Adaptation, Learning, and Optimization over Networks

Ali H. Sayed
University of California at Los Angeles
Editorial Scope

Topics

Foundations and Trends® in Machine Learning publishes survey and tutorial articles on the theory, algorithms and applications of machine learning, including the following topics:

- Adaptive control and signal processing
- Applications and case studies
- Behavioral, cognitive, and neural learning
- Bayesian learning
- Classification and prediction
- Clustering
- Data mining
- Dimensionality reduction
- Evaluation
- Game theoretic learning
- Graphical models
- Independent component analysis
- Inductive logic programming
- Kernel methods
- Markov chain Monte Carlo
- Model choice
- Nonparametric methods
- Online learning
- Optimization
- Reinforcement learning
- Relational learning
- Robustness
- Spectral methods
- Statistical learning theory
- Variational inference
- Visualization

Information for Librarians

Foundations and Trends® in Machine Learning, 2014, Volume 7, 6 issues. ISSN paper version 1935-8237. ISSN online version 1935-8245. Also available as a combined paper and online subscription.
Adaptation, Learning, and Optimization over Networks

Ali H. Sayed
University of California at Los Angeles
Contents

1 Motivation and Notation 2
  1.1 Introduction .................................. 2
  1.2 Biological Networks .......................... 3
  1.3 Distributed Processing ........................ 3
  1.4 Adaptive Networks ............................ 5
  1.5 Organization .................................. 6
  1.6 Notation and Symbols ......................... 7

2 Optimization by Single Agents 9
  2.1 Risk and Loss Functions ...................... 9
  2.2 Conditions on Risk Function .................. 13
  2.3 Optimization via Gradient Descent .......... 15
  2.4 Decaying Step-Size Sequences ............... 18
  2.5 Optimization in the Complex Domain ........ 23

3 Stochastic Optimization by Single Agents 28
  3.1 Adaptation and Learning ...................... 29
  3.2 Gradient Noise Process ........................ 33
  3.3 Stability of Second-Order Error Moment .... 36
  3.4 Stability of Fourth-Order Error Moment ...... 39
  3.5 Decaying Step-Size Sequences ............... 45
9.2 Stability of Fourth-Order Error Moment ........................................ 212
9.3 Stability of First-Order Error Moment ........................................... 221

10 Long-Term Network Dynamics ....................................................... 242
  10.1 Long-Term Error Model ............................................................ 243
  10.2 Size of Approximation Error ...................................................... 246
  10.3 Stability of Second-Order Error Moment ...................................... 250
  10.4 Stability of Fourth-Order Error Moment ...................................... 253
  10.5 Stability of First-Order Error Moment ........................................ 256
  10.6 Comparing Consensus and Diffusion Strategies ............................ 258

11 Performance of Multi-Agent Networks .............................................. 264
  11.1 Conditions on Costs and Noise .................................................. 265
  11.2 Performance Metrics ............................................................... 271
  11.3 Mean-Square-Error Performance ................................................ 273
  11.4 Excess-Risk Performance .......................................................... 298
  11.5 Comparing Consensus and Diffusion Strategies ............................ 305

12 Benefits of Cooperation ............................................................... 314
  12.1 Doubly-Stochastic Combination Policies .................................... 315
  12.2 Left-Stochastic Combination Policies ........................................ 318
  12.3 Comparison with Centralized Solutions ...................................... 325
  12.4 Excess-Risk Performance .......................................................... 331

13 Role of Informed Agents ............................................................... 336
  13.1 Informed and Uninformed Agents ................................................ 336
  13.2 Conditions on Cost Functions .................................................... 338
  13.3 Mean-Square-Error Performance ................................................ 340
  13.4 Controlling Degradation in Performance ..................................... 349
  13.5 Excess-Risk Performance .......................................................... 350

14 Combination Policies ................................................................. 352
  14.1 Static Combination Policies ....................................................... 353
  14.2 Need for Adaptive Policies ......................................................... 355
  14.3 Hastings Policy ........................................................................ 357
  14.4 Relative-Variance Policy ............................................................ 358
15 Extensions and Conclusions 373
15.1 Gossip and Asynchronous Strategies .......... 373
15.2 Noisy Exchanges of Information ............... 376
15.3 Exploiting Temporal Diversity ................. 377
15.4 Incorporating Sparsity Constraints ............. 380
15.5 Distributed Constrained Optimization .......... 381
15.6 Distributed Recursive Least-Squares .......... 386
15.7 Distributed State-Space Estimation .......... 391

