Coreset Stochastic Variance-Reduced Gradient with Application to Optimal Margin Distribution Machine

Zhi-Hao Tan, Teng Zhang, Wei Wang
National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China
Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing 210023, China
{tanzh, zhangt, wangw}@lamda.nju.edu.cn

Abstract

A major problem for kernel-based predictors is the prohibitive computational complexity, which limits their application in large-scale datasets. Coreset, an approximation method which tries to cover the given examples with a small set of points, can be used to remain the prominent information and accelerate the kernel method. In this paper, we provide perhaps the first coreset-based kernel-accelerating optimization method that has a linear convergence rate, which is much faster than existing approaches. Our method can be used to train kernel SVM-style problems and obtain sparse solutions efficiently. Specifically, the method uses SVRG as the framework, and utilizes the core points to approximate the gradients, so it can significantly reduce the complexity of the kernel method. Furthermore, we apply the method to train ODM, a kernel machine enjoying better statistical property than SVM, so that we can reduce the risk of compromising the performance while encouraging the sparsity. We conduct extensive experiments on several large-scale datasets and the results verify that our method outperforms the state-of-the-art coreset approximation method in both efficiency and generalization, while simultaneously achieving significant speed-up compared to non-approximation baselines.

Introduction

The kernel method provides a powerful and unified framework for applying linear methods to general learning problems. The key idea is to map data to a higher dimensional kernel feature space, where linear relationships correspond to nonlinear relationships in the original data. In the past decades, quite a lot of kernel methods have been developed, among which the representatives are kernel SVMs (Cortes and Vapnik 1995), kernel regression (Smola and Schölkopf 2004), kernel PCA (Schölkopf, Smola, and Müller 1998), Gaussian process (Rasmussen 2004), and so on.

Given m data points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m$, the $m \times m$ kernel matrix K is formed where K_{ij} is the inner product between $\phi(\boldsymbol{x}_i)$ and $\phi(\boldsymbol{x}_j)$ in the high-dimensional space, computed by the kernel function $k(\cdot, \cdot)$. Then all inner product required by linear methods are performed by the kernel matrix K. Unfortunately, the kernel method brings huge cost. Specifically, just generating the entries of K requires $\Theta(m^2)$ computation time and memory storage, which is prohibitive for large-scale datasets.

Alleviating this issue has motivated a variety of practical approaches, including *random Fourier feature* methods (Rahimi and Recht 2008; 2009; Le, Sarlós, and Smola 2013), the *Nyström* methods (Williams and Seeger 2001; Drineas and Mahoney 2005; Zhang, Tsang, and Kwok 2008; Gittens and Mahoney 2016), and *coreset approximation* methods (Tsang, Kwok, and Cheung 2005; Tsang, Kwok, and Zurada 2006; Loosli and Canu 2007; Asharaf, Murty, and Shevade 2007; Le et al. 2017), etc. RFF aims to approximate the shift-invariant kernel function through orthogonal trigonometric function family. However, this devised kernel mapping is data-independent, which leads to poorer generalization performance than the Nyström method (Yang et al. 2012). On the other hand, the Nyström method focuses on constructing a low-rank approximation kernel matrix using a subset of examples. In the classic variants, since these points are randomly selected without considering their position or importance, it may destroy the spectral structure of the kernel matrix and result in unstable performance.

Coreset approximation is a method originated in computational geometry. The basic idea is to use core points to approximate the shape of all samples. It could significantly reduce the size of kernel matrix in kernel method, especially in kernel SVM. Notable works include the Core Vector Machine (CVM) (Tsang, Kwok, and Cheung 2005), the Ball Vector Machine (BVM) (Tsang, Kocsor, and Kwok 2007) and the Approximation Vector Machine (AVM) (Le et al. 2017). The main idea of CVM is to reformulate SVM as a *minimum enclosing ball* (MEB) problem and obtain the approximation solution of the MEB using the coreset-based algorithm in computational geometry (Bădoiu and Clarkson 2008). However, the state-of-the-art coreset-based kernel machine, AVM (Le et al. 2017), solves the primal problem directly and utilizes an online way to construct coreset with overlapping hyperballs. Moreover, the AVM uses core points to approximate all gradients required in SGD and easily obtains a sparse model in the form of $\hat{\boldsymbol{w}} = \sum_{i=1}^{r} \sigma_i \phi(\boldsymbol{c}_i)$, where r is the coreset size and satisfies $r \ll m$. Despite using approximated sparse gradients, Le et al. (2017) prove that the AVM has a convergence rate of $O(1/T)$, and the gap between the approximated and optimal solutions can be controlled by the diameter of hyperballs. However, the SGD

suffers the suboptimal convergence due to the inherent variance (Johnson and Zhang 2013), which limits the efficiency of the method.

Our contributions In this work, we aim to further improve the efficiency of coreset-based kernel machine for large-scale datasets and simultaneously keep competitive generalization performance.

Theoretically, inspired by introducing full gradient to control variance explicitly in SVRG (Johnson and Zhang 2013), we propose an optimization method called CSVRG (**C**oreset **S**tochastic **V**ariance-**R**educed **G**radient), which can be used to optimize the kernel SVM-style problems efficiently. The main result is that we prove the linear convergence of CSVRG despite only a coreset-estimated full gradient is available. Thus it's much faster than the AVM.

Empirically, we apply CSVRG to train ODM (Optimal Margin Distribution Machine) (Zhang and Zhou 2016; 2017; 2018), a kernel machine which aims to optimize the margin distribution for better statistical property than SVM (Gao and Zhou 2013; Zhang and Zhou 2014; Zhou and Zhou 2016), so that we can achieve the "best of both worlds", i.e., the best efficiency as well as the best generalization. We conduct extensive experiments on several large-scale datasets and the results verify that our method outperforms the state-of-the-art coreset approximation method in both efficiency and generalization, while simultaneously achieving significant speed-up compared to non-approximation baselines.

Paper outline Our optimization method is built on the coreset approximation, so we first introduce some definitions and how to construct coreset in preliminaries. Then we present a unified formulation to represent SVMs and ODM and give our optimization algorithm for SVMs and ODM.

After that, we theoretically prove the linear convergence rate of CSVRG for optimizing SVMs and ODM. Finally, we introduce the empirical study by applying CSVRG to train ODM on several large-scale datasets and conclude the paper.

Preliminaries

Notations

We denote $\mathcal{X}$ as the instance space and $\mathcal{Y} = \{+1, -1\}$ as the label set. Let $\mathcal{D}$ be an unknown (underlying) distribution over $\mathcal{X} \times \mathcal{Y}$. A training set $S = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_m, y_m)\} \in (\mathcal{X} \times \mathcal{Y})^m$ is drawn identically and independently (i.i.d.) according to $\mathcal{D}$. We assume that a positive semi-definite and isotropic kernel is used, i.e., $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = k(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2)$, where $k : \mathbb{R} \mapsto \mathbb{R}$ is a monotonically decreasing function. Let $\phi : \mathcal{X} \mapsto \mathbb{H}$ be a feature mapping where $\mathbb{H}$ is a Reproducing Kernel Hilbert Space (RKHS) associated to the kernel K, i.e., $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi(\boldsymbol{x}_i)^\top \phi(\boldsymbol{x}_j)$. Besides, we denote r as the size of the coreset, and denote δ as the diameter of the coverage.

Constructing Coreset

A coreset is a subset of input points such that we can get a good approximation to the origin input. Given an input space $\mathcal{X}$, we introduce the concept of δ-coverage and coreset as follows.

Definition 1 (δ-coverage)**.** *The collection of sets $\mathcal{P} = (P_i)_{i \in I}$ is said to be a δ-coverage of the set S iff $S \subset \cup_{i \in I} P_i$ and $D(P_i) \leq \delta$, $\forall i \in I$, where I is the index set and $D(P_i) = \sup_{x, x' \in P_i} \|x - x'\|$, which is the maximal pairwise distance between any two points in P_i. Each element $P_i \in \mathcal{P}$ is further referred to as a cell.*

Definition 2 (coreset, core point)**.** *Given an δ-coverage $\mathcal{P} = (P_i)_{i \in I}$ over a given input space $\mathcal{X}$, for each $i \in I$, we select an arbitrary point $\boldsymbol{c}_i$ from the set P_i, then the collection of all $\boldsymbol{c}_i$ is called the coreset $\mathcal{C}$ of the δ-coverage $\mathcal{P}$. Each point $\boldsymbol{c}_i \in \mathcal{C}$ is further referred to as a core point.*

Remark 1. There are two representative ways to construct a δ-coverage, i.e., a coreset. The CVM (Tsang, Kwok, and Cheung 2005) adds the points lying furthest from the current core points to the coreset each time. Differently, the AVM (Le et al. 2017) adopts the online constructing method and the key idea is that only if the arrived point falls outside the existing cells, we will add it to the coreset and create a new cell. The shape of cells is a hyperball when the Euclidean distance is used, and corresponds to a hyperrectangle when Chebyshev distance is used. Considering the low computational complexity of the online constructing method in the AVM and the better performance of Euclidean distance (Le et al. 2017), we use these to construct the δ-coverage in this paper.

Formulation

SVM aims to learn a large margin separator, i.e., maximizing the smallest distance from the instances to the classfication boundary in a RKHS. A more robust strategy is to consider the whole data, i.e., optimizing the margin distribution. Moreover, a recent study (Gao and Zhou 2013) on margin theory proved that maximizing the margin mean and minimizing the margin variance simultaneously can yeild a tighter generalization bound. By fixing the margin mean and optimizing the margin distribution, we obtain the formulation of Optimal Margin Distribution Machine (ODM) (Zhang and Zhou 2016),

$$\begin{aligned} \min_{\boldsymbol{w}, \xi_i, \epsilon_i} \ & \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{\lambda}{m} \sum_{i=1}^{m} \frac{\xi_i^2 + \mu \epsilon^2}{(1-\theta)^2} \\ \text{s.t.} \ & y_i \boldsymbol{w}^\top \phi(\boldsymbol{x}_i) \geq 1 - \theta - \xi_i, \ i = 1, \ldots, m \\ & y_i \boldsymbol{w}^\top \phi(\boldsymbol{x}_i) \leq 1 + \theta + \epsilon_i, \ i = 1, \ldots, m \end{aligned} \tag{1}$$

where $\lambda > 0$ is the trade-off parameter, $\theta \in [0, 1)$ is a parameter of the zero loss band, which can control the sparsity of the solution. $\mu \in (0, 1]$ is a parameter to trade off two different kinds of deviation.

