Fifty-third Annual Allerton Conference
Allerton House, UIUC, Illinois, USA
September 29 - October 2, 2015

Distributed Proportional Stochastic Coordinate Descent With Social Sampling

Mohsen Ghassemi Anand D. Sarwate
Department of Electrical and Computer Engineering
Rutgers, the State University of New Jersey
Piscataway, NJ
Email: m.ghassemi@rutgers.edu, asarwate@ece.rutgers.edu

***Abstract*—We consider stochastic message passing algorithms that limit the communication required for decentralized and distributed convex optimization and provide convergence guarantees on the objective value. We first propose a centralized method that modifies the coordinate-sampling distribution for stochastic coordinate descent, which we call *proportional stochastic coordinate descent*. This method treats the gradient of the function as a probability distribution to sample the coordinates, and may be useful in so-called lock-free decentralized optimization schemes. For general distributed optimization in which agents jointly minimize the sum of local objectives, we propose treating the iterates as gradients and propose a stochastic coordinate-wise primal averaging algorithm for optimization.**

I. Introduction

Large-network paradigms for communication and distributed computation have driven renewed interest in opinion and belief formation models from mathematical sociology and psychology. One such recent work is the novel message passing protocol called *social sampling* [1] that uses limited communication to perform distributed estimation. This protocol is similar to consensus-based multi-agent optimization models – the goal of this work is to investigate the connection between the two. The idea is that every agent performs local processing based on its local objective function, then samples its *belief* or *state* of the global at random to send to its neighbors. Subsequently, agents update their belief based on the messages they receive from their neighbors. Transmitting samples of the belief instead of the complete information makes this method suitable for distributed settings with limited communication resources.

The social sampling setup is similar to several existing stochastic optimization methods, especially stochastic coordinate descent were social samples are the partial derivatives [2], [3]. Distributed optimization has received significant interest in recent years especially consensus-based algorithms under various assumptions and constraints [4]–[8]. Many of these variants build on general analyses [9]–[11] are among remarkable works in the distributed optimization literature. Many other authors have have studied non-uniform sampling algorithms that differ from ours [12]–[15]. Of particular note is the seminal work of Nesterov, who proves linear convergence rate for his non-uniform method for strongly convex objective functions [12]. In our centralized setting we consider optimization of convex and smooth objectives rather than strongly convex objectives.

We first propose a method for the simpler case of centralized optimization that uses a novel non-uniform sampling of the coordinates. In this scheme, the chance of a coordinate j being selected is proportional to partial gradients $\frac{\partial f(w)}{\partial w_j}$. We show that for convex smooth objective functions, our algorithm, with constant step size, achieves $\mathcal{O}\left(\frac{1}{t}\right)$ convergence rate in expectation. Our centralized analysis is based on the analysis of the uniform scheme by Sahlev-Shwartz and Tewari [3]. The recent survey of Wright [16] summarizes much of the early work on coordinate descent methods.

Our proportional method can also be adopted for shared memory systems where the nodes (computational agents) are arranged in a star network. For this setup, our algorithm is based on the framework used by Recht et al. [17] where a central node (memory node) keeps the current global decision vector and the rest of nodes (computing nodes) access to this value and update it in an asynchronous manner. In this framework, it is assumed that while each working node is computing its update based on its local objective function value and transmitting it to the central node, other working nodes can also access or update the decision vector. This means that the estimates of the gradient vector that are transmitted to the central node could be obsolete. We propose that each node evaluates its estimate of the gradient according to the PSCD update rule. Assuming that the gap between the access time and the update time of each node is limited, the suboptimality gap is bounded.

This approach to decentralized optimization is like a star network with several nodes connected to a single memory node. Several authors have also considered star network models [18], [19]. For general connected networks, the gradient information from other nodes is useful only if their current states are not very different which for strongly convex objective function is the case if the estimates are close to the optimum. However, we are interested in methods that guarantee convergence (at least in expectation) to the optimal point regardless of the initial estimate given to the algorithm.

For distributed settings we propose social sampling treating the primal iterate as a probability distribution and exchanging samples in the network. This solution might be

978-1-5090-1824-6/15/$31.00 ©2015 IEEE

useful for networks with limited computation and communication resources. Our methods build on the framework developed by Nedić and Ozdaglar [20]. We assume that the computational nodes broadcast information about their current local decision vectors to their neighbors to cooperatively optimize the global objective function which is the average of the local objective functions. However, in contrast to the mentioned works, our methods rely on sharing partial information with the neighbors, namely information about a small subset of the coordinates.

II. Preliminaries

A. Notation

Throughout this paper, superscript i indicates node i of a network, except for e^j that denotes the j-th standard coordinate vector. Furthermore, subscript t indicates the discrete time (iterations index). All element indexes in matrices and vectors are demonstrated as subscripts, as well. We denote the set $\{1, \cdots, k\}$ by $[k]$. The vector $\mathbf{1}_A$ for $A \subseteq [d]$ is a d-dimensional vector with 1's for indices $i \in A$ and 0 elsewhere.

B. Optimization

Definition 1. *A function $f : \mathcal{R}^d \to \mathcal{R}$ is convex if for all vectors u and v,*

$$f(u) - f(v) \geq \nabla f(v)^\top (u - v) . \tag{1}$$

Definition 2. *A function $f : \mathcal{R}^d \to \mathcal{R}$ is λ-strongly convex if for all vectors u and v,*

$$f(u) - f(v) \geq \nabla f(v)^\top (u - v) + \frac{\lambda}{2}\|u - v\|^2. \tag{2}$$

Definition 3. *A function $f : \mathcal{R}^d \to \mathcal{R}$ is L-Lipschitz continuous if for all vectors u and v,*

$$\|f(u) - f(v)\| \leq L \|u - v\| . \tag{3}$$

Definition 4. *A function $f : \mathcal{R}^d \to \mathcal{R}$ is L-smooth if it is twice differentiable and has L-Lipschitz continuous gradients.*

The optimal solution to an optimization problem is denoted by w^*. A solution w_t (also referred to in this paper as estimate, belief, or decision vector), found by an optimization algorithm after t iterations is "ϵ-accurate" if $f(w_t) \leq f(w^*) + \epsilon$. Let $\mathcal{F}_t$ be the sigma algebra of all the random events up to time t.

