Online Partial Least Square Optimization: Dropping Convexity for Better Efficiency and Scalability

Zhehui Chen [1] **Lin F. Yang** [2] **Chris J. Li** [3] **Tuo Zhao** [1]

Abstract

Multiview representation learning is popular for latent factor analysis. Many existing approaches formulate the multiview representation learning as convex optimization problems, where global optima can be obtained by certain algorithms in polynomial time. However, many evidences have corroborated that heuristic nonconvex approaches also have good empirical computational performance and convergence to the global optima, although there is a lack of theoretical justification. Such a gap between theory and practice motivates us to study a nonconvex formulation for multiview representation learning, which can be efficiently solved by a simple stochastic gradient descent method. By analyzing the dynamics of the algorithm based on diffusion processes, we establish a global rate of convergence to the global optima. Numerical experiments are provided to support our theory.

1. Introduction

Multiview data have become increasingly available in many popular real-world data analysis and machine learning problems. These data are collected from diverse domains or different feature extractors, which share latent factors. Existing literature has demonstrated different scenarios. For instance, the pixels and captions of images can be considered as two-view data, since they are two different features describing the same contents. More motivating examples involving two or more data sets simultaneously can be found in computer vision, natural language processing, and acoustic recognition. See (Hardoon et al., 2004; Socher and Fei-Fei, 2010; Kidron et al., 2005; Chaudhuri et al., 2009; Arora and Livescu, 2012; Bharadwaj et al., 2012; Vinokourov et al., 2002; Dhillon et al., 2011). Although these data are usually unlabeled, there exist underlying association and dependency between different views, which allows us to learn useful representations in a unsupervised manner. Here we are interested in finding a representation that reveals intrinsic low-dimensional structures and decomposes underlying confounding factors. One ubiquitous approach is partial least square (PLS) for multiview representation learning. Specifically, given a data set of n samples of two sets of random variables (views), $X \in \mathbb{R}^m$ and $Y \in \mathbb{R}^d$, PLS aims to find an r-dimensional subspace ($r \ll \min(m, d)$) that preserves most of the covariance between two views. Existing literature has shown that such a subspace is spanned by the leading r components of the singular value decomposition (SVD) of $\Sigma_{XY} = \mathbb{E}_{(X,Y)\sim\mathcal{D}}\left[XY^\top\right]$ (Arora et al., 2012), where we sample (X, Y) from some unknown distribution $\mathcal{D}$. Throughout the rest of the paper, if not clear specified, we denote $\mathbb{E}_{(X,Y)\sim\mathcal{D}}$ by $\mathbb{E}$ for notational simplicity.

A straightforward approach for PLS is "Sample Average Approximation" (SAA, (Abdi, 2003; Ando and Zhang, 2005)), where we run an offline (batch) SVD algorithm on the empirical covariance matrix after seeing sufficient data samples. However, in the "big data" regime, this approach requires unfeasible amount of storage and computation time. Therefore, it is much more practical to consider the multiview learning problem in a "data laden" setting, where we draw independent samples from an underlying distribution $\mathcal{D}$ over $\mathbb{R}^m \times \mathbb{R}^d$, one at a time. This further enables us to formulate PLS as a stochastic (online) optimization problem. Here we only consider the rank-1 case ($r = 1$) for simplicity, and solve

$$(\widehat{u},\, \widehat{v}) = \underset{u\in\mathbb{R}^m, v\in\mathbb{R}^d}{\operatorname{argmax}} \; \mathbb{E}\left(v^\top Y X^\top u\right) \quad \text{subject to} \quad u^\top u = 1, v^\top v = 1. \tag{1.1}$$

Several nonconvex stochastic approximation (SA) algorithms have been proposed in (Arora et al., 2012). These algorithms work great in practice, but are lack of theoretic justifications, since the nonconvex nature of (1.1) makes the theoretical analysis very challenging. To overcome this obstacle, (Arora et al., 2016) propose a convex relaxation of (1.1) by lifting (Lasserre, 2001). Specifically, by a reparametrization $M = uv^\top$ (Recall that we are interested

[1]Georgia Institute of Technology; [2]Johns Hopkins University; [3]Princeton University. Correspondence to: Tuo Zhao <tourzhao@gatech.edu.>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017.

in the rank-1 PLS), they rewrite (1.1) as

$$\widehat{M} = \underset{M\in\mathbb{R}^{d\times m}}{\operatorname{argmax}} \ \langle M, \Sigma_{XY}\rangle \quad \text{subject to} \quad \|M\|_* \leq 1 \text{ and } \|M\|_2 \leq 1. \tag{1.2}$$

where $\Sigma_{XY} = \mathbb{E}XY^\top$, and $\|M\|_2$ and $\|M\|_*$ are the spectral norm(i.e., the largest singular value of M) and nuclear norm(i.e., the sum of all singular values of M) of M respectively. By examining the KKT conditions of (1.2), one can verify that $\widehat{M} = \widehat{u}\widehat{v}^\top$ is the optimal solution, where $\widehat{u}, \widehat{v}$ are the leading left and right singular vectors of Σ_{XY}, i.e., a pair of global optimal solutions to (1.1) for $r = 1$. Accordingly, they propose a projected stochastic gradient-type algorithm to solve (1.2), which is often referred to the Matrix Stochastic Gradient (MSG) algorithm. Particularly, at the $(k+1)$-th iteration, MSG takes

$$M_{k+1} = \Pi_{\text{Fantope}}(M_k + \eta X_k Y_k^\top),$$

where X_k and Y_k are independently sampled from $\mathcal{D}$, and $\Pi_{\text{Fantope}}(\cdot)$ is a projection operator to the feasible set of (1.2). They further prove that given a pre-specified accuracy ϵ, MSG requires

$$N = \ \mathcal{O}(\epsilon^{-2}\log(1/\epsilon))$$

iterations such that $\langle \widehat{M}, \mathbb{E}xy^\top\rangle - \langle M_N, \mathbb{E}xy^\top\rangle \leq \epsilon$ with high probability.

