Stochastic Non-Convex Ordinal Embedding with Stabilized Barzilai-Borwein Step Size

Ke Ma,[1,2] **Jinshan Zeng,**[3,4] **Jiechao Xiong,**[5] **Qianqian Xu,**[1]
Xiaochun Cao,[1*] **Wei Liu,**[5] **Yuan Yao**[4*]

[1] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences
[2] School of Cyber Security, University of Chinese Academy of Sciences
[3] School of Computer Information Engineering, Jiangxi Normal University
[4] Department of Mathematics, Hong Kong University of Science and Technology [5] Tencent AI Lab
{make, xuqianqian, caoxiaochun}@iie.ac.cn, jsh.zeng@gmail.com
jcxiong@tencent.com, wliu@ee.columbia.edu, yuany@ust.hk

Abstract

Learning representation from relative similarity comparisons, often called ordinal embedding, gains rising attention in recent years. Most of the existing methods are batch methods designed mainly based on the convex optimization, say, the projected gradient descent method. However, they are generally time-consuming due to that the singular value decomposition (SVD) is commonly adopted during the update, especially when the data size is very large. To overcome this challenge, we propose a stochastic algorithm called SVRG-SBB, which has the following features: (a) SVD-free via dropping convexity, with good scalability by the use of stochastic algorithm, i.e., stochastic variance reduced gradient (SVRG), and (b) adaptive step size choice via introducing a new stabilized Barzilai-Borwein (SBB) method as the original version for convex problems might fail for the considered stochastic *non-convex* optimization problem. Moreover, we show that the proposed algorithm converges to a stationary point at a rate $\mathcal{O}(\frac{1}{T})$ in our setting, where T is the number of total iterations. Numerous simulations and real-world data experiments are conducted to show the effectiveness of the proposed algorithm via comparing with the state-of-the-art methods, particularly, much lower computational cost with good prediction performance.

Introduction

Ordinal embedding aims to learn representation of data objects as points in a low-dimensional space. The distances among these points agree with a set of relative similarity comparisons as well as possible. Relative comparisons are often collected by workers who are asked to answer the following question:

"Is the similarity between object i and j larger than the similarity between l and k?"

The feedback of individuals provide us a set of quadruplets, *i.e.*, $\{(i, j, l, k)\}$, which can be treated as the supervision for ordinal embedding. Without prior knowledge, the relative similarity comparisons always involve all objects and the number of potential quadruplet could be $\mathcal{O}(n^4)$.

The ordinal embedding problem was firstly studied by (Shepard 1962a; 1962b; Kruskal 1964a; 1964b) in the psychometric society. In recent years, it has drawn a lot of attention in machine learning (Jamieson and Nowak 2011b; Kleindessner and Luxburg 2014; Arias-Castro 2017; Jain, Jamieson, and Nowak 2016), statistic ranking (McFee and Lanckriet 2011; Jamieson and Nowak 2011a), artificial intelligence (Heikinheimo and Ukkonen 2013; Wilber, Kwak, and Belongie 2014), information retrieval (Song et al. 2015), and computer vision (Wah et al. 2014; Wilber et al. 2015), etc.

One of the typical methods for ordinal embedding problem is the well-known Generalized Non-Metric Multidimensional Scaling (GNMDS) (Agarwal et al. 2007), which aims at finding a low-rank Gram (or kernel) matrix $\mathbf{G}$ in Euclidean space such that the pairwise distances between the embedding of objects in Reproducing Kernel Hilbert Space (RKHS) satisfy the relative similarity comparisons. As GNMDS uses hinge loss to model the relative similarity between the objects, it neglects the information provided by satisfied constraints in finding the underlying structure in the low-dimensional space. To alleviate this issue, the Crowd Kernel Learning (CKL) was proposed by (Tamuz et al. 2011) via employing a scale-invariant loss function. However, the objective function used in CKL only considers the constraints which are strongly violated. Latter, (van der Maaten and Weinberger 2012) proposed the Stochastic Triplet Embedding (STE) that jointly penalizes the violated constraints and rewards the satisfied constraints, via using the logistic loss function. Note that the aforementioned three typical methods are based on the convex formulations, and also employ the projected gradient descent method and singular value decomposition (SVD) to obtain the embedding. However, a huge amount of comparisons and the computational complexity of SVD significantly inhibit their usage to large scale and on-line applications. Structure Preserving Embedding (SPE) (Shaw and Jebara 2009) and Local Ordinal Embedding (LOE) (Terada and Luxburg 2014) embed unweighted nearest neighbor graphs to Euclidean spaces with convex and non-convex objective functions. The nearest neighbor adjacency matrix

---

can be transformed into ordinal constraints, but it is not a standard equipment in those scenarios which involve relative comparisons. With this limitation, SPE and LOE are not suitable for ordinal embedding via quadruplets or triple comparisons.

In contrast to the kernel-learning or convex formulation of ordinal embedding, the aforementioned methods have the analogous non-convex counterparts. The non-convex formulations directly obtain the embedding instead of the Gram matrix. Batch gradient descent is not suitable for solving these large scale ordinal embedding problems because of the expense of full gradients in each iteration. Stochastic gradient descent (SGD) is a common technology in this situation as it takes advantage of the stochastic gradient to devise fast computation per iteration. In (Ghadimi and Lan 2013), the $\mathcal{O}(\frac{1}{\sqrt{T}})$ convergence rate of SGD for the stochastic non-convex optimization problem was established, in the sense of convergence to a stationary point, where T is the total number of iterations. As SGD has slow convergence due to the inherent variance, stochastic variance reduced gradient (SVRG) method was proposed in (Johnson and Zhang 2013) to accelerate SGD. For the strongly convex function, the linear convergence of SVRG with Option-II was established in (Johnson and Zhang 2013), and latter the linear convergence rates of SVRG with Option-I and SVRG incorporating with the Barzilai-Borwein (BB) step size were shown in (Tan et al. 2016). In the non-convex case, the $\mathcal{O}(\frac{1}{T})$ convergence rates of SVRG in the sense of convergence to a stationary point were shown in (Reddi et al. 2016; Allen-Zhu and Hazan 2016) under certain conditions.

