

Stochastic Gradient Descent Algorithms for Resource Allocation

Amrit Singh Bedi

Supervisor:
Dr. Ketan Rajawat

Department of Electrical Engineering,
Indian Institute of Technology, Kanpur
Kanpur, Uttar Pradesh



Outline

- Gradient descent algorithm



Outline

- Gradient descent algorithm
- Subgradient descent algorithm



Outline

- Gradient descent algorithm
- Subgradient descent algorithm
- Stochastic subgradient algorithm



Outline

- Gradient descent algorithm
- Subgradient descent algorithm
- Stochastic subgradient algorithm
- Incremental Stochastic subgradient algorithm



Outline

- Gradient descent algorithm
- Subgradient descent algorithm
- Stochastic subgradient algorithm
- Incremental Stochastic subgradient algorithm
- Ergodic stochastic algorithm



Outline

- Gradient descent algorithm
- Subgradient descent algorithm
- Stochastic subgradient algorithm
- Incremental Stochastic subgradient algorithm
- Ergodic stochastic algorithm
- Applications in wireless communication, smart grid.



Outline

- Gradient descent algorithm
- Subgradient descent algorithm
- Stochastic subgradient algorithm
- Incremental Stochastic subgradient algorithm
- Ergodic stochastic algorithm
- Applications in wireless communication, smart grid.
- Future work



Introduction

Standard convex optimization problem

$$\text{minimize } f(\mathbf{x}) \tag{1}$$

$$\text{subject to } g(\mathbf{x}) \leq 0 \tag{2}$$

$$\mathbf{x} \in \mathcal{X}$$



Introduction

Standard convex optimization problem

$$\text{minimize } f(\mathbf{x}) \tag{1}$$

$$\text{subject to } g(\mathbf{x}) \leq 0 \tag{2}$$

$$\mathbf{x} \in \mathcal{X}$$

- \mathbf{x} is the optimization variable



Introduction

Standard convex optimization problem

$$\text{minimize } f(\mathbf{x}) \tag{1}$$

$$\text{subject to } g(\mathbf{x}) \leq 0 \tag{2}$$

$$\mathbf{x} \in \mathcal{X}$$

- \mathbf{x} is the optimization variable
- $f(\mathbf{x})$ is the objective function



First-order methods



First-order methods

- Only the first order derivative is required



First-order methods

- Only the first order derivative is required
- Every iteration is inexpensive, does not require second derivative



First-order methods

- Only the first order derivative is required
- Every iteration is inexpensive, does not require second derivative
- Useful for large scale optimization problems



First-order methods

- Only the first order derivative is required
- Every iteration is inexpensive, does not require second derivative
- Useful for large scale optimization problems
- Can be easily extended to include uncertainty cases



First-order methods

- Only the first order derivative is required
- Every iteration is inexpensive, does not require second derivative
- Useful for large scale optimization problems
- Can be easily extended to include uncertainty cases
- Useful to take optimal decisions on-the-fly



Gradient Descent Algorithm [1]

- **Motivation:** Very useful for large scale problems, much faster



- Convergence properties of the algorithm is governed by $\epsilon(t)$
- Too small values of $\epsilon(t)$ will cause the algorithm to converge slowly
- Too large values could cause the algorithm to overshoot and diverge

A simple convergence analysis for constant step size is discussed here:

Definition: A function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is called L-Lipschitz continuous gradient if and only if

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\|_2 \leq L\|\mathbf{x} - \mathbf{y}\|_2, \quad \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n \quad (5)$$



- Convergence properties of the algorithm is governed by $\epsilon(t)$
- Too small values of $\epsilon(t)$ will cause the algorithm to converge slowly
- Too large values could cause the algorithm to overshoot and diverge

A simple convergence analysis for constant step size is discussed here:

Definition: A function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is called L-Lipschitz continuous gradient if and only if

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\|_2 \leq L\|\mathbf{x} - \mathbf{y}\|_2, \quad \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n \quad (5)$$

Implications:

- Lipschitz continuous gradient, denoted as $f \in C_L$
- Speed at which gradient varies is bounded
- Objective function has bounded curvature



Lemma

Let $f \in C_L$, then the following upper bound holds

$$f(\mathbf{y}) \leq f(\mathbf{x}) + \nabla f(\mathbf{x})^T (\mathbf{y} - \mathbf{x}) + \frac{L}{2} \|\mathbf{y} - \mathbf{x}\|_2^2 \quad (6)$$



Proof: Using the lemma, put $\mathbf{y} = \mathbf{x}^{(t+1)}$, we get

$$\begin{aligned}
 f(\mathbf{x}^{(t+1)}) &\leq f(\mathbf{x}^{(t)}) + \nabla f(\mathbf{x}^{(t)})^T (\mathbf{x}^{(t+1)} - \mathbf{x}^{(t)}) + \frac{L}{2} \|\mathbf{x}^{(t+1)} - \mathbf{x}^{(t)}\|^2 \\
 &= f(\mathbf{x}^{(t)}) - \epsilon \|\nabla f(\mathbf{x}^{(t)})\|^2 + \frac{\epsilon^2 L}{2} \|\nabla f(\mathbf{x}^{(t)})\|^2 \\
 &= f(\mathbf{x}^{(t)}) - \epsilon \left(1 - \frac{\epsilon}{2} L\right) \|\nabla f(\mathbf{x}^{(t)})\|^2
 \end{aligned}$$

$$\Rightarrow \|\nabla f(\mathbf{x}^{(t)})\|^2 \leq \frac{1}{\epsilon(1 - \frac{\epsilon}{2} L)} \left(f(\mathbf{x}^{(t)}) - f(\mathbf{x}^{(t+1)}) \right) \quad (8)$$

$$\sum_{t=0}^{T-1} \|\nabla f(\mathbf{x}^{(t)})\|^2 \leq \frac{1}{\epsilon(1 - \frac{\epsilon}{2} L)} \left(f(\mathbf{x}^{(0)}) - f(\mathbf{x}^{(T)}) \right) \quad (9)$$

$$\leq \frac{1}{\epsilon(1 - \frac{\epsilon}{2} L)} \left(f(\mathbf{x}^{(0)}) - f^* \right) \quad (10)$$