Despite of the attractive theoretic guarantee, MSG does not present superior performance to other heuristic nonconvex stochastic optimization algorithms for solving (1.1). Although there is a lack of theoretical justification, many evidences have corroborated that heuristic nonconvex approaches not only converge to the global optima in practice, but also enjoy better empirical computational performance than the convex approaches (Zhao et al., 2015; Candes et al., 2015; Ge et al., 2015; Cai et al., 2016). Another drawback of MSG is the complicated projection step at each iteration. Although (Arora et al., 2016) further propose an algorithm to compute the projection with a computational cost cubically depending on the rank of the iterates (the worst case: $\mathcal{O}(d^3)$), such a sophisticated implementation significantly decreases the practicability of MSG. Furthermore, MSG is also unfavored in a memory-restricted scenario, since storing the update $M_{(k)}$ requires $\mathcal{O}(md)$ real number storage. In contrast, the heuristic algorithm analyzed in this paper requires only $\mathcal{O}(m+d)$ real number storage.

We aim to bridge the gap between theory and practice for solving multiview representation learning problems by nonconvex approaches. Specifically, we analyze the convergence properties of one heuristic stochastic optimization algorithm for solving (1.1) based on diffusion processes. Our analysis takes advantage of the strong Markov properties of the stochastic optimization algorithm updates and casts the trajectories of the algorithm as a diffusion process (Ethier and Kurtz, 2009; Li et al., 2016b). By leveraging the weak convergence from discrete Markov chains to their continuous time limits, we demonstrate that the trajectories are essentially the solutions to stochastic differential equations (SDE). Such SDE-type analysis automatically incorporates the geometry of the objective and the randomness of the algorithm, and eventually demonstrates three phases of convergence.

1. Starting from an unstable equilibrium with negative curvature, the dynamics of the algorithm can be described by an Ornstein-Uhlenbeck process with a steady driven force pointing away from the initial.

2. When the algorithm is sufficiently distant from initial unstable equilibrium, the dynamics can be characterized by an ordinary differential equation (ODE). The trajectory of this phase is evolving directly toward the desired global maximum until it reaches a small basin around the optimal.

3. In this phase, the trajectory can be also described by an Ornstein-Uhlenbeck process oscillating around the global maximum. The process has a drifting term that gradually dies out and eventually becomes a nearly unbiased random walk centered at the maximum.

   The sharp characterization in these three phases eventually allows us to establish strong convergence guarantees. Specifically, we show the nonconvex stochastic gradient algorithm guarantees an ϵ-optimal solution in

   $$\mathcal{O}(\epsilon^{-1}\log(1/\epsilon))$$

   iterations with high probability, which is a significant improvement over convex MSG by a factor of ϵ^{-1}. Our theoretical analysis reveals the power of the nonconvex optimization in PLS. The simple heuristic algorithm drops the convexity, but achieves much better efficiency.

Notations: Given a vector $v = (v^{(1)}, \ldots, v^{(d)})^\top \in \mathbb{R}^d$, we define vector norms: $\|v\|_1 = \sum_j |v^{(j)}|$, $\|v\|_2^2 = \sum_j (v^{(j)})^2$, and $\|v\|_\infty = \max_j |v^{(j)}|$. Given a matrix $A \in \mathbb{R}^{d\times d}$, we use $A_j = (A_{1j}, ..., A_{dj})^\top$ to denote the j-th column of A and define the matrix norms $\|A\|_{\text{F}}^2 = \sum_j \|A_j\|_2^2$ and $\|A\|_2$ as the largest singular value of A.

2. Nonconvex Stochastic Optimization

Recall that we solve (1.1)

$$(\widehat{u},\widehat{v}) = \underset{u,v}{\operatorname{argmax}}\ u^\top \mathbb{E}XY^\top v$$
$$\text{subject to}\quad \|u\|_2^2 = 1,\ \|v\|_2^2 = 1, \tag{2.1}$$

where (X, Y) follows some unknown distribution $\mathcal{D}$. Due to symmetrical structure of (2.1), $(-\widehat{u}, -\widehat{v})$ is also a pair of global optimum. All our analysis holds for both optima. Throughout the rest of paper, if it is not clearly specified, we consider $(\widehat{u}, \widehat{v})$ as the global optimum for simplicity.

We apply a straightforward projected stochastic gradient algorithm (PSG). Specifically, at the k-th iteration, we have the iterates u_k and v_k. We then independently sample X_k and Y_k from $\mathcal{D}$, and take

$$u_{k+1} = \Pi(u_k + \eta X_k Y_k^\top v_k), \tag{2.2}$$
$$v_{k+1} = \Pi(v_k + \eta Y_k X_k^\top u_k), \tag{2.3}$$

where $\eta > 0$ is the step size parameter, and $\Pi(\cdot)$ is the projection operator on the unit sphere. As can be seen from (2.2), we have $X_k Y_k^\top v_k$ as a unbiased estimator of the gradient of the objective function. The projected stochastic gradient algorithm has been studied for convex optimization, and their rates of convergence have been characterized in (Ben-Tal and Nemirovski, 2001; Nemirovski et al., 2009). The problem (2.1), however, is nonconvex, and existing literature in optimization only shows that the stochastic gradient algorithms converge to a stationary solution.

3. Global Convergence by ODE

Before we proceed with our analysis, we first impose some mild assumptions on the problem.

Assumption 3.1. $X_k, Y_k,\ k = 1, ..., N$ are data samples identically independently distributed as $X,\ Y \in \mathbb{R}^d$ respectively satisfying the following conditions:

1. $\|X\|_2^2 \le Bd,\ \|Y\|_2^2 \le Bd$ for a constant B;
2. $\lambda_1 > \lambda_2 \ge \lambda_3 \ge ... \ge \lambda_d > 0$, where λ_i's are the singular values of $\Sigma_{XY} = \mathbb{E}XY^\top$.

Note that here we assume X and Y are of the same dimensions (i.e., $m = d$) and Σ_{XY} is full rank for convenience of analysis. The extension to $m \neq d$ and rank deficient settings is straightforward.

