Realistic Compression of Kinetic Sensor Data

Sorelle A. Friedler*
sorelle@cs.umd.edu
http://www.cs.umd.edu/~sorelle

David M. Mount†
mount@cs.umd.edu
http://www.cs.umd.edu/~mount

Dept. of Computer Science, University of Maryland, College Park, MD 20742

Abstract

We introduce a realistic analysis for a framework for storing and processing kinetic data observed by sensor networks. The massive data sets generated by these networks motivates a significant need for compression. We are interested in the kinetic data generated by a finite set of objects moving through space. Our previously introduced framework and accompanying compression algorithm assumed a given set of sensors, each of which continuously observes these moving objects in its surrounding region. The model relies purely on sensor observations; it allows points to move freely and requires no advance notification of motion plans. Here, we extend the initial theoretical analysis of this framework and compression scheme to a more realistic setting. We extend the current understanding of empirical entropy to introduce definitions for joint empirical entropy, conditional empirical entropy, and empirical independence. We also introduce a notion of limited independence between the outputs of the sensors in the system. We show that, even with this notion of limited independence and in both the statistical and empirical settings, the previously introduced compression algorithm achieves an encoding size on the order of the optimal.

1 Introduction

Wireless sensor networks are ubiquitous; they continuously observe road-traffic conditions [16], world-wide environmental variables [11, 12], and the locations of fish, birds, moose, and other wildlife [3, 9, 14, 17]. As sensors continue to become smaller, cheaper, and more reliable, and as cell phone embedded sensors become more common [13], the need for efficient analysis and storage of data generated by sensor networks will increase.

Wireless sensor networks record vast amounts of data. For example, road-traffic camera systems [16] 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 the more immediate slow in traffic directly after a crash, 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.

With the goal of handling observed data from these sensor networks, it is important to create a realistic system that can store and retrieve the large amounts of data generated. Here we consider

*The work of Sorelle Friedler has been supported in part by the AT&T Labs Fellowship Program and the Ann G. Wylie Dissertation Fellowship.

†The work of David Mount has been supported in part by the National Science Foundation under grant CCR-0635099 and the Office of Naval Research under grant N00014-08-1-1015

the first issue, that of compression for storage. Compression methods may either be *lossless* (allow the original data to be fully reconstructed) or *lossy* (some of the data may be permanently lost through approximation methods). Since 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 this reason, we focus on the less studied problem of lossless compression of sensor network data. Virtually all lossless compression techniques that operate on a single stream rely on the information redundancy present in the stream in order to achieve high compression rates [8,15,18]. In the context of sensor networks, this redundancy arises naturally due to correlations in the outputs of sensors that are spatially close to each other. As with existing methods for lossy compression [2,7], our approach is based on aggregating correlated streams and compressing these aggregated streams.

We are particularly interested in *kinetic data*, by which we mean data arising from the observation of a discrete set of objects moving in time (as opposed to continuous phenomena such as temperature). The data sets generated by sensor networks have a number of spatial, temporal, and statistical properties that render them interesting for study. We assume that we use sensors at given locations to observe the motion of a discrete set of objects over some domain of interest. Thus, it is to be expected that the entities observed by one sensor will also likely be observed by nearby sensors, albeit at a slightly different time. For example, many of the vehicles driving by one traffic camera are likely to be observed by nearby cameras, perhaps a short time later or earlier. It is reasonable to expect the data streams generated by nearby sensors to be related. If so, the information content of the assembled data will be significantly smaller than the size of the raw data. In other words, the raw sensor streams, when considered in aggregate, will contain a great deal of redundancy. Well-designed storage and processing systems should capitalize on this redundancy to optimize space and processing times. In this paper we propose a realistic model of kinetic data as observed by a collection of fixed sensors. We will analyze a method for the lossless compression of the resulting data sets to show that this method is within a constant factor of the asymptotically optimal bit rate, subject to the assumptions of our model.

In an earlier paper we considered these issues under a theoretical analysis [6]. The framework we presented is purely observational; it relies on no assumptions or advance knowledge about the underlying object motion. We introduced a lossless compression algorithm within this framework and showed that, when considered in terms of the Shannon entropy, the compression algorithm achieved storage space on the order of the optimal joint entropy bound. This compression algorithm relied on the assumption that a sensor's output is statistically dependent only on the output of other locally close sensors. More information about this framework can be found in Section 2.

While our earlier framework provided theoretical guarantees on the compression rates it achieves, it is based on a theoretical model of sensor networks that may not be satisfied in practice. In particular, it has two significant drawbacks. The first is an analysis based in the statistical setting using Shannon entropy and its extensions. These entropy definitions assume an underlying random process that generates the data, and when analyzing a specific data set the probabilities associated with each random variable are known in advance; when considering observed sensor data, this assumption is unrealistic. We extend the framework analysis to hold under the more realistic definition of empirical entropy [10] that has the advantage of not assuming an underlying stationary, ergodic random process. Empirical entropy relies only on the observed probabilities of the sensor data values. In order to perform the complex analyses for the framework in the empirical setting, we also introduce new definitions for empirical entropy constructs that are analogous to existing statistical ones: joint empirical entropy, conditional empirical entropy, and empirical independence.

The second modification to the previously introduced framework that should be made in order to create a more realistic analysis concerns its assumptions of independence. The framework makes the assumption that sensor outputs are dependent only on their neighbors and are purely independent of all other outputs. However, it may be the case that there is some underlying dependence that may be common to many or all sensor outputs. For example, if the sensors are detecting and reporting car traffic counts, while nearby sensors may be more likely to see the same traffic patterns at consecutive time intervals, all sensors are likely to see a decrease in traffic at night and increases during rush hours. In order to analyze these underlying commonalities in the context of the framework for kinetic sensor data we introduce a notion of limited independence in both the statistical and empirical settings.

With the addition of the realistic assumptions of empirical entropy and limited independence, we revisit the space bounds for the framework compression algorithm and prove that the encoding size is on the order of the optimal size under an assumption of limited independence for both the statistical and empirical settings. These extensions confirm that the framework and its accompanying compression scheme are realistic for use with kinetic sensor data.

In summary, this paper makes the following contributions to the understanding and realistic analysis of kinetic sensor data:

1. An extension of the definition of empirical entropy to definitions for joint empirical entropy, conditional empirical entropy, and empirical independence. (See Section 4.)

2. An empirical entropy based analysis of a lossless compression algorithm within a sensor-based framework for kinetic data. (See Section 6.)

