of acoustic features via deep canonical correlation analysis. In ICASSP, 2015b.

Wang, Weiran, Arora, Raman, Srebro, Nati, and Livescu, Karen. Stochastic optimization for deep CCA via nonlinear orthogonal iterations. In 53nd Annual Allerton Conf. Communication, Control and Computing, 2015c.

Wang, Weiran, Arora, Raman, Livescu, Karen, and Bilmes, Jeff. On deep multi-view representation learning: Objectives and optimization. arXiv preprint arXiv:1602.01024, 2016.

Warmuth, Manfred K and Kuzmin, Dima. Online variance minimization. In Learning theory, pp. 514–528. Springer, 2006.

Warmuth, Manfred K and Kuzmin, Dima. Randomized online PCA algorithms with regret bounds that are logarithmic in the dimension. JMLR, 2008.

Westbury, JR. X-ray microbeam speech production database users handbook. Waisman Center, University of Wisconsin, Madison, WI, 1994.