Distributed Weighted Parameter Averaging for SVM Training on Big Data

Ayan Das
Dept. of Computer Science and Engineering
IIT Kharagpur, Kharagpur
W.B. - 721032, India
ayand@cse.iitkgp.ernet.in

Sourangshu Bhattacharya
Dept. of Computer Science and Engineering
IIT Kharagpur, Kharagpur
W.B. - 721032, India
sourangshu@cse.iitkgp.ernet.in

Abstract

Two popular approaches for distributed training of SVMs on big data are parameter averaging and ADMM. Parameter averaging is efficient but suffers from loss of accuracy with increase in number of partitions, while ADMM in the feature space is accurate but suffers from slow convergence. In this paper, we report a hybrid approach called weighted parameter averaging (WPA), which optimizes the regularized hinge loss with respect to weights on parameters. The problem is shown to be same as solving SVM in a projected space. We also demonstrate an $O(\frac{1}{N})$ stability bound on final hypothesis given by WPA, using novel proof techniques. Experimental results on a variety of toy and real world datasets show that our approach is significantly more accurate than parameter averaging for high number of partitions. It is also seen the proposed method enjoys much faster convergence compared to ADMM in features space.

1 Introduction

With the growing popularity of Big Data platforms [1] for various machine learning and data analytics applications [9, 12], distributed training of Support Vector Machines (SVMs)[4] on Big Data platforms have become increasingly important. Big data platforms such as Hadoop [1] provide simple programming abstraction (Map Reduce), scalability and fault tolerance at the cost of distributed iterative computation being slow and expensive [9]. Thus, there is a need for SVM training algorithms which are efficient both in terms of the number of iterations and volume of data communicated per iteration.

The problem of distributed training of support vector machines (SVM) [6] in particular, and distributed regularized loss minimization (RLM) in general [2, 9], has received a lot of attention in the recent times. Here, the training data is partitioned into M-nodes, each having L datapoints. Parameter averaging (PA), also called "mixture weights" [9] or "parallelized SGD" [12], suggests solving an appropriate RLM problem on data in each node, and use average of the resultant parameters. Hence, a single distributed iteration is needed. However, as shown in this paper, the accuracy of this approach reduces with increase in number of partitions. Another interesting result described in [9] is a bound of $O(\frac{1}{ML})$ on the stability of the final hypothesis, which results in a bound on deviation from optimizer of generalization error.

Another popular approach for distributed RLM is *alternating direction method of multipliers* (ADMM) [2, 6]. This approach tries to achieve consensus between parameters at different nodes while optimizing the objective function. It achieves optimal performance irrespective of the number of partitions. However, this approach needs many distributed iterations. Also, number of parameters to be communicated among machines per iteration is same as the dimension of the problem. This can be $\sim$ millions for some practical datasets, e.g. webspam [3].

In this paper, we propose a hybrid approach which uses weighted parameter averaging and proposes to learn the weights in a distributed manner from the data. We propose a novel SVM-like formulation for learning the weights of the weighted parameter averaging (WPA) model. The dual of WPA turns out to be same as SVM dual, with data projected in a lower dimensional space. We propose an ADMM [2] based distributed algorithm (DWPA), and an accelerated version (DWPAacc), for learning the weights.

Another contribution is a $O(\frac{1}{ML})$ bound on the stability of final hypothesis leading to a bound on deviation from optimizer of generalization error. This requires a novel proof technique as both the original parameters and the weights are solutions to optimization problems (section 2.4). Empirically, we show that that accuracy of parameter averaging degrades with increase in the number of partitions. Experimental results on real world datasets show that DWPA and DWPAacc achieve better accuracies than PA as the number of partitions increase, while requiring lower number of iterations and time per iteration compared to ADMM.

2 Distributed Weighted Parameter Averaging (DWPA)

In this section, we describe the distributed SVM training problem, the proposed solution approach and a distributed algorithm. We describe a bound on stability of the final hypothesis in section 2.4. Note that, we focus on the distributed SVM problem for simplicity. The techniques described here are applicable to other distributed regularized risk minimization problems.

2.1 Background

Given a training dataset $S = \{(\mathbf{x}_i, y_i) : i = 1, \cdots, ML, y_i \in \{-1, +1\}, \mathbf{x_i} \in \mathbf{R^d}\}$, the linear SVM problem [4] is given by:

$$\min_{\mathbf{w}} \lambda \|\mathbf{w}\|_2^2 + \frac{1}{m} \sum_{i=1}^{ML} loss(\mathbf{w}; (\mathbf{x}_i, y_i)), \tag{1}$$

where, λ is the regularization parameter and the hinge loss is defined as $loss(\mathbf{w}; (\mathbf{x}_i, y_i)) = max(0, 1 - y_i \mathbf{w}^T \mathbf{x}_i)$. The separating hyperplane is given by the equation $\mathbf{w}^T \mathbf{x} + b = 0$. Here we include the bias b within $\mathbf{w}$ by making the following transformation, $\mathbf{w} = \begin{bmatrix} \mathbf{w}^T, b \end{bmatrix}^T$ and $\mathbf{x}_i = \begin{bmatrix} \mathbf{x}_i^T, 1 \end{bmatrix}^T$.

The above SVM problem can be posed to be solved in a distributed manner, which is interesting when the volume of training data is too large to be effectively stored and processed on a single computer. Let the dataset be which is partitioned into M partitions ($\mathcal{S}_m, \ m = 1, \ldots, M$), each having L datapoints. Hence, $\mathcal{S} = \mathcal{S}_1 \cup, \ldots, \cup \mathcal{S}_M$, where $\mathcal{S}_m = \{(\mathbf{x}_{ml}, y_{ml})\}, l = 1, \ldots, L$. Under this setting, the SVM problem (Eqn 1), can be stated as:

$$\min_{\mathbf{w}_m, \mathbf{z}} \sum_{m=1}^{M} \sum_{l=1}^{L} loss(\mathbf{w}_m; (\mathbf{x}_{ml}, y_{ml})) + r(\mathbf{z}) \tag{2}$$

$$\text{s.t.} \mathbf{w}_m - \mathbf{z} = 0, m = 1, \cdots, M, \ l = 1, \ldots, L$$

where $loss()$ is as described above and $r(\mathbf{z}) = \lambda \|\mathbf{z}\|^2$. This problem is solved in [2] using ADMM (see section 2.3).

