PERFORMANCE LIMITS OF SINGLE-AGENT AND MULTI-AGENT SUB-GRADIENT STOCHASTIC LEARNING

Bicheng Ying and Ali H. Sayed

Department of Electrical Engineering
University of California, Los Angeles

ABSTRACT

This work examines the performance of stochastic sub-gradient learning strategies, for both cases of stand-alone and networked agents, under weaker conditions than usually considered in the literature. It is shown that these conditions are automatically satisfied by several important cases of interest, including support-vector machines and sparsity-inducing learning solutions. The analysis establishes that sub-gradient strategies can attain exponential convergence rates, as opposed to sub-linear rates, and that they can approach the optimal solution within $O(\mu)$, for sufficiently small step-sizes, μ. A realizable exponential-weighting procedure is proposed to smooth the intermediate iterates and to guarantee these desirable performance properties.

Index Terms— sotochastic sub-gradient method, affine-Lipschitz, exponential rate, diffusion strategy, SVM, LASSO, gradient noise.

1. INTRODUCTION AND RELATED WORK

The minimization of *non-differentiable* convex cost functions is a critical step in the solution of many important design problems [1–3], including the design of sparse-aware (LASSO) solutions [4, 5], support-vector machine (SVM) learners [6–10], or total-variation based image denoising solutions [11, 12]. The sub-gradient technique is a popular choice for minimizing such non-differentiable costs; it is closely related to the traditional gradient-descent method where the actual gradient vector is replaced by a sub-gradient at points of non-differentiability. It is one of the simplest methods in current practice but is known to suffer from slow convergence. In particular, it is shown in [3] that, for convex cost functions, the optimal convergence rate that can be delivered by sub-gradient methods in *deterministic* optimization problems cannot be faster than the $O(1/\sqrt{i})$, where i is the iteration index.

However, the results in subsequent sections will show that when used in the context of *stochastic* optimization, sub-gradient descent algorithms turn out to have superior performance than suggested by traditional analyses in the deterministic context. In particular, under constant step-size adaptation, these algorithms will be shown to converge at the faster exponential rate of $O(\alpha^i)$ for some $\alpha \in (0, 1)$ when the cost function is strongly-convex. This rate is much faster than the $O(1/i)$ rate that would be observed under a diminishing step-size implementation for strongly-convex costs. We will clarify these favorable properties for both cases of stand-alone agents and networked agents [13–16].

There are at least two main reasons that motivate a closer examination of the limits of performance of sub-gradient learning algorithms. First, the explosive interest in large-scale and big data scenarios favors the use of simple and computer-efficient algorithmic structures, of which the sub-gradient technique is a formidable example. Second, it is becoming increasingly evident that more sophisticated optimization iterations do not necessarily ensure improved performance when dealing with complex models and data structures [2, 17–19]. Motivated by these consideration, in our analysis of stochastic sub-gradient descent algorithms, we diverge in a noticeable way from conditions that are commonly used in the literature. First, we introduce weaker assumptions than usually adopted in prior works and, more importantly, we show that our assumptions are automatically satisfied for important cases of interest (such as SVM, LASSO, Total Variation). In contrast, these same problem formulations do not satisfy the traditional assumptions used in the literature and, hence, conclusions derived based on these earlier studies are not directly applicable to SVM or LASSO problems. For example, it is common in the literature to assume that the cost function has a bounded gradient [2, 16, 20–22]; this condition is unreasonable and is not satisfied even by quadratic costs whose gradient vectors are affine in their parameter. The condition is also in direct conflict with strongly-convex costs. By relaxing the conditions, the conclusions in our work become stronger and applicable to a broader class of algorithms and scenarios.

A second aspect of our study is that we focus on the use of *constant* step-sizes in order to enable continuous adaptation and learning. Since the step-size is assumed to remain constant, the effect of gradient noise is always present and does not die out, as would occur if we were using instead a diminishing step-size, say, of the form $\mu(i) = \tau/i$ [7, 16, 21, 23]. The challenge in analyzing the performance under constant-rate adaptation is to show that the algorithm is able to counter the effect of gradient noise and ensure convergence of the iterates at exponential rate to within $O(\mu)$ of the desired optimal solution.