Although the BB step size has been incorporated into SVRG and its effectiveness has been shown in (Tan et al. 2016) for the strongly convex case, it might not work when applied to some stochastic non-convex optimization problems. Actually, in our latter simulations, we found that the absolute value of the original BB step size is unstable when applied to the stochastic non-convex ordinal embedding problem studied in this paper (see, Figure 1(a)). The absolute value of the original BB step size varies dramatically with respect to the epoch number. Such phenomenon is mainly due to without the strong convexity, the denominator of BB step size might be very close 0, and thus the BB step size broken up. This motivates us to investigate some new stable and adaptive strategies of step size for SVRG when applied to the stochastic non-convex ordinal embedding problem.

In this paper, we introduce a new adaptive step size strategy called stabilized BB (SBB) step size via adding another positive term to the absolute of the denominator of the original BB step size to overcome the instability of the BB step size, and then propose a new stochastic algorithm called SVRG-SBB via incorporating the SBB step size for fast solving the considered non-convex ordinal embedding model. In a summary, our main contribution can be shown as follows:

- We propose a non-convex framework for the ordinal embedding problem via considering the optimization problem with respect to the original embedding variable but not its Gram matrix. By exploiting this idea, we get rid of the positive semi-definite (PSD) constraint on the Gram matrix, and thus, our proposed algorithm is SVD-free and has good scalability.

- The introduced SBB step size can overcome the instability of the original BB step size when the original BB step size does not work. More importantly, the proposed SVRG-SBB algorithm outperforms most of the the state-of-the-art methods as shown by numerous simulations and real-world data experiments, in the sense that SVRG-SBB often has better generalization performance and significantly reduces the computational cost.

- We establish $O(\frac{1}{T})$ convergence rate of SVRG-SBB in the sense of convergence to a stationary point, where T is the total number of iterations. Such convergence result is comparable with the existing best convergence results in literature.

Stochastic Ordinal Embedding

A. Problem Description

There is a set of n objects $\{o_1, \ldots, o_n\}$ in abstract space $\mathbf{O}$. We assume that a certain but unknown dissimilarity function $\xi : \mathbf{O} \times \mathbf{O} \to \mathbb{R}^+$ assigns the dissimilarity value ξ_{ij} for a pair of objects (o_i, o_j). With the dissimilarity function ξ, we can define the ordinal constraint (i, j, l, k) from a set $\mathcal{P} \subset [n]^4$, where

$$\mathcal{P} = \{(i, j, l, k) \mid \text{if exist } o_i, o_j, o_k, o_l \text{ satisfy } \xi_{ij} < \xi_{lk}\}$$

and $[n]$ is the set of $\{1, \ldots, n\}$. Our goal is to obtain the representations of $\{o_1, \ldots, o_n\}$ in Euclidean space $\mathbb{R}^p$ where p is the desired embedding dimension. The embedding $\mathbf{X} \in \mathbb{R}^{n \times d}$ should preserve the ordinal constraints in $\mathcal{P}$ as much as possible, which means

$$0(i, j, l, k) \in \mathcal{P} \Leftrightarrow \xi_{ij} < \xi_{lk} \Leftrightarrow d_{ij}^2(\mathbf{X}) < d_{lk}^2(\mathbf{X})$$

where $d_{ij}^2(\mathbf{X}) = \|\mathbf{x}_i - \mathbf{x}_j\|^2$ is the squared Euclidean distance between $\mathbf{x}_i$ and $\mathbf{x}_j$, and $\mathbf{x}_i$ is the ith row of $\mathbf{X}$.

Let $\mathbf{D} = \{d_{ij}^2(\mathbf{X})\}$ be the distance matrix of $\mathbf{X}$. There are some existing methods for recovering $\mathbf{X}$ given ordinal constraints on distance matrix $\mathbf{D}$. It is known that $\mathbf{D}$ can be determined by the Gram matrix $\mathbf{G} = \mathbf{X}\mathbf{X}^T = \{g_{ij}\}_{i,j=1}^n$ as $d_{ij}^2(\mathbf{X}) = g_{ii} - 2g_{ij} + g_{jj}$, and

$$\mathbf{D} = diag(\mathbf{G}) \cdot \mathbf{1}^T - 2\mathbf{G} + \mathbf{1} \cdot diag(\mathbf{G})^T$$

where $diag(\mathbf{G})$ is the column vector composed of the diagonal of $\mathbf{G}$ and $\mathbf{1}^T = [1, \ldots, 1]$. As $\text{rank}(\mathbf{G}) \leq \min(n, d)$ and it always holds $p \ll n$, these methods (Agarwal et al. 2007; Tamuz et al. 2011; van der Maaten and Weinberger 2012) can be generalized by a semidefinite programming (SDP) with low rank constraint,

$$\min_{\mathbf{G} \in \mathbb{R}^{n \times n}} L(\mathbf{G}) + \lambda \cdot \text{tr}(\mathbf{G}) \quad \text{s.t.} \quad \mathbf{G} \succeq 0 \qquad (1)$$

where $L(\mathbf{G}) = \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} l_p(\mathbf{G})$ is a convex function of $\mathbf{G}$ which satisfies

$$l_p(\mathbf{G}) : \begin{cases} > 0, & d_{ij}^2(\mathbf{X}) > d_{lk}^2(\mathbf{X}) \\ \leq 0, & \text{otherwise.} \end{cases}$$

tr($\mathbf{G}$) is the trace of matrix $\mathbf{G}$. To obtain the embedding $\mathbf{X} \in \mathbb{R}^{n \times p}$, the projected gradient descent is performed. The basic idea of the projected gradient descent method is: the batch gradient descent step with all $p \in \mathcal{P}$ is firstly used to learn the Gram matrix $\mathbf{G}$,

$$\mathbf{G}'_t = \mathbf{G}_{t-1} - \eta_t(\nabla L(\mathbf{G}_{t-1}) + \lambda \mathbf{I})$$

where t denotes the current iteration, η_t is the step size; then $\mathbf{G}'_t$ is projected onto a positive semi-definite (PSD) cone $\mathbb{S}_+$, $\mathbf{G}_t = \Pi_{\mathbb{S}_+}(\mathbf{G}'_t)$; and latter, once the iterates converge, the embedding X is obtained by projecting $\mathbf{G}$ onto the subspace spanned by the largest p eigenvectors of $\mathbf{G}$ via SVD.