Another method for solving distributed RLM problems, called parameter averaging (PA), was proposed by Mann et. al. [9], in the context of *conditional maximum entropy* model. Let $\hat{\mathbf{w}}_m = \operatorname{argmin}_{\mathbf{w}} \frac{1}{L} \sum_{l=1}^{L} loss(\mathbf{w}; \mathbf{x}_{ml}, y_{ml}) + \lambda \|\mathbf{w}\|^2$, $m = 1, \ldots, M$ be the standard SVM solution obtained by training on partition $\mathcal{S}_m$. Mann et al. [9] suggest the approximate final parameter to be the arithmetic mean of the parameters learnt on individual partitions, ($\hat{\mathbf{w}}_m$). Hence:

$$\mathbf{w}_{PA} = \frac{1}{M} \sum_{m=1}^{M} \hat{\mathbf{w}}_m \tag{3}$$

Zinekevich et al. [12] have also suggested a similar approach where $\hat{\mathbf{w}}_m$'s are learnt using SGD. We tried out this approach for SVM. Note that assumptions regarding differentiability of loss function made in [2] can be relaxed in case of convex loss function with an appropriate definition of bregmann divergence using sub-gradients (see [11], section 2.4). The results (reported in section 3) show that the method fails to perform well as the number of partitions increase. This drawback of the above

mentioned approach motivated us to propose the weighted parameter averaging method described in the next section.

2.2 Weighted parameter averaging (WPA)

The parameter averaging method uses uniform weight of $\frac{1}{M}$ for each of the M components. One can conceive a more general setting where the final hypothesis is a weighted sum of the parameters obtained on each partition: $\mathbf{w} = \sum_{m=1}^{M} \beta_m \hat{\mathbf{w}}_{\mathbf{m}}$, where $\hat{\mathbf{w}}_{\mathbf{m}}$ are as defined above and $\beta_m \in \mathbb{R}, m = 1, \ldots, M$. Thus, $\boldsymbol{\beta} = [\beta_1, \cdots, \beta_M]^T = [\frac{1}{M}, \ldots, \frac{1}{M}]$ achieves the PA setting. Note that Mann et al. [9] proposed $\boldsymbol{\beta}$ to be in a simplex. However, no scheme was suggested for learning an appropriate $\boldsymbol{\beta}$.

Our aim is to find the optimal set of weights $\boldsymbol{\beta}$ which attains the lowest regularized loss. Let $\hat{\mathbf{W}} = [\hat{\mathbf{w}}_1, \cdots, \hat{\mathbf{w}}_M]$, so that $\mathbf{w} = \hat{\mathbf{W}}\boldsymbol{\beta}$. Substituting $\mathbf{w}$ in eqn. 1, the regularized loss minimization problem becomes:

$$\min_{\boldsymbol{\beta}, \xi} \lambda \|\hat{\mathbf{W}}\boldsymbol{\beta}\|^2 + \frac{1}{ML} \sum_{m=1}^{M} \sum_{i=1}^{l} \xi_{mi} \tag{4}$$

$$\textbf{subject to: } y_{mi}(\boldsymbol{\beta}^T \hat{\mathbf{W}}^T \mathbf{x}_{mi}) \geq 1 - \xi_{mi}, \quad \forall i, m$$
$$\xi_{mi} \geq 0, \quad \forall m = 1, \ldots, M, \ i = 1, \ldots, l$$

Note that, here the optimization is only w.r.t. $\boldsymbol{\beta}$ and $\xi_{m,i}$. $\hat{\mathbf{W}}$ is a pre-computed parameter. Next we can derive the dual formulation by writing the lagrangian and eliminating the primal variables. The Lagrangian is given by:

$$\mathcal{L}(\boldsymbol{\beta}, \xi_{mi}, \alpha_{mi}, \mu_{mi}) = \lambda \|\hat{\mathbf{W}}\boldsymbol{\beta}\|^2 + \frac{1}{ML} \sum_{m,i} \xi_{mi} + \sum_{m,i} \alpha_{mi}(y_{mi}(\boldsymbol{\beta}^T W^T \mathbf{x}_{mi}) - 1 + \xi_{mi}) - \sum_{m,i} \mu_{mi}\xi_{mi}$$

Differentiating the Lagrangian w.r.t. $\boldsymbol{\beta}$ and equating to zero, we get:

$$\boldsymbol{\beta} = \frac{1}{2\lambda}(\hat{\mathbf{W}}^T \hat{\mathbf{W}})^{-1}(\sum_{m,i} \alpha_{mi} y_{mi} \hat{\mathbf{W}}^T \mathbf{x}_{mi}) \tag{5}$$

Differentiating $\mathcal{L}$ w.r.t. ξ_{mi} and equating to zero, $\forall i \in 1, \cdots, L$ and $\forall m \in 1, \cdots, M$, implies $\frac{1}{ML} - \alpha_{mi} - \mu_{mi} = 0$. Since $\mu_{mi} \geq 0$ and $\alpha_{mi} \geq 0, 0 \leq \alpha_{mi} \leq \frac{1}{ML}$. Substituting the value of $\boldsymbol{\beta}$ in the Lagrangian $\mathcal{L}$, we get the dual problem:

$$\min_{\boldsymbol{\alpha}} \mathcal{L}(\boldsymbol{\alpha}) = \sum_{m,i} \alpha_{mi} - \frac{1}{4\lambda} \sum_{m,i} \sum_{m',j} \alpha_{mi}\alpha_{m'j} y_{mi} y_{m'j} (\mathbf{x}_{mi}^T \hat{\mathbf{W}}(\hat{\mathbf{W}}^T \hat{\mathbf{W}})^{-1} \hat{\mathbf{W}}^T \mathbf{x}_{m'j}) \tag{6}$$

$$\textbf{subject to: } 0 \leq \alpha_{mi} \leq \frac{1}{ML} \quad \forall i \in 1, \cdots, L, m \in 1, \cdots, M$$

Note that this is equivalent to solving SVM using the projected datapoint $(\mathcal{H}\mathbf{x}_{mi}, y_{mi})$, instead of $(\mathbf{x}_{mi}, y_{mi})$, where $\mathcal{H} = \hat{\mathbf{W}}(\hat{\mathbf{W}}^T \hat{\mathbf{W}})^{-1} \hat{\mathbf{W}}^T$, which is the projection on column space of $\hat{\mathbf{W}}$. Hence the performance of the method is expected to depend on size and orientation of the column space of $\hat{\mathbf{W}}$. Next, we describe distributed algorithms for learning $\boldsymbol{\beta}$.