A third aspect of our contribution is that it is known that sub-gradient methods are not *descent* methods. For this reason, it is customary to employ pocket variables (i.e., the best iterate) [1, 3, 24, 25] or arithmetic averages [7] to smooth out the output. However, the pocket method is not practical in the *stochastic* setting, and the use of arithmetic averages slows down convergence. Our analysis will suggest an alternative weighted averaging scheme that does not degrade convergence while providing the desired smoothing effect in an efficient manner.

This work was supported in part by NSF grants CCF-1011918, CIF-1524250, and DARPA project N66001–14–2–4029. Email: {ybc,sayed}@ucla.edu

978-1-4799-9988-0/16/$31.00 ©2016 IEEE

2. PROBLEM FORMULATION: SINGLE AGENT CASE

We consider the problem of minimizing a risk function, $J(w) : \mathbb{R}^M \to \mathbb{R}$, which is assumed to be expressed as the expected value of some loss function, $Q(w;\boldsymbol{x})$, namely,

$$w^\star \;\triangleq\; \arg\min_{w} J(w) \;\triangleq\; \arg\min_{w} \mathbb{E}_{\boldsymbol{x}}\, Q(w;\boldsymbol{x}) \tag{1}$$

where $w^\star$ denotes the minimizer. We first denote the sub-gradient of $J(w)$ at any arbitrary point w_0 by $g(w_0)$, and defined it as any vector $g \in \mathbb{R}^M$ that satisfies:

$$J(w) \;\geq\; J(w_0) + g^{\mathsf{T}}(w_0)(w - w_0), \;\; \forall w \tag{2}$$

In the context of adaptation and learning, we do not know the exact form of $J(w)$ because the distribution of the data is not known to enable computation of $\mathbb{E}_{\boldsymbol{x}} Q(w;\boldsymbol{x})$. As such, true sub-gradient vectors for $J(w)$ cannot be determined and they will need to be replaced by stochastic approximations evaluated from streaming data. We employ the following stochastic iteration [1,3,24,25]:

$$\boldsymbol{w}_i = \boldsymbol{w}_{i-1} - \mu\, \widehat{g}(\boldsymbol{w}_{i-1}) \tag{3}$$

where the successive iterates, $\{\boldsymbol{w}_i\}$, are now random variables (denoted in boldface) and $\widehat{g}(\cdot)$ represents an approximate sub-gradient vector at location $\boldsymbol{w}_{i-1}$ estimated from data available at time i. The difference between an actual sub-gradient vector and its approximation is referred to as *gradient noise* and is denoted by

$$\boldsymbol{s}_i(\boldsymbol{w}_{i-1}) \;\triangleq\; \widehat{g}(\boldsymbol{w}_{i-1}) - g(\boldsymbol{w}_{i-1}) \tag{4}$$

2.1. Modeling Conditions

In order to examine the performance of the stochastic sub-gradient implementation (3) for single-agent adaptation and learning, and later for multi-agent networks, it is necessary to introduce some assumptions. The first condition essentially requires that the construction of the approximate sub-gradient vector should not introduce bias and that its error variance should decrease as the quality of the iterate approaches the optimal solution. Both of these conditions are sensible and can be shown to be satisfied by, for example, SVM and LASSO constructions.

Assumption 1 (CONDITIONS ON GRADIENT NOISE) *The first and second-order conditional moments of the gradient noise process satisfy the following conditions:*

$$\mathbb{E}\left[\, \boldsymbol{s}_i(\boldsymbol{w}_{i-1}) \,|\, \boldsymbol{\mathcal{F}}_{i-1} \right] \;=\; 0 \tag{5}$$

$$\mathbb{E}\left[\, \|\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\|^2 \,|\, \boldsymbol{\mathcal{F}}_{i-1} \right] \;\leq\; \beta^2 \|w^\star - \boldsymbol{w}_{i-1}\|^2 + \sigma^2 \tag{6}$$

for some constants $\beta^2 \geq 0$ and $\sigma^2 \geq 0$, and where $\boldsymbol{\mathcal{F}}_{i-1}$ denotes the filtration corresponding to all past iterates (essentially, the conditioning in (5)–(6) is relative to the previous iterates). ■

The second condition ensures that $w^\star$ is unique so that the optimization problem is well-defined, and the third condition is more relaxed than what is traditionally imposed in the literature.