B. Stochastic Non-convex Ordinal Embedding

Although the SDP (1) is a convex optimization problem, there are some disadvantages of this approach: (i) the projection onto PSD cone $\mathbb{S}_+$, which is performed by an expensive SVD due to the absence of any prior knowledge on the structure of $\mathbf{G}$, is a computational bottleneck of optimization; and (ii) the desired dimension of the embedding $\mathbf{X}$ is d and we hope the Gram matrix $\mathbf{G}$ satisfies rank($\mathbf{G}$) $\leq d$. If rank($\mathbf{G}$) $\gg d$, the freedom degree of $\mathbf{G}$ is much larger than $\mathbf{X}$ with over-fitting. Although $\mathbf{G}$ is a global optimal solution of (1), the subspace spanned by the largest d eigenvectors of $\mathbf{G}$ will produce less accurate embedding. We can tune the regularization parameter λ to force $\{\mathbf{G}_t\}$ to be low-rank and cross-validation is the most utilized technology. This needs extra computational cost. In summary, projection and parameter tuning render gradient descent methods computationally prohibitive for learning the embedding $\mathbf{X}$ with ordinal information $\mathcal{P}$. To overcome these challenges, we will exploit the non-convex and stochastic optimization techniques for the ordinal embedding problem.

To avoid projecting the Gram matrix $\mathbf{G}$ onto the PSD cone $\mathbb{S}_+$ and tuning the parameter λ, we directly optimize $\mathbf{X}$ and propose the unconstrained optimization problem of learning embedding $\mathbf{X}$,

$$\min_{\mathbf{X} \in \mathbb{R}^{n \times d}} F(\mathbf{X}) := \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} f_p(\mathbf{X}) \qquad (2)$$

where $f_p(\mathbf{X})$ evaluates

$$\triangle_p = d_{ij}^2(\mathbf{X}) - d_{lk}^2(\mathbf{X}), \; p = (i,j,l,k).$$

and

$$f_p(\mathbf{X}) : \begin{cases} \leq 0, & \triangle_p \leq 0 \\ > 0, & \text{otherwise.} \end{cases}$$

The loss function $f_p(\mathbf{X})$ can be chosen as the hinge loss (Agarwal et al. 2007)

$$f_p(\mathbf{X}) = \max\{0, 1 + \triangle_p\}, \qquad (3)$$

the scale-invariant loss (Tamuz et al. 2011)

$$f_p(\mathbf{X}) = \log \frac{d_{lk}^2(\mathbf{X}) + \delta}{d_{ij}^2(\mathbf{X}) + d_{lk}^2(\mathbf{X}) + 2\delta}, \qquad (4)$$

where $\delta \neq 0$ is a scalar which overcomes the problem of degeneracy and preserve numerical stable, the logistic loss (van der Maaten and Weinberger 2012)

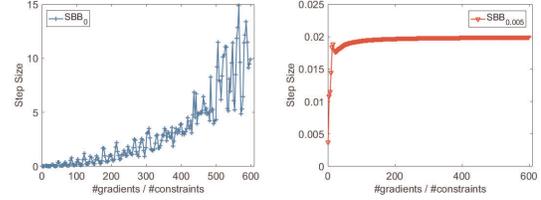$$f_p(\mathbf{X}) = -\log(1 + \exp(\triangle_p)), \qquad (5)$$

Figure 1: Step sizes along iterations of SVRG-SBB$_\epsilon$ on the synthetic data. (a) SBB step size with $\epsilon = 0$, (b) SBB step size with $\epsilon = 0.005$.

and replacing the Gaussian kernel in (5) by the Student-t kernel with degree α (van der Maaten and Weinberger 2012)

$$f_p(\mathbf{X}) = -\log \frac{\left(1 + \frac{d_{ij}^2(\mathbf{X})}{\alpha}\right)^{-\frac{\alpha+1}{2}}}{\left(1 + \frac{d_{ij}^2(\mathbf{X})}{\alpha}\right)^{-\frac{\alpha+1}{2}} + \left(1 + \frac{d_{lk}^2(\mathbf{X})}{\alpha}\right)^{-\frac{\alpha+1}{2}}}. \qquad (6)$$

Since (2) is an unconstrained optimization problem, it is obvious that SVD and parameter λ are avoided. Moreover, instead of the batch methods like the gradient descent method, we use a fast stochastic gradient descent algorithm like SVRG to solve the non-convex problem (2).

SVRG with Stabilized BB Step Size

A. Motivation

One open issue in stochastic optimization is how to choose an appropriate step size for SVRG in practice. The common method is either to use a constant step size to track, a diminishing step size to enforce convergence, or to tune a step size empirically, which can be time consuming. Recently, (Tan et al. 2016) proposed to use the Barzilai-Borwein (BB) method to automatically compute step sizes in SVRG for strongly convex objective function shown as follows

$$\eta_s = \frac{1}{m} \frac{\|\tilde{\mathbf{X}}^s - \tilde{\mathbf{X}}^{s-1}\|_F^2}{\text{vec}(\tilde{\mathbf{X}}^s - \tilde{\mathbf{X}}^{s-1})^T \text{vec}(\mathbf{g}^s - \mathbf{g}^{s-1})}, \qquad (7)$$

where $\tilde{\mathbf{X}}^s$ is the s-th iterate of the outer loop of SVRG and $\mathbf{g}^s = \nabla F(\tilde{\mathbf{X}}^s)$. However, if the objective function F is non-convex, the denominator of (7) might be close to 0 and even negative that fail the BB method. For example, Figure 1(a) shows that in simulations one can observe the instability of the absolute value of the original BB step size (called SBB$_0$ henceforth) in non-convex problems. Due to this issue, the original BB step size might not be suitable for the non-convex ordinal embedding problem.