2.3 Distributed algorithms for WPA using ADMM

In the distributed setting, we assume the presence of a central (master) computer which stores and updates the final hypothesis. The partitions of training set $\mathcal{S}_1, \ldots, \mathcal{S}_M$ are distributed to M (slave) computers, where the local optimizations are performed. The master needs to communicate to slaves and vice versa. However, no communication between slaves is necessary. Thus, the underlying networks has a star topology, which is also easily implemented using Big data platforms like Hadoop [1].

Let $\boldsymbol{\gamma}_m$, for $m = 1, \cdots, M$ be the weight values at the M different nodes and $\boldsymbol{\beta}$ be the value of the weights at the central server. The formulation given in eqn. 4 can be written as:

$$\min_{\gamma_m, \beta} \frac{1}{ML} \sum_{m=1}^{M} \sum_{l=1}^{L} loss(\hat{W}\boldsymbol{\gamma}_m; \mathbf{x}_{ml}, y_{ml}) + r(\boldsymbol{\beta}) \tag{7}$$

$$\text{s.t.} \quad \boldsymbol{\gamma}_m - \boldsymbol{\beta} = 0, \quad m = 1, \cdots, M,$$

where $r(\boldsymbol{\beta}) = \lambda \|\hat{\mathbf{W}}\boldsymbol{\beta}\|^2$. The augmented lagrangian for the above problem is: $L(\boldsymbol{\gamma}_m, \boldsymbol{\beta}, \boldsymbol{\lambda}) = \frac{1}{ML} \sum_{m=1}^{M} \sum_{l=1}^{L} loss(\hat{W}\boldsymbol{\gamma}_m; \mathbf{x}_{ml}, y_{ml}) + r(\boldsymbol{\beta}) + \sum_{i=1}^{M} \frac{\rho}{2} \|\boldsymbol{\gamma}_m - \boldsymbol{\beta}\|^2 + \sum_{i=1}^{M} \boldsymbol{\psi}_m^T(\boldsymbol{\gamma}_m - \boldsymbol{\beta})$, where

ψ_m is the lagrange multiplier vector corresponding to m^{th} constraint. Let $\mathbf{A}_m \in \mathbf{R}^{L \times d} = -\mathbf{diag}(\mathbf{y}_m)\mathbf{X}_m\hat{W}$. Using results from [2], the ADMM updates for solving the above problem can derived as:

$$\boldsymbol{\gamma}_m^{k+1} := \underset{\boldsymbol{\gamma}}{\operatorname{argmin}}(loss(\mathbf{A}_i\boldsymbol{\gamma}) + (\rho/2)\|\boldsymbol{\gamma}_m l - \boldsymbol{\beta}^k + \mathbf{u}_m^k\|_2^2) \tag{8}$$

$$\boldsymbol{\beta}^{k+1} := \underset{\boldsymbol{\beta}}{\operatorname{argmin}}(r(\boldsymbol{\beta}) + (M\rho/2)\|\boldsymbol{\beta} - \overline{\boldsymbol{\gamma}}^{k+1} - \overline{\mathbf{u}}^k\|_2^2) \tag{9}$$

$$\mathbf{u}_m^{k+1} = \mathbf{u}_m^k + \boldsymbol{\gamma}_m^{k+1} - \boldsymbol{\beta}^{k+1}. \tag{10}$$

where, $\mathbf{u}_m = \frac{1}{\rho}\psi_m, \overline{\boldsymbol{\gamma}} = \frac{1}{M}\sum_{m=1}^M \boldsymbol{\gamma}_m$ and $\overline{\mathbf{u}} = \frac{1}{M}\sum_{m=1}^M \mathbf{u}_m$ and the superscript k denotes the iteration counts. Algorithm 1 describes the full procedure.

Algorithm 1: Distributed Weighted Parameter Averaging (DWPA)

---

input : Partitioned datasets $\mathcal{S}_m$, SVM parameter learnt for each partition $\hat{\mathbf{w}}_m, \forall m = 1, \cdots, M$
output: Optimal weight vector $\boldsymbol{\beta}$

1 Initialize $\boldsymbol{\beta} = \mathbf{1}, \boldsymbol{\gamma}_m = \mathbf{1}, \mathbf{u}_m = \mathbf{1}, \forall m \in \{1, \cdots, M\}$;
2 **while** $k < T$ **do**
    /* Executed on slaves   */
3   **for** $m \leftarrow 1$ **to** M **do**
4         $\boldsymbol{\gamma}_m^k := \operatorname{argmin}_{\boldsymbol{\gamma}_m}(1^T(A_m\boldsymbol{\gamma}_m + 1)_+ + \rho/2\|\boldsymbol{\gamma}_m^{k-1} - \boldsymbol{\beta}^{k-1} - \mathbf{u}_m^{k-1}\|_2^2)$
5   **end**
    /* Executed on master   */
6     $\boldsymbol{\beta}^k := \frac{1}{2\lambda}(\hat{W}^T\hat{W} + M\rho I_m)^{-1}M\rho(\overline{\boldsymbol{\gamma}}^k + \overline{\mathbf{u}}^{k-1})$
7   **for** $m \leftarrow 1$ **to** M **do**
8         $\mathbf{u}_m^k = \mathbf{u}_m^{k-1} + \boldsymbol{\gamma}_m^k - \boldsymbol{\beta}^k$
9   **end**
10 **end**

A heuristic called *overrelaxation* [2] is ofter used for improving the convergence rate of ADMM. For overrelaxation, the updates for $\boldsymbol{\beta}^k$ (line 6 and $\mathbf{u}_m^k$ (line 8) are obtained by replacing $\overline{\boldsymbol{\gamma}}^k$ with $\hat{\boldsymbol{\gamma}}_m^k = \alpha \times \boldsymbol{\gamma}_m^k + (1 - \alpha) \times \boldsymbol{\beta}^{k-1}$, in algorithm 1. We implemented this heuristic for both DSVM and DWPA. We call them accelerated DSVM (DSVMacc) and accelarated DWPA (DWPAacc).

2.4 Bound on stability of WPA

In this section, we derive a bound of $\mathcal{O}(\frac{1}{ML})$ on stability of the final hypothesis returned by WPA algorithm described in eqn. 4. A similar bound was derived by Mann et al. [9] on the stability of PA. This leads to a $\mathcal{O}(\frac{1}{\sqrt{ML}})$ bound on deviation from optimizer of generalization error (see [9], theorem 2).

