J. Korean Soc. Ind. Appl. Math. Vol.25, No.4, 162–172, 2021 https://doi.org/10.12941/jksiam.2021.25.162

STOCHASTIC GRADIENT METHODS FOR L^2-WASSERSTEIN LEAST SQUARES PROBLEM OF GAUSSIAN MEASURES

SANGWOON YUN[1], XIANG SUN[2], AND JUNG-IL CHOI[3†]

[1]DEPARTMENT OF MATHEMATICS EDUCATION, SUNGKYUNKWAN UNIVERSITY, SEOUL, 03063, REPUBLIC OF KOREA

[2]SCHOOL OF MATHEMATICAL SCIENCES, OCEAN UNIVERSITY OF CHINA, QINGDAO, 266100, CHINA

[3]SCHOOL OF MATHEMATICS AND COMPUTING (COMPUTATIONAL SCIENCE & ENGINEERING), YONSEI UNIVERSITY, SEOUL, 03722, REPUBLIC OF KOREA
Email address: †`jic@yonsei.ac.kr`

ABSTRACT. This paper proposes stochastic methods to find an approximate solution for the L^2-Wasserstein least squares problem of Gaussian measures. The variable for the problem is in a set of positive definite matrices. The first proposed stochastic method is a type of classical stochastic gradient methods combined with projection and the second one is a type of variance reduced methods with projection. Their global convergence are analyzed by using the framework of proximal stochastic gradient methods. The convergence of the classical stochastic gradient method combined with projection is established by using diminishing learning rate rule in which the learning rate decreases as the epoch increases but that of the variance reduced method with projection can be established by using constant learning rate. The numerical results show that the present algorithms with a proper learning rate outperforms a gradient projection method.

1. INTRODUCTION

The Wasserstein distance between two Gaussian measures μ_1 and μ_2 with zero mean and covariance matrices Σ_1 and Σ_2, respectively, is given as follows [1]:

$$d_W(\Sigma_1, \Sigma_2) = \sqrt{\mathrm{Tr}(\Sigma_1 + \Sigma_2) - 2\mathrm{Tr}\left(\Sigma_1^{\frac{1}{2}}\Sigma_2\Sigma_1^{\frac{1}{2}}\right)^{\frac{1}{2}}}.$$

If $w = (w_1, \dots, w_n)$ be a positive provability vector in $\Re^n$ and $\mathcal{S} = (\Sigma_1, \dots, \Sigma_n)$ in which Σ_j (for $j = 1, \dots, n$) are positive definite matrices, then the w-weighted Wasserstein barycenter of

Received September 15 2021; Accepted December 8 2021; Published online December 25 2021.

2000 *Mathematics Subject Classification.* 90C15, 90C25, 90C30.

Key words and phrases. L^2-Wasserstein least squares problem, stochastic variance reduction method, learning rate, Gaussian measure, positive definite matrix.

† Corresponding author.

Gaussian measures μ_j with zero mean and covariance matrices Σ_j, respectively, is determined by a solution of the following minimization problem (see [2] and references therein):

$$\min_{X \succeq 0} f(X) := \sum_{j=1}^{n} w_j d_W^2(X, \Sigma_j), \tag{1.1}$$

where $X \succeq Y$ means that $X - Y$ is positive semi-definite. Agueh and Carlier (Theorem 6.1, [3]) proved that the problem (1.1) has a unique positive definite solution. Note that the problem is related to the multi-marginal optimal transport problem;

$$\min_{\gamma \in \Pi(\mu_1, \ldots, \mu_n)} \int_{(\Re)^d)^n} \left[\sum_{j=1}^{n} w_j \| x_j - C(x) \|^2 \right] d\gamma(x_1, \ldots, x_n),$$