3. The introduction of a notion of limited independence between sensor outputs in both statistical and empirical settings. (See Section 5.)

4. The analysis of compression space bounds taking the notion of limited independence into account in both statistical and empirical settings. (See Section 6.)

2 Framework for Kinetic Sensor Data

In this section we give a more formal introduction to our earlier framework [6] and lossless compression scheme for discrete kinetic sensor data. This framework will be used as a basis for the results of this paper. We begin with some basic definitions about the structure of the sensor network and the associated observed data streams. Consider a static sensor network with S sensors, monitoring the motion of a collection of moving objects. Let P be a point set indicating the sensor locations. All sensors are assumed to operate over T synchronized time steps. Each sensor observes the motion of objects in some region surrounding it, and records an *occupancy count* indicating the number of objects passing within its region during the observed time step. No assumptions are made about the nature of the point motion nor the nature of the sensor regions (e.g., their shapes, density, disjointness, etc.).

Recall that central to our framework is the notion that each sensor's output is statistically dependent on a relatively small number of nearby sensors. For some point $p \in P$, let $NN_m(p) \subseteq P$ be the m nearest neighbors of p. Sensors i and j with associated sensor positions $p_i, p_j \in P$ are said to be *mutually m-close* if $p_i \in NN_m(p_j)$ and $p_j \in NN_m(p_i)$. For a constant m, a sensor system is said to be *m-local* if all pairs of sensors that are not mutually m-close are statistically independent.

In [6] we introduced a compression algorithm, *PartitionCompress*, which operates on an m-local sensor system. It compresses the sensor outputs to within a constant factor c (depending on dimension) of the optimal joint entropy bound. Intuitively, the compression algorithm is based on the following idea. If two sensor streams are statistically independent, they may be compressed independently from each other. If not, optimal compression can only be achieved if they are compressed jointly. The algorithm works by compressing the outputs from clusters of nearest neighbor groups together, as if they were a single stream. In order to obtain the desired compression bounds, these clusters must be sufficiently well separated so that any two mutually m-close sensors are in the same cluster. *PartitionCompress* partitions the points into a constant number c (independent of m but depending on dimension) of subsets for which this is true and then compresses clusters together to take advantage of local dependencies. The compression of a single cluster may be performed using any string compression algorithm; to obtain the near optimal bound, this algorithm must compress streams to their optimal entropy bound. In Section 6, we will show that LZ78, the Lempel-Ziv dictionary compression algorithm [18], is sufficient for our purposes.

3 Statistical Setting

As a point of reference, we begin by considering entropy and independence in the traditional statistical setting. In this setting, a sensor's output stream is modeled by a stationary, ergodic random process X over an alphabet Σ of fixed size. The *statistical probability* $p(x)$ of some outcome $x \in \Sigma$ is the probability associated with that outcome by the underlying random process. The *statistical entropy* of X is defined to be $-\sum_{x\in\Sigma} p(x)\log p(x)$. (Throughout, logarithms are taken base 2.) The *normalized statistical entropy* generalizes this to strings of increasing length:

$$\mathcal{H}_k(X) \;=\; -\frac{1}{k}\sum_{x\in\Sigma^k} p(x)\log p(x),$$

where in the standard definition, k is considered in the limit:

$$\mathcal{H}(X) \;=\; \lim_{k\to\infty} \mathcal{H}_k(X).$$

A fundamental fact from information theory is that this value represents the number of bits needed to encode a single character of the stream [1]. Unless otherwise specified, all references to *entropy* will mean normalized entropy. The *normalized joint statistical entropy* of two streams X and Y is defined to be

$$\mathcal{H}(X,Y) \;=\; \lim_{k\to\infty} -\frac{1}{k}\sum_{x,y\in\Sigma^k} p(x,y)\log p(x,y),$$

where $p(x,y)$ denotes the joint probability of both x and y occurring. The normalized joint statistical entropy of a set of strings $\mathbf{X} = \{X_1,\ldots,X_Z\}$ is defined analogously and is denoted $\mathcal{H}(\mathbf{X})$.

We say that two sensor streams X and Y are *statistically independent* if, for all k and any $x,y\in\Sigma^k$, we have $p(x,y) = p(x)p(y)$. If X and Y are statistically independent then $\mathcal{H}(X,Y) = \mathcal{H}(X)+\mathcal{H}(Y)$ [1]. The following technical result will be of later use.

Lemma 3.1. *Consider two sensor outputs X and Y over the same time period. Let $X+Y$ denote the componentwise sum of these streams. Then $\mathcal{H}(X+Y) \le \mathcal{H}(X,Y) \le \mathcal{H}(X)+\mathcal{H}(Y)$.*

Proof. To prove the first inequality, let $Z = X+Y$, and observe that $p(z) = \sum_{x+y=z} p(x,y)$. Clearly, if $x+y=z$, then $p(x,y)\le p(z)$. Thus,

$$\begin{aligned}\mathcal{H}(X+Y) \;&=\; -\sum_{z} p(z)\log p(z) \;\le\; -\sum_{z}\sum_{\substack{x,y\\ x+y=z}} p(x,y)\log p(x,y)\\ &=\; -\sum_{x,y} p(x,y)\log p(x,y) \;=\; \mathcal{H}(X,Y).\end{aligned}$$

By basic properties of conditional entropy (see, e.g., [1]), we have

$$\mathcal{H}(X,Y) \;=\; \mathcal{H}(X) + \mathcal{H}(Y|X) \;\leq\; \mathcal{H}(X) + \mathcal{H}(Y),$$

which establishes the second inequality. $\square$

4 Empirical Setting

Unlike statistical entropy, *empirical entropy* is based purely on the observed string, and does not assume an underlying random process. It replaces the probabilities of normalized entropy over substrings of length k by observed probabilities, conditioned on the value of the previous k characters. Let X be a string of length T over some alphabet Σ of fixed size. For $k \geq 1$ and $x \in \Sigma^k$, let $c_0(x)$ denote the number of times x appears in X, and let $c(x)$ denote the number of times x appears without being the suffix of X. Let $\mathsf{p}_{\mathrm{X}}(x) = c(x)/(T-k)$ denote the *observed probability* of x in X. (When X is clear from context, we will express this as $\mathsf{p}(x)$.) Following the definitions of Kosaraju and Manzini [10], the 0th order empirical entropy of a string X is defined to be

$$\mathsf{H}_0(X) \;=\; -\sum_{a\in\Sigma} \mathsf{p}(a)\log\mathsf{p}(a) = -\sum_{a\in\Sigma} \frac{c_0(a)}{T}\log\frac{c_0(a)}{T}\;.$$