Let $S = \{S_1, \cdots, S_M\}$ and $S' = \{S'_1, \cdots, S'_M\}$ be two datasets with M partitions and L datapoints per partition, differing in only one datapoint. Hence, $S_m = \{z_{m1}, \cdots, z_{mL}\}$ and $S'_m = \{z'_{m1}, \cdots, z'_{mL}\}$, where $z_{ml} = (\mathbf{x}_{ml}, y_{ml})$ and $z'_{ml} = (\mathbf{x}'_{ml}, y'_{ml})$. Further, $S_1 = S'_1, \cdots, S_{M-1} = S'_{M-1}$, and S_M and S'_M differs at single point z_{ML} and z'_{ML}. Also, let $\|\mathbf{x}\| \leq R, \forall \mathbf{x}$. Moreover, let $\hat{W} = [\hat{\mathbf{w}}_{S_1}, \cdots, \hat{\mathbf{w}}_{S_M}]$ and $\hat{W}' = [\hat{\mathbf{w}}_{S'_1}, \cdots, \hat{\mathbf{w}}_{S'_M}]$ where, $\hat{\mathbf{w}}_{S_i} = \operatorname{argmin}_{\mathbf{w}} \lambda\|\mathbf{w}\|^2 + \frac{1}{L}\sum_{i \in S_i} max(0, 1 - y\mathbf{w}^T\mathbf{x})$. We also assume $\|\hat{\mathbf{W}}\|_F = \|\hat{\mathbf{W}}'\|_F = 1$. Hence, $\|\hat{\mathbf{w}}_m\|^2 = \|\hat{\mathbf{w}}'_m\|^2 = \frac{1}{M}, \forall m \in \{1, \cdots, M\}$.

We also define the following quantities:

$$\boldsymbol{\beta} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \lambda\|\hat{W}\boldsymbol{\beta}\|^2 + \frac{1}{ML}\sum_{i=1}^M \sum_{z \in S_i} max(0, 1 - y(\hat{W}\boldsymbol{\beta})^T\mathbf{x})$$

$$\boldsymbol{\beta}' = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \lambda\|\hat{W}'\boldsymbol{\beta}\|^2 + \frac{1}{ML}\sum_{i=1}^M \sum_{z' \in S'_i} max(0, 1 - y'(\hat{W}\boldsymbol{\beta})^T\mathbf{x}')$$

4

$$\tilde{\boldsymbol{\beta}} = \operatorname*{argmin}_{\boldsymbol{\beta}} \lambda \|\hat{W}'\boldsymbol{\beta}\|^2 + \frac{1}{ML} \sum_{i=1}^{M} \sum_{z \in S_i} max(0, 1 - y(\hat{W}'\boldsymbol{\beta})^T \mathbf{x})$$

Also, let $\boldsymbol{\theta} = \hat{W}\boldsymbol{\beta}$, $\boldsymbol{\theta}' = \hat{W}'\boldsymbol{\beta}'$ and $\tilde{\boldsymbol{\theta}} = \hat{W}'\tilde{\boldsymbol{\beta}}$.

We are interested in deriving a bound on $\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|$, which decompose as: $\|\boldsymbol{\theta} - \boldsymbol{\theta}'\| \leq \|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\| + \|\tilde{\boldsymbol{\theta}} - \boldsymbol{\theta}'\|$. Intuitively, the first term captures the change from $\hat{W}$ to $\hat{W}'$ and second term captures change in dataset. Lemma 2.2, shows that $\|\tilde{\boldsymbol{\theta}} - \boldsymbol{\theta}'\|$ is $\mathcal{O}(\frac{1}{ML})$. Showing bound on $\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|$ requires bounds on $\|\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}}\|$ (lemma 2.3) and $\|\hat{W} - \hat{W}'\|$ (lemma 2.1). The final proof is given in Theorem 2.4.

Lemma 2.1. $\|\hat{W} - \hat{W}'\| = \mathcal{O}(\frac{1}{ML})$

Proof (sketch): Since $\hat{w}_m = \hat{w}'_m, m = 1, \ldots, M - 1$, it suffices to show that $\|\hat{w}_M - \hat{w}'_M\| = \mathcal{O}(\frac{1}{ML})$. Since, $\hat{w}$ and $\hat{w}'$ are scaled as $\|\hat{w}_m\|^2 = \|\hat{w}'_m\|^2 = \frac{1}{M}$ it suffices to show that $M\|\hat{w} - \hat{w}'\| = \mathcal{O}(\frac{1}{L})$. This result is analogous to theorem 1 of [9]. This can be proved using a special definition of bregmann divergence shown in appendix A.

Lemma 2.2. $\|\tilde{\boldsymbol{\theta}} - \boldsymbol{\theta}'\| = \mathcal{O}(\frac{1}{ML})$

Proof (sketch): This can be shown using similar technique as proof in appendix B using $\| \cdot \|_K$, where, $K = \hat{W}'^T \hat{W}'$ instead of the Euclidean norm.

Lemma 2.3. $\|\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}}\| = \mathcal{O}(\frac{1}{ML})$

Proof: Let $F_W(\boldsymbol{\beta}) = G_W(\boldsymbol{\beta}) + L_W(\boldsymbol{\beta})$ and $F_{W'}(\tilde{\boldsymbol{\beta}}) = G_{W'}(\tilde{\boldsymbol{\beta}}) + L_{W'}(\tilde{\boldsymbol{\beta}})$. Using a similar definition of Bregmann divergence as in appendix B and its positivity:

$$B_{G_{\hat{W}}}(\tilde{\boldsymbol{\beta}}\|\boldsymbol{\beta}) + B_{G_{\hat{W}'}}(\boldsymbol{\beta}\|\tilde{\boldsymbol{\beta}}) \leq B_{F_{\hat{W}}}(\tilde{\boldsymbol{\beta}}\|\boldsymbol{\beta}) + B_{F_{\hat{W}'}}(\boldsymbol{\beta}\|\tilde{\boldsymbol{\beta}}) \tag{11}$$

The *left hand side* of the inequality 11, is given by;

$$B_{G_{\hat{W}}}(\tilde{\boldsymbol{\beta}}\|\boldsymbol{\beta}) + B_{G_{\hat{W}'}}(\boldsymbol{\beta}\|\tilde{\boldsymbol{\beta}}) = \lambda\|\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta}\|^T (\|\hat{W}^T\hat{W} + \hat{W}'^T\hat{W}'\|)\|\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta}\|$$

$$\leq \lambda\|\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}}\|^2_{K'}, \text{ where, } K' = \hat{W}^T\hat{W} + \hat{W}'^T\hat{W}'$$

