Multivariate Stream Data Reduction in Sensor Network Applications

Sungbo Seo[1,*], Jaewoo Kang[2], and Keun Ho Ryu[1]

[1] Dept. of Computer Science, Chungbuk National University, Chungbuk, Korea
{sbseo, khryu}@dblab.cbu.ac.kr
[2] Dept. of Computer Science, North Carolina State University, Raleigh, NC, USA
{kang}@csc.ncsu.edu

Abstract. We evaluated several multivariate stream data reduction techniques that can be used in sensor network applications. The evaluated techniques include Wavelet-based methods, sampling, hierarchical clustering, and singular value decomposition (SVD). We tested the reduction methods over the range of different parameters including data reduction rate, data types, number of dimensions and data window size of the input stream. Both real and synthetic time series data were used for the evaluation. The results of experiments suggested that the reduction techniques should be evaluated in the context of applications, as different applications generate different types of data and that has a substantial impact on the performance of different reduction methods. The findings reported in this paper can serve as a useful guideline for sensor network design and construction.

1 Introduction

A typical wireless sensor network (WSN) consists of small battery-powered wireless devices and sensors. Conserving battery power on such devices is crucial to improve the life span of a WSN. Among many operations that a sensor node performs, transmitting data among sensor nodes typically consumes the most energy. Many data reduction techniques have been proposed to address this problem [1, 2, 3]. However, different sensor networks have different data requirements depending on the types of applications they run and characteristics of data generated by different applications can be also different. Thus, such data reduction techniques need to be evaluated in the context of applications and the types of data they generate. In this paper, we attempt to identify such application specific requirements, and to propose different data reduction techniques for different types of application scenarios.

Three broad areas of sensor network applications are *environmental monitoring*, *object tracking*, and *object guarding* [4, 5, 6, 9]. First, examples of environmental monitoring are flood detection, home application and habitat monitoring. Long-term data analysis over low frequency data is usually used in this type of applications. Second, examples of object tracking include vehicle tracking, military applications and SCM (Supply Chain Management). These applications typically generate high

[*] Work performed while the author visited North Carolina State University.

T. Enokido et al. (Eds.): EUC Workshops 2005, LNCS 3823, pp. 198 – 207, 2005.

frequency multivariate data. Finally, examples of object guarding are emergency medical care, intrusion detection and earthquake risk assessment. These applications require real-time monitoring of outliers and detection of abnormality in the data. As we see here, different applications need different models for data acquisition, transmission, and storage. These need to be considered together with physical constraints such as limited bandwidth and power, and unreliable network, when the data reduction techniques are evaluated.

A typical sensor network example is shown in Fig. 1. A sensor node has one or more sensors. A node periodically collects data from its own sensors as well as data transmitted from other children sensor nodes. Thus, data collected by a sensor node naturally forms a multivariate time series. Previous researches on data acquisition and transmission have suggested data reduction techniques suitable for single or relatively small numbers of attributes [2, 7]. However, these techniques may not suitable for applications such as object tracking and guarding as they typically generate multivariate data with large numbers of attributes. This problem is even more exacerbated in sink nodes (see Fig. 1) where data generated by all sensor nodes in the network is collected and aggregated.

In this work, we studied efficient, multivariate approximate data transmission techniques as follows. First, we defined the hierarchical/distributed sensor network architecture and data model. Second, we classified application areas in wireless sensor networks, and then briefly introduced the multivariate data reduction techniques, such as Wavelet, HCL (Hierarchical Clustering), Sampling and SVD (Singular Value Decomposition). Finally, we experimented with data reduction methods with respect to relative error and reduction ratio.

The rest of the paper is organized as follows. Section 2 presents related work. Section 3 defines a hierarchical/distributed sensor network architecture and data model. In section 4 we suggest a simulation model and introduce some multivariate data reduction techniques. Section 5 reports the result of our experiments. Section 6 presents concluding remarks.

2 Related Work

Many previous work [1, 2, 3] in sensor networks studied data aggregation and approximate data transmission between sink nodes and base stations. Generally, data analysis and reduction techniques in sensor network include clustering, wavelet, histogram, regression, aggregation, sampling, PCA and SVD. Aggregation is an effective mean to get a synopsis (avg., max., min.), but is rather crude for applications that need detailed historical information [3]. Spectral models such as DWT, DFT and DCT are tuned for time sequence, ideally with a few low-frequency harmonics, but it is ineffective under the multi-dimensional attributes [11, 13]. Sampling has a good performance, but has some problems such as sampling ratio, relational join over arbitrary schemas and set-valued approximate queries [10, 11]. Clustering techniques for stream data is presented in [15] which analyzed the complexity and requirements of one-pass clustering over streaming data.