Proof: Using the lemma, put $\mathbf{y} = \mathbf{x}^{(t+1)}$, we get

$$\begin{aligned} f(\mathbf{x}^{(t+1)}) &\leq f(\mathbf{x}^{(t)}) + \nabla f(\mathbf{x}^{(t)})^T (\mathbf{x}^{(t+1)} - \mathbf{x}^{(t)}) + \frac{L}{2} \|\mathbf{x}^{(t+1)} - \mathbf{x}^{(t)}\|^2 \\ &= f(\mathbf{x}^{(t)}) - \epsilon \|\nabla f(\mathbf{x}^{(t)})\|^2 + \frac{\epsilon^2 L}{2} \|\nabla f(\mathbf{x}^{(t)})\|^2 \\ &= f(\mathbf{x}^{(t)}) - \epsilon \left(1 - \frac{\epsilon}{2} L\right) \|\nabla f(\mathbf{x}^{(t)})\|^2 \end{aligned}$$

$$\Rightarrow \|\nabla f(\mathbf{x}^{(t)})\|^2 \leq \frac{1}{\epsilon(1 - \frac{\epsilon}{2} L)} \left(f(\mathbf{x}^{(t)}) - f(\mathbf{x}^{(t+1)}) \right) \quad (8)$$

$$\sum_{t=0}^{T-1} \|\nabla f(\mathbf{x}^{(t)})\|^2 \leq \frac{1}{\epsilon(1 - \frac{\epsilon}{2} L)} \left(f(\mathbf{x}^{(0)}) - f(\mathbf{x}^{(T)}) \right) \quad (9)$$

$$\leq \frac{1}{\epsilon(1 - \frac{\epsilon}{2} L)} \left(f(\mathbf{x}^{(0)}) - f^* \right) \quad (10)$$

Since $f^* > -\infty$, therefore, as $T \rightarrow \infty$, the LHS must converge

$$\|\nabla f(\mathbf{x}^{(t)})\|^2 \rightarrow 0, \quad \text{as } t \rightarrow \infty. \quad (11)$$



Bound on $\|\mathbf{x} - \mathbf{x}^*\|_2$

In a similar way, if the function f is assumed to be strongly convex, $\nabla^2 f(\mathbf{x}) \geq mI$, then we could bound the term $\|\mathbf{x} - \mathbf{x}^*\|_2$ as follows:

$$\|\mathbf{x} - \mathbf{x}^*\|_2 \leq \frac{2}{m} \|\nabla f(\mathbf{x})\|_2^2 \quad (12)$$

which will follow from the inequality

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})^T (\mathbf{y} - \mathbf{x}) + \frac{m}{2} \|\mathbf{y} - \mathbf{x}\|_2^2 \quad (13)$$

by substituting $\mathbf{y} = \mathbf{x}^*$.



Convergence rate

- Smoothness of the objective controls the convergence rate of gradient based methods

Convex objective $f(\mathbf{x})$	Iterations. . .
Nondifferentiable	$O(1/\epsilon^2)$
differentiable	$O(1/\epsilon)$
Smooth (Lipschitz gradient)	$O(1/\sqrt{\epsilon})$
Strongly convex	$O(\log(1/\epsilon))$



Contrast with Newton method



- Newton's method requires fewer iterations, but each one is slow (we need to compute 2nd derivatives too)



Contrast with Newton method

In general, if we minimize a n dimensional objective function

- Gradient descent requires more iterations, but each one is fast (we only need to compute 1st derivatives)
- Newton's method requires fewer iterations, but each one is slow (we need to compute 2nd derivatives too)

Recent result

- Accelerated method are proposed in [2]
- An $O(1/k)$ Gradient Method for Network Resource Allocation is proposed in [3]

[2] Tseng P. On accelerated proximal gradient methods for convex-concave optimization. 2008. Submitted to SIAM J. Optim. 2009.

[3]Beck A, Nedic A, Ozdaglar A, Teboulle M. An Gradient Method for Network Resource Allocation Problems. IEEE TCNS. 2014.



Subgradient Methods

Subgradient method [4] is a simple algorithm for minimizing the non-differential convex function.



Subgradient Methods

Subgradient method [4] is a simple algorithm for minimizing the non-differential convex function.

- Applies directly to non-differential objective functions



Subgradient Methods

Subgradient method [4] is a simple algorithm for minimizing the non-differential convex function.

- Applies directly to non-differential objective functions
- In contrast to gradient method, function value may increase



Subgradient Methods

Subgradient method [4] is a simple algorithm for minimizing the non-differential convex function.