For $a \in \Sigma$, let $\mathsf{p}_{\mathrm{X}}(a|x) = c(xa)/c(x)$ denote the observed probability that a is the next character of X immediately following x. The kth order empirical entropy is defined to be

$$\mathsf{H}_k(X) \;=\; -\frac{1}{T}\sum_{x\in\Sigma^k} c(x)\left[\sum_{a\in\Sigma}\mathsf{p}(a|x)\log\mathsf{p}(a|x)\right].$$

As observed in Kosaraju and Manzini [10], it is easily verified that $T \cdot \mathsf{H}_k(X)$ is a lower bound to the output size of any compressor that encodes each symbol with a code that only depends on the symbol itself and the k immediately preceding symbols. In the rest of this section, we introduce new extensions of these notions of empirical entropy to concepts that are analogous to those defined for the statistical entropy. Given two strings $X, Y \in \Sigma^T$ and $x, y \in \Sigma^k$, define $c(x,y)$ to be the count of the number of indices i, $1 \leq i \leq T-k$, such that $X[i \dots i+k-1] = x$ and $Y[i \dots i+k-1] = y$. Define $\mathsf{p}_{\mathrm{X,Y}}(x,y) = c(x,y)/(T-k)$. For $a, b \in \Sigma$, define $\mathsf{p}_{\mathrm{X,Y}}(a,b|x,y) = c(xa,yb)/c(x,y)$ to be the observed probability of seeing a and b in X and Y, respectively, just after seeing x and y. The *joint empirical entropy* of X and Y is defined to be

$$\mathsf{H}_k(X,Y) \;=\; -\frac{1}{T}\sum_{x,y\in\Sigma^k} c(x,y)\left[\sum_{a,b\in\Sigma}\mathsf{p}_{\mathrm{X,Y}}(a,b|x,y)\log\mathsf{p}(a,b|x,y)\right].$$

The joint empirical entropy of a set of strings $\mathbf{X} = \{X_1, \dots, X_Z\}$ is defined analogously and is denoted $\mathsf{H}_k(\mathbf{X})$.

We define the *conditional empirical entropy* of two strings $X, Y \in \Sigma^T$ to be

$$\mathsf{H}_k(X|Y) \;=\; -\frac{1}{T}\sum_{x,y\in\Sigma^k} c(x,y)\sum_{a,b\in\Sigma}\mathsf{p}_{\mathrm{X,Y}}(a,b|x,y)\log\mathsf{p}_{\mathrm{X,Y}}(x,a|y,b),$$

where we define $\mathsf{p}_{\mathrm{X,Y}}(x,a|y,b) = \mathsf{p}_{\mathrm{X,Y}}(a,b|x,y)/\mathsf{p}_{\mathrm{Y}}(b|y)$ to be the probability that a directly follows x in X given that b directly follows y in Y.

We say that two strings X and Y are *empirically independent* if, for all $j \leq k+1$ and all $x, y \in \Sigma^j$, the observed probability of x occurring at the same time instant as y is equal to the product of the observed probabilities of each outcome individually, that is, $\mathsf{p}_{\mathrm{X,Y}}(x,y) = \mathsf{p}_{\mathrm{X}}(x)\mathsf{p}_{\mathrm{Y}}(y)$. If X and Y are empirically independent then this also implies that, for $a \in \Sigma$ and $b \in \Sigma$, $\mathsf{p}_{\mathrm{X,Y}}(a,b|x,y) = \mathsf{p}_{\mathrm{X}}(a|x)\mathsf{p}_{\mathrm{Y}}(b|y)$.

The following technical lemma provides a few straightforward generalizations regarding properties of statistical entropy to empirical entropy.

Lemma 4.1. *Consider two strings $X, Y \in \Sigma^T$. Let $X+Y$ denote the componentwise sum of these strings.*

(i) If X and Y are empirically independent, $\mathsf{H}_k(X,Y) = \mathsf{H}_k(X) + \mathsf{H}_k(Y)$.

(ii) $\mathsf{H}_k(X,Y) = \mathsf{H}_k(X) + \mathsf{H}_k(Y|X)$.

(iii) $\mathsf{H}_k(X,Y) \leq \mathsf{H}_k(X) + \mathsf{H}_k(Y)$.

(iv) $\mathsf{H}_k(X+Y) \leq \mathsf{H}_k(X) + \mathsf{H}_k(Y)$.

Proof. We will not prove (i) here, since it will follow as a special case of Lemma 5.2 below (by setting $\delta = 0$). To prove (ii), observe that by manipulation of the definitions

$$\begin{aligned}\mathsf{H}_k(X,Y) &= -\frac{1}{T}\sum_{x,y\in\Sigma^k} c(x,y)\left[\sum_{a,b\in\Sigma}\mathsf{p}_{\mathrm{X,Y}}(a,b|x,y)\log\mathsf{p}_{\mathrm{X,Y}}(a,b|x,y)\right]\\ &= \mathsf{H}_k(X) + \mathsf{H}_k(Y|X).\end{aligned}$$

Symmetrically, we have $\mathsf{H}_k(X,Y) = \mathsf{H}_k(Y) + \mathsf{H}_k(X|Y)$.

To prove (iii), using (ii) we need only prove that $\mathsf{H}_k(Y|X) \leq \mathsf{H}_k(Y)$. By definition, we have

$$\mathsf{H}_k(Y|X) = -\frac{1}{T}\sum_{x,y\in\Sigma^k} c(x,y)\left[\sum_{a,b\in\Sigma}\frac{c(xa,yb)}{c(x,y)}\log\frac{c(xa,yb)}{c(x,y)}\right] .$$

Since clearly $c(x,y) \leq c(y)$ for all x and y, this means that

$$\mathsf{H}_k(Y|X) \leq -\frac{1}{T}\sum_{y\in\Sigma^k} c(y)\left[\sum_{b\in\Sigma}\frac{c(yb)}{c(y)}\log\frac{c(yb)}{c(y)}\right] = -\frac{1}{T}\sum_{y\in\Sigma^k} c(y)\left[\sum_{b\in\Sigma} p_Y(b|y)\log p_Y(b|y)\right] = \mathsf{H}_k(Y) ,$$

which completes the proof of (iii).