Remark 2. To make the subsequent optimization algorithm and theoretical analysis concise, we represent SVMs and ODM in a unified formulation as follows

$$\min_{\boldsymbol{w}} f(\boldsymbol{w}) = \frac{1}{m} \sum_{i=1}^{m} \psi_i(\boldsymbol{w}) \tag{2}$$

$$= \frac{1}{2}\|\boldsymbol{w}\|_{\mathbb{H}}^2 + \frac{\lambda}{m} \sum_{i=1}^{m} l(\boldsymbol{w}; \boldsymbol{x}_i, y_i), \tag{3}$$

where $\psi_i(\boldsymbol{w}) = \|\boldsymbol{w}\|_{\mathbb{H}}^2/2 + \lambda l(\boldsymbol{w}; \boldsymbol{x}_i, y_i)$ and $l(\boldsymbol{w}; \boldsymbol{x}, y)$ is a convex loss function. When l is set as $\max(0, 1 - y\boldsymbol{w}^\top\phi(\boldsymbol{x}))$ or $\max(0, 1 - y\boldsymbol{w}^\top\phi(\boldsymbol{x}))^2$, we can obtain the hinge loss SVM and squared hinge loss SVM, respectively, while ODM can be derived by using $l(\boldsymbol{w}; \boldsymbol{x}, y) = \frac{1}{m(1-\theta)^2}(\max\{0, 1 - \theta - y\boldsymbol{w}^\top\phi(\boldsymbol{x})\}^2 + \mu\max\{0, y\boldsymbol{w}^\top\phi(\boldsymbol{x}) - 1 - \theta\}^2)$.

The Proposed Optimization Method

In this section, we commence with the unified formulation of the gradient of different loss functions, followed by the coreset approximation for gradients. Next, the theoretically guided projection operations are introduced. Finally we give the detailed optimization algorithm.

Unified Form of Gradient

Our method focuses on the optimization problems with the above formulation (3). For hinge loss SVM and squared hinge loss SVM, the derivative of the loss functions are as follows

$$l'_h(\boldsymbol{w}; \boldsymbol{x}, y) = -\mathbb{I}_{\{y\boldsymbol{w}^\top\phi(\boldsymbol{x})\le 1\}} y\phi(\boldsymbol{x})$$
$$l'_{h^2}(\boldsymbol{w}; \boldsymbol{x}, y) = -\mathbb{I}_{\{y\boldsymbol{w}^\top\phi(\boldsymbol{x})\le 1\}} 2y\left(1 - y\boldsymbol{w}^\top\phi(\boldsymbol{x})\right)\phi(\boldsymbol{x})$$

where $\mathbb{I}_S$ is the indicator function, which equals to 1 if the logical statement S is true and 0 otherwise. For ODM, we have

$$l'(\boldsymbol{w}; \boldsymbol{x}, y) = \frac{2}{(1-\theta)^2}\{(y\boldsymbol{w}^\top\phi(\boldsymbol{x}) + \theta - 1)y\mathbb{I}(\boldsymbol{x} \in I_1) + \mu(y\boldsymbol{w}^\top\phi(\boldsymbol{x}) - \theta - 1)y\mathbb{I}(\boldsymbol{x} \in I_2)\}\phi(\boldsymbol{x})$$

where $I_1 \equiv \{\boldsymbol{x}|y\boldsymbol{w}^\top\phi(\boldsymbol{x}) < 1 - \theta\}$, $I_2 \equiv \{\boldsymbol{x}|y\boldsymbol{w}^\top\phi(\boldsymbol{x}) > 1 + \theta\}$.

Remark 3. For ODM and SVMs, we can reformulate the gradient (or sub-gradient) of loss function for randomly sampled $(\boldsymbol{x}_t, y_t)$ as

$$l'(\boldsymbol{w}_t; \boldsymbol{x}_t, y_t) = \alpha_t\phi(\boldsymbol{x}_t) \tag{4}$$

where α_t is a scalar. This form greatly facilitates the subsequent theoretical analysis.

Coreset Approximation

Since our optimization algorithm is based on SVRG (Johnson and Zhang 2013), we need the approximation of gradient of $\nabla\psi(\boldsymbol{w})$ and $\nabla f(\boldsymbol{w})$. For sample $(\boldsymbol{x}_t, y_t)$, the gradient of $\psi_t(\boldsymbol{w})$, which satisfies $\mathbb{E}[\nabla\psi_t(\boldsymbol{w})|\boldsymbol{w}] = \nabla f(\boldsymbol{w})$, has the form

$$\nabla\psi_t(\boldsymbol{w}) = \boldsymbol{w} + \lambda\alpha_t\phi(\boldsymbol{x}_t). \tag{5}$$

Considering that the representer theorem indicates that the optimal solution of the above formulation (3) has the form $\boldsymbol{w} = \sum_{i=1}^m \alpha_i\phi(\boldsymbol{x}_i)$, we can improve the model sparsity with the following coreset approximation.

We denote $\nabla\psi_t^c(\boldsymbol{w})$ as the coreset approximation of $\psi_t(\boldsymbol{w})$, then we can obtain

$$\nabla\psi_t^c(\boldsymbol{w}) = \boldsymbol{w} + \lambda\alpha_t\phi(\boldsymbol{c}_t), \tag{6}$$

where $\boldsymbol{c}_t$ is the center of the hypersphere to which the sample $\boldsymbol{x}_t$ belongs. In addition, the approximation of full gradient can also be obtained in the same way

$$\nabla f^c(\boldsymbol{w}) = \boldsymbol{w} + \frac{\lambda}{m}\sum_{i=1}^m \alpha_i\phi(\boldsymbol{c}_i). \tag{7}$$

Projection

In this part, we show that for ODM and squared hinge loss SVM, the optimal solution $\|\boldsymbol{w}^*\|$ is bounded, so we can safely add projection operations when updating models.

Theorem 1. *If $\boldsymbol{w}^*_{odm}$ is the optimal solution of ODM, then there exists a positive constant H such that $\|\boldsymbol{w}^*_{odm}\| \le H$, where $H = \frac{\sqrt{\lambda(1-\theta)^2 + \lambda\mu(1+\theta)^2}}{1-\theta}$. Moreover, for squared hinge loss SVM, the optimal solution $\boldsymbol{w}^*_{h^2}$ satisfies $\boldsymbol{w}^*_{h^2} \le \lambda$.*

The proof of this theorem is similar to that of Theorem 1 in Shalev-Shwartz et al. (2011). Due to space limitations, the detailed proof is presented in the supplementary material.

Remark 4. According to Theorem 1, to ensure that $\|\boldsymbol{w}_t\|$ is bounded for all $t \ge 1$ in situations of square hinge loss SVM and ODM, we project $\boldsymbol{w}_t$ onto the hypersphere with the centre origin, radius λ and H, i.e., $\mathcal{B}(\boldsymbol{0}, \lambda)$ and $\mathcal{B}(\boldsymbol{0}, H)$ respectively after each round of model update. This operation could guarantee the safety of model updates and possibly result in a faster convergence.

Optimization Algorithm

Based on the coreset approximation and projection, we propose the method CSVRG(**C**oreset **S**tochastic **V**ariance-**R**educed **G**radient) for optimizing kernel SVM-style problems. The detailed procedure is showed in Algorithm 1.

At each time, we keep a snapshot of $\widetilde{\boldsymbol{w}}$ after T iterations like SVRG. However, we only maintain the coreset approximation $\nabla f^c(\boldsymbol{w})$ of full gradient, which significantly reduce the model complexity. In addition, the gradient items in update rule are all approximated with the coreset. For the convenience of convergence analysis, we denote

$$\boldsymbol{h}_t = \nabla\psi_t^c(\boldsymbol{w}_{t-1}) - \nabla\psi_t^c(\widetilde{\boldsymbol{w}}) + \nabla f^c(\widetilde{\boldsymbol{w}}) \tag{8}$$
$$= \boldsymbol{w}_{t-1} + (\alpha_t - \widetilde{\alpha}_t)\phi(\boldsymbol{c}_t) + \frac{1}{m}\sum_{i=1}^m \widetilde{\alpha}_i\phi(\boldsymbol{c}_i) \tag{9}$$

such that the update rule is $\boldsymbol{w}_t = \boldsymbol{w}_{t-1} - \eta_t\boldsymbol{h}_t$. In this way, we explicitly reduce the variance of SGD, and the learning rate η does not decay.

Remark 5. The computation still requires one pass over all data using $\widetilde{\boldsymbol{w}}$, but the most expensive calculations $\boldsymbol{w}_t\phi(\boldsymbol{x}_t) = \sum_{i=1}^m \sigma_i K(\boldsymbol{x}_i, \boldsymbol{x}_t)$ become simple and efficient. This is caused by two reasons. First, the current model after coreset approximation $\boldsymbol{w}_t = \sum_{i=1}^r \sigma_i\phi(\boldsymbol{c}_i)$ is very sparse, where r is the coreset size and $r \ll m$, so we need only $O(r)$ operations. Second, since the model is fixed as a linear combination of $\{\phi(\boldsymbol{c}_1), \ldots, \phi(\boldsymbol{c}_r)\}$, the required kernel matrix is reduced from m^2 to $m \times r$, which greatly eliminates the cost of generating the entries of kernel matrix. Moreover, referring to Cucker and Smale (2002), it is known that the model size r cannot exceed $\left(\frac{4D(\mathcal{X})}{\delta}\right)^d$.

Algorithm 1 Coreset Stochastic Variance-Reduced Gradient

Require: λ, μ, θ

Initialize $\widetilde{\boldsymbol{w}}_0 = \mathbf{0}$
for $s = 1, 2, \ldots$ **do**
  $\widetilde{\boldsymbol{w}} = \widetilde{\boldsymbol{w}}_{s-1}$
  $\nabla f^c(\widetilde{\boldsymbol{w}}) = \widetilde{\boldsymbol{w}} + \frac{1}{m}\sum_{i=1}^{m} \widetilde{\alpha}_i \phi(\boldsymbol{c}_i)$
  $\widetilde{\boldsymbol{w}}_0 = \widetilde{\boldsymbol{w}}$
  for $t = 1, 2, \ldots, T$ **do**
    Randomly sample $(\boldsymbol{x}_t, y_t)$
    Find the core point $\boldsymbol{c}_t$ closest to $\boldsymbol{x}_t$
    $\nabla \psi_t^c(\boldsymbol{w}_{t-1}) = \boldsymbol{w}_{t-1} + \alpha_t \phi(\boldsymbol{c}_t)$
    $\nabla \psi_t^c(\widetilde{\boldsymbol{w}}) = \widetilde{\boldsymbol{w}} + \widetilde{\alpha}_t \phi(\boldsymbol{c}_t)$
    $\boldsymbol{h}_t = \nabla \psi_t^c(\boldsymbol{w}_{t-1}) - \nabla \psi_t^c(\widetilde{\boldsymbol{w}}) + \nabla f^c(\widetilde{\boldsymbol{w}})$
    if ODM is used **then**
      $\boldsymbol{w}_t = \prod_{\mathcal{B}(\mathbf{0}, H)}(\boldsymbol{w}_{t-1} - \eta \boldsymbol{h}_t)$
    else if square hinge loss SVM is used **then**
      $\boldsymbol{w}_t = \prod_{\mathcal{B}(\mathbf{0}, \lambda)}(\boldsymbol{w}_{t-1} - \eta \boldsymbol{h}_t)$
    else
      $\boldsymbol{w}_t = \boldsymbol{w}_{t-1} - \eta \boldsymbol{h}_t$
    end if
  end for
  option I: set $\widetilde{\boldsymbol{w}}_s = \boldsymbol{w}_T$
  option II: set $\widetilde{\boldsymbol{w}}_s = \boldsymbol{w}_t$ for randomly chosen $t \in \{0, \ldots, T-1\}$
end for

Convergence Analysis

In this section, we provide the convergence analysis of CSVRG. The theoretical results are suitable for both SVMs and ODM. In the following, we will first show an upper bound of the coreset approximation error, and then present the detailed convergence analysis.