C. Network Model

We will consider two types of problems in this paper: centralized and distributed. For the distributed setting, we make the following assumptions on the network.

Let $\mathcal{G} = (V, E)$ represent an undirected graph with vertex set $V = \{1, \cdots, n\}$ and edge set $E \subseteq V \times V$. Let $\mathcal{N}^i \subset V$ be the set of the neighbors of node (vertex) i and $\tilde{\mathcal{N}}^i = \mathcal{N}^i \cup i$.

In the distributed setup, the optimization task is jointly accomplished by the n processing units that are arranged in a network represented by a graph $\mathcal{G} = (V, E)$ which we assume to be connected; we further assume $(i, i) \in E$ for all i. An $n \times n$ matrix Q is called graph conformant if $Q_{ik} = 0$ for $(i, k) \notin E$. We consider matrix-valued processes $Q(t)$ where $Q(t)$ is doubly stochastic. We use the notation $Q_{ik}(t) = q_k^i(t)$. We think of $q_k^i(t)$ as the weight that node i assigns to the information from node k at time t. Throughout the paper we assume that the expectation of each stochastic graph conformant matrix corresponds to a connected graph. Deterministic matrices correspond to connected graphs as well.

III. Problem Setups and Algorithms

First, we consider an optimization problem in a centralized setup. Then, We study the case of optimizing the sum of functions f_i for $i \in [n]$ where each function is associated with one node of a network.

A. Centralized problem and algorithm

In the centralized problem, we aim to minimize the following objective function:

$$\min_{w \in \mathcal{R}^d} f(w), \tag{4}$$

where $f(w)$ is a convex smooth function.

In order solve the minimization problem (4), we use a variant of the stochastic coordinate descent method, which we call centralized Proportional Stochastic Gradient Descent (centralized PSCD). At every iteration t, a coordinate j is randomly selected and the j-th coordinate of w_t is updated:

$$w_{t+1} = w_t - \eta \hat{g}_t^j, \tag{5}$$

where $\hat{g}_t^j = C_1 e^j$ is an unbiased estimate of the gradient vector on a single coordinate: $\mathbb{E}\left[\hat{g}^j\right] = g_t$. In this algorithm the coordinates are selected according to the following distribution:

$$P(j) = \frac{|g_j|}{\|g_t\|_1}, \tag{6}$$

where $g_t \in \partial f(w_t)$ is a sub-gradient of $f(w_t)$ and g_j is a sub-derivative of $f(w_t)$ w.r.t. the j-th coordinate. Considering that $\mathbb{E}\left[\hat{g}^j\right] = \mathbb{E}\left[C_1 e^j\right] = C_1 \sum_j \frac{|g_j|}{\|g_t\|_1} e^j$ while $\mathbb{E}[\mathrm{sgn}(g_j) \cdot \|g_t\|_1 \cdot e^j] = g_t$, we need to set $C_1 = \|g_t\|_1 \, \mathrm{sgn}(g_j)$. In this setup, we use constant step size $\eta = \frac{1}{\alpha L}$ where L is the maximum component-wise Lipschitz constant of $f(w)$ and α is a constant.

The pseudo-code for the centralized setup is demonstrated in Algorithm 1.

B. Shared Memory System

Our proportional sampling scheme extends naturally to shared-memory models for distributed optimization. In these models, a common memory element holding the current iterate w_t is accessed by a collection of n processors, each

Algorithm 1 Centralized PSCD

```
**Require:** $\lambda$, $L$, $N$, $T$
  arbitrarily select $w_0 \in \mathcal{R}^d$
  set $\eta = \frac{1}{\alpha L}$
  **for** $t = 1, 2, \ldots T$ **do**
    calculate $g_t \in \partial f(w_t)$
    select $J_t$ according to $P(j) = \frac{|g_j|}{\|g_t\|_1}$ for $j \in [d]$
    set $w_{t+1} = w_t - \eta_t \|g_t\|_1 \operatorname{sgn}(g_j) e^j$
  **end for**
  **return** $w_{T+1}$
```

with its own local objective function $f^i(w)$. The goal of such a system is to minimize the average of the local objectives:

$$\min_{w \in \mathcal{R}^d} f_S(w) = \frac{1}{n} \sum_{i=1}^{n} f^i(w), \tag{7}$$

where $\{f^i(w)\}_{i=1}^n$ are strongly convex functions with Lipschitz continuous gradients. In this setup, a central node has a memory that keeps a shared estimate vector and all other nodes have access to this node to read or update the estimate. If the nodes operate synchronous, the algorithm will be essentially performing the conventional unbiased stochastic gradient descent on the entire data set $\mathbb{S}$.

Here, we focus on the more challenging asynchronous setup, which can be considered as a modified version of HOGWILD! [17]. Our proposed method for this setting, called asynchronous distributed PSCD, assumes that each node reads the shared vector at arbitrary times and updates the estimate using its local gradient information. The update rule in this method is

$$w_{t^i+1} = w_{t^i} - \eta \hat{g}^i_{t^i_\tau}, \tag{8}$$

where η is a constant step size and $\hat{g}^i_{t^i_\tau} = \left\|g^i_{t^i_\tau}\right\|_1 \operatorname{sgn}((g^i_{t^i_\tau})_j) \; e^j$. Also, $w_{t^i_\tau}$, w_{t^i},and w_{t^i+1} are the values of the shared estimate when accessed by node i, when it is about to be updated by node i, and when the update by node i is used, respectively. The delay $t^i - t^i_\tau \leq \tau$ is sum of two delays, namely, the computation time of $\hat{g}^i_{t^i_\tau}$ and communication delay between node i and the central node. During this period, the estimate vector might be updated by other nodes. In fact, we assume that t keeps track of the number of updates to the shared vector by any node. Therefore, τ is essentially an upper bound on the number of updates by the other nodes while a certain node is computing and transmitting its update to the central node. Since in our algorithm only one random coordinate is updated by each node per iteration, the updates do not get overwritten by the other nodes too often.

C. Distributed coordinate-wise Primal Averaging algorithms

Similar to shared memory model, in a general connected network, we aim to minimize the average of the local objective functions associated with the nodes of the network:

$$\min_{w \in \mathcal{R}^d} f_D(w) = \frac{1}{n} \sum_{i=1}^{n} f^i(w), \tag{9}$$

where $\{f^i(w)\}$ are strongly convex functions.