These previous work focused on solving the problems with intrinsic characteristics and limitations of sensor networks, but these techniques don't take into account the application specific requirements and different types of nodes with varying capabilities. In this paper, we evaluated the multivariate data reduction methods in the context of different applications. The findings reported in this paper can serve as a useful guideline for sensor network design and construction.

3 System Architecture

Hierarchical/Distributed organization is the most widely adopted model in sensor network [4]. Fig. 1 shows its architecture. Each type of nodes has the following characteristics.

- Sensor node gathers periodically the multivariate data collected from sensors or target nodes. Data transmission is done by a multi-hop or cluster-based communication method and not typically done by a point-to-point direct communication.
- Each sensor node has a small processor and main memory, and periodically sends the data to sink node by wireless communication. Sink node collects the data from the nodes and usually contains in-memory DBMS.
- Sink nodes transmit data to a base station through a wireless communication. Aggregated data collected in a base station can be stored in a server node for archiving and for serving historical queries spanning over long period of time.
- Server node and base station use an existing network infrastructure and have a traditional DBMS. Generally, a server node has a multi-dimensional data cube in order to serve aggregate queries efficiently.

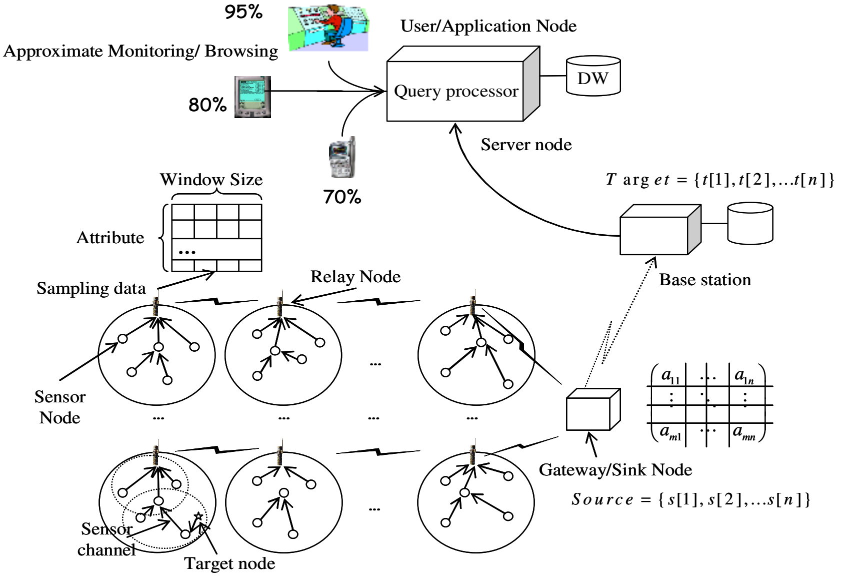


Fig. 1. General architecture and simulation model in wireless sensor network

As shown in Fig. 1, data collected by each sensor node is transmitted to the sink node. The sink node then temporarily stores the data for some time, and periodically sends the data to the base station. Data reduction is typically performed in this transmission because the size of aggregated data can be large, and depending on the applications, often times large, exact original data is out of favor to compact approximate summarization [8].

Communication between the base station and the server node typically use a wired network such as LAN, and hence the data transmission and reduction methods for these nodes should be considered differently. Unlike sensor and sink nodes, these nodes contain a powerful CPU, a large amount of memory, and reliable power sources. Efficient query processing over the large collection of aggregated data should be the more important consideration in these nodes. Similar to the transmission model, the query and data acquisition model also have to be determined according to the application requirements.

4 Multivariate Data Reduction Methods in Sensor Networks

We compared the multivariate data reduction methods, such as DWT (Discrete Wavelet Transformation), HCL (Hierarchical Clustering), Sampling, and SVD (Singular Value Decomposition) over different types of data generated from different application scenarios. In what follows, we present brief descriptions of the data reduction techniques and their characteristics.

