Computational Geometry 48 (2015) 147–168

Contents lists available at ScienceDirect

Computational Geometry: Theory and Applications

www.elsevier.com/locate/comgeo

A sensor-based framework for kinetic data compression [☆]

Sorelle A. Friedler [a,*], David M. Mount [b,1]

[a] Dept. of Computer Science, Haverford College, Haverford, PA 19041, USA
[b] Dept. of Computer Science, University of Maryland, College Park, College Park, MD 20742, USA

ARTICLE INFO

Article history:
Received 1 December 2013
Accepted 10 September 2014
Available online 16 September 2014

Keywords:
Kinetic data
Sensor data
Lossless compression
Information theory

ABSTRACT

We introduce a framework for storing and processing kinetic data observed by sensor networks. These sensor networks generate vast quantities of data, which motivates a significant need for data compression. We are given a set of sensors, each of which continuously monitors some region of space. We are interested in the kinetic data generated by a finite set of objects moving through space, as observed by these sensors. Our model relies purely on sensor observations; it allows points to move freely and requires no advance notification of motion plans. Sensor streams are represented as random processes, where nearby sensors may be statistically dependent. We model the local nature of sensor networks by assuming that two sensor streams are statistically dependent only if the two sensors are among the m nearest neighbors of each other.

We present an algorithm for the lossless compression of the data produced by the network. We show that, under the statistical dependence and locality assumptions of our framework, asymptotically this compression algorithm encodes the data to within a constant factor of the information-theoretic lower bound dictated by the joint entropy of the system. In order to justify our locality assumptions, we provide a theoretical comparison with a variant of the kinetic data structures framework and experimental results demonstrating the existence of such locality properties in real-world data. We also give a relaxed version of our sensor stream independence property where even distant sensor streams are allowed some limited dependence. We extend the current understanding of empirical entropy to introduce definitions for joint empirical entropy, conditional empirical entropy, and empirical independence. We show that, even with the notion of limited independence and in both the statistical and empirical settings, the introduced compression algorithm achieves an encoding size that is within a constant factor of the optimal.

1. Introduction

There is a growing appreciation of the importance of algorithms and data structures for processing large data sets arising from the use of sensor networks, particularly for the statistical analysis of objects in motion. Large wireless sensor networks

[☆] A preliminary version of this work titled *Compressing Kinetic Data From Sensor Networks* appeared in the *Proceedings of the 5th International Workshop on Algorithmic Aspects of Wireless Sensor Networks (AlgoSensors)*, 2009 [16].

* Corresponding author.
E-mail addresses: sorelle@cs.haverford.edu (S.A. Friedler), mount@cs.umd.edu (D.M. Mount).
URLs: http://www.cs.haverford.edu/faculty/sorelle (S.A. Friedler), http://www.cs.umd.edu/~mount (D.M. Mount).

[1] The work of David Mount has been supported by the National Science Foundation under grants CCR-0635099 and CCF-1117259 and the Office of Naval Research under grant N00014-08-1-1015.