- Applies directly to non-differential objective functions
- In contrast to gradient method, function value may increase

Definition: Vector \mathbf{g} is the **subgradient** of $f(\cdot)$ at \mathbf{x} , if

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \mathbf{g}^T(\mathbf{y} - \mathbf{x}) \quad \text{for all } \mathbf{y} \quad (14)$$

[4]Boyd S, Mutapcic A. Subgradient methods. Lecture notes of EE364b, Stanford University, Winter Quarter. 2006;2007.



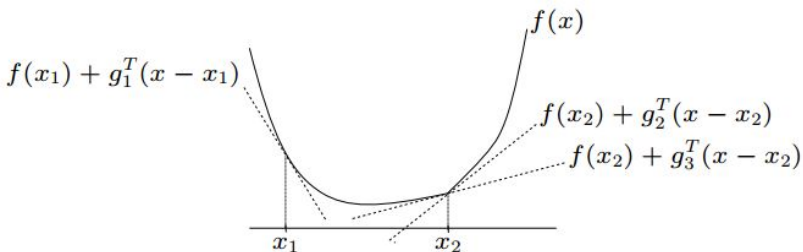


Figure: Example for one dimensional setting

$\mathbf{g}_1, \mathbf{g}_2$ and \mathbf{g}_3 are the subgradients at \mathbf{x}_1 , and \mathbf{x}_2 .

Algorithm

- For **unconstrained** convex problems, the algorithm is

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \epsilon^{(t)} \mathbf{g}^{(t)} \quad (15)$$

Here, $\mathbf{g}^{(t)}$ is any subgradient of f at $\mathbf{x}^{(t)}$ and $\epsilon^{(t)} > 0$ is the step size.



Algorithm

- For **unconstrained** convex problems, the algorithm is

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \epsilon^{(t)} \mathbf{g}^{(t)} \quad (15)$$

Here, $\mathbf{g}^{(t)}$ is any subgradient of f at $\mathbf{x}^{(t)}$ and $\epsilon^{(t)} > 0$ is the step size.

- Since, it is not a descent method, it is common to keep track of the best point found so far given by

$$f_{best}^t := \min\{f(\mathbf{x}^{(1)}), f(\mathbf{x}^{(2)}), \dots, f(\mathbf{x}^{(t)})\} \quad (16)$$





Convergence Analysis

Consider the Euclidean distance to the optimal point, we have

$$\|\mathbf{x}^{(t+1)} - \mathbf{x}^*\|_2^2 = \|\mathbf{x}^{(t)} - \epsilon \mathbf{g}^{(t)} - \mathbf{x}^*\|_2^2 \quad (18)$$

$$\leq \|\mathbf{x}^{(t)} - \mathbf{x}^*\|_2^2 - 2\epsilon \left(f(\mathbf{x}^{(t)}) - f^* \right) + \epsilon^2 G^2 \quad (19)$$

summation over t yields

$$\|\mathbf{x}^{(t+1)} - \mathbf{x}^*\|_2^2 \leq \|\mathbf{x}^{(1)} - \mathbf{x}^*\|_2^2 - 2\epsilon \sum_{i=1}^t \left(f(\mathbf{x}^{(i)}) - f^* \right) + \epsilon^2 T G^2 \quad (20)$$

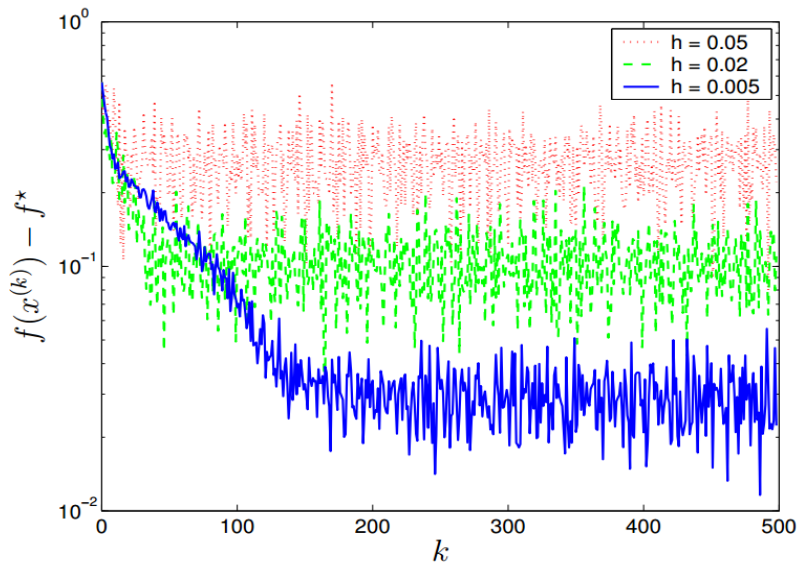
\Rightarrow

$$2\epsilon \sum_{i=1}^t \left(f(\mathbf{x}^{(i)}) - f^* \right) \leq R^2 + \epsilon^2 T G^2 \quad (21)$$

$$f_{best}^{(t)} - f^* \leq \frac{R^2 + \epsilon^2 T G^2}{2\epsilon T} \quad (22)$$



A simple example[5]



Projected subgradient method

- Consider the constrained optimization problem

$$\text{minimize } f(\mathbf{x}) \quad (23)$$

$$\text{subject to } \mathbf{x} \in \mathcal{X} \quad (24)$$

- **Projected subgradient algorithm** [4] for this problem is

$$\mathbf{x}^{(t+1)} = \mathcal{P}_{\mathcal{X}} \left[\mathbf{x}^{(t)} - \epsilon^{(t)} \mathbf{g}^{(t)} \right] \quad (25)$$

[4] Boyd S, Mutapcic A. Subgradient methods. Lecture notes of EE364b, Stanford University, Winter Quarter. 2006;2007.