Due to the space limitations, only the proof of theorem is provided, the details of lemma's proof are presented in the supplementary material.

Bounded Approximation Error

The origin update item without approximation is

$$\begin{aligned}\boldsymbol{v}_t &= \nabla \psi_t(\boldsymbol{w}_{t-1}) - \nabla \psi_t(\widetilde{\boldsymbol{w}}) + \nabla f(\widetilde{\boldsymbol{w}}) \\ &= \boldsymbol{w}_{t-1} + (\alpha_t - \widetilde{\alpha}_t)\phi(\boldsymbol{x}_t) + \frac{1}{m}\sum_{i=1}^{m} \widetilde{\alpha}_i \phi(\boldsymbol{x}_i).\end{aligned}$$

By reformulating the coreset-approximated update item $\boldsymbol{h}_t$ as $\boldsymbol{h}_t = \boldsymbol{v}_t + \Delta_t$, we can obtain

$$\begin{aligned}\Delta_t = \frac{1}{m}\sum_{i=1}^{m} \widetilde{\alpha}_i [\phi(\boldsymbol{c}_i) - \phi(\boldsymbol{x}_i)] \\ + (\alpha_t - \widetilde{\alpha}_t)[\phi(\boldsymbol{c}_t) - \phi(\boldsymbol{x}_t)].\end{aligned} \tag{10}$$

The Δ_t represents the approximation error caused by updating model with the approximated gradients.

Lemma 2. *For ODM problem, the α_t satisfies $\alpha_t^2 \le A^2$ for all t, where $A = \frac{2(H+1+\theta)}{(1-\theta)^2}$.*

Lemma 3. *For hinge loss SVM and square hinge loss SVM, the α_t satisfies $\alpha_t^2 \le A^2 = \lambda^2 B^2$ for all t, where $B = 1$ and $B = 2\lambda + 2$ respectively.*

Remark 6. Without loss of generality, lemma 2, 3 are both based on the assumption that $\|\phi(\boldsymbol{x})\| = K(\boldsymbol{x}, \boldsymbol{x})^{1/2} = 1$. To make the subsequent theorems concise, we denote all upper bounds of α_t^2 as A. This does not affect the correctness of the theorems, although the value of A is not the same under different loss functions.

Theorem 4. *Assume that the p.s.d. and isotropic kernel $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = k(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2)$ is used, where $k(.)$ is a monotonically continuous decreasing function with $k(0) = 1$, and let δ be the diameter of hyperballs. For the approximation error Δ_t as indicated in (10), we have $\|\Delta_t\| \le \frac{3}{2} A \delta_\phi$, where $\delta_\phi = 2\sqrt{2(1 - k(\delta^2/4))}$.*

Proof. First, since the core points $\{\boldsymbol{c}_i\}_{i=1}^r$ are the centers of the hyperspheres, we have $\|\boldsymbol{c}_i - \boldsymbol{x}_i\| \le \delta/2, \ \forall i$. Then according to Theorem 4 in Le et al. (2017), let $\delta_\phi = 2\sqrt{2(1 - k(\delta^2/4))}$, we can obtain

$$\begin{aligned}\|\phi(\boldsymbol{c}_i) - \phi(\boldsymbol{x}_i)\|^2 &= K(\boldsymbol{c}_i, \boldsymbol{c}_i) + K(\boldsymbol{x}_i, \boldsymbol{x}_i) - 2K(\boldsymbol{c}_i, \boldsymbol{x}_i) \\ &= 2(1 - k(\|\boldsymbol{c}_i - \boldsymbol{x}_i\|^2)) \\ &\le 2(1 - k(\delta^2/4)) = \delta_\phi^2/4\end{aligned}$$

Second, Lemma 2, 3, illustrate that for ODM and SVMs, there exists a positive constant A such that $\alpha_t^2 \le A^2$. Based on this, we have

$$\begin{aligned}\|\Delta_t\| &= \|(\alpha_t - \widetilde{\alpha}_t)[\phi(\boldsymbol{c}_t) - \phi(\boldsymbol{x}_t)] \\ &\quad + \frac{1}{m}\sum_{i=1}^{m} \widetilde{\alpha}_i [\phi(\boldsymbol{c}_i) - \phi(\boldsymbol{x}_i)]\| \\ &\le ||\alpha_t| + |\widetilde{\alpha}_t| + \frac{1}{m}\sum_{i=1}^{m} |\widetilde{\alpha}_i|| \cdot \|\phi(\boldsymbol{c}_i) - \phi(\boldsymbol{x}_i)\| \\ &\le \frac{1}{2}\delta_\phi ||\alpha_t| + |\widetilde{\alpha}_t| + \frac{1}{m}\sum_{i=1}^{m} |\widetilde{\alpha}_i|| \le \frac{3}{2} A \delta_\phi\end{aligned}$$

Hence, we gain the conclusion $\|\Delta_t\| \le \frac{3}{2} A \delta_\phi$, where $\delta_\phi = 2\sqrt{2(1 - \kappa(\delta^2/4))}$. □

Theorem 4 explains that the error caused by coreset approximation is always bounded in each iteration. This is because when using coreset approximation, the information of the instances is largely preserved by the core points. Moreover, the theorem also illustrates that the diameter of hypersphere δ can be used to efficiently control the trade-off between sparsity and approximation error.

Linear Convergence Rate

For convergence analysis, in addition to the above approximation error bound, we still need some results about the diameter of the solution domain in expectation.

Lemma 5. *When using CSVRG to train ODM, we have $\mathbb{E}\left[\|\boldsymbol{w}_t - \boldsymbol{w}^*\|^2\right] \le W^2$ for all t, where $W = 2H$.*

Lemma 6. *When using CSVRG to train hinge loss SVM or square hinge loss SVM, there exists a positive constant P such that $\mathbb{E}\left[\|\boldsymbol{w}_t\|^2\right] \le P^2$ for all t, where $P = 2A + \frac{3}{2} A \delta_\phi$.*

Lemma 7. *Assume that $f(\boldsymbol{w})$ is ν-strongly convex, when using CSVRG to train hinge loss SVM or square hinge loss SVM, we have $\mathbb{E}\left[\|\boldsymbol{w}_t - \boldsymbol{w}^*\|^2\right] \leq W^2$ for all t, where $W = \frac{3A\delta_\phi + \sqrt{9A^2\delta_\phi^2 + 16(1-\eta\nu)P^2}}{2\nu}$.*

We denote the expectation upper bound of the diameter of the model domain as W for three different loss functions. Like Lemma 2, 3, the value of W is different in different situations.

Theorem 8. *Considering CSVRG in Algorithm 1 with option II and using it to solve SVMs and ODM, assume that all $\psi_i(\boldsymbol{w})$ are convex and L-smooth, $f(\boldsymbol{w})$ is ν-strongly convex. Let $\boldsymbol{w}^* = argmin_{\boldsymbol{w}} f(\boldsymbol{w})$. Assume that T is sufficiently large so that*

$$\rho = \frac{1}{\nu\eta(1-4L\eta)T} + \frac{4L\eta}{1-4L\eta} < 1$$

then we have linear convergence in expectation for CSVRG:

$$\mathbb{E}[f(\widetilde{\boldsymbol{w}}_s) - f(\boldsymbol{w}^*)] \leq \rho^s \mathbb{E}[f(\widetilde{\boldsymbol{w}}_0) - f(\boldsymbol{w}^*)] + \frac{1-\rho^s}{1-\rho}\Omega$$

where Ω is a constant gap caused by coreset approximation, and $\Omega \to 0$ when the diameter of hyperballs approaches 0.

Proof. First, according to Theorem 1 in Johnson and Zhang (2013), conditioned on $\boldsymbol{w}_{t-1}$, we have

$$\mathbb{E}\|\boldsymbol{v}_t\|^2 \leq 4L[f(\boldsymbol{w}_{t-1}) - f(\boldsymbol{w}^*) + f(\widetilde{\boldsymbol{w}}) - f(\boldsymbol{w}^*)] \quad (11)$$

Then we use Theorem 4 and substitute (10) and (11) to obtain

$$\begin{aligned}
&\mathbb{E}\|\boldsymbol{h}_t\|^2 = \mathbb{E}\|\boldsymbol{v}_t + \Delta_t\|^2 \\
\leq& 2\mathbb{E}\|\boldsymbol{v}_t\|^2 + 2\mathbb{E}\|\Delta_t\|^2 \\
\leq& 8L[f(\boldsymbol{w}_{t-1}) - f(\boldsymbol{w}^*) + f(\widetilde{\boldsymbol{w}}) - f(\boldsymbol{w}^*)] + \frac{9}{2}A^2\delta_\phi^2
\end{aligned}$$

, where the first and third inequality uses $\|a+b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$. Second, conditioned on $\boldsymbol{w}_{t-1}$, we have

$$\begin{aligned}
&\mathbb{E}\|\boldsymbol{w}_t - \boldsymbol{w}^*\|^2 = \mathbb{E}\|\prod_B(\boldsymbol{w}_{t-1} - \eta\boldsymbol{h}_t) - \boldsymbol{w}^*\|^2 \\
\leq& \mathbb{E}\|\boldsymbol{w}_{t-1} - \eta\boldsymbol{h}_t - \boldsymbol{w}^*\|^2 \\
=& \|\boldsymbol{w}_{t-1} - \boldsymbol{w}^*\|^2 + \eta^2\mathbb{E}[\|\boldsymbol{h}_t\|^2] - 2\eta\mathbb{E}[\langle \boldsymbol{w}_{t-1} - \boldsymbol{w}^*, \boldsymbol{v}_t\rangle] \\
&- 2\eta\mathbb{E}[\langle \boldsymbol{w}_{t-1} - \boldsymbol{w}^*, \Delta_t\rangle] \\
\leq& \|\boldsymbol{w}_{t-1} - \boldsymbol{w}^*\|^2 + \eta^2\mathbb{E}[\|\boldsymbol{h}_t\|^2] - 2\eta(\boldsymbol{w}_{t-1} - \boldsymbol{w}^*)^\top\mathbb{E}[\boldsymbol{v}_t] \\
&+ 2\eta\mathbb{E}[\|\boldsymbol{w}_{t-1} - \boldsymbol{w}^*\|^2\|\Delta_t\|^2]^{1/2}
\end{aligned}$$