To prove (iv), let $Z = X+Y$. By the definition of empirical entropy we have $\mathsf{H}_k(X+Y) =$

$$-\frac{1}{T}\sum_{z\in\Sigma^k}\sum_{\substack{x,y\\ x+y=z}} c(x+y)\left[\sum_{g\in\Sigma}\sum_{\substack{a,b\\ a+b=g}}\mathsf{p}_{\mathrm{Z}}(a+b|x+y)\log\mathsf{p}_{\mathrm{Z}}(a+b|x+y)\right],$$

where $x+y$ is an outcome of length k and $a+b$ is an outcome of length 1 in the new string $X+Y$. By the same reasoning as in Lemma 3.1, $\mathsf{p}_{\mathrm{X,Y}}(x,y) \leq \mathsf{p}_{\mathrm{Z}}(x+y)$. Substituting this relationship into our equation and, since we desire an upper bound, considering only cases in which

$$-\mathsf{p}_{\mathrm{X,Y}}(a+b|x+y)\log\mathsf{p}_{\mathrm{X,Y}}(a+b|x+y) \leq -\mathsf{p}_{\mathrm{X,Y}}(a,b|x,y)\log(\mathsf{p}_{\mathrm{X,Y}}(a,b|x,y)),$$

we find that

$$\begin{aligned}
\mathsf{H}_k(X+Y) &\leq -\frac{1}{T}\sum_{x+y\in\Sigma^k} c(x,y)\left[\sum_{a,b\in\Sigma} \mathsf{p}_{\mathrm{X,Y}}(a,b|x,y)\log \mathsf{p}_{\mathrm{X,Y}}(a,b|x,y)\right] \\
&= \mathsf{H}_k(X,Y).
\end{aligned}$$

By (iii) we have $\mathsf{H}_k(X,Y) \leq \mathsf{H}_k(X) + \mathsf{H}_k(Y)$, which implies that $\mathsf{H}_k(X+Y) \leq \mathsf{H}_k(X) + \mathsf{H}_k(Y)$, as desired. □

5 Limited Independence

Perfect statistical or empirical independence is too strong an assumption to impose on sensor outputs. For example, if strings are drawn from independent sources, empirical independence will hold only in the limit. To deal with this, in this section we introduce a notion of limited independence for both the statistical and empirical settings. Given $0 \leq \delta < 1$, we say that a set of sensor streams $\mathbf{X} = \{X_1, X_2, ..., X_Z\}$ is *statistically δ-independent* if, for any k and outcomes $x_i \in \Sigma^k$,

$$(1-\delta)\prod_{i=1}^{Z} p(x_i) \leq p(x_1, x_2, ..., x_Z) \leq (1+\delta)\prod_{i=1}^{Z} p(x_i).$$

In the following lemma, we develop a relationship regarding the entropies of statistically δ-independent streams.

Lemma 5.1. *Given $0 \leq \delta < 1$ and a set of statistically δ-independent streams $\mathbf{X} = \{X_1, X_2, ..., X_Z\}$,* $(1-\delta)(\sum_{i=1}^{Z}\mathcal{H}(X_i)) - O(\delta) \ \leq\ \mathcal{H}(\mathbf{X}) \ \leq\ (1+\delta)(\sum_{i=1}^{Z}\mathcal{H}(X_i)) + O(\delta)$.

Proof. For simplicity of presentation, here we prove the lemma for sets $\mathbf{X} = \{X, Y\}$. The proof for a set of any size follows clearly from this presentation.

Recall that

$$\mathcal{H}(X,Y) \ =\ \lim_{k\to\infty} -\frac{1}{k}\sum_{x\in X, y\in Y} p(x,y)\log p(x,y).$$

By the assumption of statistical δ-independence, and by manipulation of the definitions, we have

$$\begin{aligned}
\mathcal{H}(X,Y) &\leq \lim_{k\to\infty} -\frac{1}{k}\sum_{x\in X,y\in Y} p(x)p(y)(1+\delta)\log(p(x,y)) \\
&\leq \lim_{k\to\infty} \frac{1}{k}\sum_{x\in X,y\in Y} p(x)p(y)(1+\delta)\log\frac{1}{p(x)p(y)(1-\delta)} \\
&= (1+\delta)(\mathcal{H}(X)+\mathcal{H}(Y)) + \lim_{k\to\infty}\frac{1+\delta}{k}\log\frac{1}{1-\delta}.
\end{aligned}$$

By a Taylor expansion in the neighborhood of $\delta = 0$, we see that $(1+\delta)\log\frac{1}{(1-\delta)} = O(\delta)$, which yields $\mathcal{H}(X,Y) \leq (1+\delta)(\mathcal{H}(X)+\mathcal{H}(Y)) + O(\delta)$. The proof that $(1-\delta)(\mathcal{H}(X)+\mathcal{H}(Y)) - O(\delta)$ follows symmetrically. □

We also introduce the idea of limited independence in the context of empirical entropy. Given $0 \leq \delta < 1$ a set of strings $\{X_1, X_2, ..., X_Z\}$ is *empirically δ-independent* if, for all $x_i \in \Sigma^j$ for $j \leq k+1$,

$$(1-\delta)\prod_{i=1}^{Z} \mathsf{p}(x_i) \leq \mathsf{p}(x_1, x_2, ..., x_Z) \leq (1+\delta)\prod_{i=1}^{Z} \mathsf{p}(x_i).$$

Lemma 5.2. *Given $0 \le \delta < 1$, and a set of empirically δ-independent strings $\mathbf{X} = \{X_1, X_2, ..., X_Z\}$ for $X_i \in \Sigma^j$ where $j \le k+1$,*

$$(1-\delta)\sum_{i=1}^{Z} \mathsf{H}_k(X_i) - O(\delta) \;\le\; \mathsf{H}_k(\mathbf{X}) \;\le\; (1+\delta)\sum_{i=1}^{Z} \mathsf{H}_k(X_i) + O(\delta).$$

Proof. For simplicity of presentation, here we prove the lemma for sets $\mathbf{X} = \{X, Y\}$. The general case is a straightforward generalization.

$$\mathsf{H}_k(X,Y) \;=\; -\frac{1}{T}\sum_{x,y\in\Sigma^k} c(x,y) \sum_{a,b\in\Sigma} \mathsf{p}(a,b|x,y)\log \mathsf{p}(a,b|x,y)$$