- Simple to implement and can be applied to variety of problems
- Subgradient methods were first introduced in the middle sixties by N. Z. Shor [6]
- Extensive treatment of these subgradient methods are provided in books [7, 17]
- Nemirovski & Yudin in [8] derived the worst-case complexity bound to achieve an ϵ -solution
 - where it is $O(\frac{1}{\epsilon^2})$ for Lipschitz continuous nonsmooth problems
 - and $O(\frac{1}{\sqrt{\epsilon}})$ for smooth problems with Lipschitz continuous gradient
- Mixture with primal or dual decomposition techniques, sometimes provides simple distributed algorithm [9]

[6] Shor NZ. Minimization Methods for Non-differentiable Functions. Springer, 1985.

[7] Shor NZ. Nondifferentiable Optimization and Polynomial Problems. Springer Science & Business Media; 1998.

[8]Blair C. Problem Complexity and Method Efficiency in Optimization (AS Nemirovsky and DB Yudin). SIAM Review. 1985.

[9] Palomar DP, Chiang M. A tutorial on decomposition methods for network utility maximization. IEEE JSAC. 2006.



20/65

Motivation

Motivation for solving dual problem:



Motivation

Motivation for solving dual problem:

- The dual is a convex optimization problem



Motivation

Motivation for solving dual problem:

- The dual is a convex optimization problem
- Dual may have smaller dimensions than primal



Motivation

Motivation for solving dual problem:

- The dual is a convex optimization problem
- Dual may have smaller dimensions than primal
- If duality gap is zero, primal optimal can be derived from dual



Recent results and scope



Recent results and scope

- Recent convergence results in this direction are discussed in [12]
 - averaging scheme applied
 - Provides convergence rate estimates for the approximate solutions (Slater's qualification)
 - Amount of feasibility violation
 - Provides upper and lower bound for primal objective function
- Accelerated Dual Descent for Network Flow Optimization [13]

[12] Nedic A, Ozdaglar A. Approximate primal solutions and rate analysis for dual subgradient methods. *SIAM Journal on Opt.* 2009

[13] Zargham M, Ribeiro A, Ozdaglar A, Jadbabaie A. Accelerated dual descent for network flow optimization. IEEE TAC. 2014.



Standard Convex Stochastic Optimization Problem

$$\text{minimize } \mathbb{E}[f_0(x, w)] \quad (33)$$

$$\text{subject to } \mathbb{E}[f_i(x, w)] \leq 0, \quad i = 1, 2, \dots, m. \quad (34)$$

$$x \in \mathcal{X} \quad (35)$$



History and Motivation

- History of stochastic algorithms way back to adaptive filtering algorithm by Robbins & Monro [14] and Widrow & Stearns [15]
- Extensively studied in the context of LMS and RLS algorithms [16]
- Stochastic subgradient methods with detailed analysis are discussed in [18, 19]

[14]Robbins H, Monro S. A stochastic approximation method. AMS. Sep. 1951.

[15] Widrow B, Stearns SD. Adaptive signal processing. Englewood Cliffs, NJ, Prentice-Hall, Inc., 1985.

[16]Sayed AH. Adaptive filters. John Wiley & Sons; Oct. 2011.

[18]Ermoliev Y. stochastic quasigradient methods and their application to system optimization. Stochastics: An IJPSP. 1983



Different applications

Stochastic gradient, subgradient methods have been widely applied to

- Neural networks [20],
- Parameter tracking [21],
- Large scale machine learning [22, 23],
- and Resource allocation problems [24, 25, 27, 28, 34].

[20]Bottou L. Stochastic gradient learning in neural networks. Proceedings of Neuro-Nmes. 1991.

[21]Kushner HJ, Yang J. Analysis of adaptive step size SA algorithms for parameter tracking. , Proc. of IEEE Conference on DoC, 1994.

[22]Bottou L. Large-scale machine learning with stochastic gradient descent. In Proc. of COMPSTAT'2010.

[24]Alaei S, Hajiaghayi M, Liaghat V. The online stochastic generalized assignment problem. In Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques 2013.

[27]Wang X, Gao N. Stochastic resource allocation over fading multiple access and broadcast channels. IEEE TIT, 2010.



Relation with LMS algorithm

- The goal is to minimize $\mathbb{E} [|e(t)|^2]$



Relation with LMS algorithm

- The goal is to minimize $\mathbb{E} [|e(t)|^2]$
- Utilizing the steepest descent we get

$$h(t+1) = h(t) + \mu \mathbb{E} [u(t)e(t)] \quad (36)$$

when statistics are not known, then using the approximation for average term, we get LMS

$$h(t+1) = h(t) + \mu u(t)e(t) \quad (37)$$



Stochastic Subgradient Methods



Stochastic Subgradient Methods

- Stochastic subgradient algorithms [29] are generalization of gradient one



Stochastic Subgradient Methods

- Stochastic subgradient algorithms [29] are generalization of gradient one
- Uses noisy subgradient and more limited set of step size rules



Algorithm

- For **unconstrained** minimization of a convex function $f : \mathbb{R}^n \rightarrow \mathbb{R}$, the stochastic subgradient update is given as

