Automatica 66 (2016) 254–261

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Brief paper

On reconstructability of quadratic utility functions from the iterations in gradient methods☆

Farhad Farokhi [a,1], Iman Shames [a], Michael G. Rabbat [b], Mikael Johansson [c]

[a] Department of Electrical and Electronic Engineering, University of Melbourne, Parkville, Australia
[b] Department of Electrical and Computer Engineering, McGill University, Montréal, Québec, Canada
[c] ACCESS Linnaeus Center, Electrical Engineering, KTH Royal Institute of Technology, Stockholm, Sweden

ARTICLE INFO

Article history:
Received 3 January 2015
Received in revised form
26 August 2015
Accepted 2 December 2015
Available online 28 January 2016

Keywords:
Statistical inference
Data privacy
Gradient methods
Data confidentiality
Parameter identification
Quadratic programming

ABSTRACT

In this paper, we consider a scenario where an *eavesdropper* can read the content of messages transmitted over a network. The nodes in the network are running a gradient algorithm to optimize a quadratic utility function where such a utility optimization is a part of a decision making process by an *administrator*. We are interested in understanding the conditions under which the eavesdropper can reconstruct the utility function or a scaled version of it and, as a result, gain insight into the decision-making process. We establish that if the parameter of the gradient algorithm, i.e., the step size, is chosen appropriately, the task of reconstruction becomes practically impossible for a class of Bayesian filters with uniform priors. We establish what step-size rules should be employed to ensure this.

1. Introduction

In recent decades, tremendous advances in the areas of communication and computation have facilitated the construction of complex systems. The design and analysis of these systems involve solving large optimization problems. Utility maximization, optimal flow, expenditure minimization, and traffic optimization are examples of such problems. Due to the size of these problems, it is often required that problems are solved over a network of interconnected processors. In many scenarios, the implementation of the solution to the optimization problem is in the public domain. However, from an operational point of view, it is important that the way that the decision is made remains confidential. In other words, while the optimal decision can be known by everyone, the utility function itself should remain confidential. Portfolios in portfolio optimization and local utilities in resource allocation can be considered as examples of such utility functions that need to be kept confidential. This especially becomes an issue as more computations related to operating the critical infrastructure (e.g. power distribution networks) are carried out in the cloud (Akyol, 2012). The importance of confidentiality, integrity, and availability is well understood in the security of data and ICT services (Bishop, 2002) and cloud computing (Chen, Paxson, & Katz, 2010). In these settings, confidentiality corresponds to ensuring the non-disclosure of data, integrity is related to the trustworthiness of data, and availability is concerned with the timely access to the data or system functionalities.

In this paper, we mainly focus on the question of confidentiality—particularly, the confidentiality of the utility functions even when the security of the network is compromised and an eavesdropper can listen to all the information being exchanged over the network during the course of solving the optimization problem. We consider scenarios where the utility function has a quadratic form. Specifically, the following question is answered: *when is it possible to reconstruct a utility function, or a scaled version of it, via having access to the iterations produced by an iterative method?* The iterative method considered in this paper is a gradient ascent algorithm. The choice of a gradient algorithm is inline with the recent observations that cast a favorable light on employing first-order methods to

☆ The work of I. Shames is supported by McKenzie Fellowship and the Advanced Vehicle System of Defence and Science Technology Group under research agreement MyIP.6288. The work of F. Farokhi is supported by the Australian Research Council (LP130100605). The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Hyeong Soo Chang under the direction of Editor Ian R. Petersen.

E-mail addresses: farhad.farokhi@unimelb.edu.au (F. Farokhi), iman.shames@unimelb.edu.au (I. Shames), michael.rabbat@mcgill.ca (M.G. Rabbat), mikaelj@ee.kth.se (M. Johansson).

[1] Tel.: +61 3 8344 8616; fax: +61 3 8344 6678.

http://dx.doi.org/10.1016/j.automatica.2016.01.014
0005-1098/

solve very large optimization problems (Cevher, Becker, & Schmidt, 2014). Note that the choice of quadratic programs is not very restrictive as trust-region optimization techniques allow us to solve any general optimization problem using a sequence of constrained quadratic programs recursively (Nocedal & Wright, 1999).

The problem that is addressed here is related to the one considered in the context of differential privacy (Dwork, 2008) and, to a larger extent, the application of differential privacy in optimization (Chaudhuri, Monteleoni, & Sarwate, 2011; Gupta, Ligett, McSherry, Roth, & Talwar, 2010; Mangasarian, 2011). However, it is important to note that there, the price for guaranteed confidentiality is paid in terms of data integrity and the accuracy of the solution. To ensure differential privacy, it is known that the information passed between the processing nodes at each step of the optimization algorithm should be perturbed by a random variable from a Laplace distribution (Dwork, 2008). This results in the algorithm not yielding an accurate solution. Here, we argue that the confidentiality of the objective (but not the solution) can be guaranteed in practice with no impact on the accuracy of the solution, if the algorithm parameter (the step size) is chosen appropriately, i.e., it is picked randomly from a sufficiently large set of suitable step sizes. In addition to differential privacy, other notions of privacy in optimization and machine learning have recently been pursued, e.g., see Duchi, Jordan, and Wainwright (2012), Vaidya, Yu, and Jiang (2008) and Weeraddana, Athanasiou, Fischione, and Baras (2013). Note that, in this paper, we are not directly contributing to the privacy-preserving literature, *per se*. Our main objective is to point out that, in the setups discussed, one does not need to worry about privacy since estimating the underlying parameters is practically impossible due computational restrictions (at least, with current technologies). Note that this problem is also related to the system identification and the parameter estimation literature, where the aim is to extract the parameters of the underlying utility or dynamics. However, in our setup, the eavesdropper cannot inject proper reference signals to fully probe the system (that is commonly known as the *persistent excitation* and is necessary for achieving the estimation objective Moore, 1983). Finally, in Gentry, Saligrama, and Feron (2001), the agents use their actions to learn about the strategies or the utilities of the other agents to subsequently devise optimal strategies. However, in that study, the computational aspects of the problem were largely unexplored and only linear programs were considered.