It is tempting to directly apply proportional gradient sampling to the general network setting for distributed optimization. A naïve adoption of our centralized method for distributed setups would involve communicating proportionally sampled estimates of the local gradients with the neighbors in order to converge to a common optimal point of the global objective function. We show that this method actually works if the nodes are fully connected where all of the nodes start with the same initial value. However, sending gradient information to neighbors in the network does not necessarily help them reach the global minimizer because at any instant t, various nodes have different values of estimates w^i_t, so the gradient values from the neighbors might be totally irrelevant. Hence, for general connected networks we suggest that the nodes exchange partial information about their current local estimates $\{w^i_t\}_{i=1}^n$ instead of communicating gradient information.

If all nodes are connected to each other, every node has the gradient information of all nodes. However, since the nodes are not sharing their estimate values $\{w^i_t\}$, the gradient information is not useful unless all the nodes have the same iterate value at every iteration. In order to satisfy this requirement without synchronizing the iterate values at every iteration, nodes can start the optimization algorithm with the same initial value. The update rule in this model is

$$w^i_{t+1} = w^i_t - \eta_t \sum_{k=1}^{n} \frac{\hat{g}^k_t}{n}, \tag{10}$$

where $\eta = \frac{1}{\lambda t}$ and $\hat{g}^k_t = \|g^k_t\|_1 \cdot \operatorname{sgn}\left(g^k_{J^k_t}\right) \cdot e^{J^k_t}$. In the end, $\widetilde{w}^i_T$ will be calculated.

D. Synchronous Coordinate-wise Primal Averaging

As before, we are motivated by the problem of limiting communication during the iterations. Based on the works by [20] we consider a communication scheme based on sending partial information about the current iterate w^i_t: each node only sends an estimate $\hat{w}^t_i$ of their current actual estimate w^t_i. In particular, $\hat{w}^t_i$ only contains information about a single (random) coordinate of w^i_t.

A simple model is for an oracle to select a coordinate of $\{w^i_t\}$ and have all nodes synchronized to transmit information about that same coordinate. This method can be seen as a coordinate-wise variant of the distributed primal averaging algorithm of Nedić and Ozdaglar [20]. In this algorithm, at every iteration some oracle selects a random coordinate j from $\{1, \cdots, n\}$ and orders all nodes to send the j-th coordinate of w^i_t to their neighbors. Each node i updates its j-th coordinate with a weighted average of their neighbor's coordinates. Subsequently, node i updates all coordinates of w^i_t using its full local gradient $g^i_t = \nabla f^i(w^i_t)$. Therefore, the update rule can be written as:

$$w^i_{t+1} = \sum_{j=1}^{d} \sum_{k\in\hat{\mathcal{N}}^i} Q^j_{ik}(t) D^j w^k_t - \eta^i_t g^i_t \qquad \text{for } j \in [d], \tag{11}$$

where D^j is a diagonal matrix with 1 on its j-th diagonal element and zero otherwise and

$$Q^j(t) = \begin{cases} Q & \text{if } j \text{ is selected at time } t, \\ I & \text{otherwise.} \end{cases} \tag{12}$$

E. Asynchronous Coordinate-wise Primal Averaging

The synchronous algorithm requires an oracle to select a common coordinate j so that all nodes average on the same coordinate at each time. In a more realistic scenario, however, we would like to allow each node i to select its own coordinate j^i at random. It can then send the pair $\left(j^i, (w^i_t)_{j^i}\right)$ to its neighbors. Then, node i will receive a collection of *requests* $\left\{\left(j^k, (w^i_t)_{j^k}\right) : k \in \mathcal{N}^i\right\}$. For each $k \in \mathcal{N}^i$, it will send $\left(j^k, (w^i_t)_{j^k}\right)$ to node k. This preserves the bidirectionality of the links. Each node, for every coordinate, computes a convex combination of its own belief and those of the neighbors who have sent their information about the same coordinate. For different coordinates, the assigned weights to the neighbors need not be the same.

For this algorithm the update rule is the same as (11), except that for every coordinate j, matrix $Q^j(t)$ is chosen i.i.d across time according to $\mathcal{P}^{\mathcal{G}}_j$, which is a a probability distribution over a set of doubly stochastic matrices comfortant to subgraphs of $\mathcal{G}$ each of which include all of the self-loops.

Algorithm 2 Asynchronous Coordinate-wise Primal Averaging

```
Require: $N$, $T$, graph $\mathcal{G}$, step size sequence $\{\eta_t\}$, matrix $Q$
  arbitrarily select $w^i_1 \in \mathcal{R}^d$ for all $i \in [n]$.
  for $t = 1, 2, \dots T$ do
    for all $i \in [n]$ do
      compute $g^i_t \in \partial f^i(w^i_t)$
      select $j^i$ uniformly in $[d]$
      send $\left(j^i, (w^i_t)_{j^i}\right)$ to all nodes in $\mathcal{N}^i$
    end for
    for all $i \in [n]$ do
      send $\left(j^k, (w^i_t)_{j^k}\right)$ to each node $k \in \mathcal{N}^i$
    end for
    for all $i \in [n]$ do
      for all $j \in [d]$ do
        if  $j \in \{j^k : k \in \bar{\mathcal{N}}^i\}$  then
          $(v^i_{t+1})_j = \sum_{k\in\mathcal{N}^i} Q^j_{ik}(w^k_t)_j$
        else
          $(v^i_{t+1})_j = (w^i_t)_j$
        end if
      end for
      $w^i_{t+1} = v^i_{t+1} - \eta_t g^i_t$
    end for
  end for
  return  for each $i \in [n]$ the average
              $\tilde{w}^i_T = \frac{1}{T}\sum_{t=1}^{T} w^i_t.$
  for each $i \in [n]$
```

IV. Convergence Analysis

The convergence analysis of the methods presented in the previous section is provided here. Due to lack of space, more detailed proofs will appear in the extended version of this paper. We first provide an analysis of centralized PCSD. We then state a result for the shared-memory case. We conclude with some results on the asynchronous distributed primal coordinate descent method.