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \epsilon^{(t)} \tilde{\mathbf{g}}^{(t)} \quad (40)$$

where, $\epsilon^{(t)} > 0$ is the t^{th} step size, and $\tilde{\mathbf{g}}^{(t)}$ is stochastic subgradient.

$$\mathbb{E} [\tilde{\mathbf{g}}^{(t)} | \mathbf{x}^{(t)}] = \mathbf{g}^{(t)} \in \partial f(\mathbf{x}^{(t)}) \quad (41)$$

- In this algorithm, similar to subgradient one,

$$f_{best}^t := \min\{f(\mathbf{x}^{(1)}), f(\mathbf{x}^{(2)}), \dots, f(\mathbf{x}^{(t)})\} \quad (42)$$



Convergence analysis

Assumptions:

- There exist a minimizer of f , say \mathbf{x}^* .
- There exist G such that, $\mathbb{E}\|\mathbf{g}^{(t)}\|_2 \leq G$ for all t .
- There exist R such that $\|\mathbf{x}^{(1)} - \mathbf{x}^*\| \leq R^2$.

Results:

- Convergence in Expectation

$$\mathbb{E}\{f_{best}^{(t)}\} \rightarrow f^* \text{ as } t \rightarrow \infty. \quad (43)$$

- Convergence in Probability

$$\lim_{t \rightarrow \infty} \mathbf{Prob}\left(f_{best}^{(t)} \geq f^* + \alpha\right) = 0 \text{ for any } \alpha > 0. \quad (44)$$



- Taking the summation over $t = 1, 2, \dots, T$, we get

$$\mathbb{E}\{f_{best}^{(t)}\} = \mathbb{E}\{\min_{i=1,\dots,T} f(\mathbf{x}^{(i)})\} \leq \frac{R^2 + \epsilon^2 T G^2}{2\epsilon T} \quad (46)$$



- For convergence in probability, use the Markov's inequality

$$\text{Prob}\left(f_{best}^{(t)} - f^* \geq \alpha\right) \leq \frac{\mathbb{E}(f_{best}^{(t)} - f^*)}{\alpha} \text{ for any } \alpha > 0. \quad (47)$$

The RHS goes to zero as $t \rightarrow \infty$, so the LHS as well.



- A problem of recent interest in **distributed networks** is
 - To design **decentralized** algorithms to minimize a sum of functions
 - With each component function is known only to a particular agent



Incremental Stochastic SubGradient Algorithms [30]

- A problem of recent interest in **distributed networks** is
 - To design **decentralized** algorithms to minimize a sum of functions
 - With each component function is known only to a particular agent
- Consider a network of m agents, indexed by $i = 1, 2, \dots, m$. The aim is to solve the following optimization problem:

$$\begin{aligned}
 &\text{minimize } f(\mathbf{x}) = \sum_{i=1}^m f_i(\mathbf{x}) \\
 &\text{subject to } \mathbf{x} \in \mathcal{X}
 \end{aligned} \tag{48}$$



Incremental Stochastic SubGradient Algorithms [30]

- A problem of recent interest in **distributed networks** is
 - To design **decentralized** algorithms to minimize a sum of functions
 - With each component function is known only to a particular agent
- Consider a network of m agents, indexed by $i = 1, 2, \dots, m$. The aim is to solve the following optimization problem:

$$\begin{aligned}
 &\text{minimize } f(\mathbf{x}) = \sum_{i=1}^m f_i(\mathbf{x}) \\
 &\text{subject to } \mathbf{x} \in \mathcal{X}
 \end{aligned} \tag{48}$$

- $\mathbf{x} \in \mathbb{R}^n$ is the decision parameter vector



- A problem of recent interest in **distributed networks** is
 - To design **decentralized** algorithms to minimize a sum of functions
 - With each component function is known only to a particular agent
- Consider a network of m agents, indexed by $i = 1, 2, \dots, m$. The aim is to solve the following optimization problem:

$$\begin{aligned} & \text{minimize} \quad f(\mathbf{x}) = \sum_{i=1}^m f_i(\mathbf{x}) \\ & \text{subject to} \quad \mathbf{x} \in \mathcal{X} \end{aligned} \quad (48)$$

- $\mathbf{x} \in \mathbb{R}^n$ is the decision parameter vector
- \mathcal{X} is the closed and convex subset of \mathbb{R}^n



Incremental Stochastic SubGradient Algorithms [30]

- A problem of recent interest in **distributed networks** is
 - To design **decentralized** algorithms to minimize a sum of functions
 - With each component function is known only to a particular agent
- Consider a network of m agents, indexed by $i = 1, 2, \dots, m$. The aim is to solve the following optimization problem:

$$\begin{aligned}
 &\text{minimize } f(\mathbf{x}) = \sum_{i=1}^m f_i(\mathbf{x}) \\
 &\text{subject to } \mathbf{x} \in \mathcal{X}
 \end{aligned} \tag{48}$$

- $\mathbf{x} \in \mathbb{R}^n$ is the decision parameter vector
- \mathcal{X} is the closed and convex subset of \mathbb{R}^n
- function f_i is a convex function from \mathbb{R}^n to \mathbb{R} known to only agent i

[30] Ram SS, Nedic A, Veeravalli VV. Incremental stochastic subgradient algorithms for convex optimization. in SIAM. 2009.