Assumption 2 (STRONGLY-CONVEX RISK FUNCTION) *The risk function is assumed to be $\eta-$strongly-convex, i.e.,*

$$\begin{aligned} J(\theta w_1 + (1-\theta) w_2) \;\leq\; & \theta J(w_1) + (1-\theta) J(w_2) \\ & - \frac{\eta}{2}\theta(1-\theta)\|w_1 - w_2\|^2 \end{aligned} \tag{7}$$

for any $\theta \in [0,1]$, w_1, and w_2, and where $\eta > 0$ ■

Assumption 3 (SUB-GRADIENT IS AFFINE-LIPSCHITZ) *It is assumed that the sub-gradient of the risk function, $J(w)$, is affine Lipschitz, i.e. there exist constants $c \geq 0$ and $d \geq 0$ such that*

$$\|g(w_1) - g(w_2)\| \;\leq\; c\|w_1 - w_2\| + d, \quad \forall w_1, w_2 \tag{8}$$

and for any choice $g(\cdot) \in \partial J(\cdot)$, where $\partial J(w)$ represent sub-differentials, i.e., the set of all valid sub-gradients at w. ■

Assumption 2 is rare in works on sub-gradient optimization because it is customary for these works to focus on studying *piece-wise* linear risks; these are important examples of non-smooth functions but they do not satisfy the strong-convexity condition. In our case, strong-convexity is not a restriction because in the context of adaptation and learning, it is common for the risk functions to include a regularization term, which helps ensure strong-convexity.

More critically, though, it is customary in the literature to use in place of Assumption 3 a more restrictive condition that requires the risk function itself (rather than its sub-gradient) to be Lipschitz.This condition is equivalent to requiring the sub-gradient to be bounded [1,16,20,22], i.e.,

$$\|g(w)\| \leq d_1, \quad \forall w, g \in \partial J(w) \tag{9}$$

Such a requirement does not even hold for quadratic risk functions, $J(w)$, whose gradient vectors are affine in w and, therefore, cannot be bounded. Even more, it can be easily seen that requirement (9) is conflicted with the strong-convexity assumption. One way to circumvent this problem is to restrict the domain of $J(w)$ to some bounded convex set, say, $w \in \mathcal{W}$, and then employ a projection-based sub-gradient method. However, this approach has at least three drawbacks. First, the unconstrained problem is transformed into a more demanding constrained problem involving an extra projection step. Second, the projection step may not be straightforward to carry out unless the set $\mathcal{W}$ is simple enough. Third, the bound that results on the sub-gradient vectors by limiting w to $\mathcal{W}$ can be very loose.

For these reasons, we do not rely on the restrictive condition (9) and introduce instead the more relaxed affine-Lipschitz condition (8). This condition is weaker than (9). Indeed, it can be verified that (9) implies (8) but not the other way around. The following example shows that the important problem of SVM learning satisfies condition assumption 3; a similar conclusion applies to ℓ_1-regularized least-square (LASSO) but is omitted for brevity.