DWT: The DWT is a linear signal processing technique using a hierarchical decomposition function. DWT is closely related to the DFT (Discrete Fourier Transform) and performs well with a low frequency data type. However, its performance degrades if data has several spikes or abnormal jumps [10, 13]. The advantages of DWT are the fast computation and small space complexity. A fast DWT algorithm has a complexity of *O(n)* for an input vector of length *n* [10]. Some researchers [7, 11] proposed the improved versions of the wavelet method, but it is still inefficient with the presence of multi-dimensional attributes.

HCL: Clustering is partitioning the objects into groups or clusters so that objects within a cluster are similar to one another and dissimilar to objects in other clusters [10, 15]. It can be used for data reduction as a group of similar objects in a cluster can be replaced with a single centroid. In order to cluster multivariate data set, in our experiments, we used the hierarchical clustering method using single, average and complete-linkage method. The HCL with multi-dimensional index tree can be used for hierarchical data reduction as well as for the fast approximate answers to queries.

Sampling: Sampling can be used as a data reduction technique since it allows a large data set to be represented by a much smaller random sample of the data [10, 11]. An advantage of sampling for data reduction is that the cost of obtaining a sample is proportional to the size of the sample. The complexity of sampling is potentially linear and we can easily control sampling rate according to the error ratio. But it is ineffective for ad-hoc relational joins over arbitrary schema and effectiveness for set-valued approximate queries is unclear [11].

SVD: SVD can be used for multivariate data reduction and is defined as follows.

Definition 1. (SVD): Given an *m×n* real matrix *X*, we can express it as $X=U\Sigma V^T$ where *U* and *V* are column-orthonormal and Σ is a diagonal matrix such that

$$U_{m\times m} = UU^T = U^TU = I\text{ , } V_{n\times n} = VV^T = V^TV = I \tag{1}$$

$$\Sigma_{m\times n} = [\Sigma]_{ij} = 0\text{ , } i \neq j\text{ , } [\Sigma]_{ii} = \sigma_i \geq 0\text{ , } \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min\{m,n\}} \tag{2}$$

Recall that a matrix *U* is called column-orthonormal if its columns u_i are mutually orthogonal unit vectors. So, U^T is equal to U^{-1} and $U \times U^T=I$, where *I* is the identity matrix. Σ is a diagonal matrix with values called singular values $\{\sigma_i\}$ in its diagonal. The rank *k* of *X* equals to the number of nonzero singular values of *X*. The SVD of $X=U\Sigma V^T$ can be illustrated as follows.

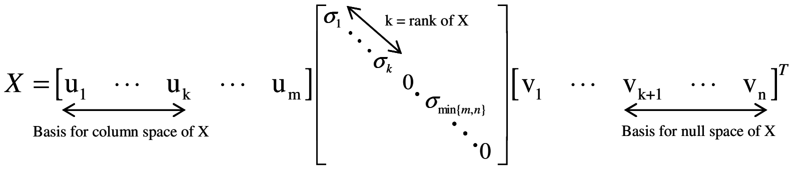

$X = [u_1 \cdots u_k \cdots u_m]$
Basis for column space of X
σ_1
k = rank of X
σ_k
0
$\sigma_{\min\{m,n\}}$
0
$[v_1 \cdots v_{k+1} \cdots v_n]^T$
Basis for null space of X

Fig. 2. Column space, rank and null space

As for the space complexity, the original matrix *X* contains *N×M* data elements while the SVD representation, after truncating to *k* principal components, will need $N\times k$ data elements for *U*, *k* data elements for the Eigen values, and *k×M* data elements for the *V* matrix. Thus, the reduced data to the original data ratio, *s_ratio,* is as follows [12, 13].

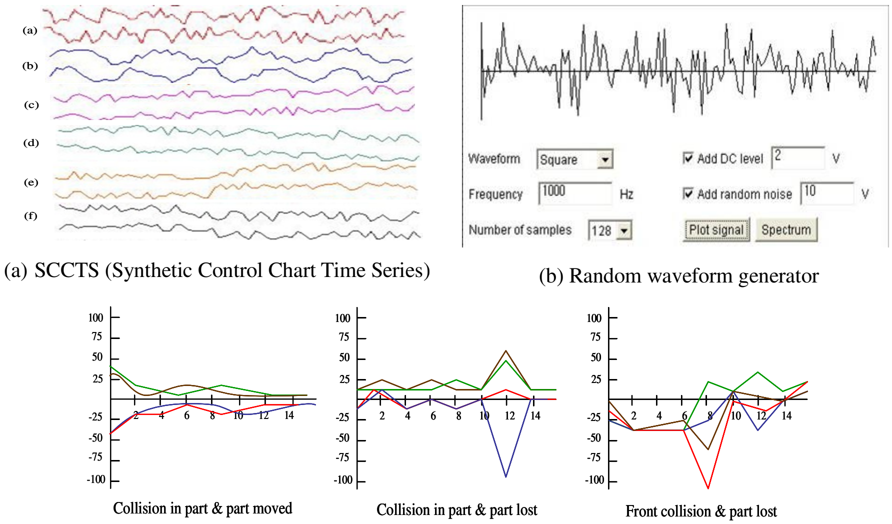
$$s\_ratio = \frac{N\times k + k + k\times M}{N\times M} \approx \frac{k}{M} \quad (N \gg M \geq k) \tag{3}$$