Acknowledgements 400

Appendices 401

A Complex Gradient Vectors 402
A.1 Cauchy-Riemann Conditions .................. 402
A.2 Scalar Arguments .......................... 404
A.3 Vector Arguments .......................... 406
A.4 Real Arguments ............................ 408

B Complex Hessian Matrices 410
B.1 Hessian Matrices for Real Arguments .......... 410
B.2 Hessian Matrices for Complex Arguments ...... 412

C Convex Functions 420
C.1 Convexity in the Real Domain ................. 421
C.2 Convexity in the Complex Domain ............ 429

D Mean-Value Theorems 434
D.1 Increment Formulae for Real Arguments ..... 434
D.2 Increment Formulae for Complex Arguments . 436

E Lipschitz Conditions 439
E.1 Perturbation Bounds in the Real Domain .... 439
E.2 Lipschitz Conditions in the Real Domain ..... 443
Abstract

This work deals with the topic of information processing over graphs. The presentation is largely self-contained and covers results that relate to the analysis and design of multi-agent networks for the distributed solution of optimization, adaptation, and learning problems from streaming data through localized interactions among agents. The results derived in this work are useful in comparing network topologies against each other, and in comparing networked solutions against centralized or batch implementations. There are many good reasons for the peaked interest in distributed implementations, especially in this day and age when the word “network” has become commonplace whether one is referring to social networks, power networks, transportation networks, biological networks, or other types of networks. Some of these reasons have to do with the benefits of cooperation in terms of improved performance and improved resilience to failure. Other reasons deal with privacy and secrecy considerations where agents may not be comfortable sharing their data with remote fusion centers. In other situations, the data may already be available in dispersed locations, as happens with cloud computing. One may also be interested in learning through data mining from big data sets. Motivated by these considerations, this work examines the limits of performance of distributed stochastic-gradient solutions and discusses procedures that help bring forth their potential more fully. The presentation adopts a useful statistical framework and derives performance results that elucidate the mean-square stability, convergence, and steady-state behavior of the learning networks. The work also illustrates how distributed processing over graphs gives rise to some revealing phenomena due to the coupling effect among the agents. These phenomena are discussed in the context of adaptive networks, along with examples from a variety of areas including distributed sensing, intrusion detection, distributed estimation, online adaptation, network system theory, and machine learning.

DOI: 10.1561/2200000051.
1

Motivation and Notation

1.1 Introduction

Network science is a fascinating field that is evolving rapidly across many domains [15, 19, 92, 121, 155, 179, 208]. As remarked in [208], and for long, classical system and learning theories have focused on optimizing stand-alone systems or learners with great success. Nevertheless, progress in recent decades in the biological sciences [16, 50, 131, 147], animal behavior studies [7, 50, 79, 90, 188, 220], and the neuroscience of the brain [20, 49, 226], has revealed remarkable patterns of organization and structured complexity in the behavior of biological networks, animal groups, and in the dynamics of brain connectivity. These studies have brought forward notable examples of complex systems that derive their sophistication from coordination among simpler units and from the aggregation and processing of decentralized pieces of information. While each unit in these systems is not capable of sophisticated behavior on its own, it is the interaction among the constituents that leads to systems that are resilient to failure and that are capable of adjusting their behavior in response to changes in their environment.

These discoveries have motivated diligent efforts towards a deeper understanding of information processing, adaptation, and learning over
complex networks in several disciplines including machine learning, optimization, control, economics, biological sciences, information sciences, and the social sciences. A common goal in these investigations has been to develop theory and tools that enable the design of networks with sophisticated learning and processing abilities, such as networks that are able to solve important inference and optimization tasks in a distributed manner by relying on agents that interact locally and do not rely on fusion centers to collect and process their information.

