Automatica 61 (2015) 282–288

Contents lists available at ScienceDirect

Automatica

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

Brief paper

Submodularity and greedy algorithms in sensor scheduling for linear dynamical systems☆

Syed Talha Jawaid, Stephen L. Smith [1]

Department of Electrical and Computer Engineering, University of Waterloo, Waterloo ON, N2L 3G1, Canada

ARTICLE INFO

Article history:
Received 4 December 2014
Received in revised form
5 May 2015
Accepted 10 August 2015
Available online 5 September 2015

Keywords:
Kalman filters
Sensor networks
Multi-sensor estimation
Sensor scheduling
Greedy algorithms

ABSTRACT

This paper focuses on sensor scheduling for state estimation, which consists of a network of noisy sensors and a discrete-time linear system with process noise. As an energy constraint, only a subset of sensors can take a measurement at each time step. These measurements are fused into a common state estimate using a Kalman filter and the goal is to schedule the sensors to minimize the estimation error at a terminal time. A simple approach is to greedily choose sensors at each time step to minimize the estimation error at the next time step. Recent work has shown that this greedy algorithm outperforms other well known approaches. Results have been established to show that the estimation error is a submodular function of the sensor schedule; submodular functions have a diminishing returns property that ensures the greedy algorithm yields near optimal performance.

As a negative result, we show that most commonly-used estimation error metrics are not, in general, submodular functions. This disproves an established result. We then provide sufficient conditions on the system for which the estimation error is a submodular function of the sensor schedule, and thus the greedy algorithm yields performance guarantees.

1. Introduction

Sensor scheduling problems arise in applications involving the long-term estimation of a physical process through a set of static sensors. Examples include monitoring CO_2 concentrations (Weimer, Sinopoli, & Krogh, 2008) and monitoring water levels in multiple tanks (Weimer, Araújo, Hernandez, & Johansson, 2011) using a wireless sensor network. In such long-term deployments, energy consumption is a priority and by turning on only a small subset of sensors at each time-step, the battery life of the network can be extended.

Sensor scheduling can be described as follows. We are given a discrete-time linear dynamical system $x_{t+1} = Ax_t + w_t$, where x_t is the n-dimensional state vector and w_t is zero mean Gaussian process noise with known covariance. The goal is to estimate the state x_t through the use of m sensors. Each sensor takes a noisy scalar measurement of the state x_t, where the noise is zero mean Gaussian. A known covariance matrix gives the correlation between sensor noises. At each time step in this sensor scheduling problem we can activate at most $k < m$ sensors to take a measurement. Sensors that are inactive at a particular time step can then sleep, extending their battery life. After each time step t, the k measurements are fused into a single state estimate $\hat{x}_{t|t}$ with covariance $\Sigma_{t|t}$ using a centralized Kalman filter.

Given an initial covariance Σ_0 a terminal time T, and a sensor schedule σ, the final covariance $\Sigma_{T|T}$ is uniquely defined. The quality of a sensor schedule σ is determined as a function of this final covariance: for example the trace, the largest eigenvalue, or the determinant. Each objective can be thought of as a function that takes as input (Σ_0, σ) and outputs a real number. Our goal is to compute the schedule σ that minimizes/maximizes this objective function.

Related work: The sensor scheduling problem (or equivalently, the actuator scheduling problem) dates back at least as far as the early 1970s (Athans, 1972). Early work cast the problem in an optimal control framework and computed optimal schedules via dynamic programming. The drawback with this approach is that the computation grows exponentially with the terminal time T. In the last decade there has been a resurgence in work on sensor

☆ This research is partially supported by the Natural Sciences and Engineering Research Council of Canada. The material in this paper was partially presented at the 2014 American Control Conference, June 4–6, 2014, Portland, OR, USA. 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: stjawaid@uwaterloo.ca (S.T. Jawaid), stephen.smith@uwaterloo.ca (S.L. Smith).

[1] Tel.: +1 519 888 4567x31215.

http://dx.doi.org/10.1016/j.automatica.2015.08.022
0005-1098/

scheduling. A catalyst for this was the study of Kalman filtering with intermittent observations (Sinopoli et al., 2004). Based on this work, the authors in Gupta, Chung, Hassibi, and Murray (2006) provided a method for stochastically selecting measurements via random walk on an appropriately optimized Markov chain.

A variety of approaches have since been proposed for sensor scheduling, including approaches based on convex optimization (Joshi & Boyd, 2009), quadratic programming (Mo, Ambrosino, & Sinopoli, 2011), and tree pruning (Vitus, Zhang, Abate, Hu, & Tomlin, 2012). The work in Mo et al. (2011) presents a general framework that allows one to include more complex network and energy constraints. Recent work has looked at periodic schedules (Orihuela, Barreiro, Gómez-Estern, & Rubio, 2014), on the connection between the sensor selection problem and compressed sensing (Carmi & Gurfil, 2013), on provably complete algorithms (Jawaid & Smith, 2014a), and on consensus-based algorithms that remove the need for a centralized filter (Yang, Chen, Wang, & Shi, 2014).