Now we solve the *right hand side* of inequality 11,

$$B_{F_{\hat{W}}}(\tilde{\boldsymbol{\beta}}\|\boldsymbol{\beta}) + B_{F_{\hat{W}'}}(\boldsymbol{\beta}\|\tilde{\boldsymbol{\beta}}) = F_{\hat{W}}(\tilde{\boldsymbol{\beta}}) - F_{\hat{W}}(\boldsymbol{\beta}) + F_{\hat{W}'}(\boldsymbol{\beta}) + F_{\hat{W}'}(\tilde{\boldsymbol{\beta}})$$

$$= \lambda[\|\hat{W}\tilde{\boldsymbol{\beta}}\|^2 - \|\hat{W}\boldsymbol{\beta}\|^2 + \|\hat{W}'\boldsymbol{\beta}\|^2 - \|\hat{W}'\tilde{\boldsymbol{\beta}}\|^2] + \tag{12}$$

$$[L_{\hat{W}}(\boldsymbol{\beta}') - L_{\hat{W}}(\boldsymbol{\beta}) + L_{\hat{W}'}(\boldsymbol{\beta}) - L_{\hat{W}'}(\boldsymbol{\beta}')] = \mathcal{R} + \mathcal{L}$$

From 12, we have,

$$\mathcal{L} = L_{\hat{W}}(\boldsymbol{\beta}') - L_{\hat{W}}(\boldsymbol{\beta}) + L_{\hat{W}'}(\boldsymbol{\beta}) - L_{\hat{W}'}(\boldsymbol{\beta}')$$

$$= \frac{1}{ML} \sum_{m,l=1}^{M,L} [max(0, 1 - y_{ml}(\hat{W}\tilde{\boldsymbol{\beta}})^T\mathbf{x}_{ml}) - max(0, 1 - y_{ml}(\hat{W}\boldsymbol{\beta})^T\mathbf{x}_{ml}) +$$

$$max(0, 1 - y_{ml}(\hat{W}'\boldsymbol{\beta})^T\mathbf{x}_{ml}) - max(0, 1 - y_{ml}(\hat{W}'\tilde{\boldsymbol{\beta}})^T\mathbf{x}_{ml})]$$

$$\leq \frac{1}{ML} \sum_{m,l=1}^{M,L} max(0, y_{ml}((\hat{W}' - \hat{W})(\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}}))^T\mathbf{x}_{ml}) \leq \frac{1}{ML} \sum_{m,l=1}^{M,L} |y_{ml}((\hat{W}' - \hat{W})(\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}}))^T\mathbf{x}_{ml}|$$

$$\leq \frac{R}{ML}\|(\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}}))\|$$

Where, first two inequalities use: $max(a, 0) - max(b, 0) \leq max(a - b, 0)$, and the last step uses lemma 2.1.

For the part $\mathcal{R}$ of the 12 involving regularizers:

$$\mathcal{R} = \lambda[\|\hat{W}\tilde{\boldsymbol{\beta}}\|^2 - \|\hat{W}\boldsymbol{\beta}\|^2 + \|\hat{W}'\boldsymbol{\beta}\|^2 - \|\hat{W}'\tilde{\boldsymbol{\beta}}\|^2] = \lambda(\tilde{\boldsymbol{\beta}} + \boldsymbol{\beta})^T(\hat{W}^T\hat{W} - \hat{W}'^T\hat{W}')(\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta})$$

$$\leq \lambda\|\tilde{\boldsymbol{\beta}} + \boldsymbol{\beta}\|\|\hat{W}\|\|(\hat{W} - \hat{W}')\|\|\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta}\| + \|\tilde{\boldsymbol{\beta}} + \boldsymbol{\beta}\|\|\hat{W}'\|\|(\hat{W} - \hat{W}')\|\|\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta}\|$$

$$\leq \frac{4\lambda R}{ML}\|\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}}\|$$

5

where for the last step, we use the constant bound $\|\boldsymbol{\beta}\| = \lambda R$ on $\boldsymbol{\beta}$ obtained from its expression of in 5. Therefore, from the *left hand side* and *right hand side* of the inequality 11, we have:

$$\lambda\|\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}}\|_2^2 \leq \frac{\lambda}{\sigma_{min}}\|\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}}\|_{K'}^2 \leq \frac{4\lambda R}{ML}\|\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}}\| + \frac{R}{ML}\|(\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}}))\| \tag{13}$$

where, σ_{min} is smallest eigenvalue of K'. This implies;$\|\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}}\|$ is $\mathcal{O}(\frac{1}{ML})$.

$\square$

Theorem 2.4. $\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|$ *is of the order of* $\mathcal{O}(\frac{1}{ML})$.

Proof: The steps involved in the proof are as follows;

$$\|\boldsymbol{\theta} - \boldsymbol{\theta}'\| \leq \|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\| + \|\tilde{\boldsymbol{\theta}} - \boldsymbol{\theta}'\| \tag{14}$$

From lemma 2.2, $\|\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|$ is $\mathcal{O}(\frac{1}{ML})$.

$$\|\tilde{\boldsymbol{\theta}} - \boldsymbol{\theta}'\| \leq \frac{1}{2}(\|(\hat{W} - \hat{W}')(\boldsymbol{\beta} + \tilde{\boldsymbol{\beta}})\| + \|(\hat{W} + \hat{W}')(\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}})\|)$$

$$\leq \frac{1}{2}((\|(\hat{W} - \hat{W}')\|)(\|\boldsymbol{\beta} + \tilde{\boldsymbol{\beta}}\|) + (\|\hat{W} + \hat{W}'\|)(\|\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}}\|)) \tag{15}$$

We have already, shown that, we have a constant bound on $\|\boldsymbol{\beta} + \tilde{\boldsymbol{\beta}}\|$ and $\|\hat{W} + \hat{W}'\|_F$, since, norms of $\boldsymbol{\beta}$, $\tilde{\boldsymbol{\beta}}$, $\hat{W}$ and $\hat{W}'$ are bounded. Also both $\|\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}}\|$ and $\|(\hat{W} - \hat{W}')\|$ are $\mathcal{O}(\frac{1}{ML})$.

Hence, from 15, we have the required result.

$\square$

3 Experimental Results

In this section, we experimentally analyze and compare the methods proposed here, *distributed weighted parameter averaging* (DWPA) and accelerated DWPA (DWPAacc) described in section 2.3, with *parameter averaging* (PA) [9], Distributed SVM (DSVM) using ADMM, and accelerated DSVM (DSVMacc) [2]. For our experimentation, we have implemented all the above mentioned algorithms in Matlab [10]. We have used the *liblinear* library [5] to obtain the SVM parameters corresponding each partition. Optimization problems which arise as subproblems in ADMM has been solved using CVX [7], [8].