where μ_j are probability measures with a finite second moment and C is the arithmetic barycenter $C(x) = \sum_{j=1}^{n} w_j x_j$ and $\Pi(\mu_1, \ldots, \mu_n)$ is the set of probability measures on $(\mathbb{R}^d)^n$ having marginals $\mu_1, \ldots, \mu_n$. There are articles [4, 5] to study connections between Wasserstein barycenters and optimal transports. Wasserstein barycenters have been attracted in applications of statistics, image processing, and machine learning [6, 7, 8, 9, 10]. Several methods have been proposed to solve the problem (1.1). Álvarez Esteban et al. [11] proposed a fixed point iteration method based on the nonlinear matrix equation;

$$X = \frac{1}{n} \sum_{j=1}^{n} (X^{\frac{1}{2}} \Sigma_j X^{\frac{1}{2}})^{\frac{1}{2}}.$$

Recently, Kum and Yun [2] proposed three gradient projection methods - classical gradient projection method with Armijo line search, gradient projection method with a fixed step size based on pre-evaluated Lipschitz constant, and accelerated one.

The objective function of the problem (1.1) is the sum of several differentiable functions, and the evaluation of the gradient of the objective function is the main computational cost for methods, such as the gradient projection method, that require finding the descent direction based on the gradient. This motivates us to adapt stochastic gradient (projection) methods for solving the problem (1.1). In the stochastic gradient method, we only need to evaluate one or a few gradients of component functions consisting of the objective function to update iterates. This paper proposes a projected version of the classical stochastic gradient method that evaluates one component function gradient at each iteration and proposes also a projected one of the stochastic variance reduced gradient method [12]. We show the global convergence property of them by using the analysis given in [13, 14].

This paper is organized as follows. In section 2, we briefly review the boundedness of the solution of the problem (1.1), the Lipschitz continuity of the gradient of the objective function in the problem (1.1), and the gradient projection method studied in [2]. In section 3, we describe stochastic gradient (projection) methods and analyze their convergence properties. Section 4 reports numerical results for finding the Wasserstein barycenter of Gaussian measures on randomly generated matrices using proposed methods. Numerical comparisons with the gradient projection method [2] are also given. Finally, concluding remarks are included in section 5.

2. Lipschitz continuity and gradient projection method

In this section, we review the boundedness of the solution of the problem (1.1), and the Lipschitz continuity of the gradient of the objective function in (1.1), referring to [2]. The gradient projection method proposed in [2] is also briefly described.

The solution of the problem (1.1) is in the Löwner order interval $[\underline{\lambda} I, \bar{\lambda} I] := \{X : \underline{\lambda} I \preceq X \preceq \bar{\lambda} I\}$, where

$$\underline{\lambda} := \left[\sum_{j=1}^{n} w_j \sqrt{\lambda_{\min}(\Sigma_j)} \right]^2, \quad \bar{\lambda} := \left[\sum_{j=1}^{n} w_j \sqrt{\lambda_{\max}(\Sigma_j)} \right]^2.$$

Moreover, $\lambda_{\min}(\Sigma)$ and $\lambda_{\max}(\Sigma)$ denote the minimum and maximum eigenvalue of Σ, respectively. Here, $X \preceq Y$ means that $Y - X$ is positive semi-definite. Hence the problem (1.1) can be expressed as the bound constraint minimization problem:

$$\min_{X \in \mathcal{D}} f(X), \tag{2.1}$$

where $\mathcal{D} := [\underline{\lambda} I, \bar{\lambda} I]$.

The above facts for the solution of the problem (1.1) hold the following proposition for the Lipschitz continuity of the slope of the objective function in (1):

Proposition 2.1. [2, Theorem 3.1] *For* $\underline{\lambda} I \preceq X, Y \preceq \bar{\lambda} I$ *with* $X \neq Y$,

$$\frac{\| \nabla f(X) - \nabla f(Y) \|_F}{\| X - Y \|_F} \leq L := \frac{L_{\max}^2}{2L_{\min}^3},$$

where $L_{\min} = \min_{1 \leq j \leq n}\{\lambda_{\min}(\Sigma_j)\}$ *and* $L_{\max} = \max_{1 \leq j \leq n}\{\lambda_{\max}(\Sigma_j)\}$.