Assumption 3.2. Given the observed random vectors X and Y, there exist two orthogonal matrices $O_X,\ O_Y \in \mathbb{R}^{d\times d}$ such that $X = O_X \overline{X},\ Y = O_Y\, \overline{Y}$, where

$$\overline{X} = (\overline{X}^{(1)}, ..., \overline{X}^{(d)})^\top, \text{ and } \overline{Y} = (\overline{Y}^{(1)}, ..., \overline{Y}^{(d)})^\top \in \mathbb{R}^d$$

are the latent variables satisfying:

1. $\overline{X}^{(i)}$ and $\overline{Y}^{(j)}$ are uncorrelated if $i \neq j$, so that O_X and O_Y are the left and right singular matrices of Σ_{XY} respectively;
2. $\mathrm{Var}(\overline{X}^{(i)}) = \gamma_i, \mathrm{Var}(\overline{Y}^{(i)}) = \omega_i$, $\mathbb{E}\big(\overline{X}^{(i)}\overline{Y}^{(i)}\overline{X}^{(j)}\overline{Y}^{(j)}\big) = \alpha_{ij}$

The next proposition characterizes the strong Markov property of our algorithm.

Proposition 3.3. Using (2.2) and (2.3), we get a sequence of $(u_k, v_k),\ k = 1, 2, ..., N$. They form a discrete-time Markov process.

With Proposition 3.3, we can construct a continuous time process to derive an ordinary differential equation to analyze the algorithmic convergence. Before that, we first compute $u_{k+1} - u_k$ and $v_{k+1} - v_k$ to see how much they change in each iteration. We denote the i-th coordinate of u_k and v_k by $u_k^{(i)}$ and $v_k^{(i)}$.

Proposition 3.4. Suppose Assumption 3.1 holds. Given $Bd\eta < \frac{1}{4}$, the following results hold:

(1) There exist random variables $R_k^{(i)}$ with $\left|R_k^{(i)}\right| \le 20B^2d^2\eta^2$ and $Q_k^{(i)}$ with $\left|Q_k^{(i)}\right| \le 20B^2d^2\eta^2$, such that the increments $u_{k+1}^{(i)} - u_k^{(i)}$ and $v_{k+1}^{(i)} - v_k^{(i)}$ are

$$u_{k+1}^{(i)} - u_k^{(i)} = \eta\left(Y_k^\top v_k X_k^{(i)} - u_k^\top X_k Y_k^\top v_k u_k^{(i)}\right) + R_k^{(i)},$$
$$v_{k+1}^{(i)} - v_k^{(i)} = \eta\left(X_k^\top u_k Y_k^{(i)} - v_k^\top Y_k X_k^\top u_k v_k^{(i)}\right) + Q_k^{(i)}.$$

(2) Furthermore, there are two deterministic functions $f_k(u, v)$ and $g_k(u, v)$ satisfying

$$\max\{|f_k(u,v)|, |g_k(u,v)|\} \le 20B^2d^2\eta^2 \text{ for } \forall u,\ v \in \mathcal{S}^{d-1},$$

such that conditioning on u_k and v_k, the expectation of the increments in (1) can be represented as

$$\begin{aligned}&\mathbb{E}[u_{k+1} - u_k \mid u_k,\ v_k]\\ &= \big(\Sigma_{XY}v_k - u_k^\top\Sigma_{XY}v_k u_k\big)\cdot\eta + f_k(u_k, v_k),\\ &\mathbb{E}[v_{k+1} - v_k \mid u_k,\ v_k]\\ &= \big(\Sigma_{XY}^\top u_k - v_k^\top\Sigma_{XY}^\top u_k v_k\big)\cdot\eta + g_k(u_k, v_k).\end{aligned}$$

Proposition 3.4 is obtained by Taylor expansion. Its proof is presented in Appendix A.1. Result (2) enables us to compute the infinitesimal conditional mean and variance for the projected stochastic gradient algorithm. Specifically, as the fixed step size $\eta \to 0^+$, two processes $U_\eta(t) = u_{\lfloor\eta^{-1}t\rfloor},\ V_\eta(t) = v_{\lfloor\eta^{-1}t\rfloor}$ based on the sequence generated by (2.2) and (2.3), converge to the solution of the following

ODE system in probability (See more details in (Ethier and Kurtz, 2009)),

$$\frac{dU}{dt} = \left(\Sigma_{XY}V - U^\top \Sigma_{XY} VU\right), \tag{3.1}$$

$$\frac{dV}{dt} = \left(\Sigma_{XY}^\top U - V^\top \Sigma_{XY}^\top UV\right), \tag{3.2}$$

where $U(0) = u_0$ and $V(0) = v_0$. To highlight the sequence generated by (2.2) and (2.3) depending on η, we redefine $u_{\eta,k} = u_k,\ v_{\eta,k} = v_k$.

Theorem 3.5. As $\eta \to 0^+$, the processes $u_{\eta,k},\ v_{\eta,k}$ weakly converge to the solution of the ODE system in (3.1) and (3.2) with initial $U(0) = u_0,\ V(0) = v_0$.

The proof of Theorem 3.5 is presented in Appendix A.2. Under Assumption 3.1, the above ODE system admits a closed form solution. Specifically, we solve U and V simultaneously, since they are coupled together in (3.1) and (3.2) . To simplify them , we define

$$W = \frac{1}{\sqrt{2}}\left(U^\top\ V^\top\right)^\top \text{ and } w_k = \frac{1}{\sqrt{2}}\left(u_k^\top\ v_k^\top\right)^\top.$$

We then rewrite (3.1) and (3.2) as

$$\frac{dW}{dt} = QW - W^\top QWW, \tag{3.3}$$

where $Q = \begin{pmatrix} 0 & \Sigma_{XY} \\ \Sigma_{XY}^\top & 0 \end{pmatrix}$. By Assumption 3.2, O_X and O_Y are left and right singular matrices of Σ_{XY} respectively, i.e.,

$$\Sigma_{XY} = \mathbb{E}XY^\top = O_X \mathbb{E}\overline{XY}^\top O_Y^\top,$$

where $\mathbb{E}\overline{XY}^\top$ is diagonal. For notational simplicity, we define $D = \mathrm{diag}(\lambda_1, \lambda_2, ..., \lambda_d)$ such that

$$\Sigma_{XY} = O_X D O_Y^\top.$$

One can verify $Q = P\Lambda P^\top$, where

$$P = \frac{1}{\sqrt{2}}\begin{pmatrix} O_X & O_X \\ O_Y & -O_Y \end{pmatrix}, \Lambda = \begin{pmatrix} D & 0 \\ 0 & -D \end{pmatrix}. \tag{3.4}$$

By left multiplying $P^\top$ both sides of (3.3), we obtain

$$H(t) = P^\top W(t) \text{ with } \frac{dH}{dt} = \Lambda H - H^\top \Lambda HH, \tag{3.5}$$

which is a coordinate separable ODE system. Accordingly, we define $h_k^{(i)}$'s as:

$$h_k = P^\top w_k \quad \text{and} \quad h_k^{(i)} = P_i^\top w_k. \tag{3.6}$$

Thus, we can obtain a closed form solution to (3.5) based on the following theorem.