1.2 Biological Networks

Examples abound for the viability of such designs in the realm of biological networks. Nature is laden with examples of networks exhibiting sophisticated behavior that arises from interactions among agents of limited abilities. For example, fish schools are unusually skilled at navigating their environment with remarkable discipline and at configuring the topology of their school in the face of danger from predators [79, 188]; when a predator is sighted or sensed, the entire school of fish adjusts its configuration to let the predator through and then coalesces again to continue its schooling behavior. It is reasonable to assume that this complex behavior is the result of sensing information spreading fast across the school of fish through local interactions among adjacent members of the school. Likewise, in bee swarms, it is observed that only a small fraction of the agents (about 5%) are informed and this small fraction of agents is still capable of guiding an entire swarm of bees to their new hive [12, 22, 125, 220]. It is a remarkable property of biological networks and animal groups that sophisticated behavior is able to arise from simple interactions among limited agents [119, 200, 229].

1.3 Distributed Processing

Motivated by these observations, this work deals with the topic of information processing over graphs and how collaboration among agents in a network can lead to superior adaptation and learning performance. The presentation covers results and tools that relate to the analysis and design of networks that are able to solve optimization, adaptation, and
learning problems in an efficient and distributed manner from streaming data through localized interactions among their agents.

The treatment extends the presentation from [208] in several directions\(^1\) and covers three intertwined topics: (a) how to perform distributed optimization over networks; (b) how to perform distributed adaptation over networks; and (c) how to perform distributed learning over networks. In these three domains, we examine and compare the advantages and limitations of non-cooperative, centralized, and distributed stochastic-gradient solutions. In the non-cooperative mode of operation, agents act independently of each other in their pursuit of their desired objective. In the centralized mode of operation, agents transmit their (collected or processed) data to a fusion center, which is capable of processing the data centrally. The fusion center then shares the results of the analysis back with the distributed agents. While centralized solutions can be powerful, they still suffer from some limitations. First, in real-time applications where agents collect data continuously, the repeated exchange of information back and forth between the agents and the fusion center can be costly especially when these exchanges occur over wireless links or require nontrivial routing resources. Second, in some sensitive applications, agents may be reluctant to share their data with remote centers for various reasons including privacy and secrecy considerations. More importantly perhaps, centralized solutions have a critical point of failure: if the central processor fails, then this solution method collapses altogether.

Distributed implementations, on the other hand, pursue the desired objective through localized interactions among the agents. In the distributed mode of operation, agents are connected by a topology and they are permitted to share information only with their immediate neighbors. There are many good reasons for the peaked interest in such distributed solutions, especially in this day and age when the word “network” has become commonplace whether one is referring to social networks, power networks, transportation networks, biological networks, or other types of networks. Some of these reasons have to do

\(^1\)The author is grateful to IEEE for allowing reproduction of material from [208] in this work.
1.4. Adaptive Networks

with the benefits of cooperation in terms of improved performance and improved robustness and resilience to failure. Other reasons deal with privacy and secrecy considerations where agents may not be comfortable sharing their data with remote fusion centers. In other situations, the data may already be available in dispersed locations, as happens with cloud computing. One may also be interested in learning and extracting information through data mining from large data sets. Decentralized learning procedures offer an attractive approach to dealing with such large data sets. Decentralized mechanisms can also serve as important enablers for the design of robotic swarms, which can assist in the exploration of disaster areas.

For these various reasons, we devote some good effort in this work towards quantifying the limits of performance of distributed solutions and towards discussing design procedures that can bring forth their potential more fully. Our emphasis is on solutions that are able to learn from streaming data. In particular, we shall study three families of distributed strategies: (a) incremental strategies, (b) consensus strategies, and (c) diffusion strategies — see Chapter 7. We shall derive expressions that quantify the behavior of the distributed algorithms and use the expressions to compare their performance and to illustrate under what conditions network cooperation is beneficial to the learning and adaptation process. While the social benefit, defined as the average performance across the network, generally improves through cooperation, it is not necessarily the case that the individual agents will always benefit from cooperation: some agents may see their performance degrade relative to the non-cooperative mode of operation [215, 277]. This observation will motivate us to seek optimized combination policies that enable all agents in a network to enhance their performance through cooperation.