Greedy algorithms and submodularity: The focus of this paper is on recent work that has shown advantages of using simple greedy algorithms for computing sensor schedules. At each time step t, the greedy algorithm chooses k sensors to minimize the estimate error at time $t+1$. The procedure begins at time step 1, and is repeated until all T time steps of the schedule have been specified. Being greedy, the approach is computationally efficient and simple to implement. Moreover, it has been shown (Shamaiah, Banerjee, & Vikalo, 2010, 2012) that the k sensors chosen at a single time step yield a reduction in estimation error that is within a constant factor of the optimal reduction. This property is due to the *submodularity* (Fisher, Nemhauser, & Wolsey, 1978; Nemhauser, Wolsey, & Fisher, 1978) of the covariance objective function over a single time step. In addition, empirical results in Shamaiah et al. (2010) have shown that the greedy algorithm often outperforms more computationally intensive alternatives based on convex optimization (Joshi & Boyd, 2009) over multiple time steps. In Huber, Kuwertz, Sawo, and Hanebeck (2009), a simulation study is performed for estimating a 31-state dynamical system that models a temperature diffusion process using sensor networks containing 20–40 sensors. The study compared three algorithms: the greedy algorithm, an optimal schedule based on exhaustive search, and a receding horizon approximate schedule. They found empirically that the runtime of the greedy algorithm was orders of magnitude better than the other two approaches, and the estimation performance exceeded that of the receding horizon approach. The authors also provided theoretical performance guarantees of the greedy algorithm.

In this paper we explore in more depth the connection between sensor scheduling, submodularity, and greedy algorithms. We characterize conditions under which the greedy algorithm gives provable performance guarantees by studying the submodularity of sensor scheduling objective functions.

Contributions: The contributions of this paper are two-fold. First, we provide negative results to show that most sensor schedule objective functions are not, in general, submodular nor monotone functions. This result holds for objectives including the trace of the covariance, the maximum eigenvalue and the log of the determinant, and it disproves the guarantees established in Huber et al. (2009). Second, we provide a set of (restrictive) conditions on the system under which the log of the determinant objective function is submodular, and thus the greedy algorithm has guaranteed performance. An early version of this paper appeared at the conference (Jawaid & Smith, 2014b). This paper expands on the preliminary version in several respects, including complete proofs of all results, details on the submodular counterexamples, and a new interpretation of sensor scheduling in terms of a submodular function over a matroid constraint in Section 6.1.

2. Preliminaries

In this section we review some essential concepts in submodular set functions from Fisher et al. (1978), Nemhauser et al. (1978) and submodular sequence functions from Alaei and Malekian (2010) and Golovin and Krause (2011).

2.1. Independence systems and matroids

Many combinatorial optimization problems can be formulated as maximizing or minimizing an objective function $f : \mathcal{F} \to \mathbb{R}$ over a set system $(E, \mathcal{F})$. The set $\mathcal{F} \subseteq 2^E$ contains all "allowable" subsets of the base set E. An *independence system* is a set system that is closed under subsets: if $A \in \mathcal{F}$ then $B \subseteq A \Longrightarrow B \in \mathcal{F}$.

Definition 1 (*Matroid*)**.** An independence system $(E, \mathcal{F})$ is a matroid if it satisfies the additional property that if $X, Y \in \mathcal{F}$ such that $|X| > |Y|$, then there is an $x \in X \setminus Y$ such that $Y \cup \{x\} \in \mathcal{F}$.

The *uniform matroid* is defined by the collection of all subsets of E with size less then or equal to $m \in \mathbb{Z}_+$, i.e., $\mathcal{F} := \{A \subseteq E : |A| \leq m\}$. Another example is the *partition matroid*. The base set E is partitioned into n disjoint sets, $\{E_i\}_{i=1}^n$. Given $k \in \mathbb{Z}_+^n$, the partition matroid is defined by the collection $\mathcal{F} := \{A \subseteq E : |A \cap E_i| \leq k_i, \forall i = 1 \dots n\}$.

2.2. Set functions

Let E be a finite set. A set function f over E assigns a value to every subset of E, i.e., $f : 2^E \to \mathbb{R}$.

Definition 2 (*Normalized and Monotone*)**.** The function f is normalized if $f(\emptyset) = 0$. The function, f, is monotone non-decreasing if for all $A \subseteq B \subseteq E, f(A) \leq f(B)$.

Definition 3 (*Submodularity*)**.** The function f is submodular if for all $A \subseteq B \subseteq E$ and for all $x \in E \setminus B$ we have $f(A \cup \{x\}) - f(A) \geq f(B \cup \{x\}) - f(B)$.

Submodular functions satisfy the property of *diminishing marginal returns*. That is, the contribution of any element x to the total value of a set decreases as the set gets bigger. More formally, let $\Delta_f(B|A) := f(A \cup B) - f(A)$. Then, $\Delta_f(x|A) \geq \Delta_f(x|B)$ for all $A \subseteq B \subseteq E$.

2.3. Sequence functions

For our purposes, a *sequence* $A = (a_1, \dots, a_k)$, of length $k \in \mathbb{Z}_{\geq 0}$ consists of k elements from a base set of elements E, i.e., $a_i \in E$. Two sequences $A = (a_1, \dots, a_k)$ and $B = (b_1, \dots, b_\ell)$ defined over the same base set can be *concatenated* into a larger sequence: $A \parallel B = (a_1, \dots, a_k, b_1, \dots, b_\ell)$. A *subsequence* of A is a sequence derived from A by deleting some elements but not changing the order of the remaining elements and is denoted $B \subseteq A$. A *sequence function* f defined over a base set E maps from sequences over E to real numbers. The value of a sequence function depends on the order of the elements in the sequence.

Definition 4 (*Sequence Monotonicity*)**.** The sequence function f is monotone non-decreasing if for all subsequences A of a sequence B, i.e., $A \subseteq B$, $f(A) \leq f(B)$.

For a sequence function f, we define the marginal reward of concatenating a sequence C to a sequence A as $\Delta_f(C|A) := f(A \parallel C) - f(A)$. The subscript f will be omitted unless there is ambiguity.

Definition 5 (*Sequence Submodularity*)**.** The function f is sequence submodular if for all $A \subseteq B$, we have $\Delta_f(C|A) \geq \Delta_f(C|B)$.