, where the second inequality is got by using Cauchy-Schwarz inequality. Then we use Lemma 5, 7, and Theorem 4. Noticing that conditioned on $\boldsymbol{w}_{t-1}$, we have $\mathbb{E}[v_t] = \nabla f(\boldsymbol{w}_{t-1})$. And these lead to

$$\begin{aligned}
&\mathbb{E}\|\boldsymbol{w}_t - \boldsymbol{w}^*\|^2 \\
\leq& \|\boldsymbol{w}_{t-1} - \boldsymbol{w}^*\|^2 + \eta^2\mathbb{E}[\|\boldsymbol{h}_t\|^2] - 2\eta(\boldsymbol{w}_{t-1} - \boldsymbol{w}^*)^\top\mathbb{E}[\boldsymbol{v}_t] \\
&+ 3\eta AW\delta_\phi \\
\leq& \|\boldsymbol{w}_{t-1} - \boldsymbol{w}^*\|^2 + \eta^2\mathbb{E}[\|\boldsymbol{h}_t\|^2] + 3\eta AW\delta_\phi \\
&- 2\eta(\boldsymbol{w}_{t-1} - \boldsymbol{w}^*)^\top\nabla f(\boldsymbol{w}_{t-1}) \\
\leq& \|\boldsymbol{w}_{t-1} - \boldsymbol{w}^*\|^2 - 2\eta(1-4L\eta)[f(\boldsymbol{w}_{t-1}) - f(\boldsymbol{w}^*)] \\
&+ 8L\eta^2[f(\widetilde{\boldsymbol{w}}) - f(\boldsymbol{w}^*)] + \frac{9}{2}\eta^2A^2\delta_\phi^2 + 3\eta AW\delta_\phi
\end{aligned}$$

, where the third inequality is got by using the previously obtained inequality for $\mathbb{E}\|\boldsymbol{h}_t\|^2$ and the convexity of $f(w)$, which implies that $-(\boldsymbol{w}_{t-1} - \boldsymbol{w}^*)\nabla f(\boldsymbol{w}_{t-1}) \leq f(\boldsymbol{w}^*) - f(\boldsymbol{w}_{t-1})$.

Then we consider a fixed stage s so that $\widetilde{\boldsymbol{w}} = \widetilde{\boldsymbol{w}}_{s-1}$ and $\widetilde{\boldsymbol{w}}_s$ is selected after all updates have completed. By summing the previous inequalities over $t = 1, \ldots, T$ and taking expectation, we obtain

$$\begin{aligned}
&\mathbb{E}\|\boldsymbol{w}_T - \boldsymbol{w}^*\|^2 + 2\eta(1-4L\eta)T\mathbb{E}[f(\widetilde{\boldsymbol{w}}_s) - f(\boldsymbol{w}^*)] \\
\leq& \mathbb{E}\|\boldsymbol{w}_0 - \boldsymbol{w}^*\|^2 + 8L\eta^2T\mathbb{E}[f(\widetilde{\boldsymbol{w}}) - f(\boldsymbol{w}^*)] \\
&+ \frac{3}{2}\eta TA\delta_\phi(3\eta A\delta_\phi + 2W) \qquad (12)
\end{aligned}$$

Utilizing the strong convexity property of $f(\boldsymbol{w})$ leads to

$$\mathbb{E}\|\boldsymbol{w}_0 - \boldsymbol{w}^*\|^2 = \mathbb{E}\|\widetilde{\boldsymbol{w}} - \boldsymbol{w}^*\|^2 \leq \frac{2}{\nu}\mathbb{E}[f(\widetilde{\boldsymbol{w}}) - f(\boldsymbol{w}^*)]$$

Substituting this into (12), we have

$$\begin{aligned}
&\mathbb{E}\|\boldsymbol{w}_T - \boldsymbol{w}^*\|^2 + 2\eta(1-4L\eta)T\mathbb{E}[f(\widetilde{\boldsymbol{w}}_s) - f(\boldsymbol{w}^*)] \\
\leq& \frac{2}{\nu}\mathbb{E}[f(\widetilde{\boldsymbol{w}}) - f(\boldsymbol{w}^*)] + 8L\eta^2T\mathbb{E}[f(\widetilde{\boldsymbol{w}}) - f(\boldsymbol{w}^*)] \\
&+ \frac{3}{2}\eta TA\delta_\phi(3\eta A\delta_\phi + 2W) \\
=& 2(\nu^{-1} + 4L\eta^2T)\mathbb{E}[f(\widetilde{\boldsymbol{w}}) - f(\boldsymbol{w}^*)] \\
&+ \frac{3}{2}\eta TA\delta_\phi(3\eta A\delta_\phi + 2W)
\end{aligned}$$

Finally, we thus obtain

$$\begin{aligned}
&\mathbb{E}[f(\widetilde{\boldsymbol{w}}_s) - f(\boldsymbol{w}^*)] \\
\leq& [\frac{1}{\nu\eta(1-4L\eta)T} + \frac{4L\eta}{1-4L\eta}]\mathbb{E}[f(\widetilde{\boldsymbol{w}}_{s-1}) - f(\boldsymbol{w}^*)] \\
&+ \frac{3A\delta_\phi(3\eta A\delta_\phi + 2W)}{4(1-4L\eta)}
\end{aligned}$$

This implies

$$\mathbb{E}[f(\widetilde{\boldsymbol{w}}_s) - f(\boldsymbol{w}^*)] \leq \rho^s\mathbb{E}[f(\widetilde{\boldsymbol{w}}_0) - f(\boldsymbol{w}^*)] + \frac{1-\rho^s}{1-\rho}\Omega,$$

where $\Omega = \frac{3A\delta_\phi(3\eta A\delta_\phi + 2W)}{4(1-4L\eta)}$. This indicates that the algorithm converges linearly.

□

The bound we obtained in Theorem 8 is much better than those obtained in AVM, which converges in $O(1/T)$. For simplicity, considering the case where the condition number $L/\nu = m$, we can take $\eta = 0.05/L$ and $T = O(m)$ to obtain a convergence rate of $\rho = 1/2$ and $\frac{1-\rho^s}{1-\rho} < 2$. Thus our method could obtain an approximated solution much faster. Theorem 8 further shows the gap between the optimal solution and the approximate solution. Moreover, this gap Ω can be controlled by the diameter δ of the hyperballs. When δ decreases to 0, the gap Ω also decreases to 0.

Empirical Study

We apply CSVRG to train ODM and conduct comprehensive experiments to evaluate the capacity and efficiency of CSVRG on binary classfication. In the following, we first introduce the experimental setting, then give the analysis of experiemental results. Moreover, we conduct additional empirical study to compare the model size of different methods.

Experimental Setting

We use nine large-scale datasets from UCI and LIBSVM in the experiments. Table 1 summerizes the statistics of these datasets. All features are normalized into the interval $[0, 1]$.

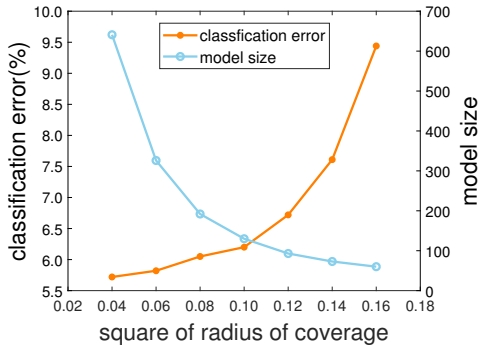
Table 1: Characteristics of 9 large-scale datasets

Dataset	#instance	#feature
magic04	19020	10
adult-a	32561	123
a9a	48842	123
w8a	49749	300
cod-rna	59535	8
mini-boo-ne	130064	50
ijcnn1	141691	22
webspam	350000	254
covtype	581012	54

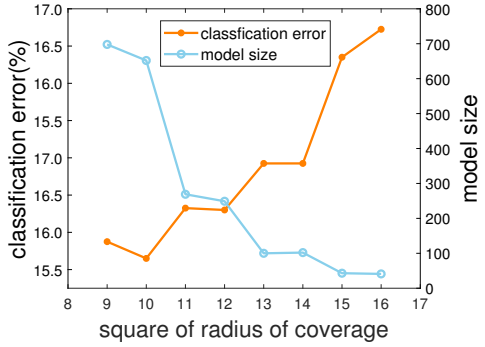
Compared Methods For the non-approximation kernel method, we compared against LIBSVM (Chang and Lin 2011), one of the most widely-used and state-of-the-art implementations for batch kernel SVM solver, and ODM, also a state-of-the-art kernel machine with better statistical property by optimizing margin distribution (Zhang and Zhou 2014; 2016). On the other hand, we also compared with AVM (Le et al. 2017), the state-of-the-art coreset-based kernel machine.

Hyperparameter Throughout the experiments, we utilize RBF kernel for all methods including ours, and the RBF width γ is selected from $\{2^{-4}, 2^{-2}, 2^0, 2^2, 2^4\}$. The regularization parameter C in LIBSVM and λ in ODM are selected from $\{2^1, \ldots, 2^{11}\}$. For ODM, the μ and θ are both selected from $\{0.2, 0.4, 0.6, 0.8\}$. For λ in AVM, it is selected from $\{2^{-11}, 2^{-9}, \ldots, 2^{-1}\}$, which is corresponding to the λ in ODM. The hyperparameters range of our method CSVRG+ODM is the same as ODM. All the hyperparameters are specified using 5-fold cross-validation on training sets. All experiments are repeated for 10 times.

The Effect of Diameter In CSVRG, we have one extra hyperparameter, i.e., the diameter of δ-coverage, which controls the degree of approximation. We study its effect using the method in Le et al. (2017). Intuitively, the larger the radius, the smaller the number of cores and the higher the degree of approximation. Specifically, Figure 1 shows the effect of the diameter on the classfication error and model size. We could learn that the diameter of δ-coverage is a trade-off parameter between performance and model size, which let us adjust freely as needed. Furthermore, the controllable trade-off parameter gives us some guidance when the method is used in practical. For hyperparameter selection, we can first set a large diameter of hyperballs to perform cross-validation efficiently. Then we can use obtained parameters to train the model of the required complexity.



(a) cod-rna



(b) adult-a

Figure 1: The effect of δ-coverage radius on the classification error and model size

Results Analysis

In the experiments, we manually choose the appropriate diameter so that the number of core points is between 100 and 1000. The average accuracies (with standard deviations), average training time and average prediction time are reported in Table 2. Overall, the non-estimated kernel algorithms achieve the highest classfication accuracies. However, our method has a substantial speed-up while maintaining a competitive accuracy. Specifically, for each dataset, the training and prediction time costs of CSVRG are the smallest with orders of magnitude lower than SVMs and ODM,

Table 2: Classfication performance of our CSVRG and the comparison methods in batch mode. The notation $[S|D]$ next to the dataset name denotes the number of core points in CSVRG and AVM respectively. The accuracy is presented as a percentage $(\%)$. The training time and testing time are in second. The best performance and the least time cost are in **bold**. Running out of memory or running for more than two hours will terminate the program.