Let $\mathsf{p}(a,b|x,y) = \mathsf{p}_{\mathrm{X,Y}}(a,b|x,y)$, and recall that $\mathsf{p}_{\mathrm{X,Y}}(a,b|x,y) = c_0(xa,yb)/c(x,y)$ where $c_0(xa,yb)$ is the number of times the string $xa \in X$ appears at the same indices as $yb \in Y$, we have

$$\begin{aligned}\mathsf{H}_k(X,Y) &= -\frac{1}{T}\sum_{x,y\in\Sigma^k} c(x,y)\sum_{a,b\in\Sigma}\frac{c_0(xa,yb)}{c(x,y)}\log \mathsf{p}(a,b|x,y)\\ &= -\frac{1}{T}\sum_{x,y\in\Sigma^k}\sum_{a,b\in\Sigma}\frac{c_0(xa,yb)(T-k)}{T-k}\log \mathsf{p}(a,b|x,y).\end{aligned}$$

Since $\mathsf{p}(xa,yb) = \frac{c_0(xa,yb)}{T-k}$ and $\mathsf{p}(a,b|x,y) = \frac{c_0(xa,yb)}{c(x,y)}$ this is

$$\mathsf{H}_k(X,Y) \;=\; -\frac{T-k}{T}\sum_{x,y\in\Sigma^k}\sum_{a,b\in\Sigma}\mathsf{p}(xa,yb)\log\left(\frac{\mathsf{p}(xa,yb)}{\mathsf{p}(x,y)}\right).$$

Before proceeding with this analysis, we develop a useful relationship.

$$\begin{aligned}&-\frac{(T-k)}{T}\left[\sum_{x,y\in\Sigma^k}\mathsf{p}(x)\mathsf{p}(y)\sum_{a,b\in\Sigma}\mathsf{p}(a|x)\mathsf{p}(b|y)\log(\mathsf{p}(a|x)\mathsf{p}(b|y))\right]\\ &\quad= -\frac{1}{T}\left[\sum_{x\in\Sigma^k}c(x)\sum_{a\in\Sigma}p(a|x)\log p(a|x) + \sum_{y\in\Sigma^k}c(y)\sum_{b\in\Sigma}p(b|y)\log p(b|y)\right]\\ &\quad= \mathsf{H}_k(X) + \mathsf{H}_k(Y).\end{aligned}$$

Now we develop an upper bound on the earlier equation. Let $f = -\mathsf{p}(xa,yb)\log\frac{\mathsf{p}(xa,yb)}{\mathsf{p}(x,y)} = \mathsf{p}(xa,yb)\log\frac{\mathsf{p}(x,y)}{\mathsf{p}(xa,yb)}$. Then the equation we wish to bound is

$$\frac{T-k}{T}\sum_{x,y\in\Sigma^k}\sum_{a,b\in\Sigma} f,$$

where, by the definition of δ-independence,

$$f \le (1+\delta)\mathsf{p}(xa)\mathsf{p}(yb)\log\left(\frac{\mathsf{p}(x,y)}{\mathsf{p}(xa,yb)}\right) \le (1+\delta)\mathsf{p}(xa)\mathsf{p}(yb)\log\left(\frac{(1+\delta)\mathsf{p}(x)\mathsf{p}(y)}{(1-\delta)\mathsf{p}(xa)\mathsf{p}(yb)}\right).$$

Since $\mathsf{p}(xa,yb) = \mathsf{p}(a|x)\mathsf{p}(x)\mathsf{p}(b|y)\mathsf{p}(y)$, this is equal to

$$\begin{aligned}&(1+\delta)\mathsf{p}(x)\mathsf{p}(y)\mathsf{p}(a|x)\mathsf{p}(b|y)\log\left(\frac{(1+\delta)}{(1-\delta)\mathsf{p}(a|x)p(b|y)}\right)\\ &\quad= (1+\delta)\mathsf{p}(x)\mathsf{p}(y)\mathsf{p}(a|x)\mathsf{p}(b|y)\left(\log\frac{(1+\delta)}{(1-\delta)} - \log\left(\mathsf{p}(a|x)\mathsf{p}(b|y)\right)\right).\end{aligned}$$

Substituting back in for f and using our previously developed relationship, we have

$$\begin{aligned}
\mathsf{H}_k(X,Y) &\leq (1+\delta)(\mathsf{H}_k(X)+\mathsf{H}_k(Y)) + \\
&\quad \frac{(1+\delta)(T-k)}{T} \sum_{x,y\in\Sigma^k} \mathsf{p}(x)\mathsf{p}(y) \sum_{a,b\in\Sigma} \mathsf{p}(a|x)\mathsf{p}(b|y) \log \frac{1+\delta}{1-\delta} \\
&= (1+\delta)(\mathsf{H}_k(X)+\mathsf{H}_k(Y)) + \frac{(1+\delta)(T-k)}{T} \log \frac{1+\delta}{1-\delta}.
\end{aligned}$$

Let

$$g(\delta) = \log \frac{1+\delta}{1-\delta} = \log\left(1 + \frac{2\delta}{1-\delta}\right).$$

Consider the Taylor expansion for $g(\delta)$ in the neighborhood of $\delta = 0$ (i.e., the Maclaurin series). The Maclaurin series for $g(\delta)$ is within a constant factor of the expansion for $\log(1/(1-\delta)) = \delta + \delta^2/2 + \delta^3/3 + O(\delta^4)$. Since $\delta < 1$ by definition, $\delta^i > \delta^j$ for $i < j$, so $\log(1/(1-\delta)) = O(\delta)$ and $g(\delta) = O(\delta)$. Substituting back into our main inequality, we have

$$\begin{aligned}
\mathsf{H}_k(X,Y) &\leq (1+\delta)(\mathsf{H}_k(X)+\mathsf{H}_k(Y)) + \frac{(1+\delta)(T-k)}{T} O(\delta) \\
&\leq (1+\delta)(\mathsf{H}_k(X)+\mathsf{H}_k(Y)) + O(\delta).
\end{aligned}$$

The proof that

$$(1-\delta)(\mathsf{H}_k(X)+\mathsf{H}_k(Y)) - O(\delta) \leq \mathsf{H}_k(X,Y)$$

proceeds symmetrically. □

6 Compression Space Bounds

In this section will consider the encoding size that can be achieved by *PartitionCompress* [6]. Recall that *PartitionCompress* relies on a compression algorithm as a subroutine; the compression bounds for this subroutine will impact the final encoding size achieved by *PartitionCompress*. We will analyze this size in both statistical and empirical settings. In either context, we will use $\text{Enc}_{alg}(\mathbf{X})$ to denote the length of the encoded set of sensor outputs $\mathbf{X}$, where *alg* is the compression algorithm used by *PartitionCompress*.