A. Analysis of Centralized PSCD

Theorem 1. *Consider Algorithm* (1) *for solving problem* (4) *when f is convex with L-Lipschitz continuous component-wise gradients. With constant step size $\eta = \frac{1}{\alpha L}$ we have:*

$$\mathbb{E}\left[f(w_T) - f(w^*)\right] \le \frac{\alpha\left(\Psi(w_0) - \Psi(w^*)\right)}{T}, \tag{13}$$

where

$$\Psi(w) = f(w) + \frac{L}{2}\|w - w^*\|^2. \tag{14}$$

To prove this theorem we need the following Lemma, which is a corollary of Theorem 2.1.5 in the book of Nesterov [21].

Lemma 1. *Suppose that function $f(w)$ has component-wise Lipschitz continuous gradient:*

$$|\nabla_j f(w + he^j) - \nabla_j f(w)| \le L^j |h|,$$

Then we have:

$$f(w + he^j) - f(w) \le \left\langle \nabla f(w), he^j \right\rangle + \frac{L^j}{2}|h|^2. \tag{15}$$

Proof: To find an upper bound on the optimality gap in the centralized setup, we will use the preceding Lemma. Following the approach taken by Shalev-Shwartz and Tewari [3], define the potential function $\Psi(w)$ in (14), where $w^* = \operatorname{argmin}_w f(w)$ is the minimizer of the objective function $f(w)$ and L is the maximum component-wise Lipschitz constant of $f(w)$. Using this potential function we will prove that under some condition for λ our suggested method for updating w_t will converge to the optimal solution. Define $\gamma_t = \eta\|g_t\|_1 \operatorname{sgn}(g_j)$ so that the update $\eta\hat{g}_t = \gamma_t e^j$.

Consider the difference of the potential across one iteration:

$$\begin{aligned}
&\Psi(w_t) - \Psi(w_{t+1}) \\
&= f(w_t) - f(w_{t+1}) \\
&\qquad + \frac{L}{2}(\|w_t - w^*\|^2 - \|w_{t+1} - w^*\|^2) \\
&\overset{(a)}{\geq} -g_t^\top (w_{t+1} - w_t) - \frac{L\gamma_t^2}{2} \\
&\qquad + \frac{L}{2}(w_t - w_{t+1})^\top (w_t + w_{t+1} - 2w^*) \\
&\geq -g_t^\top \left(-\gamma_t e^j\right) - \frac{L\gamma_t^2}{2} \\
&\qquad + \frac{L}{2}\left(\gamma_t e^j\right)^\top \left(2w_t - 2w^* - \gamma_t e^j\right) \\
&\geq \gamma_t g_j - \frac{L\gamma_t^2}{2} + \frac{L}{2}\left(\gamma_t(2w_j - 2w_j^*) - \gamma_t^2\right) \\
&\geq \gamma_t g_j - L\gamma_t^2 + L\gamma_t\left(w_j - w_j^*\right), \qquad (16)
\end{aligned}$$

where (a) follows from Lemma 1.

Let $\mathcal{F}_t$ be the σ-algebra generated by the random coordinate choices up to time t. If we take the conditional expectation of both sides, we will have the following inequality, which is averaged over the choice J_t at time t:

$$\begin{aligned}
&\mathbb{E}\left[\Psi(w_t) - \Psi(w_{t+1}) | \mathcal{F}_t\right] \\
&\geq \sum_{j=1}^{d} \frac{\gamma_t |g_j| g_j}{\|g_t\|_1} - L\gamma_t^2 + L\sum_{j=1}^{d} \frac{\gamma_t |g_j| \left(w_j - w_j^*\right)}{\|g_t\|_1} \\
&\geq \sum_{j=1}^{d} \frac{|g_j|^2}{\alpha L} - \frac{\|g_t\|_1^2}{\alpha^2 L} + \sum_{j=1}^{d} \frac{g_j\left(w_j - w_j^*\right)}{\alpha} \\
&\geq \frac{\|g_t\|_2^2}{\alpha L} - \frac{\|g_t\|_1^2}{\alpha^2 L} + \frac{1}{\alpha} g_t^\top (w_t - w^*). \qquad (17)
\end{aligned}$$

We can see that if $\frac{\|g_t\|_2^2}{\alpha L} - \frac{\|g_t\|_1^2}{\alpha^2 L}$ has a non-negative value, then we will have the following inequality:

$$\mathbb{E}\left[\Psi(w_t) - \Psi(w_{t+1}) | \mathcal{F}_t\right] \geq \frac{1}{\alpha} g_t^\top (w_t - w^*), \qquad (18)$$

meaning that α must satisfy the following condition:

$$\frac{\|g_t\|_2^2}{\alpha L_{M_t}} - \frac{\|g_t\|_1^2}{\alpha^2 L_{M_t}} \geq 0 \Rightarrow \alpha \geq \frac{\|g_t\|_1^2}{\|g_t\|_2^2}. \qquad (19)$$

Since we have the bound $\frac{\|g_t\|_1^2}{\|g_t\|_2^2} \leq d$, it suffices to set $\alpha \geq d$.

By taking the expectation with respect to the entire history up to time t, we have

$$\mathbb{E}\left[\Psi(w_t) - \Psi(w_{t+1})\right] \geq \frac{1}{\alpha} \mathbb{E}\left[g_t^\top (w_t - w^*)\right]. \qquad (20)$$

The convexity of f implies

$$\mathbb{E}\left[f(w_t) - f(w^*)\right] \leq \alpha \mathbb{E}\left[\Psi(w_t) - \Psi(w_{t+1})\right]. \qquad (21)$$

Considering the fact that $f(w_t) - f(w^*)$ is a monotonically non-increasing sequence with respect to t, summing over t gives us

$$\begin{aligned}
T\mathbb{E}\left[f(w_T) - f(w^*)\right] &\leq \mathbb{E}\left[\sum_{t=0}^{T-1} (f(w_t) - f(w^*))\right] \\
&\leq \alpha\left(\Psi(w_0) - \Psi(w_T)\right). \qquad (22)
\end{aligned}$$

Therefore:

$$\begin{aligned}
\mathbb{E}\{F(w_T) - F(w^*)\} &\leq \frac{\alpha\left(\Psi(w_0) - \Psi(w_T)\right)}{T} \\
&\leq \frac{\alpha\left(\Psi(w_0) - \Psi(w*)\right)}{T}. \qquad (23)
\end{aligned}$$