B. Stabilized BB step size

An intuitive way to overcome the flaw of BB step size is to add another positive term in the absolute value of the denominator of the original BB step size, which leads to our introduced stabilized Barzilai-Borwein (SBB) step size shown as

follows,

$$\eta_s = \frac{1}{m} \cdot \left\| \tilde{\mathbf{X}}^s - \tilde{\mathbf{X}}^{s-1} \right\|_F^2$$
$$\times \ \left(\left| \mathrm{vec}(\tilde{\mathbf{X}}^s - \tilde{\mathbf{X}}^{s-1})^T \mathrm{vec}(\mathbf{g}^s - \mathbf{g}^{s-1}) \right| \right. \tag{8}$$
$$\left. + \ \epsilon \left\| \tilde{\mathbf{X}}^s - \tilde{\mathbf{X}}^{s-1} \right\|_F^2 \right)^{-1}, \quad \text{for some } \epsilon > 0.$$

By the use of SBB step size, the SVRG-SBB algorithm is presented in Algorithm 1.

Actually, as shown by our latter theorem (i.e., Theorem 1), if the Hessian of the objective function $\nabla^2 F(X)$ is non-singular and the magnitudes of its eigenvalues are lower bounded by some positive constant μ, then we can take $\epsilon = 0$. In this case, we call the referred step size SBB_0 henceforth. Even if we have no information of the Hessian of the objective function in practice, the SBB_ϵ step size with an $\epsilon > 0$ is just a more consecutive step size of SBB_0 step size.

From (8), if the gradient ∇F is Lipschitz continuous with constant $L > 0$, then the SBB_ϵ step size can be bounded as follows

$$\frac{1}{m(L + \epsilon)} \leq \eta_k \leq \frac{1}{m\epsilon}, \tag{9}$$

where the lower bound is obtained by the L-Lipschitz continuity of ∇F, and the upper bound is directly derived by its specific form. If further $\nabla^2 F(X)$ is nonsingular and the magnitudes of its eigenvalues has a lower bound $\mu > 0$, then the bound of SBB_0 becomes

$$\frac{1}{mL} \leq \eta_k \leq \frac{1}{m\mu}. \tag{10}$$

As shown in Figure 1 (b), SBB_ϵ step size with a positive ϵ can make SBB_0 step size more stable when SBB_0 step size is unstable and varies dramatically. Moreover, SBB_ϵ step size usually changes significantly only at the initial several epoches, and then quickly gets very stable. This is mainly because there are many iterations in an epoch of SVRG-SBB, and thus, the algorithm might close to a stationary point after only one epoch, and starting from the second epoch, the SBB_ϵ step sizes might be very close to the inverse of the sum of the curvature of objective function and the parameter ϵ used.

C. Convergence Results

In this subsection, we establish the convergence rate of SVRG-SBB as shown in the following theorem.

Theorem 1. *Let $\{\{\mathbf{X}_t^s\}_{t=1}^m\}_{s=1}^S$ be a sequence generated by Algorithm 1. Suppose that F is smooth, and ∇F is Lipschitz continuous with Lipschitz constant $L > 0$ and bounded. For any $\epsilon > 0$, if*

$$m > \max \left\{ 2L^2 \left(1 + \frac{2L}{\epsilon} \right), 1 + \sqrt{1 + \frac{8L^2}{\epsilon^2}} \right\} \cdot \epsilon^{-1}, \tag{11}$$

Algorithm 1: SVRG-SBB for (2)

Input: $\epsilon \geq 0$, update frequency m, maximal number of iterations S, initial step size η_0 (only used in the first epoch), initial point $\tilde{\mathbf{x}}^0 \in \mathbb{R}^{n \times d}$, and $N := |\mathcal{P}|$

Output: $\mathbf{x}_{\mathrm{out}}$ is chosen uniformly from $\{\{\mathbf{X}_t^s\}_{t=1}^m\}_{s=1}^S$

1 **for** $s=0$ *to* $S-1$ **do**

2 $\quad$ $\mathbf{g}^s = \nabla F(\tilde{\mathbf{X}}^s) = \frac{1}{N} \sum_{i=1}^N \nabla f_i(\tilde{\mathbf{X}}^s)$;

3 $\quad$ **if** $s > 0$ **then**

4

$$\eta_s = \frac{1}{m} \cdot \frac{\|\tilde{\mathbf{X}}^s - \tilde{\mathbf{X}}^{s-1}\|_F^2}{|\mathrm{vec}(\tilde{\mathbf{X}}^s - \tilde{\mathbf{X}}^{s-1})^T \mathrm{vec}(\mathbf{g}_s - \mathbf{g}_{s-1})| + \epsilon \|\tilde{\mathbf{X}}^s - \tilde{\mathbf{X}}^{s-1}\|_F^2} \tag{8}$$

5 $\quad$ **end**

6 $\quad$ $\mathbf{X}_0^s = \tilde{\mathbf{X}}^s$;

7 $\quad$ **for** $t=0$ *to* $m-1$ **do**

8 $\quad\quad$ uniformly randomly pick $i_t \in \{1, ..., N\}$;

9 $\quad\quad$ $\mathbf{X}_{t+1}^s = \mathbf{X}_t^s - \eta_s (\nabla f_{i_t}(\mathbf{X}_t^s) - \nabla f_{i_t}(\tilde{\mathbf{X}}^s) + \mathbf{g}^s)$;

10 $\quad$ **end**

11 $\quad$ $\tilde{\mathbf{X}}^{s+1} = \mathbf{X}_m^s$;

12 **end**

---

then for the output $\mathbf{X}_{\mathrm{out}}$ of Algorithm 1, we have

$$\mathbb{E}[\|\nabla F(\mathbf{X}_{\mathrm{out}})\|^2] \leq \frac{F(\tilde{\mathbf{X}}^0) - F(\mathbf{X}^*)}{T \cdot \gamma_S}, \tag{12}$$

where $\mathbf{X}^*$ is an optimal solution of (2), $T = m \cdot S$ is the total number of iterations, γ_S is some positive constant satisfying

$$\gamma_S \geq \min_{0 \leq s \leq S-1} \left\{ \eta_s \left[\frac{1}{2} - \eta_s \left(1 + 4(m-1)L^3 \eta_s^2 \right) \right] \right\},$$

and $\{\eta_s\}_{s=0}^{S-1}$ are SBB step sizes specified in (8).