Theorem 3.6. Given (3.5), we write the ODE in each component $H^{(i)}$,

$$\frac{d}{dt}H^{(i)} = H^{(i)}\sum_{j=1}^{2d}(\lambda_i - \lambda_j)(H^{(j)})^2, \tag{3.7}$$

where $\lambda_i = -\lambda_{i-d}$ when $i > d$. This ODE System has a closed form solution as follows:

$$H^{(i)}(t) = \left(C(t)\right)^{-\frac{1}{2}} H^{(i)}(0)\exp\left(\lambda_i t\right), \tag{3.8}$$

for $i = 1, 2, ..., 2d$, where

$$C(t) = \sum_{j=1}^{2d}\left(\left[H^{(j)}(0)\right]^2 \exp\left(2\lambda_j t\right)\right)$$

is a normalization function such that $\|H(t)\|_2 = 1$.

The proof of Theorem 3.6 is presented in Appendix A.3. Without loss of generalization, we assume $H^{(1)}(0) > 0$. We can get $H^{(1)}(t) \to 1$, as $t \to \infty$. We have successfully characterized the global convergence performance of our algorithm with an approximate error $o(1)$. The solution to the ODE system in (3.8), however, does not fully reveal the algorithmic behavior (more precisely, the rate of convergence) near the equilibria of the ODE system. This further motivates us to exploit the stochastic differential equation approach to characterize the dynamics of the algorithm.

4. Global Dynamics by SDE

We analyze the dynamics of the algorithm near the equilibria based on stochastic differential equation by rescaling analysis. Specifically, we characterize three stages for the trajectories of solutions:

1. Neighborhood around unstable equilibria — minimizers and saddle points of (2.1),
2. Neighborhood around stable equilibria — maximizers of (2.1), and
3. deterministic traverses between equilibria. Moreover, we provide the approximate the number of iterations in each phase until convergence.

4.1. Phase I: Escaping from unstable equilibria

Suppose that the algorithm starts to iterate around a unstable equilibrium, (e.g. saddle point). Different from our previous analysis, we rescale two aforementioned processes $U_\eta(t)$ and $V_\eta(t)$ by a factor of $\eta^{-1/2}$. This eventually allows us to capture the uncertainty of the algorithm updates by stochastic differential equations. Roughly speaking, the ODE approximation is essentially a variant of law of large

number for Markov process, while the SDE approximation serves as a variant of central limit theorem accordingly.

Recall that P is an orthonormal matrix for diagonalizing Q, and H is defined in (3.5). Let $Z_\eta^{(i)}$ and $z_{\eta,k}^{(i)}$ denote the i-th coordinates of

$$Z_\eta = \eta^{-1/2} H_\eta \text{ and } z_{\eta,k} = \eta^{-1/2} h_{\eta,k}$$

respectively. The following theorem characterizes the dynamics of the algorithm around the unstable equilibrium.

Theorem 4.1. Suppose $z_{\eta,0}$ is initialized around some saddle point or minimizer (e.g. j-th column of P with $j \neq 1$), i.e., $Z^{(j)}(0) \approx \eta^{-1}$ and $Z^{(i)}(0) \approx 0$ for $i \neq j$. Then as $\eta \to 0^+$, for all $i \neq j$, $z_{\eta,k}^{(i)}$ weakly converges to a diffusion process $Z^{(i)}(t)$ satisfying the following SDE,

$$dZ^{(i)}(t) = -(\lambda_j - \lambda_i) Z^{(i)}(t) dt + \beta_{ij} dB(t), \qquad (4.1)$$

where $B(t)$ is a brownian motion, β_{ij} is defined as:

$$\beta_{ij} = \begin{cases} \frac{1}{2}\sqrt{\gamma_i\omega_j + \gamma_j\omega_i + 2\alpha_{i,j}} & \text{if } 1 \le i,j \le d \\ & \text{or } d+1 \le i,j \le 2d, \\ \frac{1}{2}\sqrt{\gamma_i\omega_j + \gamma_j\omega_i - 2\alpha_{i,j}} & \text{otherwise,} \end{cases}$$

where $\gamma_i = \gamma_{i-d}$ for $i > d$, $\omega_j = \omega_{j-d}$ for $j > d$, similar definition of α_{ij} for $i > d$ or $j > d$.

The proof of Theorem 4.1 is provided in Appendix B.1. Note that (4.1) is a Fokker-Planck equation, which admits a closed form solution as follows,

$$Z^{(i)}(t) = \underbrace{\left[Z^{(i)}(0) + \beta_{i,j} \int_0^t \exp\left[(\lambda_j - \lambda_i)s\right] dB(s) \right]}_{T_1} \cdot \underbrace{\exp\left[(\lambda_i - \lambda_j)t\right]}_{T_2} \quad \text{for } i \neq j. \qquad (4.2)$$

Such a solution is well known as the Ornstein-Uhlenbeck process (Øksendal, 2003), and also implies that the distribution of $z_{\eta,k}^{(i)}$ can be well approximated by the normal distribution of $Z^{(i)}(t)$ for a sufficiently small step size. This continuous approximation further has the following implications:

[a] For $\lambda_i > \lambda_j$, $T_1 = \beta_{ij} \int_0^t \exp\left[(\lambda_j - \lambda_i)s\right] dB(s) + Z^{(i)}(0)$ is essentially a random variable with mean $Z^{(i)}(0)$ and variance smaller than $\frac{\beta_{ij}^2}{2(\lambda_i - \lambda_j)}$. The larger t is, the closer its variance gets to this upper bound. While $T_2 = \exp\left[(\lambda_i - \lambda_j)t\right]$ essentially amplifies T_1 by a factor exponentially increasing in t. This tremendous amplification forces $Z^{(i)}(t)$ to quickly get away from 0, as t increases.