We used both toy datasets (section 3.1) and real world datasets (described in table 1) for our experiments. Real world datasets were obtained from LIBSVM website [3]. Samples for real world datasets were selected randomly. The datasets were selected to have various ranges of feature count and sparsity. Section 3.1 describes a specially construc

Table 1: Training and test dataset size

Dataset Name	Number of training instances	Number of test instances	Number of features	Domain
epsilon	6000	1000	2000	mixed
gisette	6000	1000	5000	mixed
real-sim	3000	5000	20958	text

3.1 Results on toy dataset

The main purpose of the toy dataset was to visually observe the effect of change in the number of partitions on the final hypothesis for various algorithms. Datapoints are generated from a 2 dimensional mixture of gaussians. In figure 1, the red and blue dots indicate the datapoints from two different classes. The upper red blob contains only 20% of red points. Hence as the number of partitions increase, many partitions will not have any data points from upper blob. For these partitions, the separating hyperplane passes throught the upper red blob. These cause the average hyperplane to pass through upper red blob, thus decreasing the accuracy. This effect is visible in the left plot of figure 1. This effect is mitigated in weighted parameter averaging as the weights learnt for the hyperplanes passing through upper red blob are lesser. This is shown in middle plot of figure 1. Finally, the right plot of figure 1 shows the resultant decrease in accuracy for PA with increase in number of partitions.
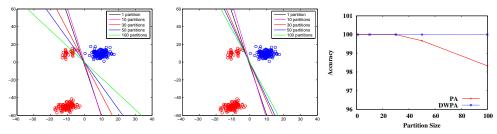
Figure 1: Comparison of performance of *PA*(left) and *DWPA*(middle) on toy dataset. The graph on the *right* shows the change in accuracy of PA and DWPA with change in partition size for the toy dataset

Bias of a learning algorithm is $E[|w - w*|]$, where w and $w*$ are minimizers of regularized loss and generalization error, and the expectation is over all samples of fixed size say N. An criticism against PA is the lack of bound on bias [9]. In table 2, we compare bias of PA, WPA and SVM as a function of N. Data samples were generated from the same toy distribution as above. $w*$ was computed by training on a large sample size and ensuring that training set error and test set error are very close. w was computed 100 times by randomly sampling from the distribution. The average of $|w - w*|$ is reported in table 2. We observe that bias of PA is indeed much higher than SVM or WPA.

Table 2: Variation of mean bias with increase in dataset size for PA, DWPA and DSVM

Sample size	Mean bias(PA)	Mean bias(DWPA)	Mean bias(DSVM)
3000	0.868332	0.260716	0.307931
6000	0.807217	0.063649	0.168727

3.2 Comparison of Accuracies

In this section, we compare accuracies obtained by various algorithms on real world datasets, with increase in number of paritions. Figure 2 reports test set accuracies for PA, WPA and SVM on three real world datasets with varying size of partitions. It is clear that performance of PA degrades dramatically as the number of parition increases. Thus, the effect demonstrated in section 3.1 is also observed on real world datasets.

We also observe that performance of WPA improves with increase in number of paritions. This is due to fact that dimension of space on which x_{ml}'s are projected using $\mathcal{H}$ (section 2.2) increases, thus reducing the information loss caused by projection. Finally, as expected WPA performs slightly worse than SVM.
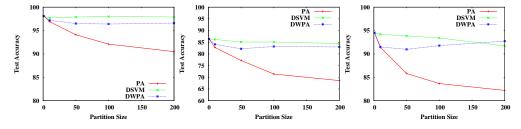

Figure 2: Variation of accuracy with number of partitions for *gisette(left), epsilon(middle)* and *real-sim(right)* for partition size 1, 10, 50, 100 and 200. The results were recorded upto 500 iterations

3.3 Convergence Analysis and time comparison

In this section, we compare the convergence properties of DSVM, DSVMacc, DWPA, and DW-PAacc. Here we report results on real-sim due to lack of space. Results on other real world datasets are provided in appendix C. Top row of figure 3 shows variation of primal residual (disagreement between parameters on various partitions) with iterations. It is clear that DWPA and DWPAacc

7

show much lesser disagreement compared to DSVM and DSVMacc, thus showing faster convergence. Bottom row fo figure 3 shows variation of test set accuracy with iterations. The same behaviour is apparent here, with testset accuracy of DWPA and DWPAacc converging much faster than DSVM and DSVMacc. One of the reasons is also that DWPA has an obvious good starting point of $\boldsymbol{\beta} = [\frac{1}{M}, \ldots, \frac{1}{M}]$ corresponding to PA.
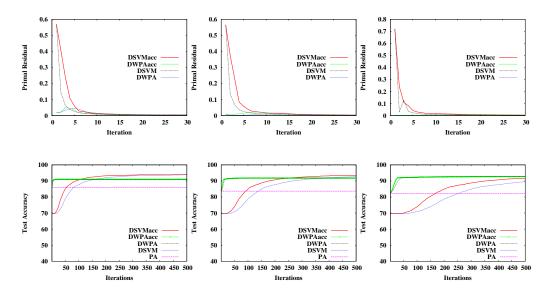

Figure 3: Convergence of primal residual (top) and test accuracy (bottom) for **real-sim**

Table 3: Average time per iteration(in seconds)

Number of partitions	DWPA			DSVM		
	epsilon	real-sim	gisette	epsilon	real-sim	gisette
10	27.4313	13.9004	31.2253	624.0198	622.6011	1653.0
50	23.1451	23.1181	37.3698	125.0944	125.0944	525.7135
100	37.3016	47.4931	65.1963	116.8604	116.8604	440.6123

Table 3 reports the average time taken by DWPA and DSVM for completing one iteration as a function of number of paritions. It is clear that DWPA takes much lesser time due to much smaller number of variables in the local optimization problem (Feature dimensions for DSVM, number of paritions for DWPA). There is slight increase in time per iteration with increase in number of paritions due to increase in number of variables.

4 Conclusion

We propose a novel approach for training SVM in a distributed manner by learning an optimal set of weights for combining the SVM parameters independently learnt on partitions of the entire dataset. Experimental results show that our method is much more accurate than *parameter averaging* and is much faster than training SVM in feature space. Moreover, our method reaches an accuracy close to that of SVM trained in feature space in a much shorter time. We propose a novel proof to show that the stability final SVM parameter learnt using DWPA is $\mathcal{O}(\frac{1}{ML})$. Also, our method requires much less network band-width as compared to DSVM when the number of features for a given dataset is very large as compared to the number of partitions, which is the usual scenerio for Big Data.