If further the Hessian $\nabla^2 F(X)$ exists and μ is the lower bound of the magnitudes of eigenvalues of $\nabla^2 F(X)$ for any bounded X, then the convergence rate (12) still holds for SVRR-SBB with ϵ replaced by $\mu + \epsilon$. In addition, if $\mu > 0$, then we can take $\epsilon = 0$, and (12) still holds for SVRR-SBB_0 with ϵ replaced by μ.

Theorem 1 is an adaptation of (Reddi et al. 2016, Theorem 2) via noting that the used SBB step size specified in (8) satisfies (9). The proof of this theorem is presented in supplementary material. Theorem 1 shows certain non-asymptotic rate of convergence of the Algorithm 1 in the sense of convergence to a stationary point. Similar convergence rates of SVRG under different settings have been also shown in (Reddi et al. 2016; Allen-Zhu and Hazan 2016).

Note that the Lipschitz differentiability of the objective function is crucial for the establishment of the convergence rate of SVRG-SBB in Theorem 1. In the following, we give a lemma to show that a part of aforementioned objective functions (4), (5) and (6) in the ordinal embedding problem are Lipschitz differentiable. Considering the limited space of this paper, the readers are refered to (https://github.com/alphaprime/Stabilized_Stochastic_BB) for detailed proofs.

Lemma 1. *The ordinal embedding functions (4), (5) and (6) are Lipschitz differentiable for any bounded variable X.*

Experiments

In this section, we conduct a series of simulations and real-world data experiments to demonstrate the effectiveness of the proposed algorithms. Three models including *GNMDS*, *STE* and *TSTE* are taken into consideration. Our source code could be found on the web[1].

A. Simulations

Table 1: Computational complexity (second) comparisons on the synthetic dataset.

GNMDS				
	min	mean	max	std
cvx	1.6140	1.7881	1.9390	0.0844
ncvx Batch	4.5070	4.9372	5.2910	0.1857
ncvx SGD	5.3070	5.5882	5.8970	0.1216
ncvx SVRG	2.3020	2.8919	3.5280	0.2911
ncvx SVRG-SBB$_0$	**0.3500**	**0.4347**	**0.5340**	**0.0367**
ncvx SVRG-SBB$_\epsilon$	*0.3570*	*0.4858*	*0.6070*	*0.0621*
STE				
	min	mean	max	std
cvx	3.6740	3.9442	4.1870	0.1709
ncvx Batch	2.4610	2.6326	2.9110	0.1162
ncvx SGD	1.5000	2.1312	2.7190	0.2740
ncvx SVRG	1.9930	2.4068	2.8350	0.1935
ncvx SVRG-SBB$_0$	*0.5000*	*0.6052*	*0.6980*	*0.0660*
ncvx SVRG-SBB$_\epsilon$	**0.4510**	**0.5773**	**0.6780**	**0.0515**
TSTE				
	min	mean	max	std
ncvx Batch	3.9380	4.1815	4.4790	0.1146
ncvx SGD	6.0410	8.2870	9.4770	0.6863
ncvx SVRG	1.6090	1.9250	2.3470	0.1807
ncvx SVRG-SBB$_0$	*0.4580*	*0.7906*	*1.2480*	*0.1969*
ncvx SVRG-SBB$_\epsilon$	**0.3800**	**0.4726**	**0.5470**	**0.0420**

We start with a small-scale synthetic experiment to show how the methods perform in an idealized setting, which provides sufficient ordinal information in noiseless case.

Settings. The synthesized dataset consists of 100 points $\{\mathbf{x}_i\}_{i=1}^{100} \subset \mathbb{R}^{10}$, where $\mathbf{x}_i \sim \mathcal{N}(\mathbf{0}, \frac{1}{20}\mathbf{I})$, where $\mathbf{I} \in \mathbb{R}^{10 \times 10}$ is the identity matrix. The possible similarity triple comparisons are generated based on the Euclidean distances between $\{\mathbf{x}_i\}$. As (Jain, Jamieson, and Nowak 2016) has proved that the Gram matrix $\mathbf{G}$ can be recovered from $\mathcal{O}(pn \log n)$ triplets, we randomly choose $|\mathcal{P}| = 10,000$ triplets as the training set and the rest as test set. The regularization parameter and step size settings for the convex formulation follow the default setting of the STE/TSTE implementation[2], so we do not choose the step size by line search or the halving heuristic for convex formulation. The embedding dimension is selected just to be equal to 10 without variations, because the results of different embedding dimensions have been discussed in the original papers of GNMDS, STE and TSTE.

---

[1]https://github.com/alphaprime/Stabilized_Stochastic_BB
[2]http://homepage.tudelft.nl/19j49/ste/Stochastic_Triplet_Embedding.html

Evaluation Metrics. The metrics that we used in the evaluation of various algorithms include the generalization error and running time. As the learned embedding $\mathbf{X}$ from partial triple comparisons set $\mathcal{P} \subset [n]^3$ may be generalized to unknown triplets, the percentage of held-out triplets which is satisfied in the embedding $\mathbf{X}$ is used as the main metric for evaluating the quality. The running time is the duration of a algorithm when the training error is larger than 0.15.