The gradient projection method (GPM) proposed in [2] uses the Armijo rule along the feasible direction for selecting a stepsize. We compare this method with our proposed method for the numerical experiments. The algorithm framework is described below.

Algorithm 1 Gradient projection method (GPM)

```
Choose $X^0 \in \mathcal{D}$. Initialize $k = 0$. Update $X^{k+1}$ from $X^k$ by the following template:
    Step 1.: Find $\bar{X}^k = [X^k - \nabla f(X^k)]^+$,
    Step 2.: Select a stepsize $t^k$,
    Step 3.: $X^{k+1} = X^k + t^k(\bar{X}^k - X^k)$.
Here, $[\cdot]^+$ denotes the projection on the set $[\underline{\lambda} I, \bar{\lambda} I]$.
```

The stepsize t^k is chosen by the following Armijo rule over the interval $[0, 1]$:

Let t^k be the largest element of $\{\xi^j\}_{j=0,1,\ldots}$ satisfying

$$f(X^k + t^k D^k) \leq f(X^k) - \sigma t^k \langle \nabla f(X^k),\, D^k \rangle,$$

where $0 < \xi < 1$, $0 < \sigma < 1$, and $D^k = \bar{X}^k - X^k$.

3. STOCHASTIC GRADIENT (PROJECTION) METHOD

In this section, we describe stochastic gradient (projection) methods for solving the problem (1.1), more precisely the problem (2.1), and analyze their convergence properties.

The first stochastic gradient (projection) method (SGM) is a projected version of the classical stochastic gradient method that uses only one randomly selected component function gradient at each iteration. Hence the computational cost to evaluate the direction is $\frac{1}{n}$ that of the gradient projection method. This method is formally described below.

Algorithm 2 Stochastic gradient (projection) method (SGM)

```
Choose a positive definite matrix $X^0 \in \mathcal{D}$.
for $k = 0, 1, \ldots,$
        $\hat{X}_0^k = X^k$.
        for $t = 1, \ldots, n$
            Step 1.: Randomly pick $i_t \in \{1, \ldots, n\}$.
            Step 2.: $\hat{X}_t^k = [\hat{X}_{t-1}^k - \eta_t^k w_{i_t} \nabla d_W^2(\hat{X}_{t-1}^k, \Sigma_{i_t})]^+$.
        end
        $X^{k+1} = \hat{X}_n^k$.
end
```

The second step of SGM is the projection of the matrix $S \in \mathcal{S}^d$, where $\mathcal{S}^d$ is the set of $d \times d$ symmetric matrices, onto the set $\mathcal{D}$ and is formulated as the following minimization problem,

$$\min_{X \in \mathcal{D}} \left\| X - S_t^k \right\|_F, \tag{3.1}$$

where $S_t^k = \hat{X}_{t-1}^k - \eta_t^k w_{i_t}(\nabla d_W^2(\hat{X}_{t-1}^k, \Sigma_{i_t}))$. The solution of this problem is

$$U_t^k \mathrm{Diag}\left(\min(\max(\underline{\lambda}, \lambda_1), \bar{\lambda}), \ldots, \min(\max(\underline{\lambda}, \lambda_d), \bar{\lambda})\right) (U_t^k)^T,$$

where $\lambda_1 \geq \cdots \geq \lambda_d$ are the eigenvalues of S_t^k and U_t^k is a corresponding orthogonal matrix of eigenvalues of S_t^k. This result can be found in [15].

The problem (3.1) can also be formulated as

$$\hat{X}_t^k = \arg\min_{X \in \mathcal{S}^d} \frac{1}{2\eta_t} \left\| X - [\hat{X}_{t-1}^k - \eta_t^k w_{i_t} \nabla d_W^2(\hat{X}_{t-1}^k, \Sigma_{i_t})] \right\|_F^2 + \iota_{\mathcal{D}}(X),$$

where $\iota_{\mathcal{D}}(X)$ is an indicator function of $\mathcal{D}$, i.e., $\iota_{\mathcal{D}}(X) = 0$ if $X \in \mathcal{D}$ or ∞ otherwise. Hence the proposed SGM can be considered as a specific version of a proximal stochastic gradient method (PSGM). Now, we establish the convergence property of SGM in the following theorem by using the convergence analysis for PSGM. Its proof can be induced from [13, Proposition 9].