6.1 Statistical Setting

Given a set of streams $\mathbf{X} = \{X_1, X_2, ..., X_Z\}$ in a statistical setting, standard information theory results [1] tell us that the optimal encoded space is $\sum_{i=1}^{Z} \mathcal{H}(X_i)$ bits. Call this $S_{opt}(\mathbf{X})$. From Section 5, we know that the optimal space used by an encoded set of statistically δ-independent streams $\mathbf{X}$ is $(1-\delta)\left(\sum_{i=1}^{Z} \mathcal{H}(X_i)\right) - O(\delta)$ bits. Call this $S_{opt}(\mathbf{X}, \delta)$. Let *opt* be some compression algorithm that achieves the optimal statistical entropy encoding length, for example LZ78. We know from [6] that $\text{Enc}_{opt}(\mathbf{X}) = O(\mathcal{H}(\mathbf{X}))$ bits for a set of observations from an m-local sensor system, where the hidden constant is exponential in m and the doubling dimension. We define a *statistically* (δ, m)*-local sensor system* to be the same as an m-local sensor system but with an assumption of δ-independence between the clusters instead of pure independence. We have the following theorem regarding the space used by *PartitionCompress*:

Theorem 6.1. *Given a set* $\mathbf{X}$ *of sensor outputs from a statistically* (δ, m)*-local sensor system, for any* $0 \leq \delta < 1 - \Omega(1)$*,*

$$\text{Enc}(\mathbf{X}) = O(\max\{\delta T, S_{opt}(\mathbf{X}, \delta)\}) \text{ bits.}$$

Proof. The optimal space bound is

$$S_{opt}(\mathbf{X}, \delta) = T(1-\delta)\sum_{i=1}^{Z} \mathcal{H}(X_i) \quad - \quad T \cdot O(\delta)$$

while *PartitionCompress* achieves a bound of

$$O(S_{opt}(\mathbf{X})) = O\left(T \cdot \sum_{i=1}^{Z} \mathcal{H}(X_i)\right).$$

The ratio is

$$\frac{O\left(\sum_{i=1}^{Z} H(X_i)\right)}{(1-\delta)\left[\sum_{i=1}^{Z} H(X_i)\right] - O(\delta)}.$$

The rest of the proof proceeds similarly to the proof of Theorem 6.3, but for $\mathcal{H}(X_i)$ instead of $\mathsf{H}_k(X_i)$ and with an extra constant factor hidden in the final bound. □

The space established in Theorem 6.1 is the basic statistical encoded space bound. It hides constants that are exponential in m and the doubling dimension. As a direct consequence of Lemma 3.1 and Theorem 6.1 we have the following corollary:

Corollary 6.1. *Consider two sensor outputs X and Y over the same time period. Let $X+Y$ denote the componentwise sum of these streams over some commutative semigroup. Then $Enc_{opt}(X+Y) \leq Enc_{opt}(X) + Enc_{opt}(Y)$ in the statistical setting.*

6.2 Empirical Setting

In the rest of this section, we extend the results of Section 6.1 to the empirical setting. In order to reason about the empirically optimal space bound for a set of strings $\mathbf{X}$, consider the string X^* over the alphabet Σ^Z created from the original set of strings by letting the ith character of the new string, for $1 \leq i \leq T$, be equal to a new character created by concatenating the ith character of each string in the original set. As mentioned earlier, the new string's optimal encoded space bound is $T \cdot \mathsf{H}_k(X^*)$.

Lemma 6.1. *Given a set of strings $\mathbf{X}$ and a string X^* created from $\mathbf{X}$ as described above, $\mathsf{H}_k(X^*) = \mathsf{H}_k(\mathbf{X})$.*

Proof. Recall that the definition of joint empirical entropy is based on the observed probability that single characters occur in all strings at the same string index directly after specific substrings of length k. Observe that by the construction of X^*, simultaneous occurrences appear for the same indices at which a single combined character appears in X^*. This observation implies that if $\mathsf{H}_k(\mathbf{X})$ is restated to refer to the characters appearing in X^*, $\mathsf{H}_k(X^*) = \mathsf{H}_k(\mathbf{X})$. □

Corollary 6.2. *The minimum number of bits to encode a set $\mathbf{X}$ of strings, assuming that each character depends only on the preceding k characters, is $S_{opt}(\mathbf{X}) = T \cdot \mathsf{H}_k(\mathbf{X})$.*

We will rely on context to distinguish between $S_{opt}(\mathbf{X})$ in statistical and empirical contexts. Although this construction suggests a compression procedure, it is impractical because in order to capture the repetitive nature of the strings in $\mathbf{X}$, the window size k would need to grow exponentially based on the size of the alphabet for each additional sensor stream. Instead, we use the more local approach of *PartitionCompress*.

We define an *empirically m-local sensor system* to be analogous to the definition of an m-local sensor system, but with an assumption of empirical independence instead of statistical independence. Similarly, an *empirically (δ, m)-local sensor system* assumes empirical δ-independence instead of statistical independence. The algorithm *PartitionCompress* relies on an entropy encoding algorithm as a subroutine. In the context of an empirical entropy-based analysis it would be appropriate to use the data structure developed by Ferragina and Manzini [4] as the subroutine that jointly compresses the streams from a single cluster. The Ferragina and Manzini structure [4] gives an optimal space bound of $O(T \cdot \mathsf{H}_k(X_i)) + T \cdot o(1)$ where X_i is the merged stream for that single cluster. We are interested in developing a lower bound on the compression that can be achieved using *PartitionCompress* in an empirical setting. Instead of using a specific algorithm we use the bound of $S_{opt}(\mathbf{X})$ discussed earlier and call the algorithm that achieves this bound *opt*. Assuming empirical independence of the set of strings $\mathbf{X}$ from z separate clusters within a single partition, compressing these clusters separately achieves the optimal bound of $S_{opt}(\mathbf{X}) = T \cdot \sum_{i=1}^{z} \mathsf{H}_k(X_i)$ space for a single partition. As a direct consequence, we have the following theorem.

Theorem 6.2. *Given a set $\mathbf{X}$ of sensor outputs from an empirically m-local sensor system,* $\mathrm{Enc}_{opt}(\mathbf{X}) = O(S_{opt}(\mathbf{X}))$ *bits.*

The hidden constants from Theorem 6.2 and for Theorems 6.3 and 6.4 grow exponentially in m and the doubling dimension of the space in which the sensors reside. If we consider empirical δ-independence, then the lower bound achieved by the compression algorithm (over Z total clusters in all partitions) remains $O(T \cdot \sum_{i=1}^{Z} \mathsf{H}_k(X_i))$, but $\sum_{i=1}^{Z} \mathsf{H}_k(X_i)$ is not generally equal to $\mathsf{H}_k(\mathbf{X})$, and so an optimal algorithm may be able to reduce the bound due to the δ dependence allowed. By application of Lemma 5.2, an optimal algorithm's bound is $S_{opt}(\mathbf{X}, \delta) = T(1-\delta)\left(\sum_{i=1}^{Z} \mathsf{H}_k(X_i)\right) + T \cdot O(\delta)$. We have the following theorem regarding the compressed size of the sensor outputs.