Competitors. We evaluate both convex and non-convex formulations of three objective functions (i.e. GNMDS, STE and TSTE). We set the two baselines as : (1) the convex objective function whose results are denoted as "convex", and (2) these non-convex objective functions solved by batch gradient descent denoted as "ncvx batch". We compare the performance of SVRG-SBB$_\epsilon$ with SGD, SVRG with a fixed step size (called SVRG for short henceforth) as well as the batch gradient descent methods. As SVRG and its variant (SVRG-SBB$_\epsilon$) runs $2m + |\mathcal{P}|$ times of (sub)gradient in each epoch, the batch and SGD solutions are evaluated with the same numbers of (sub)gradient of SVRG. In Figure 2, the x-axis is the computational cost measured by the number of gradient evaluations divided by the total number of triple-wise constraints $|\mathcal{P}|$. The generalization error is the result of 50 trials with different initial $\mathbf{X}_0$. For each epoch, the median number of generalization error over 50 trials with [0.25, 0.75] confidence interval are plotted. The experiment results are shown in Figure 2 and Table 1.

Results. From to Figure 2, the following phenomena can be observed. First, the algorithm SVRG-SBB$_0$ will be unstable at the initial several epochs for three models, and latter get very stable. The eventual performance of SVRG-SBB$_0$ and that of SVRG-SBB$_\epsilon$ are almost the same in three cases. Second, compared to the batch methods, all the stochastic methods including SGD, SVRG and SVRG-SBB$_\epsilon$ ($\epsilon = 0$ or $\epsilon > 0$) converge fast at the initial several epoches and quickly get admissible results with relatively small generalization error. This is one of our main motivations to use the stochastic methods. Particularly, for all three models, SVRG-SBB$_\epsilon$ outperforms all the other methods in the sense that it not only converges fastest but also achieves almost the best generalization error. Moreover, the outperformance of SVRG-SBB$_\epsilon$ in terms of the cpu time can be also observed from Table 1. Specifically, the speedup of SVRG-SBB$_\epsilon$ over SVRG is about 4 times for all three models.

Table 1 shows the computational complexity achieved by SGD, SVRR-SBB$_\epsilon$ and batch gradient descent for convex and non-convex objective functions. All computation is done using MATLAB® R2016b, on a desktop PC with Windows® 7 SP1 64 bit, with 3.3 GHz Intel® Xeon® E3-1226 v3 CPU, and 32 GB 1600 MHz DDR3 memory. It is easy to see that for all objective functions, SVRG-SBB$_\epsilon$ gains speed-up compared to the other methods. Besides, we notice that the convex methods could be effective when n is small as the projection operator will not be the bottleneck of the convex algorithm.
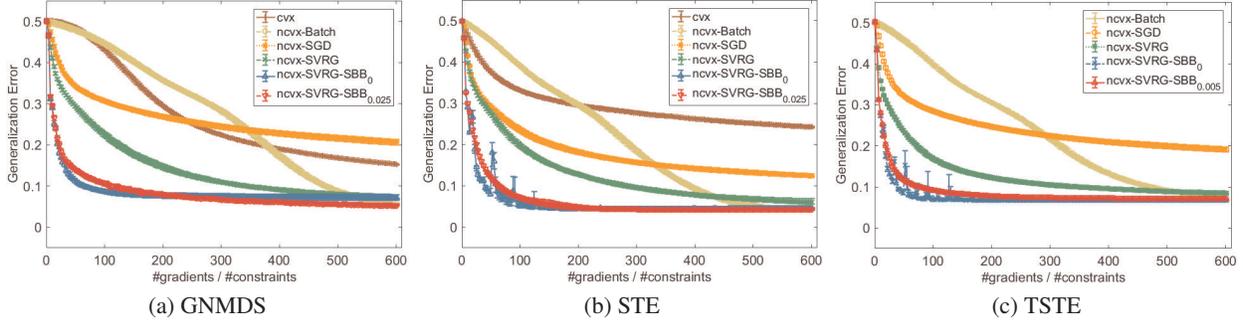
Figure 2: Generalization errors of SGD, SVRG, SVRG-SBB and batch methods on the synthetic dataset.
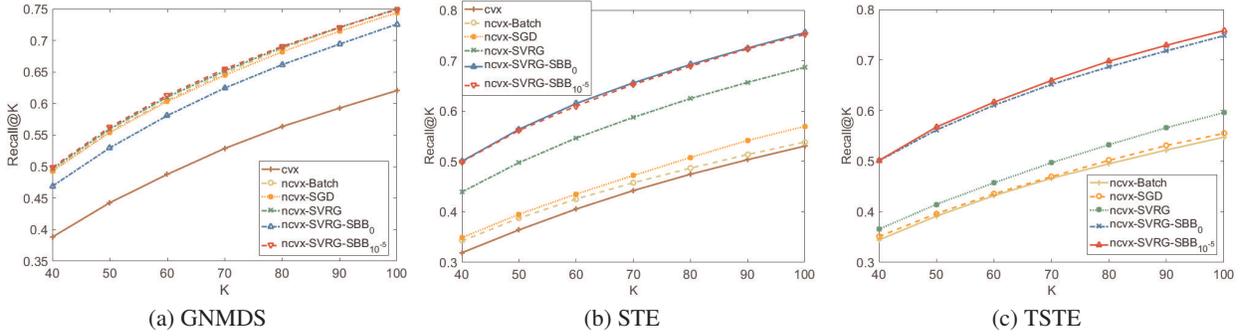

Figure 3: Recall@K with 10% noise on SUN397.

B. Image Retrieval on SUN397

Here, we apply the proposed SVRG-SBB algorithm for a real-world dataset, i.e., SUN 397, which is generally used for the image retrieval task. In this experiment, we wish to see how the learned representation characterizes the "relevance" of the same image category and the "discrimination" of different image categories. Hence, we use the image representation obtained by ordinal embedding for image retrieval.