2.2. Example: Single-Agent SVM Learning

The two-class SVM formulation deals with the problem of determining a separating hyperplane, $w \in \mathbb{R}^M$, in order to classify feature vectors, denoted by $\boldsymbol{h} \in \mathbb{R}^M$, into one of two classes: $\boldsymbol{\gamma} = +1$ or $\boldsymbol{\gamma} = -1$. The regularized SVM risk function is strongly-convex and of the form:

$$J^{\rm svm}(w) \;\triangleq\; \frac{\rho}{2}\|w\|^2 + \mathbb{E}\left(\max\left\{0, 1 - \boldsymbol{\gamma}\boldsymbol{h}^{\mathsf{T}} w\right\}\right) \tag{10}$$

where $\rho > 0$ is a regularization parameter. We are generally given a collection of independent training data, $\{\boldsymbol{\gamma}(i), \boldsymbol{h}_i\}$, consisting of feature vectors and their class designations and assumed to arise from joint wide-sense stationary processes. One choice to approximate the sub-gradient vector of $J^{\rm svm}(w)$ is to employ the following instantaneous approximation:

$$\widehat{g}^{\rm svm}(\boldsymbol{w}_{i-1}) = \rho \boldsymbol{w}_{i-1} + \boldsymbol{\gamma}(i)\boldsymbol{h}_i\, \mathbb{I}[\boldsymbol{\gamma}(i)\boldsymbol{h}_i^{\mathsf{T}}\boldsymbol{w}_{i-1} \leq 1] \tag{11}$$

In this expression, the indicator function $\mathbb{I}[a]$ is 1 if the statement a is true; otherwise it equals 0. Then, the gradient noise process in the SVM formulation is given by

$$\boldsymbol{s}_i(\boldsymbol{w}_{i-1}) = \boldsymbol{\gamma}(i)\boldsymbol{h}_i\,\mathbb{I}[\boldsymbol{\gamma}(i)\boldsymbol{h}_i^{\mathsf{T}}\boldsymbol{w}_{i-1} \le 1] - \mathbb{E}\,\boldsymbol{\gamma}\boldsymbol{h}\,\mathbb{I}[\boldsymbol{\gamma}\boldsymbol{h}^{\mathsf{T}}\boldsymbol{w}_{i-1} \le 1] \tag{12}$$

It is easy to verify that Assumption 1 is satisfied with $\beta^2 = 0$ and $\sigma^2 = \text{Tr}(R_h)$, where $R_h = \mathbb{E}\,\boldsymbol{h}\boldsymbol{h}^{\mathsf{T}}$. Likewise Assumption 3 is satisfied with parameters $c = \rho$ and $d = 2[\text{Tr}(R_h)]^{1/2}$.

2.3. Performance Analysis

In preparation for the analysis, we first conclude from (8) that:

$$\|g(w_1) - g(w_2)\|^2 \le e^2\|w_1 - w_2\|^2 + f^2 \quad \forall w_1, w_2,\ g \in \partial J \tag{13}$$

where

$$e^2 \;\triangleq\; c^2 + \frac{2cd}{R} \ge 0, \qquad f^2 \;\triangleq\; d^2 + 2cdR \ge 0 \tag{14}$$

and the constant R is any positive number that we are free to choose.

At every iteration i, the risk value that corresponds to the iterate $\boldsymbol{w}_i$ is $J(\boldsymbol{w}_i)$. This value is obviously a random variable due to the randomness in the data used to run the algorithm. We denote the mean risk value by $\mathbb{E}\,J(\boldsymbol{w}_i)$. The next theorem shows how fast and how close this mean value approaches the optimal value, $J(w^\star)$. To do so, the statement in the theorem relies on the *best pocket* iterate, denoted by $\boldsymbol{w}_i^{\text{best}}$, and which is defined as:

$$\boldsymbol{w}_i^{\text{best}} \;\triangleq\; \arg\min_{0\le j\le i}\ \mathbb{E}\ J(\boldsymbol{w}_j) \tag{15}$$

Theorem 1 (Single agent performance) *Consider using the stochastic sub-gradient algorithm (3) to seek the unique minimizer, $w^\star$, of problem (1), where the risk function satisfies Assumptions 1–3. If the step-size parameter is sufficiently small, then it holds that*

$$\lim_{i\to\infty} \mathbb{E}\,J(\boldsymbol{w}_i^{\text{best}}) - J(w^\star) \;\le\; \mu(f^2 + \sigma^2)/2 \tag{16}$$