[b] For $\lambda_i < \lambda_j$, we have

$$\mathbb{E}[Z^{(i)}(t)] = Z^{(i)}(0) \exp\left[-(\lambda_j - \lambda_i)t\right],$$
$$\mathrm{Var}[Z^{(i)}(t)] = \frac{\beta_{ij}^2}{2(\lambda_j - \lambda_i)}\left[1 - \exp\left[-2(\lambda_j - \lambda_i)t\right]\right].$$

As has been shown in **[a]** that t does not need to be large for $Z^{(i)}(t)$ to get away from 0. Here we only consider relatively small t. Since the initial drift for $Z^{(i)}(0) \approx 0$ is very small, $Z^{(i)}$ tends to stay at 0. As t increases, the exponential decay term makes the drift quickly become negligible. Moreover, by mean value theorem, we know that the variance is bounded, and increases far slower than the variance in **[a]**. Thus, roughly speaking, $Z^{(i)}(t)$ oscillates near 0.

[c] For $\lambda_j = \lambda_i$, we have $\mathbb{E}[Z^{(i)}(t)] = Z^{(i)}(0)$ and $\mathrm{Var}[Z^{(i)}(t)] = \beta_{ij}^2$. This implies that $Z^{(i)}(t)$ also tends to oscillate around 0, as t increases.

Overall speaking, **[a]** is dominative so that it is the major driving force for the algorithm to escape from this unstable equilibrium. More precisely, let us consider one special case for Phase I, that is we start from the second maximum singular value, with $h_{\eta,k}^{(2)}(0) = 1$. We then approximately calculate the number of iterations to escape Phase I using the algorithmic behavior of $h_{\eta,k}^{(1)} = \eta^{1/2} z_{\eta,k}^{(1)} \approx \eta^{1/2} Z_\eta^{(1)}(t)$ with $t = k\eta$ by the following proposition.

Proposition 4.2. Given pre-specified $\nu > 0$ and sufficiently small η, there exists some $\delta \asymp \eta^\mu$, where $\mu \in (0, 0.5)$ is a generic constant, such that the following result holds: We need at most

$$N_1 = \frac{\eta^{-1}}{2(\lambda_1 - \lambda_2)} \log\left(\frac{2\eta^{-1}\delta^2(\lambda_1 - \lambda_2)}{\Phi^{-1}\left(\frac{1+\nu}{2}\right)^2 \beta_{12}^2} + 1 \right)$$

iterations such that $(h_{\eta,N_1}^{(2)})^2 \le 1 - \delta^2$ with probability at least $1 - \nu$, where $\Phi(x)$ is the CDF of standard normal distribution.

The proof of Proposition 4.2 is provided in Appendix B.2. Proposition 4.2 suggests that SGD can escape from unstable equilibria in a few iterations. After escaping from the saddle, SGD gets into the next phase, which is a deterministic traverse between equilibria.

4.2. Phase II: Traverse between equilibria

When the algorithm is close to neither the saddle points nor the optima, the algorithm's performance is nearly deterministic. Since $Z(t)$ is a rescaled version of $H(t)$, their trajectories are similar. Like before, we have the following proposition to calculate the approximate iterations, N_2, following our results in Section 3. We restart the counter of iteration by Proposition 3.3.

Proposition 4.3. After restarting counter of iteration, given sufficiently small η and δ defined in Proposition 4.2, we need at most

$$N_2 = \frac{\eta^{-1}}{2(\lambda_1 - \lambda_2)} \log \frac{1-\delta^2}{\delta^2}$$

iterations such that $\left(h^{(1)}_{\eta,N_2}\right)^2 \geq 1-\delta^2$.

The proof of Proposition 4.3 is provided in Appendix B.3. Combining Propositions 4.2 and 4.3, we know that after $N_1 + N_2$ iteration numbers, SGD is close to the optimum with high probability, and gets into its third phase, i.e., convergence to stable equilibria.

4.3. Phase III: Convergence to stable equilibria

Again, we restart the counter of iteration by the strong Markov property. The trajectory and analysis are similar to Phase I, since we also characterize the convergence using an Ornstein-Uhlenbeck process. The following theorem characterizes the dynamics of the algorithm around the stable equilibrium.

Theorem 4.4. Suppose $z_{\eta,0}$ is initialized around some maximizer (the first column of P), i.e., $Z^{(1)}(0) \approx \eta^{-\frac{1}{2}}$ and $Z^{(i)}(0) \approx 0$ for $i \neq 1$. Then as $\eta \to 0^+$, for all $i \neq 1$, $z^{(i)}_{\eta,k}$ weakly converges to a diffusion process $Z^{(i)}(t)$ satisfying the following SDE for $i \neq 1$,

$$dZ^{(i)}(t) = -(\lambda_1 - \lambda_i) Z^{(i)}(t) dt + \beta_{i1} dB(t), \qquad (4.3)$$

where $B(t)$ is a brownian motion, and

$$\beta_{i1} = \begin{cases} \frac{1}{2}\sqrt{\gamma_i\omega_1 + \gamma_1\omega_i + 2\alpha_{i1}} & \text{if } 1 \leq i \leq d, \\ \frac{1}{2}\sqrt{\gamma_i\omega_1 + \gamma_1\omega_i - 2\alpha_{i1}} & \text{otherwise.} \end{cases}$$

The proof of Theorem 4.4 is provided in Appendix B.4. Similar to (4.2), the closed form solution to (4.3) for $i \neq 1$ is as follow:

$$Z^{(i)}(t) = Z^{(i)}(0) \exp\left[-(\lambda_1 - \lambda_i)t\right] + \beta_{i1} \int_0^t \exp\left[(\lambda_1 - \lambda_i)(s-t)\right] dB(s).$$

By the property of the O-U process, we characterize the expectation and variance of $Z^{(i)}(t)$ for $i \neq 1$.

$$\mathbb{E} Z^{(i)}(t) = Z^{(i)}(0) \exp\left[-(\lambda_1 - \lambda_i)t\right],$$

$$\mathbb{E}\left(Z^{(i)}(t)\right)^2 = \frac{\beta^2_{i1}}{2(\lambda_1 - \lambda_i)} + \exp\left[-2(\lambda_1 - \lambda_i)t\right] \cdot \left[\left(Z^{(i)}(0)\right)^2 - \frac{\beta^2_{i1}}{2(\lambda_1 - \lambda_i)}\right].$$

Recall that the distribution of $z^{(i)}_{\eta,k}$ can be well approximated by the normal distribution of $Z^{(i)}(t)$ for a sufficiently small step size. This further implies that after sufficiently many iterations, SGD enforces $z^{(i)}_{\eta,k} \to 0$ except $i = 1$. Meanwhile, SGD behaves like a biased random walk towards the optimum, when it iterates within a small neighborhood the optimum. But unlike Phase I, the variance gradually becomes a constant.