Algorithms

- **Cyclic incremental subgradient algorithm** in a network where agents are connected in a directed ring structure, the updates are

$$\mathbf{z}_{0,t+1} = \mathbf{z}_{m,t} = \mathbf{x}_t \quad (49)$$

$$\mathbf{z}_{i,t+1} = \mathcal{P}_{\mathcal{X}} \left[\mathbf{z}_{i-1,t+1} - \alpha^{(t+1)} (\nabla f_i(\mathbf{z}_{i-1,t+1} + \epsilon_{i,t+1})) \right] \quad (50)$$



- Randomized incremental subgradient algorithm** In this algorithm, agent i that updates is selected randomly according to a distribution. Formally the updates are

$$\mathbf{x}^{(t+1)} = \mathcal{P}_{\mathcal{X}} \left[\mathbf{x}(t) - \alpha^{(t+1)} \left(\nabla f_{s(t+1)}(\mathbf{x}^{(t+1)} + \epsilon_{s(t+1), t+1}) \right) \right] \quad (51)$$

The integer $s(t + 1)$ is the index of the agent that performs the update at time $t + 1$.



Convergence analysis

Assumptions:

- The Set $\mathcal{X} \subset \mathbb{R}^n$ is closed and convex. The function $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ is convex for each $i \in \{1, 2, \dots, m\}$.



Convergence analysis

Assumptions:

- The Set $\mathcal{X} \subset \mathbb{R}^n$ is closed and convex. The function $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ is convex for each $i \in \{1, 2, \dots, m\}$.
- There exists scalar sequences μ_t and ν_t such that



Convergence analysis

Assumptions:

- The Set $\mathcal{X} \subset \mathbb{R}^n$ is closed and convex. The function $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ is convex for each $i \in \{1, 2, \dots, m\}$.
- There exists scalar sequences μ_t and ν_t such that

$$\|\mathbb{E} [\epsilon_{i,t} \mid F_t^{i-1}]\| \leq \mu_t \quad (52)$$

$$\mathbb{E} [\|\epsilon_{i,t}\|^2 \mid F_t^{i-1}] \leq \nu_t^2 \quad (53)$$





100

- **Primal iteration:**



- Primal iteration:



Convergence results

The following asymptotic results are established in [31].



- **Almost sure feasibility**

$$\lim_{t \rightarrow \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbf{s}_{\tau}(\mathbf{p}_{\tau}, \mathbf{x}_{\tau}) \geq 0 \quad (61)$$





Recent results

A recent technique for large scale machine learning problems is proposed in [32]

- The decentralized double stochastic averaging gradient (DSA) algorithm is proposed as a solution alternative that relies on:
 - The use of local stochastic averaging gradients.
 - Determination of descent steps as differences of consecutive stochastic averaging gradients.
- **Strong convexity** of local functions and **Lipschitz continuity** of local gradients is shown to guarantee linear convergence of the sequence generated by DSA in **expectation**.
- **Future scope:**



Applications

- Resource allocation in OFDM networks [26, 27, 28]



Applications

- Resource allocation in OFDM networks [26, 27, 28]
- Load shedding in smart grid with real time pricing (RTP) [33, 34]

[27] Wang X, Gao N. Stochastic resource allocation over fading multiple access and broadcast channels. IEEE TIT, 2010.

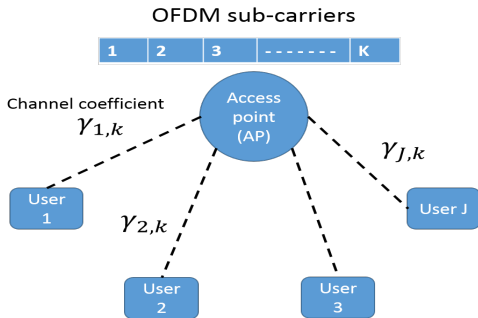
[26] Wang X, Giannakis GB. Resource allocation for wireless multiuser OFDM networks. IEEE TIT, Jul. 2011.

[33] Gatsis N, Marques AG. A stochastic approximation approach to load shedding in power networks. IEEE ICASSP, 2014.



Resource allocation in OFDM networks

Modeling preliminaries



Resources:

- Subcarriers, $\{1, 2, \dots, K\}$
- Rate, $r_{j,k}$
- Power, $p_{j,k}$



Utility based resource allocation

- Optimization problem

$$\max_{\bar{\mathbf{r}} \geq \mathbf{r}_{th}, (\boldsymbol{\alpha}, \mathbf{p}) \in \mathcal{F}} U(\bar{\mathbf{r}}) \quad (63)$$

$$\text{s.t.} \quad \hat{r}_j \leq \mathbb{E} \left[\sum_{k=1}^K c_{j,k}(\alpha_{j,k}^t, p_{j,k}^t) \right], \quad \forall j$$

$$\mathbb{E} \left[\sum_{j=1}^J \sum_{k=1}^K p_{j,k}^t \right] \leq P$$



Explanation

- $\alpha_{j,k}^t \geq 0$, is the time sharing fraction of a slot
- $p_{j,k}^t$, average transmit power allocated
-

$$\sum_{j=1}^J \alpha_{j,k}^t \leq 1, \quad \forall k = 1, \dots, K. \quad (64)$$

- assuming, AWGN at receiver with unit variance and sub-bandwidth $\nabla = 1$, the maximum achievable rate is

$$c_{j,k}^t = \begin{cases} \alpha_{j,k}^t \log_2 \left(1 + \frac{\gamma_{j,k}^t p_{j,k}^t}{\alpha_{j,k}^t} \right), & \alpha_{j,k}^t > 0 \\ 0, & \alpha_{j,k}^t = 0. \end{cases} \quad (65)$$