Settings. We evaluate the effectiveness of the proposed stochastic non-convex ordinal embedding method for visual search on the SUN397 dataset. SUN397 consists of around 108K images from 397 scene categories. In SUN397, each image is represented by a $1,600$-dimensional feature vector extracted by principle component analysis (PCA) from $12,288$-dimensional Deep Convolution Activation Features (Gong et al. 2014). We form the training set by randomly sampling $1,080$ images from 18 categories with 60 images in each category. Only the training set is used for learning an ordinal embedding and a nonlinear mapping from the original feature space to the embedding space whose dimension is $p = 18$. The nonlinear mapping is used to predict the embedding of images which do not participate in the relative similarity comparisons. We use Regularized Least Square and Radial basis function kernel to obtain the nonlinear mapping. The test set consists of 720 images

randomly chose from 18 categories with 40 images in each category. We use ground truth category labels of training images to generate the triple-wise comparisons without any error. The ordinal constraints are generated like (Song et al. 2015): if two images i, j are from the same category and image k is from the other categories, the similarity between i and j is larger than the similarity between i and k, which is indicated by a triplet (i, j, k). The total number of such triplets is $70,000$. Errors are then synthesized to simulate the human error in crowd-sourcing. We randomly sample 5% and 10% triplets to exchange the positions of j and k in each triplet (i, j, k).

Evaluation Metrics. To measure the effectiveness of various ordinal embedding methods for visual search, we consider three evaluation metrics, i.e., precision at top-K positions (Precision@K), recall at top-K positions (Recall@K), and Mean Average Precision (MAP). Given the mapping feature $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{720}\}$ of test images and chosen an image i belonging to class c_i as a query, we sort the other images according to the distances between their embeddings and $\mathbf{x}_i$ in an ascending order as $\mathcal{R}_i$. True positives (TP_i^K) are images correctly labeled as positives, which involve the images belonging to c_i and list within the top K in $\mathcal{R}_i$. False positives (FP_i^K) refer to negative examples incorrectly labeled as positives, which are the images belonging to $c_l (l \neq i)$ and list within the top K in

Table 2: Image retrieval performance (MAP and Precision@40) on SUN397 when $p = 19$.

	0%		5%		10%	
	MAP	Precision@40	MAP	Precision@40	MAP	Precision@40
GNMDS						
cvx	0.2691	0.3840	0.2512	0.3686	0.2701	0.3883
ncvx Batch	0.3357	0.4492	0.3791	0.4914	0.3835	0.4925
ncvx SGD	0.3245	0.4379	0.3635	0.4772	0.3819	0.4931
ncvx SVRG	0.3348	0.4490	*0.3872*	*0.4974*	*0.3870*	*0.4965*
ncvx SVRG-SBB$_0$	**0.3941**	**0.5040**	0.3700	0.4836	0.3550	0.4689
ncvx SVRG-SBB$_\epsilon$	*0.3363*	*0.4500*	**0.3887**	**0.4981**	**0.3873**	**0.4987**
STE						
cvx	0.2114	0.3275	0.1776	0.2889	0.1989	0.3190
ncvx Batch	0.2340	0.3525	0.2252	0.3380	0.2297	0.3423
ncvx SGD	0.3369	0.4491	0.2951	0.4125	0.2390	0.3488
ncvx SVRG	0.3817	0.4927	0.3654	0.4804	0.3245	0.4395
ncvx SVRG-SBB$_0$	**0.3968**	**0.5059**	**0.3958**	**0.5054**	*0.3895*	**0.5002**
ncvx SVRG-SBB$_\epsilon$	*0.3940*	*0.5036*	*0.3921*	*0.5012*	**0.3896**	*0.4992*
TSTE						
ncvx Batch	0.2268	0.3470	0.2069	0.3201	0.2275	0.3447
ncvx SGD	0.2602	0.3778	0.2279	0.3415	0.2402	0.3514
ncvx SVRG	0.3481	0.4617	0.3160	0.4332	0.2493	0.3656
SVRG-SBB$_0$	**0.3900**	**0.4980**	**0.3917**	**0.5018**	**0.3914**	*0.5007*
ncvx SVRG-SBB$_\epsilon$	*0.3625*	*0.4719*	*0.3845*	*0.4936*	*0.3897*	**0.5013**

$\mathcal{R}_i$. True negatives (TN_i^K) correspond to negatives correctly labeled as negatives, which refer to the images belonging to $c_l (l \neq i)$ and list after the top K in $\mathcal{R}_i$. Finally, false negatives (FN_i^K) refer to positive examples incorrectly labeled as negatives, which are relevant to the images belonging to class c_i and list after the top K in $\mathcal{R}_i$. We are able to define Precision@K and Recall@K used in this paper as: $\text{Precision@}K = \frac{1}{n}\sum_i^n p_i^K = \frac{1}{n}\sum_i^n \frac{\text{TP}_i^K}{\text{TP}_i^K + \text{FP}_i^K}$ and $\text{Recall@}K = \frac{1}{n}\sum_i^n r_i^K = \frac{1}{n}\sum_i^n \frac{\text{TP}_i^K}{\text{TP}_i^K + \text{FN}_i^K}$. Precision and recall are single-valued metrics based on the whole ranking list of images determined by the Euclidean distances among the embedding $\{\mathbf{x}_i\}_{i=1}^n$. It is desirable to also consider the order in which the images from the same category are embedded. By computing precision and recall at every position in the ranked sequence of the images for query i, one can plot a precision-recall curve, plotting precision $p_i(r)$ as a function of recall r_i. Average Precision(AP) computes the average value of $p_i(r)$ over the interval from $r_i = 0$ to $r_i = 1$: $\text{AP}_i = \int_0^1 p_i(r_i)dr_i$, which is the area under precision-recall curve. This integral can be replaced with a finite sum over every position q in the ranked sequence of the embedding: $\text{AP}_i = \sum_{q=1}^{40} p_i(q) \cdot \triangle r_i(q)$, where $\triangle r_i(q)$ is the change in recall from items $q - 1$ to q. The MAP used in this paper is defined as $\text{MAP} = \frac{1}{n}\sum_{i=1}^n \text{AP}_i$.