Based on theorem 4.4, we establish an iteration complexity bound for SGD in following proposition.

Proposition 4.5. Given a pre-specified $\epsilon > 0$, a sufficiently small η, and δ defined in Proposition 4.2, after restarting counter of iteration, we need at most

$$N_3 = \frac{\eta^{-1}}{2(\lambda_1 - \lambda_2)} \log\left(\frac{4(\lambda_1 - \lambda_2)\delta^2}{(\lambda_1 - \lambda_2)\epsilon\eta^{-1} - 4d \max\limits_{1\leq i\leq d} \beta^2_{i1}}\right),$$

iterations such that $\sum_{i=2}^{2d}\left(h^{(i)}_{\eta,N_3}\right)^2 \leq \epsilon$ with probability at least $3/4$.

The proof of Proposition 4.5 is provided in Appendix B.5. Combining Propositions 4.2, 4.3, and 4.5, we obtain a more refined result in the following corollary.

Corollary 4.6. Given a sufficiently small pre-specified $\epsilon > 0$, we choose

$$\eta \asymp \frac{\epsilon(\lambda_1 - \lambda_2)}{d \max_{1\leq i\leq d} \beta^2_{i1}}.$$

We need at most

$$N = O\left[\frac{d}{\epsilon(\lambda_1 - \lambda_2)^2} \log\left(\frac{d}{\epsilon}\right)\right]$$

iterations such that we have $\|u_{\eta,n} - \widehat{u}\|_2^2 + \|v_{\eta,n} - \widehat{v}\|_2^2 \leq 3\epsilon$ with probability at least $2/3$.

The proof of Corollary 4.6 is provided in Appendix B.6. We can further improve the probability to $1-\nu$ for some $\nu > 0$ by repeating $\mathcal{O}(\log 1/\nu)$ replicates of SGD. We then compute the geometric median of all outputs. See more details in (Cohen et al., 2016).

5. Numerical Experiments

We first provide a simple example to illustrate our theoretical analysis. Specifically, we choose $m = d = 3$. We first generate the joint covariance matrix for the latent factors $\overline{X}$

and $\overline{Y}$ as

$$\mathrm{Cov}(\overline{X}) = \Sigma_{\overline{XX}} = \begin{bmatrix} 6 & 2 & 1 \\ 2 & 6 & 2 \\ 1 & 2 & 6 \end{bmatrix},$$

$$\mathrm{Cov}(\overline{X}, \overline{Y}) = \Sigma_{\overline{XY}} = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0.5 \end{bmatrix},$$

and $\Sigma_{\overline{YY}} = \Sigma_{\overline{XX}}$. We then generate two matrices $\widetilde{U}$ and $\widetilde{V}$ with each entry independently sampled from $N(0,1)$. Then we convert $\widetilde{U}$ and $\widetilde{V}$ to orthonormal matrices U and V by Grand-Schmidt transformation. At last, we generate the joint covariance matrix for the observational random vectors X and Y using the following covariance matrix

$$\mathrm{Cov}(X) = U^\top \Sigma_{\overline{XX}} U, \quad \mathrm{Cov}(X,Y) = U^\top \Sigma_{\overline{XY}} V,$$
$$\text{and} \quad \mathrm{Cov}(Y) = V^\top \Sigma_{\overline{YY}} V.$$

We consider the total sample size as $n = 2 \times 10^5$ and choose $\eta = 5 \times 10^{-5}$. The initialization solution (u_0, v_0) is a pair of singular vectors associated with the second largest singular value of Σ_{XY}, i.e., saddle point. We repeat the simulation with update (2.2) and (2.3) for 100 times, and plot the obtained results.

Figure 1(a) illustrates the three phases of the SGD algorithm. Specifically, the horizontal axis is the number of iterations, and the vertical axis is $h_k^{(1)}$ defined in (3.6). As $h_k^{(1)} \to \pm 1$, we have $u_k \to \pm\widehat{u}$ and $v_k \to \pm\widehat{v}$, e.g., global optima. This is due to the symmetric structure of the problem as mentioned in Section 1. Figure 1(a) is consistent with our theory: In Phase I, the algorithm gradually escapes from the saddle point; In Phase II, the algorithm quickly moves towards the optimum; In Phase III, the algorithm gradually converges to the optimum.

Figure 1(b) further zooms in Phase I of Figure 1(a). We see that the trajectories of all 100 simulations behave very similar to an O-U process. Figure 1(c) illustrates the three phases by $h_k^{(2)}$. As our analysis suggests, when $h_k^{(1)} \to \pm 1$, we have $h_k^{(2)} \to 0$. We see that the trajectories of all 100 simulations also behave very similar to an O-U process in Phase III. These experimental results are consistent with our theory.

Also, we illustrate $h^{(1)}$ in Phase I and $h^{(2)}$ in Phase III are O-U process by showing that 100 simulations of $h^{(1)}$ follow a gaussian distribution in 10, 100 and 1000 iteration and those of $h^{(1)}$ follow a gaussian distribution in 10^5, 1.5×10^5 and 2×10^5 iteration. This is consistent with the Theorems 4.1 and 4.4 in Section 4. Also as we can see that in the Phase I, the variance of $h^{(1)}$ becomes larger and larger when iteration number increases. Similarly, in the Phase III, the variance of $h^{(2)}$ becomes closer to a fixed number.