5 Experiments and Analysis

5.1 Data Sets

Our results are based on experiments over three data sets obtained from [16, 19]. The first data set, SCCTS (Synthetic Control Chart Time Series), contains 600 examples of control charts synthetically generated by the process introduced by Alcock and Manolopoulos in [16].

The SSCTS consists of the six different classes of control charts (Normal (a), Cyclic (b), Increasing trend (c), Decreasing trend (d), Upward shift (e), Downward shift (f)). The second data set include five synthetic data sets generated using the waveform generator. Each data set is created applying different combinations of parameters including waveform (one of sine, cosine, square, and saw-tooth), frequency (in Hz), DC level and random noise [19]. The third data set is the robot traces containing force and torque measurements on a robot moving an object from one location to another. Each movement is characterized by 15 force/torque samples collected at regular time intervals [14, 16].



Waveform Square

Frequency 1000 Hz

Number of samples 128

☑ Add DC level 2 V

☑ Add random noise 10 V

(a) SCCTS (Synthetic Control Chart Time Series)

(b) Random waveform generator

Collision in part & part moved

Collision in part & part lost

Front collision & part lost

(c) Examples of typical robot traces

Fig. 3. Data sets of multivariate time series and sensor data

In order to measure the relative error (σ) between the original matrix A and its approximation $\hat{A}$, we used the following metric.

$$\sigma = \frac{\left\|\hat{A} - A\right\|_F}{\left\|A\right\|_F}\text{, where } \left\|A\right\|_F = \left(\sum_{ij}\left|a_{ij}\right|^2\right)^{\frac{1}{2}} \quad (4)$$

In order to compute the relative errors for the reduction methods, we need to be able to recover the original matrix from its reduced form. The recovered matrix, $\hat{A}$, is an approximation of the original matrix, A, and has the same dimensionality as A. Computing $\hat{A}$ is straightforward for all reduction methods by their definition except the sampling method. We interpolated the sample points to approximate the missing values in the time points where the samples were not taken. For the experiments, we used the multivariate data reduction algorithms available from [17, 18] after some modification.

5.2 SSCTS Data (Data Size vs. Performance)

Fig. 4 shows the result of experiments where we compared the relative errors of the reduction methods over the range of different parameters. Fig. 4 (top left) compares the relative errors over the range of different data reduction ratios from 50% to 95% (e.g., 95% means the size of data after reduction is just 5% of the original). HCL was the worst performer while sampling showed the best performance. Fig. 4(top right) compared the reduction methods over the varying numbers of attributes (or dimensions) in the input data. For example, at x=50 (the first data point in the x axis), the algorithms are compared over data with 50 sensor readings in each time point. In this test and the next test (shown on the bottom left), we fixed the reduction ratio to 90%. As the figure shows, all methods are not affected much by dimension size.