Dataset $[S\|D]$	`magic04` [359|1000]			`adult-a` [259|269]			`a9a` [128|338]		
Algorithm	Train	Test	Accuracy	Train	Test	Accuracy	Train	Test	Accuracy
LIBSVM	50.88	1.11	**87.00±0.28**	13.12	24.92	84.50±0.28	108.61	13.11	84.79±0.00
ODM	5.52	0.25	86.55±0.27	32.88	0.68	**84.61±0.24**	105.75	1.69	**85.29±0.03**
AVM	**1.12**	0.64	82.05±0.13	2.47	1.39	83.21±0.32	5.88	1.71	81.46±0.35
CSVRG+ODM	1.32	**0.09**	84.43±0.40	**1.68**	**0.33**	84.08±0.22	**1.70**	**0.18**	84.26±0.20
Dataset $[S\|D]$	`w8a` [213|498]			`cod-rna` [275|876]			`mini-boo-ne` [229|526]		
Algorithm	Train	Test	Accuracy	Train	Test	Accuracy	Train	Test	Accuracy
LIBSVM	14.50	29.22	**98.67±0.05**	37.96	3.78	95.52±0.10	243.03	84.36	**92.22±0.07**
ODM	111.34	1.93	98.57±0.08	50.79	2.79	**95.57±0.09**	-	-	-
AVM	19.79	29.15	97.09±0.08	2.94	1.61	91.32±1.39	9.03	5.52	83.50±0.25
CSVRG+ODM	**2.44**	**0.81**	97.14±0.07	**1.78**	**0.15**	94.22±0.14	**8.40**	**1.47**	85.21±0.53
Dataset $[S\|D]$	`ijcnn1` [296|502]			`webspam` [643|560]			`covtype` [132|132]		
Algorithm	Train	Test	Accuracy	Train	Test	Accuracy	Train	Test	Accuracy
LIBSVM	80.00	16.32	**99.31±0.03**	-	-	-	-	-	-
ODM	-	-	-	-	-	-	-	-	-
AVM	**7.03**	3.70	90.45±0.01	157.40	196.11	78.30±1.05	22.79	9.57	70.25±0.28
CSVRG+ODM	9.32	**0.95**	91.33±0.07	**31.32**	**24.17**	**83.90±0.21**	**14.44**	**1.68**	**73.22±0.17**

and at the same time, it ensures that the generalization performance is very close to SVMs and ODM. On the other hand, the LIBSVM and ODM with RBF kernel could not be trained within acceptable amount of time and memory on large-scale datasets.

In the comparison between CSVRG and AVM, we find that CSVRG is more efficient than AVM while ensuring better performance. According to Table 2, for *a9a*, *cod-rna*, *mini-boo-ne*, etc., to achieve similar accuracy, CSVRG need less number of core points, which verifies that margin distribution is more crucial than minimum margin for generalization. For *adult-a*, *webspam* and *covtype* datasets, CSVRG is more efficient while the number of core points of two algorithms is close, which indicates that CSVRG has faster convergence than SGD. In a nutshell, the experimental results show that we can achieve the "best of both worlds", i.e., the best efficiency as well as the best generalization.

Comparison of Model Size

Figure 2 shows the logarithmic comparison of model size in above experiments. For LIBSVM, the model size corresponds to the number of support vectors. For AVM and CSVRG, the model size corresponds to the size of coreset.

The figure indicates that the model complexity of our method and AVM is serveral orders of magnitude lower than LIBSVM, and our method achieves the best sparsity. Furthermore, we find that CSVRG+ODM can use only half the core points to achieve better generalization performance than AVM, which implies that our method can do better with even smaller coreset. This result verifies the better statistical property of ODM than SVM.

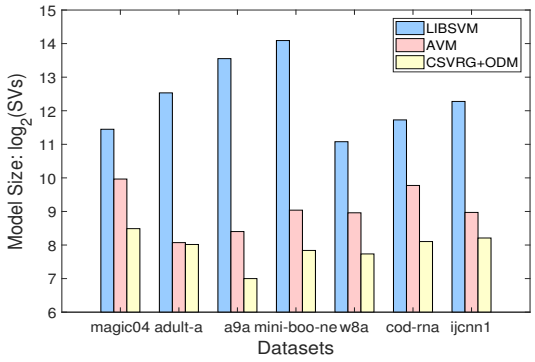


Figure 2: Comparison of the model size

Conclusion

In this paper, we propose a novel large-scale kernel-accelerating method CSVRG (**C**oreset **S**tochastic **V**ariance-**R**educed **G**radient) by applying coreset approximation to SVRG. Then we theoretically prove that our method converges linearly. By applying CSVRG to ODM, the experimental results show the superiority of our method in both efficiency and generalization compared to the state-of-the-art methods. The theoretical analysis show that there is a gap between the optimal solution and the approximated solution, how to reduce this gap will be an interesting future work.

Acknowledgments

This research was supported by the National Key R&D Program of China (2018YFB1004300), the NSFC (61673202), and the Fundamental Research Funds for the Central Universities.

References

Asharaf, S.; Murty, M. N.; and Shevade, S. K. 2007. Multiclass core vector machine. In *Proceedings of the 24th International Conference on Machine Learning*, 41–48.

Bădoiu, M., and Clarkson, K. L. 2008. Optimal core-sets for balls. *Computational Geometry* 40(1):14–22.

Chang, C.-C., and Lin, C.-J. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* 2(3):27:1–27:27.

Cortes, C., and Vapnik, V. 1995. Support-vector networks. *Machine learning* 20(3):273–297.

Cucker, F., and Smale, S. 2002. On the mathematical foundations of learning. *Bulletin of the American Mathematical Society* 39(1):1–49.

Drineas, P., and Mahoney, M. W. 2005. On the nyström method for approximating a gram matrix for improved kernel-based learning. *Journal of Machine Learning Research* 6(Dec):2153–2175.

Gao, W., and Zhou, Z.-H. 2013. On the doubt about margin explanation of boosting. *Artificial Intelligence* 203:1–18.

Gittens, A., and Mahoney, M. W. 2016. Revisiting the nyström method for improved large-scale machine learning. *Journal of Machine Learning Research* 17(1):3977–4041.

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

Le, T.; Nguyen, T. D.; Nguyen, V.; and Phung, D. 2017. Approximation vector machines for large-scale online learning. *Journal of Machine Learning Research* 18(1):3962–4016.

Le, Q.; Sarlós, T.; and Smola, A. 2013. Fastfood-computing hilbert space expansions in loglinear time. In *Proceedings of the 30th International Conference on Machine Learning*, 244–252.

Loosli, G., and Canu, S. 2007. Comments on the"core vector machines: Fast svm training on very large data sets. *Journal of Machine Learning Research* 8(Feb):291–301.

Rahimi, A., and Recht, B. 2008. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, 1177–1184.

Rahimi, A., and Recht, B. 2009. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In *Advances in Neural Information Processing Systems*, 1313–1320.

Rasmussen, C. E. 2004. Gaussian processes in machine learning. In *Advanced Lectures on Machine Learning*, 63–71.

Schölkopf, B.; Smola, A.; and Müller, K.-R. 1998. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation* 10(5):1299–1319.

Shalev-Shwartz, S.; Singer, Y.; Srebro, N.; and Cotter, A. 2011. Pegasos: Primal estimated sub-gradient solver for svm. *Mathematical Programming* 127(1):3–30.

Smola, A. J., and Schölkopf, B. 2004. A tutorial on support vector regression. *Statistics and computing* 14(3):199–222.

Tsang, I. W.; Kocsor, A.; and Kwok, J. T. 2007. Simpler core vector machines with enclosing balls. In *Proceedings of the 24th International Conference on Machine Learning*, 911–918.

Tsang, I. W.; Kwok, J. T.; and Cheung, P.-M. 2005. Core vector machines: Fast svm training on very large data sets. *Journal of Machine Learning Research* 6(Apr):363–392.

Tsang, I. W.; Kwok, J. T.; and Zurada, J. M. 2006. Generalized core vector machines. *IEEE Transactions on Neural Networks* 17(5):1126–1140.

Williams, C. K., and Seeger, M. 2001. Using the nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems*, 682–688.

Yang, T.; Li, Y.-F.; Mahdavi, M.; Jin, R.; and Zhou, Z.-H. 2012. Nyström method vs random fourier features: A theoretical and empirical comparison. In *Advances in Neural Information Processing Systems*, 476–484.

Zhang, T., and Zhou, Z.-H. 2014. Large margin distribution machine. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 313–322.

Zhang, T., and Zhou, Z.-H. 2016. Optimal margin distribution machine. *CoRR* abs/1604.03348.

Zhang, T., and Zhou, Z.-H. 2017. Multi-class optimal margin distribution machine. In *Proceedings of the 34th International Conference on Machine Learning*, 4063–4071.

Zhang, T., and Zhou, Z.-H. 2018. Optimal margin distribution clustering. In *Proceedings of the 20th National Conference on Artificial Intelligence*, 4474–4481.

Zhang, K.; Tsang, I. W.; and Kwok, J. T. 2008. Improved nyström low-rank approximation and error analysis. In *Proceedings of the 25th International Conference on Machine Learning*, 1232–1239.

Zhou, Y.-H., and Zhou, Z.-H. 2016. Large margin distribution learning with cost interval and unlabeled data. *IEEE Transactions on Knowledge and Data Engineering* 28(7):1749–1763.

Supplementary Materials for Coreset Stochastic Variance-Reduced Gradient with Application to Optimal Margin Distribution Machine

Zhi-Hao Tan and **Teng Zhang** and **Wei Wang**
National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China
Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing 210023, China
{tanzh, zhangt, wangw}@lamda.nju.edu.cn

In the supplementary material, we will give the detailed proofs of lemma and theorems in main paper. Without loss of generality, we assume that $\|\phi(x)\| = K(x,x)^{1/2} = 1,\ \forall x \in \mathcal{D}$. And we only consider the binary classification, so the label y is either -1 or 1 which implies $|y| = y^2 = 1$.

Bounded Approximation Error

In this section, we will present the detailed proofs of Theorem 1, Lemma 2 and Lemma 3, i.e., the intermediate results when we give the upper bound of approximation error.