We then provide a real data experiment for comparing the computational performance our nonconvex stochastic gradient algorithm for solving (2.1) with the convex stochastic gradient algorithm for solving (1.2). We choose a subset of the MNIST dataset, whose labels are $3, 4, 5,$ or 9. The total sample size is $n = 23343$, and $m = d = 392$. As (Arora et al., 2016) suggest, we choose $\eta_k = 0.05/\sqrt{k}$ or 2.15×10^{-5}, for the convex stochastic gradient algorithm. For our nonconvex stochastic gradient algorithm, we choose either $\eta_k = 0.05/k$ or 10^{-4}, 2×10^{-5}, 3×10^{-5}. Figure 3 illustrates the computational performance in terms of iterations and wall clock time. As can be seen, our nonconvex stochastic gradient algorithm outperforms the convex counterpart in iteration complexity, and significantly outperforms in wall clock time, since the nonconvex algorithm does not need the computationally expensive projection in each iteration. This suggests that dropping convexity for PLS can boost both computational scalability and efficiency.

6. Discussions

We establish the convergence rate of stochastic gradient descent (SGD) algorithms for solving online partial least square (PLS) problems based on diffusion process approximation. Our analysis indicates that for PLS, dropping convexity actually improves efficiency and scalability. Our convergence results are tighter than existing convex relaxation based method by a factor of $O(1/\epsilon)$, where ϵ is a prespecified error. We believe the following directions should be of wide interests:

1. Our current results hold only for the top pair of left and right singular vectors, i.e., $r = 1$. For $r > 1$, we need to solve

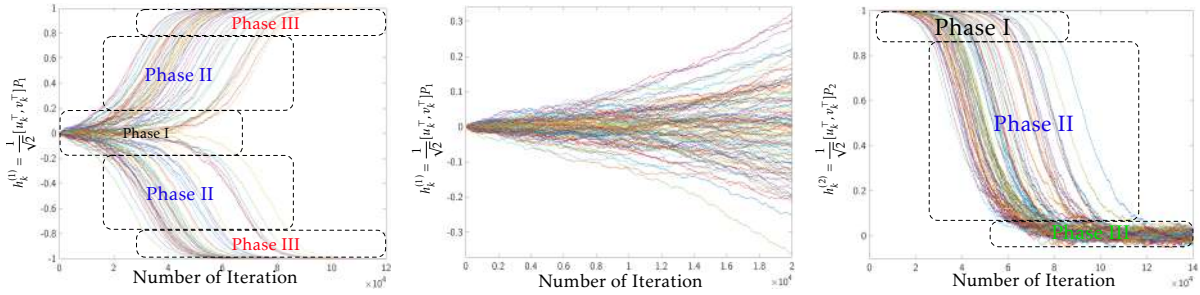
$$(\widehat{U}, \widehat{V}) = \underset{U \in \mathbb{R}^{m\times r}, V \in \mathbb{R}^{d\times r}}{\operatorname{argmax}} \ \mathbb{E}\ \mathrm{tr}(V^\top Y X^\top U)$$
$$\text{subject to} \quad U^\top U = I_r, \quad V^\top V = I_r. \tag{6.1}$$

   Our approximations using ODE and SDE, however, do not admit an unique solution due to rotation. Thus, extension to $r > 1$ is a challenging, but also an important future direction.

2. Our current results are only applicable to a fixed step size $\eta \asymp \epsilon(\lambda_1 - \lambda_2)d^{-1}$. Our experiments suggest that the diminishing step size

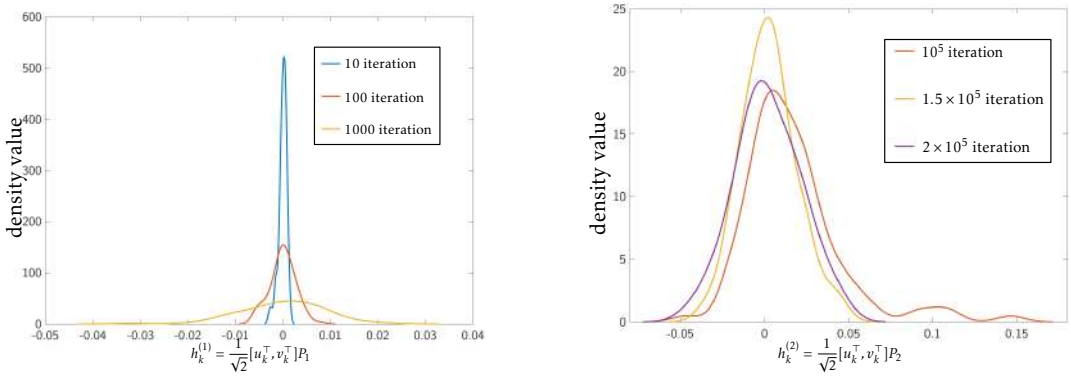
$$\eta_k \asymp k^{-1}(\lambda_1 - \lambda_2)^{-1} \log d,$$

   k from 1 to N, where N is the sample complexity from theory, achieves a better empirical performance. One possible probability tool is Stein's method (Ross et al., 2011).



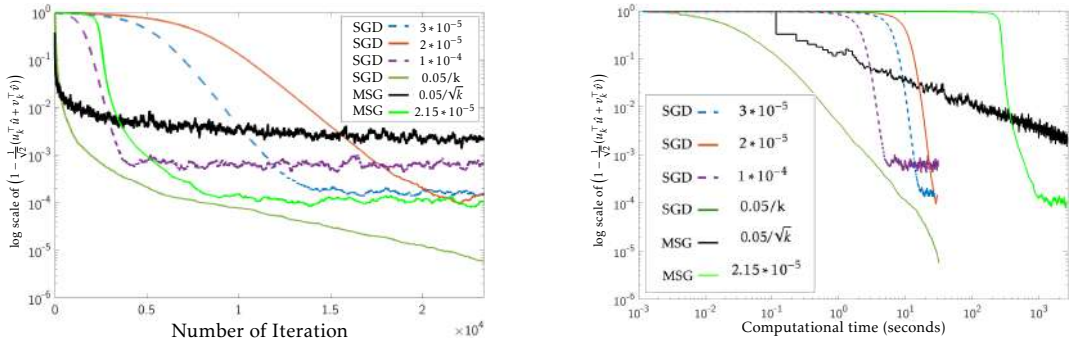
(a) All Three Phases of $h_k^{(1)}$. (b) Phase I of $h_k^{(1)}$. (c) All Three Phases of $h_k^{(2)}$.