■

B. Distributed optimization for general connected networks

Theorem 2 provides the convergence analysis for the case with deterministic Q^j across time for every coordinate which provides the basis for analyzing our proposed methods proposed in Section III-C where the weight matrices are random. Our analysis relates several average quantities such as the time average

$$\widetilde{w}_T^i = \frac{1}{T}\sum_{t=1}^{T} w_t^i, \qquad (24)$$

the network average

$$\bar{w}_t = \frac{1}{n}\sum_{t=1}^{n} w_t^i \qquad (25)$$

and the time-and-network average

$$\widetilde{w} = \sum_{t=1}^{n} \frac{\widetilde{w}^i}{n} = \sum_{t=1}^{T} \frac{\bar{w}_t}{T}. \qquad (26)$$

Theorem 2. *Consider solving problem (9). Suppose that every node uses update rule (11) with the same step size $\eta_t = \frac{1}{\lambda t}$ across the network and $Q^j(t) = Q^j$ for all $j \in [d]$. Furthermore, assume that our objective functions are λ-strongly convex and have bounded gradients, that is for any vector $w \in \mathcal{R}^d$ we have $\|\nabla f^i(w)\| \leq M$ and $|\nabla_j f^i(w)| \leq M_j$. Then, for $T \geq \max_{j\in[d]}\{-2e\log(\sqrt{\lambda_2(Q^j)})\}$,*

$$\mathbb{E}\left[f_D(\widetilde{w}_T^i) - f_D(w^*)\right] \leq (C_2 + C_3\log(T))\frac{\log(T)}{T},$$

where $C_2 = \frac{M^2}{2n\lambda}$ and $C_3 = \frac{18MM'\sqrt{n}}{\lambda}$ with $M' = \sum_{j=1}^{d} \frac{M_j}{-\log(\sqrt{\lambda_2(Q^j)})}$.

Proof: We take an approach similar to Nedić and Ozdaglar [20]. We first find a bound on the expected distance of the network average (25) from the optimal point w^*. Note that since Q^j is doubly stochastic for all $j \in [d]$, we have the following recursive relation:

$$\bar{w}_{t+1} = \bar{w}_t - \eta_t \sum_{t=1}^{n} \frac{g_t^i}{n}. \qquad (27)$$

This implies the following recursion:

$$\begin{aligned}
&\|\bar{w}_{t+1} - w^*\|^2 \\
&= \left\| \bar{w}_t - w^* - \eta_t \sum_{i=1}^{n} \frac{g_t^i}{n} \right\|^2 \\
&= \|\bar{w}_t - w^*\|^2 + \left\| \eta_t \sum_{i=1}^{n} \frac{g_t^i}{n} \right\|^2 - 2\eta_t (\bar{w}_t - w^*)^\top \sum_{i=1}^{n} \frac{g_t^i}{n} \\
&\overset{(a)}{\leq} \|\bar{w}_t - w^*\|^2 + \eta_t^2 \frac{M^2}{n} - 2\eta_t \sum_{i=1}^{n} \frac{(\bar{w}_t - w^*)^\top g_t^i}{n}, \quad (28)
\end{aligned}$$

where in (a) we made use of the finite form of Jensen's inequality. For the the summands of the third term above we have:

$$\begin{aligned}
&(\bar{w}_t - w^*)^\top g_t^i = (\bar{w}_t - w^*)^\top \nabla f^i(w_t^i) \\
&\quad = (\bar{w}_t - w_t^i)^\top \nabla f^i(w_t^i) + (w_t^i - w^*)^\top \nabla f^i(w_t^i) \\
&\quad \overset{(a)}{\geq} - \left\|\nabla f^i(w_t^i)\right\| \|\bar{w}_t - w^*\| + f^i(w_t^i) - f^i(w^*) \\
&\qquad + \frac{\lambda}{2} \left\| w_t^i - w^* \right\|^2 \\
&\quad = - \left\|\nabla f^i(w_t^i)\right\| \|\bar{w}_t - w^*\| + f^i(w_t^i) - f^i(\bar{w}_t) \\
&\qquad + \frac{\lambda}{2} \left\| w_t^i - w^* \right\|^2 + f^i(\bar{w}_t) - f^i(w^*) \\
&\quad \overset{(b)}{\geq} - \left\|\nabla f^i(w_t^i)\right\| \left\|\bar{w}_t - w_t^i\right\| + \nabla f^i(\bar{w}_t)^\top (w_t^i - \bar{w}_t) \\
&\qquad + \frac{\lambda}{2} \left\| w_t^i - w^* \right\|^2 + f^i(\bar{w}_t) - f^i(w^*) \\
&\quad \overset{(c)}{\geq} - \left(\left\|\nabla f^i(w_t^i)\right\| + \left\|\nabla f^i(\bar{w}_t)\right\|\right) \left\|\bar{w}_t - w_t^i\right\| \\
&\qquad + \frac{\lambda}{2} \left\| w_t^i - w^* \right\|^2 + f^i(\bar{w}_t) - f^i(w^*), \qquad (29)
\end{aligned}$$

where (a) follows from Cauchy-Shwartz inequality and strong convexity of f^i, (b) is a result of convexity of f^i and (c) also results from Cauchy-Shwartz inequality. Using inequality (29) we can upper-bound the third term in the r.h.s of (28):

$$\begin{aligned}
&- 2\eta_t \sum_{i=1}^{n} \frac{\langle (\bar{w}_t - w^*), g_t^i \rangle}{n} \\
&\leq 2\eta_t \sum_{i=1}^{n} \frac{\left(\left\|\nabla f^i(w_t^i)\right\| + \left\|\nabla f^i(\bar{w}_t)\right\|\right) \left\|\bar{w}_t - w_t^i\right\|}{n} \\
&\quad - \lambda\eta_t \sum_{i=1}^{n} \frac{\left\| w_t^i - w^* \right\|^2}{n} - 2\eta_t \sum_{i=1}^{n} \frac{f^i(\bar{w}_t) - f^i(w^*)}{n} \\
&\overset{(a)}{\leq} 2\eta_t \sum_{i=1}^{n} \frac{\left(\left\|\nabla f^i(w_t^i)\right\| + \left\|\nabla f^i(\bar{w}_t)\right\|\right) \left\|\bar{w}_t - w_t^i\right\|}{n} \\
&\quad - \lambda\eta_t \left\|\bar{w}_t - w^*\right\|^2 - 2\eta_t \left(f_D(\bar{w}_t) - f_D(w^*)\right), \qquad (30)
\end{aligned}$$