Theorem 6.3. *Given a set $\mathbf{X}$ of sensor outputs from an empirically (δ, m)-local sensor system for* $0 \leq \delta < 1 - \Omega(1)$, $\mathrm{Enc}_{opt}(\mathbf{X}) = O(\max\{\delta T, S_{opt}(\mathbf{X})\})$ *bits.*

Proof. An optimal algorithm would compress each partition to take the greatest advantage of the dependence between clusters. It would achieve a space bound of

$$S_{opt}(\mathbf{X}, \delta) = T(1-\delta)\left[\sum_{i=1}^{Z} \mathsf{H}_k(X_i)\right] - T \cdot O(\delta)$$

for each partition. The *PartitionCompress* algorithm compresses each partition to space

$$S_{opt}(\mathbf{X}) = T \sum_{i=1}^{Z} \mathsf{H}_k(X_i).$$

The ratio is

$$\rho = \frac{S_{opt}(\mathbf{X})}{S_{opt}(\mathbf{X}, \delta)} = \frac{\sum_{i=1}^{Z} \mathsf{H}_k(X_i)}{(1-\delta)\left[\sum_{i=1}^{Z} \mathsf{H}_k(X_i)\right] - O(\delta)}.$$

Here we consider the two possible cases for the relationship of $O(\delta)$ to $\sum_{i=1}^{Z} \mathsf{H}_k(X_i)$:

1. *Case $O(\delta) \geq \sum_{i=1}^{Z} \mathsf{H}_k(X_i)$:*

$$\frac{\sum_{i=1}^{Z} \mathsf{H}_k(X_i)}{(1-\delta)\left[\sum_{i=1}^{Z} \mathsf{H}_k(X_i)\right] - O(\delta)} = \left[\frac{1}{1-\delta}\right] O(\delta) = O(\delta).$$

For this case, *PartitionCompress*'s space bound is within $O(\delta)$ of the optimal for a single one of the c partitions, or a total of $O(\delta)$ times $S_{opt}(\mathbf{X}, \delta)$, which is $O(\delta \cdot T)$, since $O(\delta) \geq \sum_{i=1}^{Z} \mathsf{H}_k(X_i)$.

2. *Case* $O(\delta) < \sum_{i=1}^{Z} \mathsf{H}_k(X_i)$:

$$\frac{1}{1-\delta}\left[\frac{1}{1 - O(\delta)/(1-\delta)\sum_{i=1}^{Z} \mathsf{H}_k(X_i)}\right] \leq \frac{1}{1-\delta}\left[\frac{1}{1-\frac{1}{1-\delta}}\right] = O\left(\frac{1}{1-\delta}\right) = O(1+\delta).$$

For this case, *PartitionCompress*'s space bound is $O((1+\delta)S_{opt}(\mathbf{X}, \delta))$, which is $O(S_{opt}(\mathbf{X}))$ since $\delta < 1 - \Omega(1)$.

The final total space bound is $\max\{O(\delta T), O(S_{opt}(\mathbf{X}))\}$. □

For this paper, we will also be interested in the LZ78 algorithm, since the dictionary created in the process of compression is useful for searching compressed text without uncompressing it. While Kosaraju and Manzini [10] show that LZ78 does not achieve the optimal bound of $T \cdot \mathsf{H}_k(X)$, they show that it uses space at most $T \cdot \mathsf{H}_k(X) + O((T \log\log T)/\log T)$. In our context, this means that each cluster uses space $T \cdot \mathsf{H}_k(X) + O((T \log\log T)/\log T)$.

Theorem 6.4. *Given a set* $\mathbf{X} = \{X_1, X_2, ..., X_Z\}$ *of sensor outputs taken over a sufficiently long time* T *from an empirically* (δ, m)*-local sensor system, for any* $0 \leq \delta < 1 - \Omega(1)$,

$$\begin{aligned}\mathrm{Enc}(\mathbf{X}) &= cT\sum_{i=1}^{Z}\left(\mathsf{H}_k(X_i) + O\left(\frac{\log\log T}{\log T}\right)\right) \\ &= O\left(\max\left\{\delta T, S_{opt}(X,\delta), \frac{T\log\log T}{\log T}\right\}\right) \textit{ bits.}\end{aligned}$$

Proof. An optimal algorithm would compress each partition to take the most advantage of the dependence between clusters. It would achieve a space bound of

$$T(1-\delta)\left[\sum_{i=1}^{Z} \mathsf{H}_k(X_i)\right] - T \cdot O(\delta)$$

while using LZ78 as the basis for *PartitionCompress* compresses each partition to

$$T\sum_{i=1}^{Z} \mathsf{H}_k(X_i) + \sum_{i=1}^{Z} O((T\log\log T)/\log T)$$

where Z is the total number of clusters over all partitions. The ratio is

$$\frac{\sum_{i=1}^{Z} \mathsf{H}_k(X_i) + \sum_{i=1}^{Z} O((\log\log T)/\log T)}{(1-\delta)\left[\sum_{i=1}^{Z} \mathsf{H}_k(X_i)\right] - O(\delta)}$$

$$= \frac{1}{1-\delta}\left[\frac{\sum_{i=1}^{Z} \mathsf{H}_k(X_i) + \sum_{i=1}^{Z} O((\log\log T)/\log T)}{\left[\sum_{i=1}^{Z} \mathsf{H}_k(X_i)\right] - O(\delta)}\right].$$

Here we consider the two possible cases for the relationship of $O(\delta)$ to $\sum_{i=1}^{Z} \mathsf{H}_k(X_i)$:

1. *Case* $O(\delta) \geq \sum_{i=1}^{Z} \mathsf{H}_k(X_i)$:

$$\frac{1}{1-\delta}\left[\frac{\sum_{i=1}^{Z} \mathsf{H}_k(X_i) + \sum_{i=1}^{Z} O((\log\log T)/\log T)}{\left[\sum_{i=1}^{Z} \mathsf{H}_k(X_i)\right] - O(\delta)}\right]$$

$$\leq \frac{1}{1-\delta}\left[O(\delta) + \frac{\sum_{i=1}^{Z} O((\log\log T)/\log T)}{O(\delta)}\right]$$

Choose T large enough so that $O((\log\log T)/\log T) < O(\delta)$. Then the ratio is

$$\leq \left[\frac{1}{1-\delta}\right] O(\delta) = O(\delta)$$

for a single one of the c partitions, or $O(\delta)$ total times $S_{opt}(\mathbf{X}, \delta)$, which is $O(\delta T)$ since $O(\delta) \geq \sum_{i=1}^{Z} \mathsf{H}_k(X_i)$.