Theorem 1. *If $\boldsymbol{w}^*_{odm}$ is the optimal solution of ODM, then there exists a positive constant H such that $\|\boldsymbol{w}^*_{odm}\| \leq H$, where $H = \frac{\sqrt{\lambda(1-\theta)^2+\lambda\mu(1+\theta)^2}}{1-\theta}$. Moreover, for squared hinge loss SVM, the optimal solution $\boldsymbol{w}^*_{h^2}$ satisfies $\boldsymbol{w}^*_{h^2} \leq \lambda$.*

Proof. **1 For ODM**

The optimization problem of ODM can be reformulated as follows:

$$\begin{aligned}\min_{\boldsymbol{w},\boldsymbol{\xi},\boldsymbol{\epsilon}}\ & \frac{1}{2}\boldsymbol{w}^\top\boldsymbol{w} + \frac{\lambda}{m(1-\theta)^2}\boldsymbol{\xi}^\top\boldsymbol{\xi} + \frac{\lambda\mu}{m(1-\theta)^2}\boldsymbol{\epsilon}^\top\boldsymbol{\epsilon} \\ \text{s.t. }\ & \boldsymbol{Y}\boldsymbol{x}^\top\boldsymbol{w} \geq (1-\theta)\boldsymbol{e} - \boldsymbol{\xi}, \\ & \boldsymbol{Y}\boldsymbol{x}^\top\boldsymbol{w} \leq (1+\theta)\boldsymbol{e} + \boldsymbol{\epsilon}.\end{aligned} \tag{1}$$

where $\boldsymbol{x}$ is the matrix whose i-th column is $\phi(\boldsymbol{x}_i)$, i.e., $\boldsymbol{x} = [\phi(\boldsymbol{x}_1), \ldots, \phi(\boldsymbol{x}_m)]$, $\boldsymbol{Y}$ is a $m \times m$ diagonal matrix with $y_1, \ldots, y_m$ as the diagonal elements and $\boldsymbol{e}$ stands for the all-one vector.

Introduce the Lagrange multipliers $\boldsymbol{\zeta} \geq 0$ and $\boldsymbol{\beta} \geq 0$ for the two constraints respectively, the Largrangian of (1) leads to

$$\begin{aligned}&\mathcal{L}(\boldsymbol{w},\boldsymbol{\xi},\boldsymbol{\epsilon},\boldsymbol{\alpha},\boldsymbol{\beta}) \\ =&\frac{1}{2}\boldsymbol{w}^\top\boldsymbol{w} + \frac{\lambda}{m(1-\theta)^2}\boldsymbol{\xi}^\top\boldsymbol{\xi} + \frac{\lambda\mu}{m(1-\theta)^2}\boldsymbol{\epsilon}^\top\boldsymbol{\epsilon} \\ &- \boldsymbol{\zeta}^\top\left(\boldsymbol{Y}\boldsymbol{X}^\top\boldsymbol{w} - (1-\theta)\boldsymbol{e} + \boldsymbol{\xi}\right) \\ &+ \boldsymbol{\beta}^\top\left(\boldsymbol{Y}\boldsymbol{X}^\top\boldsymbol{w} - (1+\theta)\boldsymbol{e} - \boldsymbol{\epsilon}\right).\end{aligned} \tag{2}$$

By setting the partial derivative of $\boldsymbol{w}, \boldsymbol{\xi}, \boldsymbol{\epsilon}$ to zero, we have

$$\begin{aligned}&\frac{\partial\mathcal{L}}{\partial\boldsymbol{w}} = \boldsymbol{w} - \boldsymbol{X}\boldsymbol{Y}\boldsymbol{\zeta} + \boldsymbol{X}\boldsymbol{Y}\boldsymbol{\beta} = 0 \Longrightarrow \boldsymbol{w} = \boldsymbol{X}\boldsymbol{Y}(\boldsymbol{\zeta} - \boldsymbol{\beta}) \\ &\frac{\partial\mathcal{L}}{\partial\boldsymbol{\xi}} = \frac{2\lambda}{m(1-\theta)^2}\boldsymbol{\xi} - \boldsymbol{\zeta} = 0 \Longrightarrow \boldsymbol{\xi} = \frac{m(1-\theta)^2}{2\lambda}\boldsymbol{\zeta} \\ &\frac{\partial\mathcal{L}}{\partial\boldsymbol{\epsilon}} = \frac{2\lambda\mu}{m(1-\theta)^2}\boldsymbol{\epsilon} - \boldsymbol{\beta} = 0 \Longrightarrow \boldsymbol{\epsilon} = \frac{m(1-\theta)^2}{2\lambda\mu}\boldsymbol{\beta}\end{aligned}$$

Substituting the above to (2), we gain the dual form

$$\begin{aligned}\mathcal{G}(\boldsymbol{\alpha}) = &-\frac{1}{2}\boldsymbol{w}^\top\boldsymbol{w} - \frac{m(1-\theta)^2}{4\lambda}\boldsymbol{\zeta}^\top\boldsymbol{\zeta} - \frac{m(1-\theta)^2}{4\lambda\mu}\boldsymbol{\beta}^\top\boldsymbol{\beta} \\ &+ (1-\theta)\boldsymbol{\zeta}^\top\boldsymbol{e} - (1+\theta)\boldsymbol{\beta}^\top\boldsymbol{e}\end{aligned} \tag{3}$$

Let us denote $(\boldsymbol{w}^*, \boldsymbol{\xi}^*, \boldsymbol{\epsilon}^*)$ and $(\boldsymbol{\zeta}^*, \boldsymbol{\beta}^*)$ be the primal and dual solutions of (1) and (3), respectively. Since the strong duality holds, we have

$$\begin{aligned}&\frac{1}{2}\|\boldsymbol{w}^*\|^2 + \frac{\lambda}{m(1-\theta)^2}\sum_{i=1}^m\left(\xi_i^{*2} + \mu\epsilon_i^{*2}\right) \\ =&-\frac{1}{2}\|\boldsymbol{w}^*\|^2 - \frac{m(1-\theta)^2}{4\lambda}\sum_{i=1}^m\zeta_i^{*2} - \frac{m(1-\theta)^2}{4\lambda\mu}\sum_{i=1}^m\beta_i^{*2} \\ &+ (1-\theta)\sum_{i=1}^m\zeta_i^* - (1+\theta)\sum_{i=1}^m\beta_i^*\end{aligned}$$

Since the loss of ODM is non-negative, we have

$$\begin{aligned}\|\boldsymbol{w}^*\|^2 \leq &-\frac{m(1-\theta)^2}{4\lambda}\sum_{i=1}^m\zeta_i^{*2} - \frac{m(1-\theta)^2}{4\lambda\mu}\sum_{i=1}^m\beta_i^{*2} \\ &+ (1-\theta)\sum_{i=1}^m\zeta_i^* - (1+\theta)\sum_{i=1}^m\beta_i^* \\ =&\sum_{i=1}^m\left((1-\theta)\zeta_i^* - \frac{m(1-\theta)^2}{4\lambda}\zeta_i^{*2}\right) \\ &+ \sum_{i=1}^m\left((1+\theta)\beta_i^* - \frac{m(1-\theta)^2}{4\lambda\mu}\beta_i^{*2}\right) \\ \leq&\sum_{i=1}^m\frac{\lambda}{m} + \sum_{i=1}^m\frac{\lambda\mu(1+\theta)^2}{m(1-\theta)^2}\end{aligned}$$

$$=\frac{\lambda\left(1-\theta\right)^2+\lambda\mu\left(1+\theta\right)^2}{\left(1-\theta\right)^2}=H^2$$

We note that the second inequality uses that the maximum of the quadratic concave function is obtained when $\zeta_i^*=\frac{2\lambda}{m(1-\theta)}$ and $\beta_i^*=\frac{2\lambda\mu(1+\theta)}{m(1-\theta)^2}$. Therefore, for ODM, $\|\boldsymbol{w}_{odm}^*\|\leq H$, where $H=\frac{\sqrt{\lambda(1-\theta)^2+\lambda\mu(1+\theta)^2}}{1-\theta}$.

2 For Square Hinge Loss SVM

The optimization of square hinge loss SVM is as follows:

$$\begin{aligned}&\min_{\boldsymbol{w},\boldsymbol{\xi}}\ \frac{1}{2}\|\boldsymbol{w}\|^2+\frac{\lambda}{m}\sum_{i=1}^m\xi_i^2\\&\text{s.t. } y_i\boldsymbol{w}^\top\phi(\boldsymbol{x}_i)\geq 1-\xi_i,\ i=1,\ldots,m\end{aligned}\tag{4}$$

where $\boldsymbol{\xi}=[\xi_i]_{i=1}^m$. The Lagrange function is of the following form

$$\begin{aligned}\mathcal{L}(\boldsymbol{w},\boldsymbol{\xi},\boldsymbol{\alpha})=&\frac{1}{2}\|\boldsymbol{w}\|^2+\frac{\lambda}{m}\sum_{i=1}^m\xi_i^2\\&+\sum_{i=1}^m\alpha_i\left(1-\xi_i-y_i\boldsymbol{w}^\top\phi(\boldsymbol{x}_i)\right)\end{aligned}\tag{5}$$

By setting the partial derivative of $\boldsymbol{w}$, $\boldsymbol{\xi}$ to zero, we have

$$\begin{aligned}&\frac{\partial\mathcal{L}}{\partial\boldsymbol{w}}=\boldsymbol{w}-\sum_{i=1}^m\alpha_i y_i\phi(\boldsymbol{x}_i)=0\Longrightarrow\boldsymbol{w}=\sum_{i=1}^m\alpha_i y_i\phi(\boldsymbol{x}_i)\\&\frac{\partial\mathcal{L}}{\partial\xi_i}=\frac{2\lambda}{m}\xi_i-\alpha_i=0\Longrightarrow\xi_i=\frac{m}{2\lambda}\alpha_i\end{aligned}$$

Substituting the above to (5), we gain the dual form

$$\mathcal{W}(\boldsymbol{\alpha})=-\frac{1}{2}\|\boldsymbol{w}\|^2-\frac{m}{4\lambda}\sum_{i=1}^m\alpha_i^2+\sum_{i=1}^m\alpha_i\tag{6}$$

Let us denote $(\boldsymbol{w}^*,\boldsymbol{\xi}^*)$ and $\boldsymbol{\alpha}^*$ be the primal and dual solutions of (4) and (5) respectively. Since the strong duality holds, we have

$$\frac{1}{2}\|\boldsymbol{w}^*\|^2+\frac{\lambda}{m}\sum_{i=1}^m\xi_i^{*2}=-\frac{1}{2}\|\boldsymbol{w}^*\|^2-\frac{m}{4\lambda}\sum_{i=1}^m\alpha_i^{*2}+\sum_{i=1}^m\alpha_i^*$$

Since the loss of square hinge loss SVM is non-negative, we have

$$\begin{aligned}\|\boldsymbol{w}^*\|^2=&-\frac{m}{4\lambda}\sum_{i=1}^m\alpha_i^{*2}+\sum_{i=1}^m\alpha_i^*-\frac{\lambda}{m}\sum_{i=1}^m\xi_i^{*2}\\\leq&\sum_{i=1}^m\left(-\frac{m}{4\lambda}\alpha_i^{*2}+\alpha_i^*\right)\\\leq&\lambda\end{aligned}$$

Therefore, for square hinge loss SVM, we have $\boldsymbol{w}_{h^2}^*\leq\lambda$. In summary, we gain the conclusion. The proof of Theorem 1 is similar to that of Theorem 1 in (Shalev-Shwartz et al. 2011). □