Theorem 3.1. *Let $\{X^k\}$ be the sequence generated by SGM with the learning rate η_t^k satisfying $\eta_n^{k-1} \geq \eta_1^k$, $\eta_{t-1}^k \geq \eta_t^k$, $\eta_1^k \to 0$, and $\sum_{k=0}^{\infty} \eta_1^k = \infty$. Then, with probability* 1,

$$\lim_{k\to\infty} \inf f(X^k) = f^*,$$

where f^* is the optimal value. Furthermore, if $\sum_{k=0}^{\infty} (\eta_t^k)^2 < \infty$ for all $t = 1, \ldots, n$, then $\{X^k\}$ converges to the solution of the problem (1.1) with probability 1.

The proposed SGM has a disadvantage from the randomness, which causes variance; see [12] for details. In order to overcome this disadvantage, we propose a projected version of the stochastic variance reduced gradient method [12]. This is the second stochastic gradient (projection) method, which is formally described below.

Algorithm 3 Stochastic variance reduced gradient (projection) method (SVRGM)

```
Choose a positive definite matrix $X^0 \in \mathcal{D}$.
for $k = 0, 1, \ldots,$
        Step 1.: $\tilde{X} = X^k$.
        Step 2.: $g = \nabla f(\tilde{X})$.
        Step 3.: $\hat{X}_0^k = X^k$.
probability $Q = \{q_1, \ldots, q_n\}$ on $\{1, \ldots, n\}$
    for $t = 1, \ldots, m$
        Step 1.: Randomly pick $i_t \in \{1, \ldots, n\}$.
        Step 2.: $\hat{X}_t^k = \left[ \hat{X}_{t-1}^k - \eta \left( \frac{w_{i_t}(\nabla d_W^2(\hat{X}_{t-1}^k, \Sigma_{i_t}) - \nabla d_W^2(\tilde{X}, \Sigma_{i_t}))}{n q_{i_t}} + g \right) \right]^+$.
    end
    $X^{k+1} = \frac{1}{m} \sum_{j=1}^{m} \hat{X}_j^k$.
end
```

Similarly, as described in SGM, the second step of SVRGM is expressed as follows:

$$\hat{X}_t^k = \operatorname*{arg\,min}_{X \in \mathcal{S}^d} \frac{1}{2\eta} \left\| X - \left[\hat{X}_{t-1}^k - \eta \left(\frac{w_{i_t}(\nabla d_W^2(\hat{X}_{t-1}^k, \Sigma_{i_t}) - \nabla d_W^2(\tilde{X}, \Sigma_{i_t}))}{n q_{i_t}} + g \right) \right] \right\|_F^2 + \iota_{\mathcal{D}}(X).$$

Hence the proposed SVRGM can be considered as a specific version of a proximal stochastic variance reduced method (PSVRM). Before establishing a theorem for the convergence property of the proposed SVRGM, we need the strong convexity of the objective function and the Lipschitz continuity of the component functions in the problem (1.1).

The objective function $f(X)$ is strictly convex on the set of positive definite matrices [16], and so it is easily derived that the function $f(X)$ is strongly convex on the bounded and closed set $\mathcal{D}$, i.e., there exists $\mu > 0$ such that $X, Y \in \mathcal{D}$,

$$f(Y) \geq f(X) + \langle \nabla f(X),\, Y - X \rangle + \frac{\mu}{2} \| Y - X \|_F^2.$$

From Proposition 2.1, we can deduce the Lipschitz continuity of the gradient of each component function as follows:

$$\frac{\| \nabla d_W^2(X, \Sigma_j) - \nabla d_W^2(Y, \Sigma_j) \|_F}{\| X - Y \|_F} \leq L_j := \frac{\lambda_{\max}(\Sigma_j)^2}{2\lambda_{\min}(\Sigma_j)^3}, \quad j = 1, \ldots, n.$$

Now, we establish the convergence property of SVRGM in the following theorem by adapting the convergence analysis for PSVRM. Its proof can be induced from [14, Theorem 1].