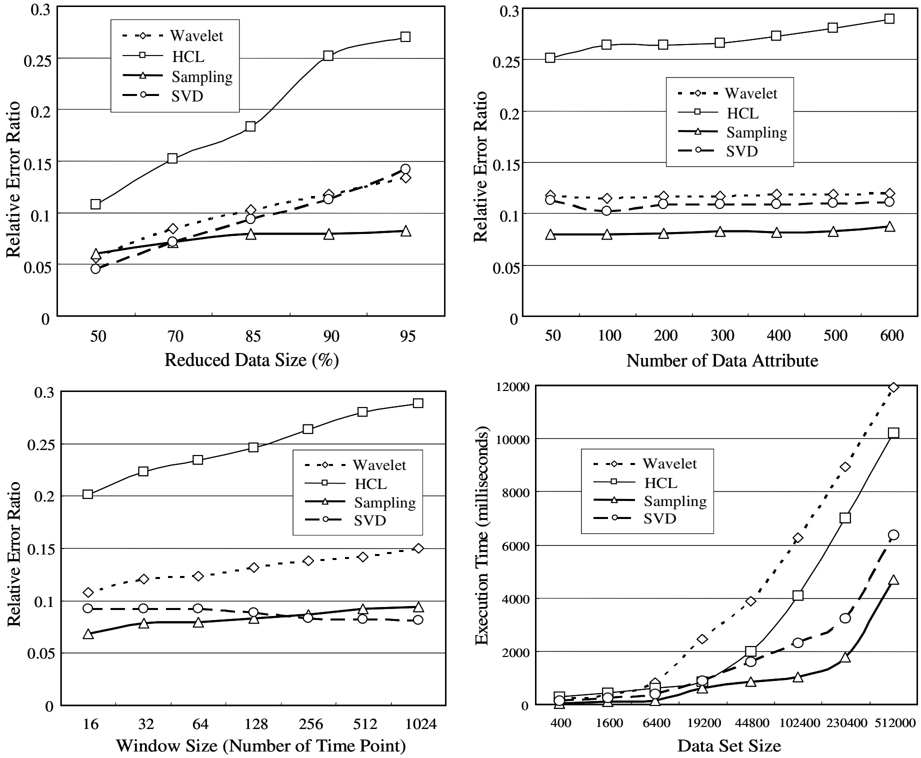


Fig. 4. Data size vs. Reduction methods performance

Fig. 4(bottom left) shows if the data window size has any influence on the performance of the methods. In each sensor node, data is accumulated for a while before transmitted to the node in the upper layer. The window size determines how much readings will be accumulated for each transmission. For example, if the window size is 10, then sensor readings are accumulated for 10 time points and transmitted as a unit. In this test, SVD showed a stable performance over the increasing window sizes while the others, especially HCL and Wavelet, showed increasing errors for larger windows.

Fig. 4(bottom right) compares the execution time for each method as the data size increases. This result shows that HCL and wavelet are more computationally expensive than others. Overall, sampling was superior to others for six different classes in SSCTS. Wavelet took longer than others and was susceptible to the increase of window size. SVD showed a reliable performance in most of the cases.

5.3 Synthetic Data (Data Type vs. Relative Error Ratio)

Fig. 5 compares the performance of the data reduction methods over the different types of data generated from different application areas. The synthetic data set generated from the waveform generator was used. In order to emulate the object tracking and object guarding scenarios, we inserted randomly generated outliers to the data. In this experiment, we fixed the data reduction ratio to 80% while varying the window size and the number of attributes. Fig. 5 (top left) shows the result with low frequency data set such as sine or cosine curves having low harmonic characteristics in the same attribute. All methods performed well in this test except HCL. HCL failed to produce comparable results.

Fig. 5 (top right) shows the result with the high frequency data set. HCL was the worst while sampling was the best. SVD and Wavelet performed reasonably well. Fig. 5 (bottom left) shows the result with the mixed input data with the ratio of high frequency to low frequency being 3:2. Fig. 5 (bottom right) shows the result with the data set containing outliers and abnormal patterns. SVD performed well while HCL and wavelet did not.