Lemma 2. *For ODM problem, the α_t satisfies $\alpha_t^2\leq A^2$ for all t, where $A=\frac{2(H+1+\theta)}{(1-\theta)^2}$.*

Proof. We first remind the representation of α_t

$$\begin{aligned}\alpha_t=&\frac{2\lambda}{(1-\theta)^2}\{\left(y_t\boldsymbol{w}^\top\phi\left(\boldsymbol{x}_t\right)+\theta-1\right)y_t\mathbb{I}\left(t\in I_1\right)\\&+\mu\left(y_t\boldsymbol{w}^\top\phi\left(\boldsymbol{x}_t\right)-\theta-1\right)y_t\mathbb{I}\left(t\in I_2\right)\}\end{aligned}$$

where $I_1\equiv\left\{i\mid y_t\boldsymbol{w}^\top\phi\left(\boldsymbol{x}_t\right)<1-\theta\right\}$ and $I_2\equiv\left\{i\mid y_t\boldsymbol{w}^\top\phi\left(\boldsymbol{x}_t\right)>1+\theta\right\}$. Then we can obtain

$$\begin{aligned}\|\alpha_t\|\leq&\frac{2\lambda}{\left(1-\theta\right)^2}\left(\left\|\left(y_t\boldsymbol{w}_t^\top\phi\left(\boldsymbol{x}_t\right)+\theta-1\right)\right\|\mathbb{I}(t\in I_2)\right.\\&\left.+\mu\left\|\left(y_t\boldsymbol{w}_t^\top\phi\left(\boldsymbol{x}_t\right)-\theta-1\right)\right\|\mathbb{I}(t\in I_3)\right)\\\leq&\frac{2\lambda}{\left(1-\theta\right)^2}\left(\left(\left\|\boldsymbol{w}_t^\top\phi\left(\boldsymbol{x}_t\right)\right\|+\|\theta-1\|\right)\mathbb{I}(t\in I_2)\right.\\&\left.+\mu\left(\left\|\boldsymbol{w}_t^\top\phi\left(\boldsymbol{x}_t\right)\right\|+\|\theta+1\|\right)\mathbb{I}(t\in I_3)\right)\\\leq&\frac{2\lambda}{\left(1-\theta\right)^2}\left(\left(\|\boldsymbol{w}_t\|\left\|\phi\left(\boldsymbol{x}_t\right)\right\|+1-\theta\right)\mathbb{I}(t\in I_2)\right.\\&\left.+\mu\left(\|\boldsymbol{w}_t\|\left\|\phi\left(\boldsymbol{x}_t\right)\right\|+1+\theta\right)\mathbb{I}(t\in I_3)\right)\\=&\frac{2\lambda}{\left(1-\theta\right)^2}\left(\left(\|\boldsymbol{w}_t\|+1-\theta\right)\mathbb{I}(t\in I_2)\right.\\&\left.+\mu\left(\|\boldsymbol{w}_t\|+1+\theta\right)\mathbb{I}(t\in I_3)\right)\\\leq&\frac{2\lambda}{\left(1-\theta\right)^2}\left(\|\boldsymbol{w}_t\|+1+\theta\right)\\\leq&\frac{2\lambda\left(H+1+\theta\right)}{\left(1-\theta\right)^2}\end{aligned}$$

Note that the first and second inequalities use Minkowski inequality, the third inequality uses Cauchy-Schwarz inequality, and the fourth inequality uses that $\mu\leq 1$ and $0\leq\theta\leq 1$. Therefore, we have $\alpha_t^2\leq A^2$, where $A=\frac{2(H+1+\theta)}{(1-\theta)^2}$. □

Lemma 3. *For hinge loss SVM and square hinge loss SVM, the α_t satisfies $\alpha_t^2\leq A^2=\lambda^2B^2$ for all t, where $B=1$ and $B=2\lambda+2$ respectively.*

Proof. For hinge loss SVM, we have

$$\begin{aligned}l\left(\boldsymbol{w};\boldsymbol{x},y\right)&=\max\left\{0,1-y\boldsymbol{w}^\top\phi\left(\boldsymbol{x}\right)\right\}\\l'\left(\boldsymbol{w};\boldsymbol{x},y\right)&=-\mathbb{I}_{\{y\boldsymbol{w}^\top\phi(\boldsymbol{x})\leq 1\}}y\phi\left(\boldsymbol{x}\right)\end{aligned}$$

where $\mathbb{I}_S$ is the indicator function, which equals 1 if the logical statement S is true and 0 otherwise. Therefore, by taking $B=1$, we have

$$\left\|l'\left(\boldsymbol{w};\boldsymbol{x},y\right)\right\|\leq\left\|\phi\left(\boldsymbol{x}\right)\right\|\leq 1=B$$

For squared hinge loss SVM, we have

$$\begin{aligned}l\left(\boldsymbol{w};\boldsymbol{x},y\right)&=\max\left\{0,1-y\boldsymbol{w}^\top\phi\left(\boldsymbol{x}\right)\right\}^2\\l'\left(\boldsymbol{w};\boldsymbol{x},y\right)&=-\mathbb{I}_{\{y\boldsymbol{w}^\top\phi(\boldsymbol{x})\leq 1\}}2y\phi\left(\boldsymbol{x}\right)\left(1-y\boldsymbol{w}^\top\phi\left(\boldsymbol{x}\right)\right)\end{aligned}$$

Then by taking $B = 2\lambda + 2$ we can obtain

$$\begin{aligned}\|l'(\boldsymbol{w};\boldsymbol{x},y)\| &\le \left\|2y\phi(\boldsymbol{x})\left(1-y\boldsymbol{w}^\top\phi(\boldsymbol{x})\right)\right\| \\ &\le 2\left\|1-y\boldsymbol{w}^\top\phi(\boldsymbol{x})\right\|\|\phi(\boldsymbol{x})\||y| \\ &\le 2\left\|y\boldsymbol{w}^\top\phi(\boldsymbol{x})\right\|+2 \\ &\le 2\|\phi(\boldsymbol{x})\|\|\boldsymbol{w}\||y|+2 \\ &= 2\|\boldsymbol{w}\|+2 \\ &\le 2\lambda+2=B\end{aligned}$$

The last inequality uses Theorem 1.

Therefore, there exists a positive constant B such that $\|l'(\boldsymbol{w};\boldsymbol{x},y)\| \le B$ for hinge loss SVM and square hinge loss SVM. Based on this, we have the following proof.

Note that $\|\phi(\boldsymbol{x})\| = K(x,x) = 1$, by taking $A = \lambda B$, we have

$$\alpha_t^2 = \alpha_t^2 K(\boldsymbol{x}_t,\boldsymbol{x}_t) = \lambda^2\|l'(\boldsymbol{w}_t;\boldsymbol{x}_t,y_t)\|^2 \le \lambda^2 B^2 = A^2$$

Therefore we gain the conclusion $\alpha_t^2 \le A^2$. □

Based on te above results, we obtain the Theorem 4, whose proof is in the main paper.

Theorem 4. *Assume that the p.s.d. and isotropic kernel $K(\boldsymbol{x}_i,\boldsymbol{x}_j) = k(\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2)$ is used, where $k(.)$ is a monotonically continuous decreasing function with $k(0) = 1$. And let δ be the diameter of coreset coverage. Then for the approximation error Δ_t as indicated in (10), we have $\|\Delta_t\| \le \frac{3}{2}A\delta_\phi$, where $\delta_\phi = 2\sqrt{2(1-\kappa(\delta^2/4))}$.*

Convergence Analysis

In this section, we will present the proof of intermediate conclusions when analysing the convergence of CSVRG. The proofs of Lemma 6,7 are similar to the proof of Lemma 21, Lemma 22 in (Le et al. 2017).

Lemma 5. *When using CSVRG to train ODM, we have $\mathbb{E}\left[\|\boldsymbol{w}_t-\boldsymbol{w}^*\|^2\right] \le W^2$ for all t, where $W = 2H$.*

Proof. According to Theorem 1, by taking $W = 2H$, we have the following

$$\begin{aligned}\mathbb{E}\left[\|\boldsymbol{w}_t-\boldsymbol{w}^*\|^2\right] &\le 2\mathbb{E}\left[\|\boldsymbol{w}_t\|^2\right]+2\mathbb{E}\left[\|\boldsymbol{w}^*\|^2\right] \\ &\le 4H^2 = W^2\end{aligned}$$

□

Lemma 6. *When using CSVRG to train hinge loss SVM or square hinge loss SVM, there exists a positive constant P such that $\mathbb{E}\left[\|\boldsymbol{w}_t\|^2\right] \le P^2$ for all t, where $P = 2A+\frac{3}{2}A\delta_\phi$.*

Proof. We prove by induction that $\mathbb{E}\left[\|\boldsymbol{w}_t\|^2\right] \le P^2$ where $P = 2A+\frac{3}{2}A\delta_\phi$ for all t. Assume that the claim is holding for $t-1$, we have the detailed proof in Figure 1.

The third inequality uses $\sqrt{\mathbb{E}\|\boldsymbol{a}+\boldsymbol{b}\|^2} \le \sqrt{\mathbb{E}\|\boldsymbol{a}\|^2}+\sqrt{\mathbb{E}\|\boldsymbol{b}\|^2}$. The fourth inequality uses Lemma 3. And the fifth inequality uses $\mathbb{E}\|\boldsymbol{\xi}-\mathbb{E}\boldsymbol{\xi}\|^2 = \mathbb{E}\|\boldsymbol{\xi}\|^2-\|\mathbb{E}\boldsymbol{\xi}\|^2 \le \mathbb{E}\|\boldsymbol{\xi}\|^2$ for any random vector $\boldsymbol{\xi}$. Finally, the sixth inequality uses Theorem 4. □

Lemma 7. *Assume that $f(\boldsymbol{w})$ is ν-strongly convex, when using CSVRG to train hinge loss SVM or square hinge loss SVM, we have $\mathbb{E}\left[\|\boldsymbol{w}_t-\boldsymbol{w}^*\|^2\right] \le W^2$ for all t, where $W = \frac{3A\delta_\phi+\sqrt{9A^2\delta_\phi^2+16(1-\eta\nu)P^2}}{2\nu}$.*

Proof. We first remind the definitions of the relevant variables

$$\begin{aligned}\boldsymbol{g} &= \boldsymbol{w}+\lambda l'(\boldsymbol{w};\boldsymbol{x}_t,y_t) = \boldsymbol{w}+\alpha_t\phi(\boldsymbol{x}_t) \\ \boldsymbol{v}_t &= \nabla\psi_t(\boldsymbol{w}_{t-1})-\nabla\psi_t(\widetilde{\boldsymbol{w}})+\widetilde{\boldsymbol{\mu}} \\ &= \boldsymbol{w}_{t-1}+(\alpha_t-\widetilde{\alpha}_t)\phi(\boldsymbol{x}_t)+\frac{1}{m}\sum_{i=1}^m\widetilde{\alpha}_i\phi(\boldsymbol{x}_i) \\ \Delta_t &= (\alpha_t-\widetilde{\alpha}_t)[\phi(\boldsymbol{c}_t)-\phi(\boldsymbol{x}_t)] \\ &\quad+\frac{1}{m}\sum_{i=1}^m\widetilde{\alpha}_i[\phi(\boldsymbol{c}_i)-\phi(\boldsymbol{x}_i)] \\ \boldsymbol{h}_t &= \boldsymbol{v}_t+\Delta_t\end{aligned}$$