Moreover, the convergence of $\mathbb{E}\ J(\boldsymbol{w}_i^{\text{best}})$ towards $J(w^\star)$ occurs at an exponential rate, $O(\alpha^i)$, where

$$\alpha \;\triangleq\; 1 - \mu\eta + \mu^2(e^2 + \beta^2) = 1 - O(\mu) \tag{17}$$

Proof: Omitted due to space limitations — see [26] ■

The above theorem only clarifies the performance of the best pocket value, which is not readily available during the algorithm implementation since the risk function itself cannot be evaluated due to the lack of knowledge about the probability distribution of the data. However, a more practical conclusion can be deduced from the statement of the theorem as follows. Suppose we choose a parameter κ that satisfies $\alpha \le \kappa < 1$. Next, we introduce the convex-combination coefficients:

$$r_L(j) \;\triangleq\; \frac{\kappa^{L-j}}{S_L}, \quad j = 0, 1, \ldots, L, \text{ where } S_L \;\triangleq\; \sum_{j=0}^{L}\kappa^{L-j} \tag{18}$$

Using these coefficients, we define the weighted iterate

$$\bar{\boldsymbol{w}}_L \;\triangleq\; \sum_{j=0}^{L} r_L(j)\boldsymbol{w}_j \tag{19}$$

Observe that, in contrast to $\boldsymbol{w}_L^{\text{best}}$, the above weighted iterate is computable since its value depends on the successive iterates $\{\boldsymbol{w}_j\}$ and these are available during the operation of the algorithm. Observe further that $\bar{\boldsymbol{w}}_L$ satisfies the recursive construction:

$$\bar{\boldsymbol{w}}_L \;=\; \left(1 - \frac{1}{S_L}\right)\bar{\boldsymbol{w}}_{L-1} \;+\; \frac{1}{S_L}\boldsymbol{w}_L \tag{20}$$

Now, since $J(\cdot)$ is a convex function, it holds that

$$J(\bar{\boldsymbol{w}}_L) \;=\; J\left(\sum_{j=0}^{L} r_L(j)\boldsymbol{w}_j\right) \;\le\; \sum_{j=0}^{L} r_L(j)J(\boldsymbol{w}_j) \tag{21}$$

Using this fact, we can derive a result similar to (16) albeit applied to $\bar{\boldsymbol{w}}_L$. Specifically, under the same conditions as in Theorem 1, it holds that

$$\lim_{L\to\infty} \mathbb{E}\,J(\bar{\boldsymbol{w}}_L) - J(w^\star) \le \mu(f^2 + \sigma^2)/2 \tag{22}$$

and the convergence of $\mathbb{E}\,J(\bar{\boldsymbol{w}}_L)$ towards $J(w^\star)$ continues to occur at an exponential rate, $O(\kappa^L)$.

2.4. Simulation: Single-Agent SVM Learning

We compare the performance of the stochastic sub-gradient SVM implementation against LIBSVM (a popular SVM solver that uses quadratic programming on dual problem) [27]. The test data is obtained from the LIBSVM website[1] and also from the UCI dataset[2]. We first use the Adult dataset after preprocessing [28] with 11,220 training data and 21,341 testing data in 123 feature dimensions. To ensure a fair comparison, we use linear LIBSVM with the exact same parameters as the sub-gradient method. Hence, we choose $C = 5 \times 10^2$ for LIBSVM, which corresponds to $\rho = \frac{1}{C} = 2 \times 10^{-3}$. We also set $\mu = 0.05$. We can see from Fig. 1 that the stochastic sub-gradient algorithm is able converge to the performance of LIBSVM quickly. Since we only use each data point once, and since each iteration is computationally simpler, the sub-gradient implementation ends up being computationally more efficient. Similar performance results can be obtained for LASSO (ℓ_1-regularized least-squares) problems and for total-variation based image denoising problems. We omit these two examples due to space limitations — see [26].