- Set \mathcal{F} is

$$\mathcal{F} := \left\{ (\alpha, \mathbf{p}) \left| \alpha_{j,k} \geq 0, p_{j,k} \geq 0, \sum_{j=1}^J \alpha_{j,k}^t \leq 1, \mathbb{E} \left[\sum_{j=1}^J \sum_{k=1}^K p_{j,k} \right] \leq P \right. \right\} \quad (66)$$

- **Subproblem I:**

$$\max_{\bar{\mathbf{r}} > \mathbf{r}_{th}} U(\bar{\mathbf{r}}) - \mu^T \bar{\mathbf{r}} \quad (68)$$

47/65

Online version

- **Primal updates:** with $\gamma[t]$, $\hat{\lambda}[t]$ and $\hat{\mu}[t]$ available per slot, the AP schedules according to allocation $\alpha^{t*}(\hat{\lambda}[t], \hat{\mu}[t], \gamma[t])$ and $\mathbf{p}^{t*}(\hat{\lambda}[t], \hat{\mu}[t], \gamma[t])$
- **Dual updates:**

$$\hat{\lambda}[t+1] = \left[\hat{\lambda}[t] + \beta \left(\sum_{k=1}^K \sum_{j=1}^J p^{t*}(\hat{\lambda}[t], \hat{\mu}[t]) - P \right) \right]^+ \quad (72)$$

$$\hat{\mu}[t+1] = \left[\hat{\mu}[t] + \beta \left(\bar{r}_j^*(\hat{\mu}[t]) - \sum_{k=1}^K r_{j,k}^{t*}(\hat{\lambda}[t], \hat{\mu}[t]) \right) \right]^+ \quad (73)$$



Future work

- Almost sure convergence results for the Incremental stochastic algorithms.
- Convergence results for the Cyclo-stationary case are not available.



Thank you



References

- [7] Shor NZ. Nondifferentiable Optimization and Polynomial Problems. Springer Science & Business Media; 1998.
- [8] Blair C. Problem Complexity and Method Efficiency in Optimization (AS Nemirovsky and DB Yudin). SIAM Review. 1985.
- [9] Palomar DP, Chiang M. A tutorial on decomposition methods for network utility maximization. IEEE JSAC. 2006.
- [10] Kelly F. Charging and rate control for elastic traffic. European transactions on Telecommunications. 1997 Jan 1.
- [11] Bertsekas DP, Nedi A, Ozdaglar AE. Convex analysis and optimization.
- [12] Nedic A, Ozdaglar A. Approximate primal solutions and rate analysis for dual subgradient methods. SIAM Journal on Opt. 2009.
- [13] Zargham M, Ribeiro A, Ozdaglar A, Jadbabaie A. Accelerated dual descent for network flow optimization. IEEE TAC. 2014.



References

- [14] Robbins H, Monro S. A stochastic approximation method. The annals of mathematical statistics. Sep. 1951.
- [15] Widrow B, Stearns SD. Adaptive signal processing. Englewood Cliffs, NJ, Prentice-Hall, Inc., 1985.
- [16] Sayed AH. Adaptive filters. John Wiley & Sons; Oct. 2011.
- [17] Bertsekas, D.P., Nonlinear programmingm, 1999.
- [18] Ermoliev Y. stochastic quasigradient methods and their application to system optimization. Stochastics: An International Journal of Probability and Stochastic Processes. 1983 Jan.
- [19] Bertsekas DP, Tsitsiklis JN. Neuro-dynamic programming: an overview. In Decision and Control, 1995., Proceedings of the 34th IEEE Conference on 1995 Dec 13.
- [20] Bottou L. Stochastic gradient learning in neural networks. Proceedings of Neuro-Nmes. 1991.





References

- [26] Wang X, Giannakis GB. Resource allocation for wireless multiuser OFDM networks. *IEEE Transactions on Information Theory*, Jul. 2011.
- [27] Wang X, Gao N. Stochastic resource allocation over fading multiple access and broadcast channels. *IEEE Transactions on Information Theory*, 2010.
- [28] Ribeiro A. Optimal resource allocation in wireless communication and networking. *EURASIP Journal on Wireless Communications and Networking*, Dec. 2012.
- [29] Weng, Lingjie, and Yutian Chen. "Stochastic Subgradient Methods".
- [30] Ram SS, Nedic A, Veeravalli VV. Incremental stochastic subgradient algorithms for convex optimization. *SIAM Journal on Optimization*. Jun. 2009.



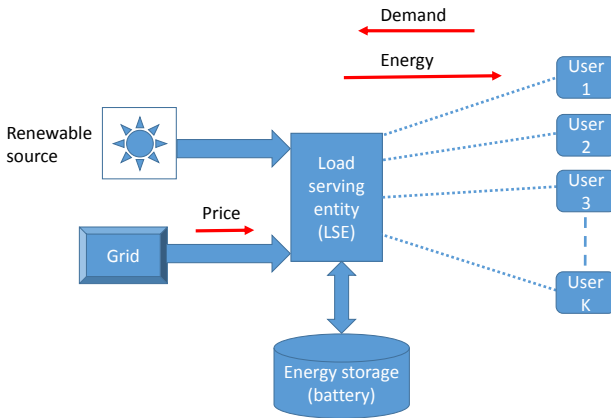
References

- [31] Ribeiro A. Ergodic stochastic optimization algorithms for wireless communication and networking. IEEE Transactions on Signal Processing, Dec. 2010.
- [32] Mokhtari A, Ribeiro A. DSA: Decentralized Double Stochastic Averaging Gradient Algorithm. arXiv preprint arXiv:1506.04216. 2015 Jun 13.
- [33] Gatsis N, Marques AG. A stochastic approximation approach to load shedding in power networks. IEEE International Conference in Acoustics, Speech and Signal Processing (ICASSP), 2014.
- [34] Deng R, Yang Z, Chen J, Chow MY. Load scheduling with price uncertainty and temporally-coupled constraints in smart grids. IEEE Transactions on Power Systems, Nov. 2014.