8

References

[1] Apache Hadoop. http://hadoop.apache.org, accessed on 06/06/2014.

[2] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, January 2011.

[3] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[4] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[5] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

[6] Pedro A. Forero, Alfonso Cano, and Georgios B. Giannakis. Consensus-based distributed support vector machines. *J. Mach. Learn. Res.*, 11:1663–1707, August 2010.

[7] Michael Grant and Stephen Boyd. Graph implementations for nonsmooth convex programs. In V. Blondel, S. Boyd, and H. Kimura, editors, *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pages 95–110. Springer-Verlag Limited, 2008. http://stanford.edu/~boyd/graph_dcp.html.

[8] Michael Grant and Stephen Boyd. CVX: Matlab software for disciplined convex programming, version 2.0 beta. http://cvxr.com/cvx, September 2013.

[9] Gideon Mann, Ryan McDonald, Mehryar Mohri, Nathan Silberman, and Dan Walker. Efficient large-scale distributed training of conditional maximum entropy models. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 1231–1239. 2009.

[10] MATLAB. *version 7.10.0 (R2010a)*. The MathWorks Inc., Natick, Massachusetts, 2010.

[11] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. The MIT Press, 2012.

[12] Martin Zinkevich, Markus Weimer, Alexander J. Smola, and Lihong Li. Parallelized stochastic gradient descent. In *NIPS*, pages 2595–2603, 2010.

Appendix A

Theorem 4.1. *For any two arbitrary training samples of size L differing by one sample point, the stability bound that holds for the parameter vectors returned by support vector machine is:*

$$\|\Delta \mathbf{w}\| \text{ is } \mathcal{O}(\frac{1}{ML}). \tag{16}$$

Proof. Suppose we have two training datasets $S = (z_1, \cdots, z_{L-1}, z_L)$ and $S' = (z_1, \cdots, z_{L-1}, z'_L)$, where $\mathbf{z} = (\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}$, such that $\mathcal{X} \subset \mathbf{R}^d$ and $\mathcal{Y} = \{-1, +1\}$. The two sets differ at a single data point: $\mathbf{z}_L = (\mathbf{x}_i, y_i)$ and $\mathbf{z}'_L = (\mathbf{x}'_i, y'_i)$. Let B_F be the Bregman divergence associated with a convex and non-diffentiable function F defined for all x, y by; $B_F(x, y) = F(x) - F(y) - <g_y, (x-y)>$, where $g \in \partial F_y$ and ∂F_y is the set of subdifferentials of F at y. Since, the minima is achieved at a point y if $0 \in \partial F_y$. We define g as follows;

$$g = \begin{cases} 0 & \text{if } 0 \in \partial F_y \\ h & \text{subject to,} \quad h \in \partial F_y \end{cases} \tag{17}$$

Let $L_s : \mathbf{x} \to \sum_{i=1}^{L} H_{z_i}(\mathbf{x})$, where, $H_{z_i}(\mathbf{x}) = max(0, 1 - y\mathbf{w}^T\mathbf{x})$ denote the loss function and $G : \mathbf{x} \to \lambda\|\mathbf{x}\|^2$ denote the regularizer corresponding to the SVM problem. Clearly, the function, $F_S = G + L_S$, is the objective function for SVM. L_S is convex and non-differentiable while G is convex and differentiable. Since, Bregman divergence is non-negative ($B_F \geq 0$),

$$B_{F_S} = B_G + B_{L_S} \geq B_G \tag{18}$$

$$B_{F_{S'}} \geq B_{G'}. \tag{19}$$

9

Thus,
$$B_G(\mathbf{w}'\|\mathbf{w}) + B_G(\mathbf{w}\|\mathbf{w}') \leq B_{F_S}(\mathbf{w}'\|\mathbf{w}) + B_{F_{S'}}(\mathbf{w}\|\mathbf{w}'). \tag{20}$$

If $\mathbf{w}$ and $\mathbf{w}'$ are minimizers for of B_{F_S} and $B_{F_{S'}}$, then, $g_S(\mathbf{w}) = g_{S'}(\mathbf{w}') = 0$ and
$$
\begin{aligned}
B_{F_S}(\mathbf{w}'\|\mathbf{w}) + B_{F_{S'}}(\mathbf{w}\|\mathbf{w}') &= F_S(\mathbf{w}') - F_S(\mathbf{w}) + F_{S'}(\mathbf{w}) - F_{S'}(\mathbf{w}') \\
&= \frac{1}{L}\left[H_{z_L}(\mathbf{w}') - H_{z_L}(\mathbf{w}) + H_{z'_L}(\mathbf{w}) - H_{z'_L}(\mathbf{w}')\right] \\
&\leq -\frac{1}{L}\left[g_{z_L}\cdot(\mathbf{w}')(\mathbf{w}-\mathbf{w}') + g_{z'_L}\cdot(\mathbf{w})(\mathbf{w}'-\mathbf{w})\right] \\
&= -\frac{1}{L}\left[g_{z'_L}(\mathbf{w}) - g_{z_L}(\mathbf{w}')\right]\cdot(\mathbf{w}'-\mathbf{w}) \\
&= -\frac{1}{L}\left[g_{z'_L}(\mathbf{w}) - g_{z_L}(\mathbf{w}')\right]\cdot(\Delta\mathbf{w}) \tag{21}
\end{aligned}
$$

From definition,
$$B_G((w')\|\mathbf{w}) + B_G((w)\|\mathbf{w}') = \lambda\|\Delta\mathbf{w}\|^2. \tag{22}$$

Hence, from derivation 21 and equation 22 and Cauchy-Schwarz inequality we have,
$$\lambda\|\mathbf{w}\| \leq \frac{1}{L}\|g_{z'_L}(\mathbf{w}) - g_{z_L}(\mathbf{w}')\| \leq \frac{1}{L}\left[\|g_{z'_L}(\mathbf{w})\| + \|g_{z_L}(\mathbf{w}')\|\right] \tag{23}$$

By definition, $H_{z_i}(\mathbf{w}) = max(0, 1 - y_i\mathbf{w}^T\mathbf{x_i})$ and $g_{z_i}(\mathbf{w}) \in \partial H_{z_i}(\mathbf{w})$. Therefore,
$$
\begin{aligned}
\partial H_{z_i}(w) &\leq \|y_i\mathbf{x_i}\| \\
\Rightarrow \partial H_{z_i}(w) &\leq \|\mathbf{x_i}\| \\
\Rightarrow \|g_{z_i}(\mathbf{w})\| &\leq \|\mathbf{x_i}\|.
\end{aligned}
$$

If we assume that the feature vectors are bounded i.e., there exists a positive integer $R > 0$ such that for all training instances $(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}$, $\|\mathbf{x}\| \leq R$, then we may state that,
$$\|\mathbf{w}\| \leq \frac{R}{\lambda L} \tag{24}$$

Since, $\mathbf{w}$ is normalized and scaled by $\frac{1}{M}$. So, the bound on $\|\Delta\mathbf{w}\|$ in our case, is $\mathcal{O}(\frac{1}{ML})$.