where (a) results from finite form Jensen's Inequality as well as the definition of f_D. Substituting this result in (28) and taking expectation w.r.t. the entire history up to time t we get

$$\begin{aligned}
&\mathbb{E}\left[\|\bar{w}_{t+1} - w^*\|^2\right] \\
&\leq \mathbb{E}\left[\|\bar{w}_t - w^*\|^2\right] + \eta_t^2 \frac{M^2}{n} \\
&\quad + 2\eta_t \sum_{i=1}^{n} \frac{\mathbb{E}\left[\left(\left\|\nabla f^i(w_t^i)\right\| + \left\|\nabla f^i(\bar{w}_t)\right\|\right) \left\|\bar{w}_t - w_t^i\right\|\right]}{n} \\
&\quad - \lambda\eta_t \mathbb{E}\left[\left\|\bar{w}_t - w^*\right\|^2\right] - 2\eta_t \mathbb{E}\left[f(\bar{w}_t) - f_D(w^*)\right] \\
&\leq (1 - \lambda\eta_t)\, \mathbb{E}\left[\|\bar{w}_t - w^*\|^2\right] + \eta_t^2 \frac{M^2}{n} \\
&\quad + 4\eta_t M \mathbb{E}\left[\left\|\bar{w}_t - w_t^i\right\|\right] - 2\eta_t \mathbb{E}\left[f_D(\bar{w}_t) - f_D(w^*)\right]. \\
&\qquad (31)
\end{aligned}$$

By rearranging the terms we get:

$$\begin{aligned}
&\mathbb{E}\left[f_D(\bar{w}_t) - f_D(w^*)\right] \\
&\leq \frac{1 - \lambda\eta_t}{2\eta_t} \mathbb{E}\left[\|\bar{w}_t - w^*\|^2\right] - \frac{1}{2\eta_t} \mathbb{E}\left[\|\bar{w}_{t+1} - w^*\|^2\right] \\
&\quad + \frac{\eta_t}{2n} M^2 + 2M \mathbb{E}\left[\left\|\bar{w}_t - w_t^i\right\|\right]. \qquad (32)
\end{aligned}$$

The following Lemma provides us with a bound on $\mathbb{E}\left[\left\|\bar{w}_t - w_t^i\right\|\right]$.

Lemma 2. *Suppose all the assumptions in Theorem 2 hold. For any node i at any time t the network average $\bar{w}_t$ in (25) satisfies*

$$\mathbb{E}\left[\left\|\bar{w}_{t+1} - w_{t+1}^i\right\|\right] \leq \frac{2\sqrt{n}}{\lambda} \sum_{j=1}^{d} M_j \frac{\log(2b_j e\, t^2)}{b_j\, t}, \qquad (33)$$

where $b_j = -\log(\sqrt{\lambda_2(Q^j)})$.

Substituting the result of Lemma 2 and $\eta_t = \frac{1}{\lambda t}$ in (32):

$$\begin{aligned}
&\mathbb{E}\left[f_D(\bar{w}_t) - f_D(w^*)\right] \\
&\leq \frac{\lambda(t-1)}{2} \mathbb{E}\left[\|\bar{w}_t - w^*\|^2\right] - \frac{\lambda t}{2} \mathbb{E}\left[\|\bar{w}_{t+1} - w^*\|^2\right] \\
&\quad + \frac{M^2}{2n\lambda t} + \frac{4M\sqrt{n}}{\lambda} \sum_{j=1}^{d} M_j \frac{\log(2b_j e\, t^2)}{b_j\, t}. \qquad (34)
\end{aligned}$$

This provides a bound on $\mathbb{E}\left[f_D(\bar{w}_t) - f_D(w^*)\right]$. However, in order to analyze the asymptotic behavior of our algorithms, we are interested in $\mathbb{E}\left[f_D(\tilde{w}^t) - f_D(w^*)\right]$. Consider the time-and-network average (26). From convexity of f_D and Jensen's inequality we have:

$$\begin{aligned}
&\mathbb{E}\left[f_D(\tilde{w}_T) - f_D(w^*)\right] \\
&\leq \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\left[f_D(\bar{w}_t) - f_D(w^*)\right] \\
&\leq \sum_{t=1}^{T} \frac{\lambda(t-1)\mathbb{E}\left[\|\bar{w}_t - w^*\|^2\right] - \lambda t \mathbb{E}\left[\|\bar{w}_{t+1} - w^*\|^2\right]}{2T} \\
&\quad + \frac{1}{T} \sum_{t=1}^{T} \frac{M^2}{2n\lambda t} + \frac{1}{T} \sum_{t=1}^{T} \frac{4M\sqrt{n}}{\lambda} \sum_{j=1}^{d} M_j \frac{\log(2b_j e\, t^2)}{b_j\, t}. \\
&\qquad (35)
\end{aligned}$$

Using convexity, Jensen's inequality, and some algebra we can establish that for $T > \max_{j\in[d]}\{2b_j e\}$,

$$\begin{aligned}
&\mathbb{E}\left[f_D(\widetilde{w}_T) - f_D(w^*)\right] \\
&\leq \frac{1}{T}\sum_{t=1} -\frac{\lambda T}{2}\mathbb{E}\left[\|\bar{w}_{T+1} - w^*\|^2\right] \\
&\quad + \frac{1}{T}\sum_{t=1}^{T}\left(\frac{M^2}{2n\lambda} + \frac{12MM'\sqrt{n}}{\lambda}\log(T)\right)\frac{1}{t} \\
&\leq \frac{C_T}{T}\sum_{t=1}^{T}\frac{1}{t} \\
&\leq C_T\frac{\log(T)}{T}
\end{aligned} \tag{36}$$

where $M' = \sum_{j=1}^{d}\frac{M_j}{b_j}$ and $C_T = \frac{M^2}{2n\lambda} + \frac{12MM'\sqrt{n}}{\lambda}\log(T)$. This lets us relate the time average $\widetilde{w}_T^i$ at a node to the network time average:

$$\begin{aligned}
&\mathbb{E}\left[f_D(\widetilde{w}_T^i) - f_D(w^*)\right] \\
&\leq \mathbb{E}\left[f_D(\widetilde{w}_T) - f_D(w^*) + \nabla f_D(\widetilde{w}_t^i)^\top(\widetilde{w}_t^i - \widetilde{w}_t)\right] \\
&\overset{(a)}{\leq} \mathbb{E}\left[f_D(\widetilde{w}_T) - f_D(w^*) + \|\nabla f_D(\widetilde{w}_T^i)\|\,\|\widetilde{w}_t^i - \widetilde{w}_T\|\right] \\
&\overset{(b)}{\leq} \mathbb{E}\left[f_D(\widetilde{w}_T) - f_D(w^*)\right] \\
&\qquad + \mathbb{E}\left[\|\nabla f_D(\widetilde{w}_T^i)\|\sum_{t=1}^{T}\frac{\|w_t^i - \bar{w}_t\|}{T}\right] \\
&\leq \mathbb{E}\left[f_D(\widetilde{w}_T) - f_D(w^*)\right] + \frac{M}{T}\sum_{t=1}^{T}\mathbb{E}\left[\|w_t^i - \bar{w}_t\|\right].
\end{aligned} \tag{37}$$

where in the inequality (a) we used Cauchy-Shwartz and the (b) results from Jensen's inequality and the definition of $\widetilde{w}_T^i$ and $\widetilde{w}_T$.

Combining the results from (37), (36), and Lemma 2 gives us the desired upper bound on the loss:

$$\mathbb{E}\left[f_D(\widetilde{w}_T^i) - f_D(w^*)\right] \tag{38}$$

$$\leq C_T\frac{\log(T)}{T} + \frac{2M\sqrt{n}}{\lambda T}\sum_{t=1}^{T}\sum_{j=1}^{d}\frac{M_j\log(2b_j e\, t^2)}{b_j\, t} \tag{39}$$

$$\begin{aligned}
&\leq C_T\frac{\log(T)}{T} + \frac{6MM'\sqrt{n}\log(T)}{\lambda T}\sum_{t=1}^{T}\frac{1}{t} \\
&\leq \left(\frac{M^2}{2n\lambda} + \frac{12MM'\sqrt{n}}{\lambda}\log(T)\right)\frac{\log(T)}{T} \\
&\quad + \frac{6MM'\sqrt{n}\log(T)}{\lambda}\frac{\log(T)}{T} \\
&= \left(\frac{M^2}{2n\lambda} + \frac{18MM'\sqrt{n}}{\lambda}\log(T)\right)\frac{\log(T)}{T}.
\end{aligned} \tag{40}$$

■

Now that we have established the results for the case with time-invariant weight matrices, we go on to study the case where $Q^j(t)$ is selected i.i.d. from distribution $\mathcal{P}_j^{\mathcal{G}}$. Note that for every coordinate j, the quantity $\mathcal{P}_j^{\mathcal{G}}$ is a function of $\{\mathcal{P}_i^C\}_{i=1}^n$ that are probability distributions over coordinate indexes at each node. The following lemma is used in the sequel to analyze the behavior of our methods.

Lemma 3. *Suppose the same assumptions as theorem* (2) *hold except that in* (11)*, $Q^j(t)$ is an i.i.d sequence of doubly stochastic matrices drawn from distribution $\mathcal{P}_j^{\mathcal{G}}$. Then, for w_t^i and $\bar{w}_t$ we have that:*

$$\mathbb{E}\left[\|\bar{w}_{t+1} - w_{t+1}^i\|\right] \leq \sum_{j=1}^{d}\frac{2M_j\sqrt{n}}{\lambda}\frac{\log(2b_j e\, t^2)}{b_j\, t}, \tag{41}$$

where $b_j = -\log\left(\mathbb{E}\left[\sigma_2\left(Q^j(t)\right)\right]\right)$ and $\sigma_2\left(Q^j\right)$ is the second largest singular value of $Q^j(t)$.

Now, we are ready to find the upper bound on $\mathbb{E}\left[f_D(\widetilde{w}_T^i) - f_D(w^*)\right]$ in the setup with random weight matrices.

Theorem 3. *Assume that all conditions in Lemma* (3) *hold. Also, we have the following upper bound on the loss of algorithm* (11) *for the optimization problem* (9) *if $T \geq -2\, e\log\left(\mathbb{E}\left[\sigma_2\left(Q^j(t)\right)\right]\right)$:*

$$\begin{aligned}
&\mathbb{E}\left[f_D(\widetilde{w}_T^i) - f_D(w^*)\right] \\
&\leq \left(\frac{M^2}{2n\lambda} + \frac{18Mc\sqrt{n}}{\lambda}\log(T)\right)\frac{\log(T)}{T},
\end{aligned} \tag{42}$$

where $c = \sum_{j=1}^{d}\frac{M_j}{-\log(\mathbb{E}[\sigma_2(Q^j(t))])}$ and M and M_j are as defined in theorem (2)*.*

Proof: By applying Lemma 3 to (32) and following the same procedure as that of the proof of Theorem 2, we get the stated result. ■

We remark that for both distributed algorithms suggested in Section III-C, the analysis in Theorem 3 holds. The synchronous algorithm is a special case where Q^j is randomly a stochastic matrix which takes value from $\{Q, I\}$. In the asynchronous method the sample space is larger, i.e. Q^j, which is defined in subsection III-E. Our analysis, with minor changes, extends to the case when each node updates its belief using unbiased estimates of the local gradient instead of the full local gradient.

V. Conclusion

In this paper, we used social sampling to limit the communication in cooperative multi-agent optimization settings. For centralized and shared memory systems, we proposed a new nonuniform variant of stochastic coordinate descent and provided upper bounds on the expected sub-optimality gap. This method requires full knowledge of local gradient vectors, which seems computationally wasteful. However, this method may be useful for shared memory systems with limited communication resources where computing local gradient vectors by each node is inexpensive and we are more concerned about the amount of communication or contention among nodes rather than the computation cost.

We note that in distributed models sharing gradient information is not necessarily beneficial; this suggests that nodes should share samples of their current estimates $\{w_t^i\}$. We proposed a stochastic coordinate-wise consensus-based optimization method that requires nodes to share random coordinates of their estimates with their neighbors. We provided convergence analysis and explicit error bounds in expectation for this method.

An interesting question raised in the centralized model is that how using "stale" gradient values from previous iterations would affect the convergence rate of the algorithm. Less frequent full gradient evaluation drastically reduces the computational cost of the algorithm, therefore a *delayed* PSCD method might solve the intrinsic issue of PSCD that it requires evaluation of the full gradient at every iteration. Analyzing such a scheme would build on recent results of Scmidt et al. [22]. Finally, an empirical evaluation of our methods on typical objective functions, especially in machine learning, may shed more light on when nonuniform sampling can help in practice.