3. PROBLEM FORMULATION: MULTI-AGENT CASE

We now extend the previous results to multi-agent networks where a collection of agents cooperate with each other to seek the minimizer of an aggregate cost of the form:

$$\min_{w} \sum_{k=1}^{N} J_k(w), \quad \text{where } J_k(w) \;\triangleq\; \mathbb{E}_{\boldsymbol{x}_k} Q_k(w; \boldsymbol{x}_k) \tag{23}$$

where k refers to the agent index. Extension of the earlier results to the multi-agent case requires some nontrivial effort due to the coupling that exists among neighboring agents. Nevertheless, the same broad conclusion will continue to hold with proper adjustments. We continue to assume that the individual costs satisfy Assumptions 2 and 3, i.e., each $J_k(w)$ is strongly-convex and its sub-gradient vectors are affine-Lipschitz with parameters $\{\eta_k, c_k, d_k\}$. We further

[1] http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets

[2] http://archive.ics.uci.edu/ml/

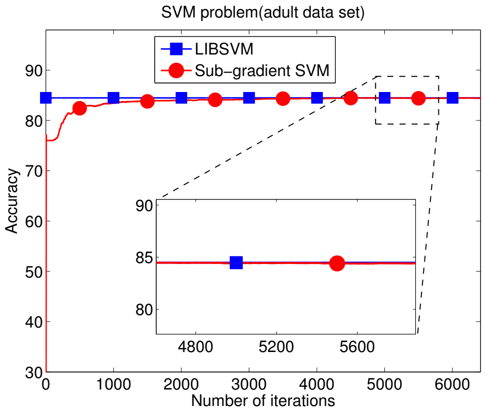

Fig. 1. SVM solvers applied to the Adult data set. Comparison of the performance accuracy, percentage of correct prediction over test dataset, for LIBSVM [27] and a stochastic sub-gradient implementation.

assume that the individual risks share a common minimizer, $w^\star$, which will therefore agree with the global minimizer for (23). This scenario corresponds to the important situation in which agents have a common objective (or task), namely, that of estimating the same parameter vector, $w^\star$, in a distributed manner through *localized* interactions and cooperation.

Thus, consider a network consisting of N separate agents connected by a topology. As described in [13, 29], we assign a pair of nonnegative weights, $\{a_{k\ell}, a_{\ell k}\}$, to the edge connecting any two agents k and ℓ. The scalar $a_{\ell k}$ is used by agent k to scale the data it receives from agent ℓ and similarly for $a_{k\ell}$. There are several strategies that the agents can employ to seek the minimizer, $w^\star$, including consensus and diffusion strategies [13–16, 29–31]. In this work, we focus on the latter class since diffusion implementations have been shown to have superior stability and performance properties when used in the context of adaptation and learning from streaming data. [13, 29, 32]. We therefore consider the following diffusion strategy in its adapt-then-combine (ATC) form:

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} - \mu\, \widehat{g}_k(\boldsymbol{w}_{k,i-1}) \qquad (24)$$

$$\boldsymbol{w}_{k,i} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\boldsymbol{\psi}_{\ell,i} \qquad (25)$$

The entries $A = [a_{\ell k}]$ define a left-stochastic matrix. Since the network is strongly-connected, the combination matrix A will be primitive [13, 33]. The eigenvectors of A corresponding to the eigenvalue at one are denoted by $Ap = p$ and $A^{\mathsf{T}}\mathbb{1} = \mathbb{1}$. It follows from the Perron-Frobenius theorem [33] that the entries of p are all strictly positive and we normalize them to add up to one. We denote the individual entries of p by $\{p_k\}$.

