Pervasive and Mobile Computing 22 (2015) 113–125

Contents lists available at ScienceDirect

Pervasive and Mobile Computing

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

Adaptive compressive sensing based sample scheduling mechanism for wireless sensor networks

Jie Hao [a], Baoxian Zhang [a,*], Zhenzhen Jiao [a], Shiwen Mao [b]

[a] Research Center of Ubiquitous Sensor Networks, University of Chinese Academy of Sciences, Beijing 100049, China

[b] Department of Electrical and Computer Engineering, Auburn University, Auburn, AL 36849-5201, USA

ARTICLE INFO

Article history:
Available online 14 February 2015

Keywords:
Compressive sensing
Sample scheduling
Energy efficiency
Wireless sensor network

ABSTRACT

Sample scheduling is a crucial issue in wireless sensor networks (WSNs). The design objectives of efficient sample scheduling are in general two-folds: to achieve a low sample rate and also high sensing quality. Recently, compressive sensing (CS) has been regarded as an effective paradigm for achieving high sensing quality at a low sample rate. However, most existing work in the area of CS for WSNs use fixed sample rates, which may make sensor nodes in a WSN unable to capture significant changes of target phenomenon, unless the sample rate is sufficiently high, and thus degrades the sensing quality. In this paper, to pursue high sensing quality at low sample rate, we propose an adaptive CS based sample scheduling mechanism (ACS) for WSNs. ACS estimates the minimum required sample rate subject to given sensing quality on a per-sampling-window basis and accordingly adjusts sensors' sample rates. ACS can be useful in many applications such as environment monitoring, and spectrum sensing in cognitive sensor networks. Extensive trace-driven experiments are conducted and the numerical results show that ACS can obtain high sensing quality at low sample rate.

1. Introduction

Wireless sensor networks (WSN) can be deployed to monitor physical phenomena, e.g., primary user signals in a cognitive sensor network. In such networks, massive sensor nodes work to sense their surrounding environments and report their sensed data to a fusion center. Sensor nodes are usually powered by batteries that are energy limited. Since all the sensor nodes are targeted to monitor the same physical phenomena and the sensor nodes are usually deployed in a sufficient high density to guarantee high sensing quality, spatial correlation among the sensing measurements from neighboring sensor nodes is highly expected. In addition, for each individual sensor node, temporal correlation usually exists among its measurements since its monitored physical phenomenon usually changes continuously. Such spatial and temporal correlations have been exploited in various WSN technologies (e.g., spectrum sensing in a cognitive sensor networks, data aggregation and compression, route selection, clustering, and etc.) while meeting certain sensing quality requirements.

To this end, compressive sensing (CS) allows a sparse analog signal to be represented by much fewer samples than that required by the Nyquist sampling theorem [1]. In other words, for a sparse signal consisting of N samples, only a small number $M \ll N$ of encoded samples, generated by using the original N samples and properly chosen transform coefficients, are needed to be reported and used to recover the original signal at the sink side. The mapping matrix from the original

* Corresponding author.
E-mail address: bxzhang@ucas.ac.cn (B. Zhang).

http://dx.doi.org/10.1016/j.pmcj.2015.02.002
1574-1192/

N-sample signal to the M-sample signal is called measurement matrix or projection matrix. The ratio between M and N is called the sample rate and the degree of consistency between the recovered signal and the original signal is called recovery quality or sensing quality. The objective of CS design is to use the minimum sample rate to recover the original signal subject to given recovery quality. Three key factors are involved in the CS design: a representation basis Ψ to represent the interested signal in a sparse form, a measurement matrix Φ to transform the original N-sample signal into an M-sample signal with efficient transform coefficients and a recovery algorithm to recover the original signal. The recovery quality depends on all these three factors and the properties of the signal per se. It has been proven that the sparser the signal as represented by the representation basis, the better the recovery quality; the higher the incoherence between the representation basis and measurement matrix, the better the recovery quality. In general, almost all existing CS work (see [2–24]) focuses on the sophisticated design of the representation basis, measurement matrix, and recovery algorithm to pursue high recovery quality.