http://dx.doi.org/10.1016/j.comgeo.2014.09.002
0925-7721/

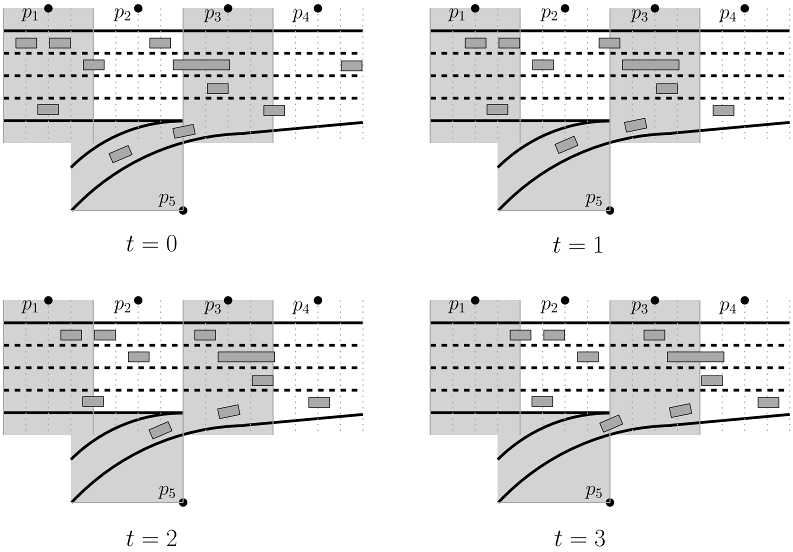

Fig. 1. An example of our sensor observation framework on a highway. Five sensors are placed at points p_1 through p_5 and observe the regions in front of them as indicated by the alternating shading patterns. Assuming that each vehicle stays in its lane and moves to the right at each time step by one unit (as marked by the light gray dotted lines), we have the following sensor counts for the four time steps shown (where the observation stream X_i is a 4-element sequence whose jth element is the number of cars that overlap p_i's region at time $t = j$): $X_1 = \langle 4, 3, 2, 1 \rangle$, $X_2 = \langle 3, 2, 3, 4 \rangle$, $X_3 = \langle 4, 4, 4, 4 \rangle$, $X_4 = \langle 2, 1, 1, 3 \rangle$, and $X_5 = \langle 2, 1, 1, 1 \rangle$.

are used in areas such as road-traffic monitoring [35], environment surveillance [28], and wildlife tracking [29,39]. With the development of sensors of lower cost and higher reliability, the prevalence of applications and the need for efficient processing will increase. We are interested not in the networking aspect of these sensor systems, but rather in the fact that data is gathered from a spatially distributed set of sensors each with a limited observation region.

Before reviewing the existing literature, we give a high-level overview of our sensor-based framework for data arising from moving objects, which will be described in greater detail in Section 2. We assume we are given a fixed set of sensors, which are modeled as points in some metric space. (An approach based on metric spaces, in contrast to standard Euclidean space, offers greater flexibility in how distances are defined between objects. This is useful in wireless settings, where transmission distance may be a function of non-Euclidean considerations, such as topography and the presence of buildings and other structures.) Each sensor is associated with a region of space, which it monitors. The moving entities are modeled as points that move over time. At regular time intervals, each sensor computes statistical information about the points within its region, which are streamed as output. We refer each of these as the sensor's *observation stream*. For the purposes of this paper, we assume that this information is simply an *occupancy count* of the number of entities that overlap the sensor's region at the given time instant (for an example, see Fig. 1). Thus, we follow the minimal assumptions made by Gandhi et al. [18] and do not rely on a sensor's ability to accurately record distance, angle, etc.

Wireless sensor networks record vast amounts of data. For example, road-traffic camera systems [35] that videotape congestion produce many hours of video or gigabytes of data for analysis even if the video itself is never stored and is instead represented by its numeric content. In order to analyze trends in the data, perhaps representing the daily rush hour or weekend change in traffic patterns, many weeks or months of data from many cities may need to be stored. As the observation time or number of sensors increases, so does the total data that needs to be stored in order to perform later queries, which may not be known in advance.

In this paper we consider the problem of how to compress the massive quantities of data that are streamed from large sensor networks. Compression methods can be broadly categorized as being either *lossless* (the original data is fully recoverable), or *lossy* (information may be lost through approximation). Because lossy compression provides much higher compression rates, it is by far the more commonly studied approach in sensor networks. Our ultimate interest is in scientific applications involving the monitoring of the motion of objects in space, where the loss of any data may be harmful to the subsequent analysis. For example, in habitat monitoring [28] if the data is collected and then studied later at a time when it is not possible to re-collect earlier data, it would be important not to lose any information. In such applications it is appropriate to focus on the less studied problem of lossless compression of sensor network data. Virtually all lossless compression techniques that operate on a single stream (such as Huffman coding [24], arithmetic coding [33], Lempel–Ziv [44]) rely on the statistical redundancy present in the data stream in order to achieve high compression rates. In the context of sensor networks, this redundancy arises naturally due to correlations in the streams of sensors that are spatially close to each other. As with existing methods for lossy compression [12,19], our approach is based on aggregating correlated streams and compressing these aggregated streams.