1.4 Adaptive Networks

We shall study distributed solutions in the context of adaptive networks [208, 209, 215], which consist of a collection of agents with adaptation and learning abilities. The agents are linked together through a topol-
ogy and they interact with each other through localized *in-network* processing to solve inference and optimization problems in a fully distributed and online manner. The continuous sharing and diffusion of information across the network enables the agents to respond in real-time to drifts in the data and to changes in the network topology. Such networks are scalable, robust to node and link failures, and are particularly suitable for learning from big data sets by tapping into the power of collaboration among distributed agents. The networks are also endowed with cognitive abilities [108, 208] due to the sensing abilities of their agents, their interactions with their neighbors, and an embedded feedback mechanism for acquiring and refining information. Each agent is not only capable of experiencing the environment directly, but it also receives information through interactions with its neighbors and processes this information to drive its learning process.

Adaptive networks are well-suited to perform decentralized information processing tasks. They are also well-suited to model several forms of complex behavior exhibited by biological [16, 50, 131, 147] and social networks [15, 77, 92, 121, 230] such as fish schooling [188], prey-predator maneuvers [105, 171], bird formations [110, 119], bee swarming [12, 22, 125, 220], bacteria motility [25, 189, 258], and social and economic interactions [98, 103]. Examples of references that discuss applications of the *diffusion* distributed algorithms studied in this work to problems involving biological and social networks include [56, 65, 156, 213, 215, 246, 247, 250, 276]. Examples of references that discuss applications of *consensus* implementations include [2, 18, 64, 80, 118, 122, 123, 181, 184, 185, 199, 200, 255]. We do not discuss biological networks in this work and refer the reader instead to the above references; the survey article [215] provides some further motivation.

1.5 Organization

This work is largely self-contained. It provides an extended treatment of topics presented in condensed form in the survey [208], and of several other additional topics. For maximal benefit, readers may review
first the background material in Appendices A through G on complex
gradient vectors and Hessian matrices, convex functions, mean-value
theorems, Lipschitz conditions, matrix theory, and logistic regression.

In preparation for the study of multi-agent networks, Chapters 2–4 review
some fundamental results on optimization, adaptation, and
learning by single stand-alone agents. The emphasis is on stochastic-
gradient constructions. The presentation in these chapters provides
insights that will be useful in our subsequent study of adaptation and
learning by a collection of networked agents. This latter study is more
demanding due to the coupling among interacting agents, and due to
the fact that networks are generally sparsely connected. The results
in this work will help clarify the effect of network topology on perfor-
ance and will develop tools that enable designers to compare various
strategies against each other and against the centralized solution.

1.6 Notation and Symbols

All vectors are column vectors, with the exception of the regression
vector (denoted by the letters $u$ or $\mathbf{u}$), which will be taken to be a row
vector for convenience of presentation. Table 1.1 lists the main conven-
tions used in our exposition. In particular, note that we use boldface
letters to refer to random quantities and normal font to refer to their
realizations or deterministic quantities. We also use $\mathbf{T}$ for matrix or
vector transposition and $\ast$ for complex-conjugate transposition.