The outline of this paper is as follows. In Section 2, the problem that is considered in this paper is formulated. In Section 3, we consider the case where the eavesdropper has access to the iterates that are generated during the course of solving an unconstrained quadratic program. In this section, different choices of the step size are considered and conditions for which the utility function cannot be constructed successfully are discussed. Next, in Section 4, we consider the case where the problem is constrained. Concluding remarks are given in Section 5.

1.1. Notation

The sets of reals, nonnegative reals, integers, and nonnegative integers are, respectively, denoted by $\mathbb{R}$, $\mathbb{R}_{\geq 0}$, $\mathbb{Z}$, and $\mathbb{Z}_{\geq 0}$. The rest of the sets are denoted by calligraphic Roman letters, such as $\mathcal{M}$. Specifically, $\mathcal{S}_+^n$ is defined to be the set of symmetric positive-definite matrices in $\mathbb{R}^{n\times n}$. We define vec : $\mathbb{R}^{n\times m} \to \mathbb{R}^{nm}$ to be a vectorization operator that puts all the columns of a matrix into a vector sequentially. Finally, we use $A \otimes B$ to be the Kronecker product of matrices A and B.

2. Problem formulation

Consider the following optimization problem:

$$\max_{x\in\mathbb{R}^n} \quad -\frac{1}{2}x^\top Qx - q^\top x, \tag{1a}$$

$$\text{s.t.} \quad Cx \leq d, \tag{1b}$$

where $Q \in \mathcal{S}_+^n$, $q \in \mathbb{R}^n$, $C \in \mathbb{R}^{m\times n}$, $d \in \mathbb{R}^m$, and $\mathcal{X} \triangleq \{x \in \mathbb{R}^n \mid Cx < d\} \neq \emptyset$. The optimization problem (1) is solved by an *administrator* over a network via an optimization method, $\mathcal{F}(\cdot)$, given by

$$x[k+1] = \mathcal{F}(x[k]), \quad x[0] \in \mathcal{X}. \tag{2}$$

Throughout this paper, we assume that $\mathcal{F}(\cdot)$ is the gradient ascent algorithm in which different step-size selection methods can be used. This assumption is, partly, motivated by favorable results on first-order methods for solving large-scale optimization problems (Cevher et al., 2014). However, this assumption is also in place to greatly simplify the proofs and the presentation.

Remark 1. At first glance, the update rule in (2) appears to be a centralized implementation. However, distributed algorithms using primal decomposition as well as the inner problems for distributed algorithms using dual decomposition (see Bertsekas & Tsitsiklis, 1997) can be both rewritten, albeit in an aggregated form, in the form of (2).

Remark 2. The results presented in this work, at least in part, are applicable to more general utility functions, e.g., logarithmic functions. However, the selection of the quadratic utility functions results in linear operators that greatly simplify the proofs. Moreover, the quadratic utility functions, although partially conservative, have many applications and are widely used in signal processing, e.g., weighted least squares, and machine learning, e.g. support vector machines (Vaidya et al., 2008).

The measurement model of the eavesdropper is as follows. For any two consecutive measurements of the optimization variable $x[k]$ and $x[k+1]$, for some $k \in \mathbb{Z}_{\geq 0}$, the eavesdropper can construct a measurement of the form

$$y[k] = x[k] - x[k+1]. \tag{3}$$

Therefore, at time step $k+1$, the eavesdropper has access to measurement pairs $(x[t], y[t])_{t=0}^k$. Providing the solution to the following problems is of interest.

Problem 3 (*Utility Function Reconstruction*)**.** Assuming that the eavesdropper can measure $x[k]$ for all k and the values of A and b are known, under what conditions on the step size selection of the gradient descent algorithms can the eavesdropper estimate $(\hat{Q}, \hat{q})$ such that $Q = \gamma\hat{Q}$ and $q = \gamma\hat{q}$ for some $\gamma > 0$?

Solving the problem above enables the eavesdropper to determine the way that decisions are made. For example, it can be determined which variable has a bigger impact on the solution of the optimization problem (1). Hence, it is not necessary to exactly estimate γ.

Remark 4. In this paper, we assume that the communication is carried out over real and noiseless channels. Alternatively, one may consider the effects of quantization and noise on the utility reconstruction problem. However, this is beyond the scope of this paper.

Finally, we have the following standing assumption.

Assumption 5. The parameters $(Q, q) \in \mathcal{Q} \subseteq \mathcal{S}_+^n \times \mathbb{R}^n$ are randomly generated according to the non-degenerate probability density function $p : \mathcal{Q} \to \mathbb{R}_{\geq 0}$. Further, we assume the distribution of (Q, q) is independent of the initialization of the algorithm $x[0]$, which is uniformly selected from $\{x | x^\top x \leq 1\}$. The eavesdropper knows these probability distributions.