The next result extends Theorem 1 to the network case. The result establishes that the distributed strategy is stable and converges exponentially fast for sufficiently small step-sizes. For each agent, we again introduce a *best pocket* iterate, denoted by $\boldsymbol{w}_{k,i}^{\text{best}}$:

$$\boldsymbol{w}_{k,i}^{\text{best}} \triangleq \arg\min_{0\le j\le i} \mathbb{E}\, J_k(\boldsymbol{w}_{k,j}) \qquad (26)$$

Theorem 2 (NETWORK PERFORMANCE) *Consider using the stochastic sub-gradient diffusion algorithm (24)–(25) to seek the unique minimizer, $w^\star$, of problem (23), where the risk functions, $J_k(w)$, satisfy Assumptions 1–3 with parameters $\{\eta_k, \beta_k^2, \sigma_k^2, e_k^2, f_k^2\}$. Assume the step-size parameter is sufficiently small. It holds that*

$$\lim_{i\to\infty} \mathbb{E}\left(\sum_{k=1}^N p_k J_k(\boldsymbol{w}_{k,i}^{\text{best}}) - \sum_{k=1}^N p_k J_k(w^\star)\right) \le \frac{\mu}{2}\sum_{k=1}^N \left(p_k f_k^2 + p_k^2\sigma_k^2 + 2p_k f_k h\right) = O(\mu) \qquad (27)$$

for some finite constant h. Moreover, the convergence occurs at an exponential rate, $O(\alpha_q^i)$, where

$$\alpha_q \triangleq \max_k \left\{1 - \mu\eta_k + \mu^2 e_k^2 + \mu^2\beta_k^2 p_k + \mu^2 h\frac{e_k^2}{f_k}\right\} = 1 - O(\mu) \qquad (28)$$

Proof: Omitted for brevity — see [26]. ■

A conclusion similar to (22) also holds in the multi-agent case [26]. Examining the bound in (27), and comparing it with result (22) for the single-agent case, we observe that the topology of the network is now reflected in the bound through the Perron entries, p_k. Moreover, the bound in (27) involves three terms (rather than only two as in the single-agent case): (1) $p_k f_k^2$, which arises from the non-smoothness of the risk function; (2) $p_k^2\sigma_k^2$, which is due to gradient noise and the approximation of the true sub-gradient vector; (3) $2hp_k f_k$, which is an extra term in comparison to the single agent case. This term reflects the small average variations in performance that arise across agents over network.

3.1. Simulation: Multi-Agent SVM Learning

We examine the Adult dataset again. We distribute 32561 training data over a network consisting of 20 agents. We set $\rho = 0.002$, equivalent to $C = 500$ in LIBSVM, and $\mu = 0.15$ for all agents and we choose $\kappa = 1 - 0.9\mu\rho$. Figure 2 shows that cooperation among the agents outperforms the non-cooperative solution. Moreover, the distributed network can almost match the performance of the centralized LIBSVM solution.

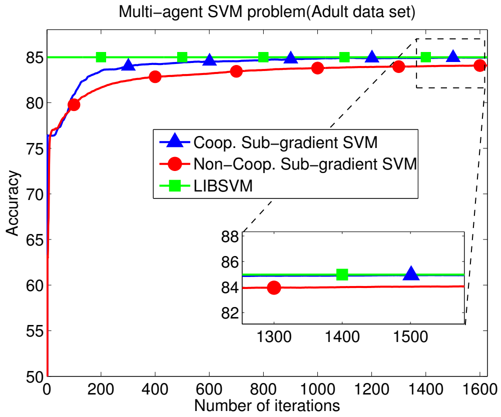

Fig. 2. Performance of multi-agent SVM solution(Adult dataset).

4. REFERENCES

[1] D. P. Bertsekas, *Nonlinear Programming*, Athena scientific Belmont, 1999.

[2] B. T. Polyak, *Introduction to Optimization*, Optimization Software New York, 1987.

[3] Y. Nesterov, *Introductory Lectures on Convex Optimization*, Springer, 2004.

[4] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, Springer, 2009.

[5] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society, Series B (Methodological)*, pp. 267–288, 1996.

[6] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[7] S. Shalev-Shwartz, Y. Singer, N. Srebro, and A. Cotter, "Pegasos: Primal estimated sub-gradient solver for SVM," *Mathematical Programming*, vol. 127, no. 1, pp. 3–30, 2011.

[8] V. Vapnik, *Statistical Learning Theory*, Wiley NY, 1998.

[9] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.

[10] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, Academic Press, 4th edition, 2008.