Let $\boldsymbol{d}_t = \boldsymbol{h}_t-\boldsymbol{w}_{t-1}$, according to the proof of lemma 2, we have

$$\begin{aligned}(1-\eta)\sqrt{\mathbb{E}\|\boldsymbol{w}_{t-1}\|^2}+\eta\sqrt{\mathbb{E}\|\boldsymbol{d}_t\|^2} &\le P \\ \Longrightarrow \sqrt{\mathbb{E}\|\boldsymbol{d}_t\|^2} &\le P\end{aligned}$$

Therefore, we can obtain

$$\sqrt{\mathbb{E}\|\boldsymbol{h}_t\|^2} \le \sqrt{\mathbb{E}\|\boldsymbol{w}_{t-1}\|^2}+\sqrt{\mathbb{E}\|\boldsymbol{d}_t\|^2} \le 2P \quad (7)$$

Conditioned on $\boldsymbol{w}_{t-1}$, we have $\mathbb{E}[\boldsymbol{v}_t] = \nabla f(\boldsymbol{w}_{t-1})$. Then we can obtain

$$\begin{aligned}&\mathbb{E}\left[\|\boldsymbol{w}_t-\boldsymbol{w}^*\|^2\right] = \mathbb{E}\left\|\prod_S(\boldsymbol{w}_{t-1}-\eta\boldsymbol{h}_t)-\boldsymbol{w}^*\right\|^2 \\ \le&\mathbb{E}\left[\|\boldsymbol{w}_{t-1}-\eta\boldsymbol{h}_t-\boldsymbol{w}^*\|^2\right] \\ =&\|\boldsymbol{w}_{t-1}-\boldsymbol{w}^*\|^2+\eta^2\mathbb{E}\left[\|\boldsymbol{h}_t\|^2\right]-2\eta\mathbb{E}[\langle\boldsymbol{w}_{t-1}-\boldsymbol{w}^*,\boldsymbol{v}_t\rangle] \\ &-2\eta\mathbb{E}[\langle\boldsymbol{w}_{t-1}-\boldsymbol{w}^*,\Delta_t\rangle] \\ =&\|\boldsymbol{w}_{t-1}-\boldsymbol{w}^*\|^2+\eta^2\mathbb{E}\left[\|\boldsymbol{h}_t\|^2\right]-2\eta(\boldsymbol{w}_{t-1}-\boldsymbol{w}^*)^\top\mathbb{E}[\boldsymbol{v}_t] \\ &+2\eta\mathbb{E}\left[\|\boldsymbol{w}_{t-1}-\boldsymbol{w}^*\|^2\|\Delta_t\|^2\right]^{1/2} \\ \le&\|\boldsymbol{w}_{t-1}-\boldsymbol{w}^*\|^2+\eta^2\mathbb{E}\left[\|\boldsymbol{h}_t\|^2\right]-\eta\nu\|\boldsymbol{w}_{t-1}-\boldsymbol{w}^*\|^2 \\ &+2\eta\mathbb{E}\left[\|\boldsymbol{w}_{t-1}-\boldsymbol{w}^*\|^2\|\Delta_t\|^2\right]^{1/2}\end{aligned}$$

where $S = \mathcal{B}(\boldsymbol{0},\lambda)$. The last inequality uses the ν-strongly convexity of $f(\boldsymbol{w})$ as follows

$$\begin{aligned}&(\boldsymbol{w}_t-\boldsymbol{w}^*)^\top\nabla f(\boldsymbol{w}_t) \\ \ge& f(\boldsymbol{w}_t)-f(\boldsymbol{w}^*)+\frac{\lambda}{2}\|\boldsymbol{w}_t-\boldsymbol{w}^*\|^2 \\ \ge&\frac{\lambda}{2}\|\boldsymbol{w}_t-\boldsymbol{w}^*\|^2\end{aligned}$$

$$
\begin{aligned}
\sqrt{\mathbb{E}\left[\|\boldsymbol{w}_t\|^2\right]} \leq & \sqrt{\mathbb{E}\left[\left\|\prod_S\left(\boldsymbol{w}_{t-1}-\eta \boldsymbol{h}_t\right)\right\|^2\right]} \leq \sqrt{\mathbb{E}\left[\left\|\boldsymbol{w}_{t-1}-\eta \boldsymbol{h}_t\right\|^2\right]} \\
= & \sqrt{\mathbb{E}\left[\left\|\boldsymbol{w}_{t-1}-\eta\left(\boldsymbol{w}_{t-1}+\lambda \ell'\left(\boldsymbol{w}_{t-1} ; x_t, y_t\right)-\widetilde{\boldsymbol{g}}_t+\widetilde{\boldsymbol{\mu}}+\Delta_t\right)\right\|^2\right]} \\
\leq & (1-\eta) \sqrt{\mathbb{E}\left[\left\|\boldsymbol{w}_{t-1}\right\|^2\right]}+\eta \lambda \sqrt{\mathbb{E}\left[\left\|\ell'\left(\boldsymbol{w}_{t-1} ; x_t, y_t\right)\right\|^2\right]}+\eta \sqrt{\mathbb{E}\left[\left\|\widetilde{\boldsymbol{\mu}}-\widetilde{\boldsymbol{g}}_t\right\|^2\right]} \\
& +\eta \sqrt{\mathbb{E}\left[\left\|\Delta_t\right\|^2\right]} \\
\leq & (1-\eta) \sqrt{\mathbb{E}\left[\left\|\boldsymbol{w}_{t-1}\right\|^2\right]}+\eta A+\eta \sqrt{\mathbb{E}\left\|\widetilde{\alpha}_t \phi\left(\boldsymbol{x}_t\right)-\frac{1}{m} \sum_{i=1}^m \widetilde{\alpha}_i \phi\left(\boldsymbol{x}_t\right)\right\|^2} \\
& +\eta \sqrt{\mathbb{E}\left[\left\|\Delta_t\right\|^2\right]} \\
\leq & (1-\eta) \sqrt{\mathbb{E}\left[\left\|\boldsymbol{w}_{t-1}\right\|^2\right]}+\eta A+\eta \sqrt{\mathbb{E}\left\|\widetilde{\alpha}_t \phi\left(\boldsymbol{x}_t\right)\right\|^2}+\eta \sqrt{\mathbb{E}\left[\left\|\Delta_t\right\|^2\right]} \\
\leq & (1-\eta) \sqrt{\mathbb{E}\left[\left\|\boldsymbol{w}_{t-1}\right\|^2\right]}+2 \eta A+\frac{3}{2} \eta A \delta_\phi \\
\leq & (1-\eta) P+2 \eta A+\frac{3}{2} \eta A \delta_\phi=P
\end{aligned}
$$

Figure 1: Proof of Lemma 6

The last inequality uses $f\left(\boldsymbol{w}_t\right)-f\left(\boldsymbol{w}^*\right) \geq 0$. Then by taking expectation again and substituting (4), we have

$$
\begin{aligned}
& \mathbb{E}\left[\left\|\boldsymbol{w}_t-\boldsymbol{w}^*\right\|^2\right] \\
\leq & (1-\eta \nu) \mathbb{E}\left[\left\|\boldsymbol{w}_{t-1}-\boldsymbol{w}^*\right\|^2\right]+4 \eta^2 P^2 \\
& \quad+2 \eta \mathbb{E}\left[\left\|\boldsymbol{w}_{t-1}-\boldsymbol{w}^*\right\|^2\left\|\Delta_t\right\|^2\right]^{1 / 2} \\
\leq & (1-\eta \nu) \mathbb{E}\left[\left\|\boldsymbol{w}_{t-1}-\boldsymbol{w}^*\right\|^2\right]+4 \eta^2 P^2 \\
& \quad+3 \eta A \delta_\phi \mathbb{E}\left[\left\|\boldsymbol{w}_{t-1}-\boldsymbol{w}^*\right\|^2\right]^{1 / 2}
\end{aligned}
$$

We prove by induction in t. Choosing $W=\frac{3 A \delta_\phi+\sqrt{9 A^2 \delta_\phi^2+16(1-\eta \nu) P^2}}{2 \nu}$, which is the solution of following equation

$$
(1-\eta \nu) W^2+4 \eta^2 P^2+3 \eta A \delta_\phi W=W^2
$$

Therefore, assuming that $\mathbb{E}\left[\left\|\boldsymbol{w}_{t-1}-\boldsymbol{w}^*\right\|^2\right] \leq W^2$, we obtain

$$
\mathbb{E}\left[\left\|\boldsymbol{w}_t-\boldsymbol{w}^*\right\|^2\right] \leq(1-\eta \nu) W^2+4 \eta^2 P^2+3 \eta A \delta_\phi W=W^2
$$

Therefore, we gain the conclusion $\mathbb{E}\left[\left\|\boldsymbol{w}_t-\boldsymbol{w}^*\right\|^2\right] \leq W^2$ for all t. □

According to the Lemma 5, 6, 7, we can obtain the convergence rate as follows, the proof of which is in the main paper.

Theorem 8. *Consider CSVRG in Algorithm 1 with option II and use it to solve SVMs and ODM. Assume that all $\psi_i(\boldsymbol{w})$ are convex and L-smooth, $f(\boldsymbol{w})$ is ν-strongly convex. Let $\boldsymbol{w}^* = argmin_{\boldsymbol{w}} f(\boldsymbol{w})$. Assume that T is sufficiently large so that*

$$
\rho=\frac{1}{\nu \eta(1-4 L \eta) T}+\frac{4 L \eta}{1-4 L \eta}<1
$$

then we have linear convergence in expectation for CSVRG:

$$
\mathbb{E}[f(\widetilde{\boldsymbol{w}}_s)-f(\boldsymbol{w}^*)] \leq \rho^s \mathbb{E}[f(\widetilde{\boldsymbol{w}}_0)-f(\boldsymbol{w}^*)]+\frac{1-\rho^s}{1-\rho} \Omega
$$

where Ω is a constant gap caused by coreset approximation, and $\Omega \to 0$ when the radius of coverage approaches 0.

References

Le, T.; Nguyen, T. D.; Nguyen, V.; and Phung, D. 2017. Approximation vector machines for large-scale online learning. *The Journal of Machine Learning Research* 18(1):3962–4016.

Shalev-Shwartz, S.; Singer, Y.; Srebro, N.; and Cotter, A. 2011. Pegasos: Primal estimated sub-gradient solver for svm. *Mathematical programming* 127(1):3–30.