Figure 1. An illustrative examples of the stochastic gradient algorithm. The three phases of the algorithm are consistent with our theory: In Phase I, the algorithm gradually escapes from the saddle point; In Phase II, the algorithm quickly iterates towards the optimum; In Phase III, the algorithm gradually converges to the optimum.



(a) The estimated density of $h^{(1)}$ in Phase I. (b) The estimated density of $h^{(2)}$ in Phase III.

Figure 2. The estimated density based on 100 simulations (obtained by kernel density estimation using 10-fold cross validation) at different iterations in Phase I and Phase III shows that $h_k^{(1)}$'s in Phase I and $h_k^{(2)}$'s in Phase III behave very similar to O-U processes, which is consistent our theory.



(a) Comparison by Iteration (b) Comparison by Time

Figure 3. Comparison between nonconvex SGD and convex MSG with different step sizes. We see that SGD not only has a better iteration complexity, but also is more computationally efficient in wall clock time than convex MSG.

3. Our current results rely on the classical central limit theorem-type analysis by taking $\eta \to 0^+$. Connecting our analysis to discrete algorithmic proofs, such as (Jain et al., 2016; Shamir, 2015; Li et al., 2016a), is an important direction (Barbour and Chen, 2005). One possible tool for addressing this is Stein's method (Ross et al., 2011).

References

ABDI, H. (2003). Partial least square regression (pls regression). *Encyclopedia for research methods for the social sciences* 792–795.

ANDO, R. K. and ZHANG, T. (2005). A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research* **6** 1817–1853.

ARORA, R., COTTER, A., LIVESCU, K. and SREBRO, N. (2012). Stochastic optimization for pca and pls. In *Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on*. IEEE.

ARORA, R. and LIVESCU, K. (2012). Kernel cca for multi-view learning of acoustic features using articulatory measurements. In *MLSLP*. Citeseer.

ARORA, R., MIANJY, P. and MARINOV, T. (2016). Stochastic optimization for multiview representation learning using partial least squares. In *Proceedings of The 33rd International Conference on Machine Learning*.

BARBOUR, A. D. and CHEN, L. H. Y. (2005). *An introduction to Stein's method*, vol. 4. World Scientific.

BEN-TAL, A. and NEMIROVSKI, A. (2001). *Lectures on modern convex optimization: analysis, algorithms, and engineering applications*. SIAM.

BHARADWAJ, S., ARORA, R., LIVESCU, K. and HASEGAWA-JOHNSON, M. (2012). Multiview acoustic feature learning using articulatory measurements. In *Intl. Workshop on Stat. Machine Learning for Speech Recognition*. Citeseer.

CAI, T. T., LI, X., MA, Z. ET AL. (2016). Optimal rates of convergence for noisy sparse phase retrieval via thresholded wirtinger flow. *The Annals of Statistics* **44** 2221–2251.

CANDES, E. J., LI, X. and SOLTANOLKOTABI, M. (2015). Phase retrieval via wirtinger flow: Theory and algorithms. *IEEE Transactions on Information Theory* **61** 1985–2007.

CHAUDHURI, K., KAKADE, S. M., LIVESCU, K. and SRIDHARAN, K. (2009). Multi-view clustering via canonical correlation analysis. In *Proceedings of the 26th annual international conference on machine learning*. ACM.

COHEN, M. B., LEE, Y. T., MILLER, G., PACHOCKI, J. and SIDFORD, A. (2016). Geometric median in nearly linear time. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing*. ACM.

DHILLON, P., FOSTER, D. P. and UNGAR, L. H. (2011). Multi-view learning of word embeddings via cca. In *Advances in Neural Information Processing Systems 24* (J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira and K. Q. Weinberger, eds.). Curran Associates, Inc., 199–207.

ETHIER, S. N. and KURTZ, T. G. (2009). *Markov processes: characterization and convergence*, vol. 282. John Wiley & Sons.

EVANS, W. (1988). Partial differential equations.

GE, R., HUANG, F., JIN, C. and YUAN, Y. (2015). Escaping from saddle points-online stochastic gradient for tensor decomposition. In *COLT*.

HARDOON, D. R., SZEDMAK, S. and SHAWE-TAYLOR, J. (2004). Canonical correlation analysis: An overview with application to learning methods. *Neural computation* **16** 2639–2664.

JAIN, P., JIN, C., KAKADE, S. M., NETRAPALLI, P. and SIDFORD, A. (2016). Streaming pca: Matching matrix bernstein and near-optimal finite sample guarantees for oja's algorithm. In *29th Annual Conference on Learning Theory*.

KIDRON, E., SCHECHNER, Y. Y. and ELAD, M. (2005). Pixels that sound. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE.

LASSERRE, J. B. (2001). Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization* **11** 796–817.

LI, C. J., WANG, M., LIU, H. and ZHANG, T. (2016a). Near-optimal stochastic approximation for online principal component estimation. *arXiv preprint arXiv:1603.05305* .

LI, C. J., WANG, Z. and LIU, H. (2016b). Online ica: Understanding global dynamics of nonconvex optimization via diffusion processes. In *Advances in Neural Information Processing Systems*.

NEMIROVSKI, A., JUDITSKY, A., LAN, G. and SHAPIRO, A. (2009). Robust stochastic approximation approach to stochastic programming. *SIAM Journal on optimization* **19** 1574–1609.

ØKSENDAL, B. (2003). Stochastic differential equations. In *Stochastic differential equations*. Springer, 65–84.

ROSS, N. ET AL. (2011). Fundamentals of stein's method. *Probab. Surv* **8** 210–293.

SHAMIR, O. (2015). Fast stochastic algorithms for svd and pca: Convergence properties and convexity. *arXiv preprint arXiv:1507.08788* .

SOCHER, R. and FEI-FEI, L. (2010). Connecting modalities: Semi-supervised segmentation and annotation of images using unaligned text corpora. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE.

VINOKOUROV, A., SHAWE-TAYLOR, J. and CRISTIANINI, N. (2002). Inferring a semantic representation of text via cross-language correlation analysis. In *NIPS*, vol. 1.

ZHAO, T., WANG, Z. and LIU, H. (2015). A nonconvex optimization framework for low rank matrix estimation. In *Advances in Neural Information Processing Systems*.