[11] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1, pp. 259–268, 1992.

[12] A. Beck and M. Teboulle, "Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems," *IEEE Trans. Image Process.*, vol. 18, no. 11, pp. 2419–2434, 2009.

[13] A. H. Sayed, "Adaptation, learning, and optimization over networks," *Foundations and Trends in Machine Learning*, vol. 7, no. 4-5, pp. 311–801, 2014.

[14] J. Chen and A. H. Sayed, "On the learning behavior of adaptive networks—Part I: Transient analysis," *IEEE Trans. Inf. Thy.*, vol. 61, no. 6, pp. 3487–3517, June 2015.

[15] J. Chen and A. H. Sayed, "On the learning behavior of adaptive networks—Part II: Performance analysis," *IEEE Trans. Inf. Thy.*, vol. 61, no. 6, pp. 3518–3548, June 2015.

[16] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Trans. Autom. Control*, vol. 54, no. 1, pp. 48–61, 2009.

[17] O. Bousquet and L. Bottou, "The tradeoffs of large scale learning," in *Advances in Neural Information Processing Systems 20*, J.C. Platt, D. Koller, Y. Singer, and S.T. Roweis, Eds., pp. 161–168. Curran Associates, Inc., 2008.

[18] L. Bottou, "Stochastic gradient tricks," in *Neural Networks, Tricks of the Trade, Reloaded*, G. Montavon, G. B. Orr, and K. Müller, Eds., Lecture Notes in Computer Science (LNCS 7700), pp. 430–445. Springer, 2012.

[19] Z. J. Towfic and A. H. Sayed, "Stability and performance limits of adaptive primal-dual networks," *IEEE Trans. Signal Process.*, vol. 63, no. 11, pp. 2888–2903, June 2015.

[20] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro, "Robust stochastic approximation approach to stochastic programming," *SIAM J. Optm.*, vol. 19, no. 4, pp. 1574–1609, 2009.

[21] S. Boyd and A. Mutapcic, "Stochastic subgradient methods," *Lecture Notes for EE364b*, Stanford University, 2008.

[22] S. S. Ram, A. Nedić, and V. V. Veeravalli, "Distributed stochastic subgradient projection algorithms for convex optimization," *J. Optm. Theory and Appl.*, vol. 147, no. 3, pp. 516–545, 2010.

[23] S. S. Ram, A. Nedic, and V. V. Veeravalli, "Incremental stochastic subgradient algorithms for convex optimization," *SIAM J. Optm.*, vol. 20, no. 2, pp. 691–717, 2009.

[24] N. Z. Shor, *Minimization Methods for Non-differentiable Functions*, vol. 3, Springer-Verlag, 2012.

[25] K. Kiwiel, *Methods of Descent for Non-differentiable Optimization*, Springer-Verlag, 1985.

[26] B. Ying and A. H. Sayed, "Performance limits of online stochastic sub-gradient learning," *arXiv preprint arXiv:1511.07902*, 2015.

[27] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. on Intelligent Systems and Tech.*, vol. 2, pp. 27:1–27:27, 2011.

[28] J. Platt, "Fast training of support vector machines using sequential minimal optimization," *Advances in Kernel Methods—Support Vector Learning*, vol. 3, 1999.

[29] A. H. Sayed, "Adaptive networks," *Proceedings of the IEEE*, vol. 102, no. 4, pp. 460–497, 2014.

[30] S. Kar and J. M. F. Moura, "Distributed consensus algorithms in sensor networks with imperfect communication: Link failures and channel noise," *IEEE Trans. Signal Process.*, vol. 57, no. 1, pp. 355–369, 2009.

[31] W. Yu, G. Chen, Z. Wang, and W. Yang, "Distributed consensus filtering in sensor networks," *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 39, no. 6, pp. 1568–1577, 2009.

[32] S.-Y. Tu and A. H. Sayed, "Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks," *IEEE Trans. Signal Process.*, vol. 60, no. 12, pp. 6217–6234, 2012.

[33] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*, SIAM, PA, 2000.