Appendix B

Theorem 4.2. $\|\boldsymbol{\theta}' - \tilde{\boldsymbol{\theta}}\|$ *is of the order of* $\mathcal{O}(\frac{1}{ML})$.

Proof: From definitions we have;
$$
\begin{aligned}
\|\boldsymbol{\theta}' - \tilde{\boldsymbol{\theta}}\|^2 &= \|\hat{W}'\boldsymbol{\beta}' - \hat{W}'\tilde{\boldsymbol{\beta}}\|^2 \\
&\leq \|\boldsymbol{\beta}' - \tilde{\boldsymbol{\beta}}\|_K^2, \text{ where },K = \hat{W}'^T\hat{W}' \tag{25}
\end{aligned}
$$

Since, we have a lower bound on $\|\boldsymbol{\beta}' - \tilde{\boldsymbol{\beta}}\|^2 \leq \frac{1}{\sigma_{min}}\|\boldsymbol{\beta}' - \tilde{\boldsymbol{\beta}}\|_K^2$, where σ_{min} is the minimum eigenvalue of K.

Hence, we need to prove an upper bound on $\|\boldsymbol{\beta}' - \tilde{\boldsymbol{\beta}}\|$.

From the reasoning of theorem 2.1, we have;
$$B_G(\boldsymbol{\beta}'\|\tilde{\boldsymbol{\beta}}) + B_G(\tilde{\boldsymbol{\beta}}\|\boldsymbol{\beta}') \leq B_{F_S}(\boldsymbol{\beta}'\|\tilde{\boldsymbol{\beta}}) + B_{F'_S}(\tilde{\boldsymbol{\beta}}\|\boldsymbol{\beta}') \tag{26}$$

From the *left hand side* of the equation we have;
$$B_G(\boldsymbol{\beta}'\|\tilde{\boldsymbol{\beta}}) + B_G(\tilde{\boldsymbol{\beta}}\|\boldsymbol{\beta}') = (\tilde{\boldsymbol{\beta}} + \boldsymbol{\beta}')^T(\hat{W}'^T\hat{W}')(\boldsymbol{\beta}' - \tilde{\boldsymbol{\beta}}) \tag{27}$$

From the *right hand side* of the equation, using the similar reasoning as that used for $\|\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}}\|$, we have;

$$B_{F_S}(\boldsymbol{\beta}'\|\tilde{\boldsymbol{\beta}}) + B_{F'_S}(\tilde{\boldsymbol{\beta}}\|\boldsymbol{\beta}') = L_S(\boldsymbol{\beta}') - L_S(\tilde{\boldsymbol{\beta}}) + L_{S'}(\tilde{\boldsymbol{\beta}}) - L_{S'}(\boldsymbol{\beta})$$

$$\leq \frac{1}{ML}[max(0, (\hat{W}'\boldsymbol{\beta}')^T(y_{z'_m}\boldsymbol{x}_{z'_m} - y_{z_m}\boldsymbol{x}_{z_m}) - (\hat{W}'\tilde{\boldsymbol{\beta}})^T(y_{z'_m}\boldsymbol{x}_{z'_m} - y_{z_m}\boldsymbol{x}_{z_m})]$$

$$\leq \frac{1}{ML}|\hat{W}'(\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta}')(y_{z'_m}\boldsymbol{x}_{z'_m} - y_{z_m}\boldsymbol{x}_{z_m})|$$

$$\leq \frac{2R}{ML}\|\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta}'\| \tag{28}$$

Equating, *left hand side* and *right hand side* of the equation, we get; $\|\boldsymbol{\beta}' - \tilde{\boldsymbol{\beta}}\|$ is $\mathcal{O}(\frac{1}{ML})$, and hence, from 25, we get, $\|\boldsymbol{\theta}' - \tilde{\boldsymbol{\theta}}\|$ is $\mathcal{O}(\frac{1}{ML})$.
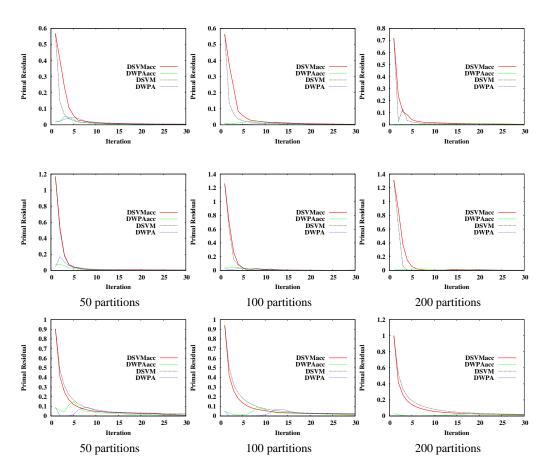
Appendix C

Figure 4: Convergence of primal residual for **real-sim**(top), **epsilon**(middle) and **gisette**(bottom)
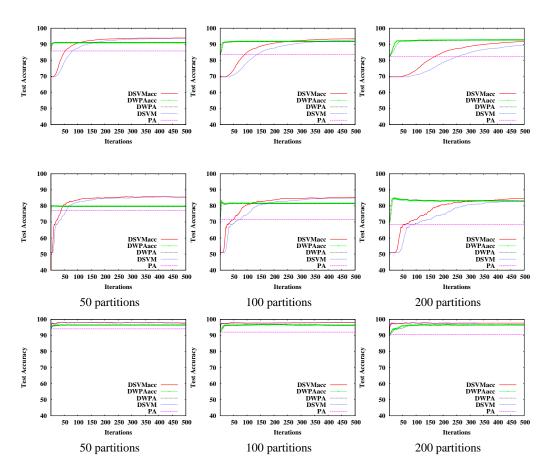
Figure 5: Testset accuracies for **real-sim**(top), **epsilon**(middle) and **gisette**(below)