For phenomena monitoring applications using WSNs, traditionally, sensor nodes are often required to take samples according to a predetermined rate (e.g., once per minute). According to CS, each sensor node is allowed to sample the environment at a much lower sample rate than the predetermined rate without sacrificing the conformance between the actual phenomenon and the sensing reading collected. To maintain conformance, the sample rate should be tuned adaptively as the phenomenon changes. However, most existing work in the field of CS based sample scheduling for WSNs assume fixed sample rate, which may degrade the compression performance under dynamic phenomenon. To pursue stably high sensing quality with a sample rate as low as possible, it is critical to investigate adaptive CS schemes to trade-off the two performance goals.

An important application of CS is spectrum sensing in cognitive radio sensor networks. This is because channel availability is generally spatially and temporally correlated in such networks. Traditional spectrum sensing mechanisms requires each spectrum sensor to sense the licensed channels at the Nyquist sample rate for accurate spectrum sensing, which may lead to excess delay and waste of energy. With CS, each sensor node can perform spectrum sensing at a much lower sampling rate while still keeping high quality in spectrum sensing. Alternatively, spatial correlation in the spectrum sensor data can be exploited to turn off some spectrum sensors to preserve energy and extend the life time of the cognitive radio sensor network [19]. Since CS and also our adaptive CS mechanism proposed in this paper can work well for both phenomena monitoring and spectrum sensing, in the rest of this paper, we shall often take the application of phenomena monitoring as example when discussing mechanism implementation details although our design can also work well for spectrum sensing. In this sense, we treat "spectrum availability" as a phenomenon to be monitored.

In this paper, we propose an adaptive CS based sample scheduling mechanism (ACS) in order to achieve a low sample rate provided that a given sensing quality requirement is expected to be met. In ACS, each sensor node adjusts its sample rate on a per-checking-window basis. The implementation of ACS can be divided into two phases: Training phase and online phase. In the training phase, ACS needs to pre-collect certain amount of original sensing data (at high sample rate) for the phenomenon of interest, based on which it can build a hash table that empirically reflects the relationship between sample rate required for meeting a given desired sensing quality requirement and a sparsity degree (or change intensity). In the online phase, each sensor can decide its sample rate on per checking window basis. More specifically, it can decide its sample rate in the next checking window based on the signal sparsity degree or change intensity in the current checking window and also the hash table built during the training phase.

We accordingly propose the detailed design of ACS. The major contributions in this paper are as follows. First, we propose an adaptive sample scheduling mechanism to overcome the drawback that fixed sample rate compressive sensing mechanisms can fail to quickly react to significant phenomena change unless the sample rate is excessively high. We further present the detailed design of ACS based on signal sparsity and change intensity, respectively. Third, we conduct extensive numerical experiments using real data trace to validate the performance of ACS. The results demonstrate that ACS can achieve high performance as compared with existing work. That is, ACS can achieve desired sensing quality by much lower sample rate than existing mechanisms. Furthermore, experiment results show that each node independently adjusts its sample rate can achieve comparatively high performance as compared with collaborated sample control at different scales such as cluster-based or network-based. Largely reduced sample rate can largely reduce the energy consumed for environmental sampling and also for wireless communications, which makes ACS attractive for energy-constrained WSNs.

The reminder of this paper is organized as follows. Section 2 briefly reviews related work. Section 3 introduces necessary preliminaries for this work and formulates the problem to be addressed. Section 4 proposes the ACS mechanism. Section 5 presents the numerical experiments to demonstrate the high performance of ACS. Section 6 concludes this paper.

2. Related work

Recently, CS has been applied to WSNs due to its high recovery quality. Fig. 1 provides a typical example illustrating how CS can be utilized to design transmission schedule in a WSN. In Fig. 1, all nodes form a chain topology where the packets $p_1, p_2, \ldots, p_N$ as generated by nodes $s_1, s_2, \ldots, s_N$, respectively, need to be transmitted to the sink node in the network. Using traditional transmission schedule, each node needs to separately transmit its own packet and all the packets of its downstream nodes (i.e., those nodes away from the sink node). As a result, the sink receives N uncoded packets, $N(N+1)/2$ transmissions in total need to be carried out in the network and the closer a node to the sink, the more energy it will consume. By introducing CS as shown in Fig. 1, the N raw packets can be represented by M ($M \ll N$) encoded packets