2. *Case* $O(\delta) < \sum_{i=1}^{Z} \mathsf{H}_k(X_i)$:

$$\frac{1}{1-\delta}\left[\frac{\sum_{i=1}^{Z} \mathsf{H}_k(X_i) + \sum_{i=1}^{Z} O((\log\log T)/\log T)}{\left[\sum_{i=1}^{Z} \mathsf{H}_k(X_i)\right] - O(\delta)}\right]$$

$$= \frac{1}{1-\delta}\left[O(1) + \frac{\sum_{i=1}^{Z} O((\log\log T)/\log T)}{O\left(\sum_{i=1}^{Z} \mathsf{H}_k(X_i)\right)}\right].$$

Here, we consider two sub-cases based on the relationship between $O((Z\log\log T)/\log T)$ and $\sum_{i=1}^{Z} \mathsf{H}_k(X_i)$.

(a) *Case* $O((\log\log T)/\log T) \geq \sum_{i=1}^{Z} \mathsf{H}_k(X_i)$:
Then the ratio is at most $O((1+\delta)(\log\log T)/\log T)$, for a total space of $O((1+\delta)((\log\log T)/\log T)S_{opt}(\mathbf{X}))$, which is $O(T(\log\log T)/\log T)$ since $O((\log\log T)/\log T) \geq \sum_{i=1}^{Z} \mathsf{H}_k(X_i) > O(\delta)$.

(b) *Case* $O((\log\log T)/\log T) < \sum_{i=1}^{Z} \mathsf{H}_k(X_i)$:
Then the ratio is

$$\leq O\left(\frac{1}{1-\delta}\right) = O(1+\delta)$$

for a single one of the c partitions, or $O(1+\delta)$ total times $S_{opt}(\mathbf{X}, \delta)$, which is $O(S_{opt}(\mathbf{X}))$ since $\delta < 1 - \Omega(1)$.

The final bound is $O(\max\{\delta T, S_{opt}(X,\delta), T(\log\log T)/\log T\})$ total space. □

As a direct consequence of Lemma 4.1(iv) and Theorem 6.4 we have the following corollary:

Corollary 6.3. *Consider two sensor outputs X and Y over the same time period. Let $X+Y$ denote the componentwise sum of these streams over some commutative semigroup. Then $Enc_{LZ78}(X+Y) \leq Enc_{LZ78}(X) + Enc_{LZ78}(Y)$ in the empirical setting.*

We have now established $\mathrm{Enc}_{\mathrm{LZ78}}(\mathbf{X})$ in both statistical and empirical settings (Theorems 6.1 and 6.4 respectively). For future analyses we may also be interested in the number of nodes (representing words) in the dictionary resulting from the LZ78 compression process, denoted d. Note that due to the nature of the dictionary, $d = O(T/\log T)$ [5], so $T = \Omega(d\log d)$. Since $d\log d$ is the total space needed to store the compressed string and dictionary, in our context $d\log d = \mathrm{Enc}_{\mathrm{LZ78}}(\mathbf{X})$.

References

[1] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley-IEEE, second edition, 2006.

[2] A. Deligiannakis, Y. Kotidis, and N. Roussopoulos. Processing approximate aggregate queries in wireless sensor networks. *Inf. Syst.*, 31(8):770–792, 2006.

[3] H. Dettki, G. Ericsson, and L. Edenius. Real-time moose tracking: an internet based mapping application using GPS/ GSM-collars. *Alces*, 40:13–21, 2004.

[4] P. Ferragina and G. Manzini. Opportunistic data structures with applications. In *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on*, pages 390–398, 2000.

[5] P. Ferragina and G. Manzini. Indexing compressed text. *Journal of the ACM*, 52(4):552–581, July 2005.

[6] S. A. Friedler and D. M. Mount. Compressing kinetic data from sensor networks. In *Proc. of the Fifth Intl. Workshop on Algorithmic Aspects of Wireless Sensor Networks (AlgoSensors)*, pages 191 – 202, 2009.

[7] S. Gandhi, S. Nath, S. Suri, and J. Liu. Gamps: Compressing multi sensor data by grouping and amplitude scaling. In *ACM SIGMOD*, 2009.

[8] D. A. Huffman. A method for the construction of minimum-redundancy codes. *Proc. of the IRE*, 40, Sept. 1952.

[9] P. Juang, H. Oki, Y. Wang, M. Martonosi, L.-S. Peh, and D. Rubenstein. Power-aware computing for wildlife tracking: Design tradeoffs and early experiences in zebranet. In *10th International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 96–107, October 2002.

[10] R. S. Kosaraju and G. Manzini. Compression of low entropy strings with Lempel–Ziv algorithms. *SIAM J. Comput.*, 29(3):893–911, 1999.

[11] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson. Wireless sensor networks for habitat monitoring. In *ACM international workshop on wireless sensor networks and applications*, pages 88–97, 2002.

[12] National Oceanic and Atmospheric Administration Earth System Research Laboratory. Science on a sphere dataset catalog. http://sos.noaa.gov/datasets/.

[13] E. Paulos, R. J. Honicky, and E. Goodman. Sensing atmosphere. In *ACM SenSys*, 2007.

[14] POST. Pacific ocean shelf tracking project. http://www.postcoml.org/.

[15] J. Rissanen. Generalized Kraft inequality and arithmetic coding. *IBM Jour. of Research and Dev.*, 20, 1976.

[16] N. Saunier and T. Sayed. Automated analysis of road safety with video data. In *Transportation Research Record*, pages 57–64, 2007.

[17] B. J. M. Stutchbury, S. A. Tarof, T. Done, E. Gow, P. M. Kramer, J. Tautin, J. W. Fox, and V. Afanasyev. Tracking long-distance songbird migration by using geolocators. *Science*, page 896, February 2009.

[18] J. Ziv and A. Lempel. Compression of individual sequences via variable-rate coding. *IEEE Transactions on Information Theory*, 24(5):530–536, 1978.