Theorem 3.2. *Let* $X^* = \arg\min_{X \in \mathcal{D}} f(X)$ *and* $L_Q = \max_i L_i/(q_i n)$. *In addition, assume that* $0 < \eta < 1/(4L_Q)$ *and* m *is sufficiently large so that*

$$\rho = \frac{1}{\mu\eta(1 - 4L_q)m} + \frac{4L_Q\eta(m+1)}{(1 - 4L_Q)m} < 1.$$

Then SVRGM has geometric convergence in expectation:

$$\mathrm{E}(f(X^k)) - f(X^*) \leq \rho^k [f(X^0) - f(X^*)].$$

4. Numerical Experiments

In this section, we report the performance of SGM, SVRGM, and GPM with the Armijo rule on n randomly generated matrices of the size $d \times d$ with $w_i = 1/n$ for all $i = 1, \ldots, n$. The $d \times d$ positive definite matrices $\Sigma_1, \ldots, \Sigma_n$ for the first test experiment are randomly generated by MATLAB pseudo-code as follows:

```
$for$  $i = 1 : n$
       $[Q,\ ] = qr(randn(d));$
       $\Sigma_i = Q * diag(0.1 + 99.9 * rand(d, 1)) * Q';$
```

The eigenvalues of generated matrices are randomly distributed in the interval $[0.1, 100]$.

In the second test experiment, each $d \times d$ positive definite matrix Σ_i is generated from a Wishart $W_d(I; d)$ distribution [11], independently of the others. We set the dimension of positive definite matrices as 10 and the number of the positive matrices as 1000 for the first experiment and 500 for the second experiment.

All runs are performed on a Laptop with Intel Core i7-10510U CPU (2.30GHz) and 16GB Memory, running 64-bit Windows 10 and MATLAB (Version 9.8). We choose the initial iterate to be $X^{(0)} = 0.5(L_{\min} + L_{\max})I$ for all algorithms throughout the experiments, where $L_{\min}$ and $L_{\max}$ are defined in Proposition 2.1. We set $\xi = 0.5$ and $\sigma = 0.1$ for GPM as suggested in [2]. The learning rate for SGM is set to

$$\eta_t^k = 10 \left(1 + 0.1 \left[k + \frac{t}{n} \right] \right)^{-1},$$

TABLE 1. Comparison of the results obtained from GPM with those obtained from SGM and SVRGM for the first test experiment. Note that obj, CPU, and epochs indicate the final objective value, the elapsed CPU time (seconds), and the number of epochs, respectively.

		$\epsilon = 5 \times 10^{-3}$			$\epsilon = 10^{-6}$		
		GPM	SGM	SVRGM	GPM	SGM	SVRGM
1	obj	-449.7121141	-449.7121212	-449.7127527	-449.7132076	-449.7131899	-449.7132076
	CPU	191.53	170.42	1.36	257.64	785.06	1.61
	epochs	329	238	1	1088	3000	3
2	obj	-447.2172693	-447.2173241	-447.2180269	-447.2183429	-447.2183179	-447.2183429
	CPU	138.61	56.84	0.86	250.58	779.06	1.33
	epochs	333	198	1	1086	3000	3
3	obj	-445.0736002	-445.0736122	-445.0744007	-445.0746755	-445.0746580	-445.0746755
	CPU	182.66	129.77	1.23	251.56	787.98	1.20
	epochs	335	344	1	1086	3000	3
4	obj	-442.6765796	-442.6765866	-442.6771942	-442.6776407	-442.6776013	-442.6776407
	CPU	152.42	134.06	0.58	251.38	850.70	1.45
	epochs	338	269	1	1085	3000	3
5	obj	-447.7992681	-447.7992682	-447.7999917	-447.8003396	-447.8003279	-447.8003396
	CPU	164.38	143.69	1.06	292.89	919.69	1.50
	epochs	332	241	1	1087	3000	3