Moreover, for generality, we treat the case in which the variables of
interest are generally complex-valued; when necessary, we show how the
results simplify in the real case. Some subtle differences in the analy-
sis arise when dealing with complex data. These differences would be
masked if we focus exclusively on real-valued data. Moreover, studying
design problems with complex data is relevant for many fields, espe-
cially in the domain of signal processing and communications problems.
**Table 1.1:** List of notation and symbols used in the text and appendices.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathbb{R}$</td>
<td>Field of real numbers.</td>
</tr>
<tr>
<td>$\mathbb{C}$</td>
<td>Field of complex numbers.</td>
</tr>
<tr>
<td>$\mathbf{1}$</td>
<td>Column vector with all its entries equal to one.</td>
</tr>
<tr>
<td>$I_M$</td>
<td>Identity matrix of size $M \times M$.</td>
</tr>
<tr>
<td>$\mathbf{d}$</td>
<td>Boldface notation denotes random variables.</td>
</tr>
<tr>
<td>$d$</td>
<td>Normal font denotes realizations of random variables.</td>
</tr>
<tr>
<td>$A$</td>
<td>Capital letters denote matrices.</td>
</tr>
<tr>
<td>$a$</td>
<td>Small letters denote vectors or scalars.</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>Greek letters denote scalars.</td>
</tr>
<tr>
<td>$d(i)$</td>
<td>Small letters with parenthesis denote scalars.</td>
</tr>
<tr>
<td>$d_i$</td>
<td>Small letters with subscripts denote vectors.</td>
</tr>
<tr>
<td>$\mathbf{T}$</td>
<td>Matrix transposition.</td>
</tr>
<tr>
<td>$\ast$</td>
<td>Complex-conjugate transposition.</td>
</tr>
<tr>
<td>$\text{Re}(z)$</td>
<td>Real part of complex number $z$.</td>
</tr>
<tr>
<td>$\text{Im}(z)$</td>
<td>Imaginary part of complex number $z$.</td>
</tr>
<tr>
<td>$\text{col}{a, b}$</td>
<td>Column vector with entries $a$ and $b$.</td>
</tr>
<tr>
<td>$\text{diag}{a, b}$</td>
<td>Diagonal matrix with entries $a$ and $b$.</td>
</tr>
<tr>
<td>$\text{vec}{A}$</td>
<td>Vector obtained by stacking the columns of $A$.</td>
</tr>
<tr>
<td>$\text{bvec}{A}$</td>
<td>Vector obtained by vectorizing and stacking blocks of $A$.</td>
</tr>
<tr>
<td>$|x|$</td>
<td>Euclidean norm of its vector argument.</td>
</tr>
<tr>
<td>$|x|_\Sigma$</td>
<td>Weighted square value $x^\ast \Sigma x$.</td>
</tr>
<tr>
<td>$|A|$</td>
<td>Two-induced norm of matrix $A$, also equal to $\sigma_{\text{max}}(A)$.</td>
</tr>
<tr>
<td>$|A|_1$</td>
<td>Maximum absolute column sum of matrix $A$.</td>
</tr>
<tr>
<td>$|A|_\infty$</td>
<td>Maximum absolute row sum of matrix $A$.</td>
</tr>
<tr>
<td>$A \geq 0$</td>
<td>Matrix $A$ is non-negative definite.</td>
</tr>
<tr>
<td>$A &gt; 0$</td>
<td>Matrix $A$ is positive-definite.</td>
</tr>
<tr>
<td>$\rho(A)$</td>
<td>Spectral radius of matrix $A$.</td>
</tr>
<tr>
<td>$\lambda_{\text{max}}(A)$</td>
<td>Maximum eigenvalue of the Hermitian matrix $A$.</td>
</tr>
<tr>
<td>$\lambda_{\text{min}}(A)$</td>
<td>Minimum eigenvalue of the Hermitian matrix $A$.</td>
</tr>
<tr>
<td>$\sigma_{\text{max}}(A)$</td>
<td>Maximum singular value of $A$.</td>
</tr>
<tr>
<td>$A \otimes B$</td>
<td>Kronecker product of $A$ and $B$.</td>
</tr>
<tr>
<td>$A \otimes_b B$</td>
<td>Block Kronecker product of block matrices $A$ and $B$.</td>
</tr>
<tr>
<td>$a \preceq b$</td>
<td>Element-wise comparison of the entries of vectors $a$ and $b$.</td>
</tr>
<tr>
<td>$\delta_{k, \ell}$</td>
<td>Kronecker delta sequence: $1$ when $k = \ell$ and $0$ when $k \neq \ell$.</td>
</tr>
<tr>
<td>$\alpha = O(\mu)$</td>
<td>Signifies that $</td>
</tr>
<tr>
<td>$\alpha = o(\mu)$</td>
<td>Signifies that $\alpha/\mu \to 0$ as $\mu \to 0$.</td>
</tr>
<tr>
<td>$\alpha(\mu) \asymp \beta(\mu)$</td>
<td>Signifies that $\alpha(\mu)$ and $\beta(\mu)$ agree to first order in $\mu$.</td>
</tr>
<tr>
<td>$\limsup\limits_{n \to \infty} a(n)$</td>
<td>Limit superior of the sequence $a(n)$.</td>
</tr>
<tr>
<td>$\liminf\limits_{n \to \infty} a(n)$</td>
<td>Limit inferior of the sequence $a(n)$.</td>
</tr>
</tbody>
</table>
References


References


References


482

References


References


References


References


References