Results. The experiment results are shown in Table 2 and Figure 3. As shown in Table 2 and Figure 3 with K varying from 40 to 100, we observe that non-convex SVRG-SBB$_\epsilon$ consistently achieves the superior Precision@K,

Recall@K and MAP results comparing against the other methods with the same gradient calculation. The results of GNMDS illustrate that SVRG-SBB$_\epsilon$ is more suitable for non-convex objective functions than the other methods. Therefore, SVRG-SBB$_\epsilon$ has a very promising potential in practice, because it generates appropriate step sizes automatically while running the algorithm and the result is robust. Moreover, under our setting and with small noise, all the ordinal embedding methods achieve the reasonable results for image retrieval. It illustrates that high-quality relative similarity comparisons can be used for learning meaningful representation of massive data, thereby making it easier to extract useful information in other applications.

Conclusions

In this paper, we propose a stochastic non-convex framework for the ordinal embedding problem. We propose a novel stochastic gradient descent algorithm called SVRG-SBB for solving this non-convex framework. The proposed SVRG-SBB is a variant of SVRG method incorporating with the so-called stabilized BB (SBB) step size, a new, stable and adaptive step size introduced in this paper. The main idea of the SBB step size is adding another positive term in the absolute value of the denominator of the original BB step size such that SVRG-SBB can overcome the instability of the original BB step size when applied to such non-convex problem. A series of simulations and real-world data experiments are implemented to demonstrate the effectiveness of the proposed SVRG-SBB for the ordinal embedding problem. It is surprising that the proposed SVRG-SBB outperforms most of the state-of-the-art methods in the perspec-

tive of both generalization error and computational cost. We also establish the $O(1/T)$ convergence rate of SVRG-SBB in terms of the convergence to a stationary point. Such convergence rate is comparable to the existing best convergence results of SVRG in literature.

Acknowledgment

References

Agarwal, S.; Wills, J.; Cayton, L.; Lanckriet, G. R.; Kriegman, D. J.; and Belongie, S. 2007. Generalized non-metric multidimensional scaling. *International Conference on Artificial Intelligence and Statistics* 11–18.

Allen-Zhu, Z., and Hazan, E. 2016. Variance reduction for faster non-convex optimization. *International Conference on Machine Learning* 699–707.

Arias-Castro, E. 2017. Some theory for ordinal embedding. *Bernoulli* 23(3):1663–1693.

Ghadimi, S., and Lan, G. 2013. Stochastic first- and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization* 23(4):2341–2368.

Gong, Y.; Wang, L.; Guo, R.; and Lazebnik, S. 2014. Multiscale orderless pooling of deep convolutional activation features. *European Conference on Computer Vision* 392–407.

Heikinheimo, H., and Ukkonen, A. 2013. The crowdmedian algorithm. *AAAI Conference on Human Computation and Crowdsourcing* 69–77.

Jain, L.; Jamieson, K. G.; and Nowak, R. 2016. Finite sample prediction and recovery bounds for ordinal embedding. *Annual Conference on Neural Information Processing Systems* 2711–2719.

Jamieson, K., and Nowak, R. 2011a. Active ranking using pairwise comparisons. *Annual Conference on Neural Information Processing Systems* 2240–2248.

Jamieson, K., and Nowak, R. 2011b. Low-dimensional embedding using adaptively selected ordinal data. *Annual Allerton Conference on Communication, Control, and Computing* 1077–1084.

Johnson, R., and Zhang, T. 2013. Accelerating stochastic gradient descent using predictive variance reduction. *Annual Conference on Neural Information Processing Systems* 315–323.

Kleindessner, M., and Luxburg, U. 2014. Uniqueness of ordinal embedding. *Conference on Learning Theory* 40–67.

Kruskal, J. B. 1964a. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika* 29(1):1–27.

Kruskal, J. B. 1964b. Nonmetric multidimensional scaling: A numerical method. *Psychometrika* 29(2):115–129.

McFee, B., and Lanckriet, G. 2011. Learning multi-modal similarity. *Journal of Machine Learning Research (JMLR)* 12:491–523.

Reddi, S. J.; Hefny, A.; Sra, S.; Poczos, B.; and Smola, A. 2016. Stochastic variance reduction for nonconvex optimization. *International Conference on Machine Learning* 314–323.

Shaw, B., and Jebara, T. 2009. Structure preserving embedding. *International Conference on Machine Learning* 937–944.

Shepard, R. N. 1962a. The analysis of proximities: Multidimensional scaling with an unknown distance function. i. *Psychometrika* 27(2):125–140.

Shepard, R. N. 1962b. The analysis of proximities: Multidimensional scaling with an unknown distance function. ii. *Psychometrika* 27(3):219–246.

Song, D.; Liu, W.; Ji, R.; Meyer, D. A.; and Smith, J. R. 2015. Top rank supervised binary coding for visual search. *IEEE International Conference on Computer Vision* 1922–1930.

Tamuz, O.; Liu, C.; Shamir, O.; Kalai, A.; and Belongie, S. 2011. Adaptively learning the crowd kernel. *International Conference on Machine Learning* 673–680.

Tan, C.; Ma, S.; Dai, Y.-H.; and Qian, Y. 2016. Barzilaiborwein step size for stochastic gradient descent. *Annual Conference on Neural Information Processing Systems* 685–693.

Terada, Y., and Luxburg, U. 2014. Local ordinal embedding. *International Conference on Machine Learning* 847–855.

van der Maaten, L., and Weinberger, K. 2012. Stochastic triplet embedding. *IEEE International Workshop on Machine Learning for Signal Processing* 1–6.

Wah, C.; Horn, G. V.; Branson, S.; Maji, S.; Perona, P.; and Belongie, S. 2014. Similarity comparisons for interactive fine-grained categorization. *IEEE Conference on Computer Vision and Pattern Recognition* 859–866.

Wilber, M.; Kwak, I.; Kriegman, D.; and Belongie, S. 2015. Learning concept embeddings with combined human-machine expertise. *IEEE International Conference on Computer Vision* 981–989.

Wilber, M.; Kwak, S.; and Belongie, S. 2014. Cost-effective hits for relative similarity comparisons. *AAAI Conference on Human Computation and Crowdsourcing* 227–233.