referring to [12]. The learning rate for SVRGM is usually set as a fixed constant $\eta = 0.1/L$ [12, 14], where L is the Lipschitz constant defined in Proposition 2.1. However, we set the rate as $\tilde{\eta}_t^k = \max\left\{\eta, \eta_t^k\right\}$ in order to improve the computational performance of SVRGM at an initial stage. Moreover, we set $m = n$ and use uniform sampling, i.e., $q_i = 1/n$ for all $i = 1, \ldots, n$.

To perform a comparison, we first ran GPM until satisfying the following criterion,

$$\left\| \left[X^k - \nabla f\left(X^k\right)\right]^+ - X^k \right\|_F < \epsilon.$$

We then run SGM and SVRGM until satisfying the criterion based on the value of the objective function obtained from GPM. Note that the value $[X^k - \nabla f(X^k)]^+ - X^k$ is a descent direction at X^k for GPM [2], and if this value is zero, then the corresponding point is optimal. In our experiments, we use $\epsilon = 5 \times 10^{-3}$, and $\epsilon = 10^{-6}$ for the first test experiment and $\epsilon = 10^{-2}$ and $\epsilon = 10^{-5}$ for the second test experiment. All the algorithms are also terminated when the epoch reaches 3000.

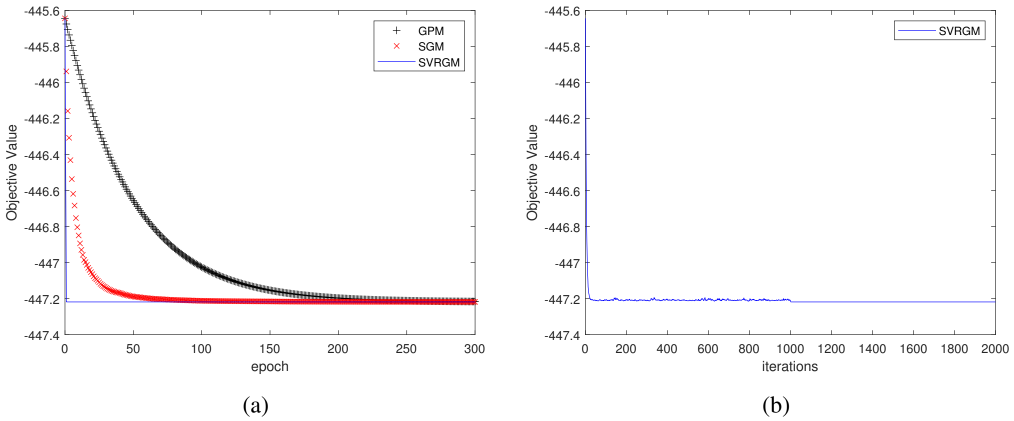


FIGURE 1. (a) Objective value versus epoch for the first test experiment. (b) Objective value versus total sub-iterations (2 epochs) of SVRGM for the first test experiment.

In Table 1, we report the final objective value, the elapsed CPU time in seconds, and the number of epochs for three methods (GPM, SGM, and SVRGM) with five random data sets for the first test experiment. Note that one epoch is the one iteration for GPM but requires n sub-iterations for SGM and SVRGM. Then the total number of gradients evaluation of component functions for GPM is equal to that for SGM, but this total number for SVRGM is two times larger than that for GPM and SGM if we set $m = n$. Table 1 indicates that SVRGM is the best algorithm for both low and medium accuracy. SGM performs better than GPM for low accuracy. However, SGM does not reach the objective value of GPM for all cases within 3000 epochs for medium accuracy. Overall, SVRGM can give a reasonable estimate for the solution in just a few epochs. Figure 1 (a) and (b) show the comparison of decrements of the objective values with epochs for three methods and the decrement of the objective value with total sub-iterations for two epochs when SVRGM is applied, respectively. From Fig. 1, we can clearly observe that SVRGM quickly converges.