Acknowledgment

The authors would like to thank Tara Javidi and Avleen Bijral for some helpful discussions.

References

[1] A. D. Sarwate and T. Javidi, "Distributed learning of distributions via social sampling," *IEEE Transactions on Automatic Control*, vol. 60, no. 1, pp. 34–45, 2015. [Online]. Available: http://dx.doi.org/10.1109/TAC.2014.2329611

[2] K.-W. Chang, C.-J. Hsieh, and C.-J. Lin, "Coordinate descent method for large-scale L2-loss linear support vector machines," *Journal of Machine Learning Research*, vol. 9, pp. 1369–1398, 2008. [Online]. Available: http://www.jmlr.org/papers/v9/chang08a.html

[3] S. Shalev-Shwartz and A. Tewari, "Stochastic methods for ℓ_1-regularized loss minimization," *Journal of Machine Learning Research*, vol. 12, pp. 1865–1892, 2011. [Online]. Available: http://www.jmlr.org/papers/v12/shalev-shwartz11a.html

[4] A. Nedić, "Asynchronous broadcast-based convex optimization over a network," *IEEE Transactions on Automatic Control*, vol. 56, no. 6, pp. 1337–1351, 2011. [Online]. Available: http://dx.doi.org/10.1109/TAC.2010.2079650

[5] I. Lobel and A. Ozdaglar, "Distributed subgradient methods for convex optimization over random networks," *IEEE Transactions on Automatic Control*, vol. 56, no. 6, pp. 1291–1306, 2011. [Online]. Available: http://dx.doi.org/10.1109/TAC.2010.2091295

[6] K. Srivastava and A. Nedić, "Distributed asynchronous constrained stochastic optimization," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 4, pp. 772–790, 2011. [Online]. Available: http://dx.doi.org/10.1109/JSTSP.2011.2118740

[7] S. S. Ram, A. Nedić, and V. V. Veeravalli, "Distributed stochastic subgradient projection algorithms for convex optimization," *Journal of Optimization Theory and Applications*, vol. 147, no. 3, pp. 516–545, 2010. [Online]. Available: http://dx.doi.org/10.1007/s10957-010-9737-7

[8] A. Nedić, A. Ozdaglar, and P. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Transactions on Automatic Control*, vol. 55, no. 4, pp. 922–938, 2010. [Online]. Available: http://dx.doi.org/10.1109/TAC.2010.2041686

[9] J. N. Tsitsiklis, D. P. Bertsekas, M. Athans *et al.*, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Transactions on Automatic Control*, vol. 31, no. 9, pp. 803–812, 1986. [Online]. Available: http://dx.doi.org/10.1109/TAC.1986.1104412

[10] V. S. Borkar, "Asynchronous stochastic approximations," *SIAM Journal on Control and Optimization*, vol. 36, no. 3, pp. 840–851, 1998. [Online]. Available: http://dx.doi.org/10.1137/S0363012995282784

[11] D. Jakovetić, J. Xavier, and J. M. F. Moura, "Fast distributed gradient methods," *IEEE Transactions on Automatic Control*, vol. 59, no. 5, pp. 1131–1146, 2014. [Online]. Available: http://dx.doi.org/10.1109/TAC.2014.2298712

[12] Y. Nesterov, "Efficiency of coordinate descent methods on huge-scale optimization problems read more: http://epubs.siam.org/doi/abs/10.1137/100802001," *SIAM Journal on Optimization*, vol. 22, no. 2, pp. 341–362, 2012. [Online]. Available: http://dx.doi.org/10.1137/100802001

[13] P. Richtárik and M. Takáč, "On optimal probabilities in stochastic coordinate descent methods," ArXiV, Tech. Rep. arXiv:1310.3438 [stat.ML], October 2013. [Online]. Available: http://arxiv.org/abs/1310.3438

[14] Z. Qu, P. Richtárik, and T. Zhang, "Randomized dual coordinate ascent with arbitrary sampling," ArXiV, Tech. Rep. arXiv:1411.5873 [math.OC], November 2014. [Online]. Available: http://arxiv.org/abs/1411.5873

[15] Z. Qu and P. Richtárik, "Coordinate descent with arbitrary sampling I: Algorithms and complexity," ArXiV, Tech. Rep. arXiv:1412.8060 [math.OC], December 2014. [Online]. Available: http://arxiv.org/abs/1412.8060

[16] S. J. Wright, "Coordinate descent algorithms," ArXiV, Tech. Rep. arXiv:1502.04759 [math.OC], February 2015. [Online]. Available: http://arxiv.org/abs/1502.04759

[17] B. Recht, C. Ré, S. Wright, and F. Niu, "HOGWILD!: a lock-free approach to parallelizing stochastic gradient descent." in *Advances in Neural Information Processing Systems 24*, J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, Eds., 2011, pp. 693–701.

[18] J. Liu, S. J. Wright, C. Re, V. Bittorf, and S. Sridhar, "An asynchronous parallel stochastic coordinate descent algorithm," in *Proceedings of the 31st International Conference on Machine Learning*, ser. JMLR Workshop and Conference Proceedings, L. Getoor and T. Scheffer, Eds., vol. 32, 2014. [Online]. Available: http://jmlr.org/proceedings/papers/v32/liud14.pdf

[19] A. Nedić, D. P. Bertsekas, and V. S. Borkar, "Distributed asynchronous incremental subgradient methods," in *Inherently Parallel Algorithms in Feasibility and Optimization and Their Applications*, ser. Studies in Computational Mathematics, D. Butnariu, Y. Censor, and S. Reich, Eds. Elsevier, 2001, pp. 381–407.

[20] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, January 2009. [Online]. Available: http://dx.doi.org/10.1109/TAC.2008.2009515

[21] Y. Nesterov, *Introductory lectures on convex optimization*, vol. 87. [Online]. Available: http://www.springer.com/us/book/9781402075537

[22] M. Schmidt, N. L. Roux, and F. Bach, "Minimizing finite sums with the stochastic average gradient," *arXiv preprint arXiv:1309.2388*. [Online]. Available: http://arxiv.org/abs/1309.2388