Load Shedding in Smart Grid

System model



System variables

• System parameters:

- π^t : actual demand - procured energy
- w^t : produced renewable energy at slot t
- a^t : real time energy price at slot t
- r^t : state of charge of battery at the end of slot t

• Optimization variables:

- s_k^t : amount of load shedded for user k at slot t
- b^t : amount of energy bought at slot t
- e_{out}^t : energy discharged from battery at slot t

[33] Gatsis N, Marques AG. A stochastic approximation approach to load shedding in power networks. IEEE (ICASSP), 2014.

[34] Deng R, Yang Z. Load scheduling with price uncertainty and temporally-coupled constraints in smart grids. IEEE TPS, 2014.



Problem formulation

- The system variables must satisfy the following relation

$$\pi^t - w^t \leq \sum_{k=1}^K s_k^t + b^t + \eta_{dis} e_{out}^t \quad (74)$$

- Battery dynamics equation

$$r^t = t^{t-1} + e_{in}^t - e_{out}^t, \quad t = 1, \dots, T \quad (75a)$$

$$0 \leq r^t \leq R, \quad t = 1, \dots, T \quad (75b)$$

$$e_{in}^t = \eta_{ch} \min\{e_{in}^{max}, [\pi^t - w^t]\}; 0 \leq e_{out}^t \leq e_{out}^{max} \quad (75c)$$

where, $[x] := \max\{-x, 0\}$.



Optimization problem

$$\min_{\{s^t\}, \{b^t\}, \{e_{out}^t\}, \{\hat{s}_k\}} \sum_{k=1}^K J_k(\hat{s}_k) + \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T a^t b^t \quad (76a)$$

$$\text{subject to } \lim_{T \rightarrow \infty} \sum_{t=1}^T s_k^t \leq \hat{s}_k, \quad k = 1, \dots, K, \quad (76b)$$

$$\lim_{T \rightarrow \infty} \sum_{t=1}^T e_{out}^t = \lim_{T \rightarrow \infty} \sum_{t=1}^T e_{in}^t, \quad (76c)$$

$$(74), (75), 0 \leq b^t \leq b^{max}, \quad \forall t \quad (76d)$$

$$0 \leq s_k^t \leq s_k^{max}, \quad \forall t \& k \quad (76e)$$



Optimization problem

$$\min_{\{s^t\}, \{b^t\}, \{e_{out}^t\}, \{\hat{s}_k\}} \sum_{k=1}^K J_k(\hat{s}_k) + \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T a^t b^t \quad (77a)$$

$$\text{subject to } \lim_{T \rightarrow \infty} \sum_{t=1}^T s_k^t \leq \hat{s}_k, \quad k = 1, \dots, K, \quad \text{assign } \sigma_k \quad (77b)$$

$$\lim_{T \rightarrow \infty} \sum_{t=1}^T e_{out}^t = \lim_{T \rightarrow \infty} \sum_{t=1}^T e_{in}^t, \quad \text{assign } \lambda \quad (77c)$$

$$(74), (75), 0 \leq b^t \leq b^{max}, \quad \forall t \quad (77d)$$

$$0 \leq s_k^t \leq s_k^{max}, \quad \forall t \& k \quad (77e)$$



- $$\hat{s}_k(\sigma_k) = \arg \min_s \{J_k(s) - \sigma_k s\} \quad (78)$$

Offline solution

- average primal variables

$$\hat{s}_k(\sigma_k) = \arg \min_s \{J_k(s) - \sigma_k s\} \quad (78)$$

- instantaneous primal variables

- If $\pi^t - w^t \leq 0$, then there is an instantaneous extra energy, to be stored in battery. $e_{in}^t = \eta_{ch} \min\{e_{in}^{max}, w^t - \pi^t\}$, while $e_{out}^{t*}(\cdot), b^{t*}(\cdot), s_k^{t*}(\cdot)$ are 0, $\forall k$.
- If $\pi^t - w^t > 0$, there is instantaneous energy deficit, optimization variables are found by solving

$$\min_{s_k^t, b^t, e_{out}^t \in S} a^t b^t + \rho e_{out}^t + \sum_{k=1}^K \sigma_k s_k^t \quad (79)$$

$$\text{subj. to } \pi^t - w^t \leq b^t + \eta_{dis} e_{out}^t + \sum_{k=1}^K s_k^t \quad (80)$$



Online version

Challenges with offline technique:

- main obstacle is to find optimal σ_k^*, ρ^*
- knowledge if joint distribution of $\{w^t, a^t\}$ is required
- algorithm will be computationally expensive

Merits of stochastic approximation

- only current samples σ_k^t, a^t is required
- computationally efficient