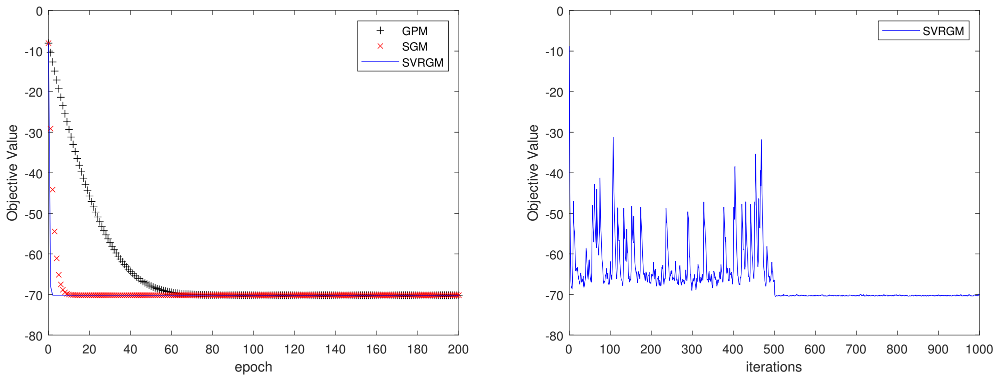
In Table 2, we report our numerical results, which address the final objective value, the elapsed CPU time in seconds, and the number of epochs for three methods with five random data sets for the second test experiment. Table 2 indicates that SVRGM is the best algorithm for both low and medium accuracy. In this experiment, GPM performs better than SGM for both low and medium accuracy. Similar to the first experiment, SVRGM can give a reasonable estimate for the solution in just a few epochs. Figure 2 shows the comparison of decrements of the objective values for three algorithms and shows the decrement of the objective value for two epochs when SVRGM is applied. In contrast to the first experiment, SVRGM shows oscillation behavior during sub-iterations of the first epoch.

TABLE 2. Comparison of the results obtained from GPM with those obtained from SGM and SVRGM for the second test experiment. Note that obj, CPU, and epochs indicate the final objective value, the elapsed CPU time (seconds), and the number of epochs, respectively.

		$\epsilon = 10^{-2}$			$\epsilon = 10^{-5}$		
		GPM	SGM	SVRGM	GPM	SGM	SVRGM
1	obj	-70.9139991	-70.9140293	-70.9145738	-70.9146316	-70.9145481	-70.9146316
	CPU	17.55	38.20	1.14	52.88	991.09	3.77
	epochs	126	269	3	241	3000	5
2	obj	-70.6459293	-70.6459457	-70.6464958	-70.6465338	-70.6464659	-70.6465338
	CPU	15.19	44.27	0.72	64.17	889.63	2.27
	epochs	121	315	3	233	3000	5
3	obj	-71.0682832	-71.0682975	-71.0689221	-71.0689281	-71.0688384	-71.0689281
	CPU	14.02	24.47	0.81	58.48	817.02	1.41
	epochs	112	169	3	224	3000	5
4	obj	-69.7738124	-69.7738181	-69.7744105	-69.7744282	-69.7743474	-69.7744282
	CPU	14.89	42.23	0.69	35.00	776.27	2.52
	epochs	117	298	3	229	3000	5
5	obj	-69.6747806	-69.6747817	-69.6753501	-69.6753602	-69.6752906	-69.6753602
	CPU	14.00	51.42	1.52	55.20	839.20	1.17
	epochs	112	333	3	223	3000	5

5. CONCLUSION

We have proposed stochastic methods, i.e., the stochastic gradient projection method (SGM) and the stochastic variance reduced gradient projection method (SVRGM), to compute the Wasserstein barycenter of Gaussian measures and analyze their convergence properties by adapting the analysis of the proximal stochastic gradient method and the proximal stochastic variance reduced method. With an appropriate choice of learning rate (stepsize), the proposed stochastic gradient projection methods can outperform the classical gradient projection method in the initial stage. Furthermore, the proposed stochastic variance reduced gradient projection method outperforms the classical gradient projection method for both low and medium accuracy. Designing the manifold version of stochastic methods for the Wasserstein barycenter of Gaussian measures is an interesting future topic.



(a) (b)

FIGURE 2. (a) Objective value versus epoch for the second test experiment. (b) Objective value versus total sub-iterations (2 epochs) of SVRGM for the second test experiment.

ACKNOWLEDGMENTS

This work was supported by the National Research Foundation of Korea (NRF) grants funded by the Ministry of Science, ICT & Future Planning (NRF-20151009350, NRF-2016R1-A5A1008055, NRF-2016R1D1A1B03934371 and NRF-2019R1F1A1057051).

REFERENCES

[1] C. R. Givens and R. M. Shortt, *A class of wasserstein metrics for probability distributions*, Mich. Math. J. **31** (1984), 231–240.

[2] S. Kum and S. Yun, *Gradient projection methods for the n-coupling problem*, J. Korean Math. Soc. **56** (2019), 1001–1016.

[3] M. Agueh and G. Carlier, *Barycenters in the wasserstein space*, SIAM J. Math. Anal. **43** (2011), 904–924.

[4] Y.-H. Kim and B. Pass, *Multi-marginal optimal transport on Riemannian manifolds*, Amer. J. Math. **137** (2015), 1045–1060.

[5] B. Pass, *The local structure of optimal measures in the multi-marginal optimal transportation problem*, Calc. Var. Partial Differential Equations **43** (2012), 529–536.

[6] G. Carlier and I. Ekeland, *Matching for teams*, Econ. Theory **42** (2010), 397–418.

[7] G. Carlier, A. Oberman, and E. Oudet, *Numerical methods for matching for teams and wasserstein barycenters*, ESAIM: M2AN **49** (2015), 1621–1642.

[8] A. Mallasto and A. Feragen, *Learning from uncertain curves: The 2-wasserstein metric for gaussian processes*, In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, Advances in Neural Information Processing Systems 30, pages 5660–5670. Curran Associates, Inc., 2017.

[9] J. Rabin, G. Peyré, J. Delon, and M. Bernot, *Wasserstein barycenter and its application to texture mixing*, In Proceedings of the Third International Conference on Scale Space and Variational Methods in Computer Vision, SSVM'11, pages 435–446, Berlin, Heidelberg, 2012.

[10] S. Srivastava, C. Li, and D. B. Dunson, *Scalable bayes via barycenter in wasserstein space*, J. Mach. Learn. Res. **19** (2018), 312–346.

[11] P. C. Álvarez Esteban, E. del Barrio, J. Cuesta-Albertos, and C. Matrán, *A fixed-point approach to barycenters in wasserstein space*, J. Math. Anal. Appl. **441** (2016), 744–762.

[12] R. Johnson and T. Zhang, *Accelerating stochastic gradient descent using predictive variance reduction*, in Adv. Neural Inf. Process. Syst. 26, NIPS'13, USA 2013.

[13] D. P. Bertsekas, *Incremental proximal methods for large scale convex optimization*, Math. Program. Ser. B **129** (2011), 163-–195.

[14] L. Xiao and T. Zhang, *A proximal stochastic gradient method with progressive variance reduction*, SIAM J. Optim. **24** (2014), 2057-–2075.

[15] A. S. Lewis and J. Malick, *Alternating projections on manifolds*, Math. Oper. Res. **33** (2008), 216–234.

[16] R. Bhatia, T. Jain, and Y. Lim, *On the bures-wasserstein distance between positive definite matrices*, Expositiones Mathematicae **37** (2019), 165–191.