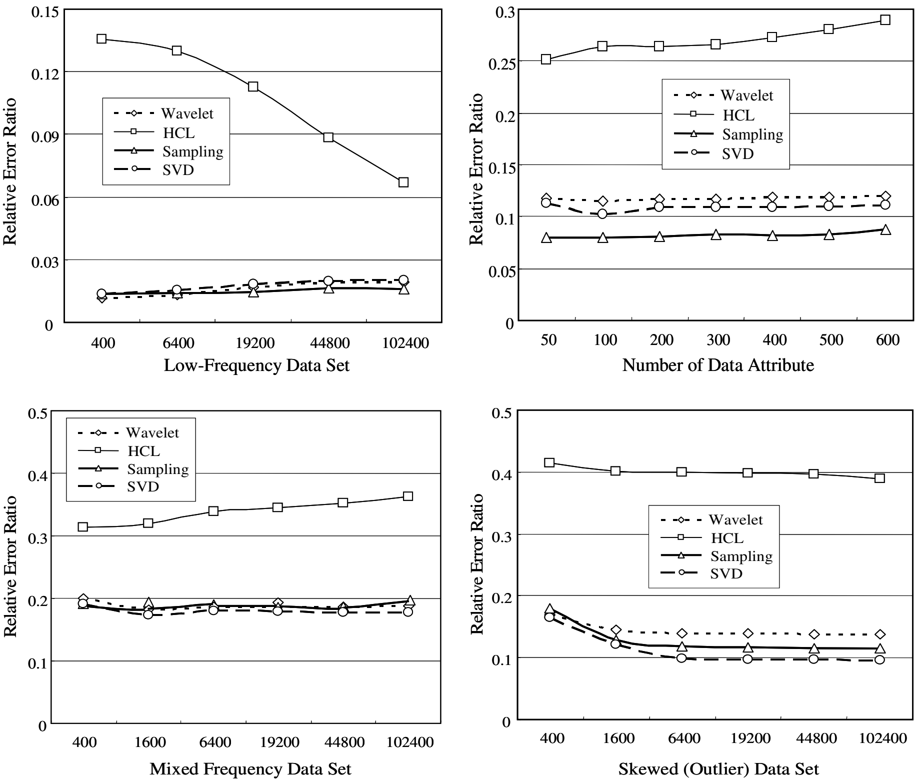


Fig. 5. Data types vs. Relative error ratio

5.4 Robot Trace Data (SVD vs. Adaptive Reduction)

Fig. 6 shows the results of experiments performed with the robot trace data (obtained from [14, 16]). In Fig. 6 (left), we compared the four methods over five different types of trace data including Normal, Collision, Obstruction, Lost, and Move as described in [16]. The reduction rate is fixed to 80% in this test. Overall, SVD showed more stable performance than others throughout the test. Fig. 6 (right) compares the SVD method (the best performer in the previous test) with the adaptive reduction method where we apply the reduction method adaptively for each window. The data set used in this test also has five different types of traces, represented as LP1 to LP5 as described in [16].

In this adaptive method, data in each window is first examined and the best reduction method for the given window is determined and applied. In order to implement this approach correctly, we need a classifier that predicts the labels for each window characterizing the properties of data in the window. Although it is an interesting and important area of research, exploring multivariate classifiers is out of scope of this paper. In our implementation of the adaptive approach, we simply assumed the correct labels for each window are given. As the result suggests, given an accurate classifier, we can achieve a significant improvement on the reduction performance over the static methods. We plan to investigate this adaptive reduction framework in our future work.

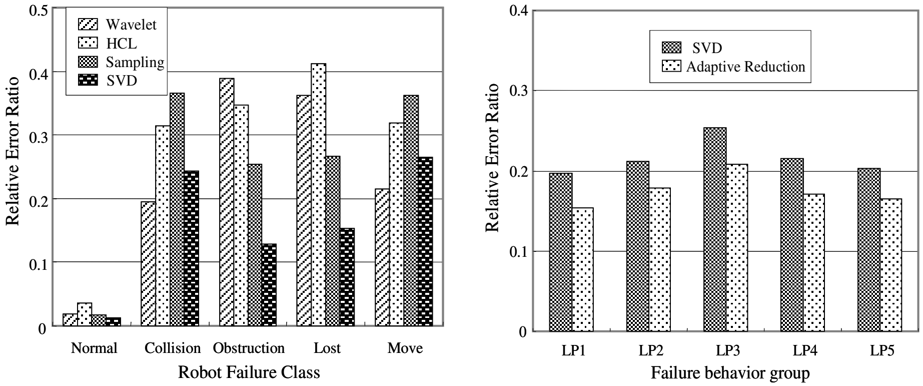


Fig. 6. Relative error ratio of robot failure behavior

6 Conclusion

We compared multivariate data reduction techniques that can be used in various sensor network applications, including wavelet, HCL, sampling and SVD methods, over both the real and synthetic time series data. We showed the relative performance of different methods vary over the data sets with different data characteristics. The findings reported in this paper can serve as a useful guideline for sensor network design and construction.

Acknowledgement. This work was partially supported by the RRC Program of MOCIE and ITEP, also ETRI (Telematics & USN Research Division) in Korea.

References

1. J. M. Hellerstein, W. Hong, and S. R. Madden.: The Sensor Spectrum: Technology, Trends, and Requirements. In SIGMOD Record (2003) 22-27
2. A. Deligiannakis, Y. Kotidis and N. Roussopoulos.: Compressing Historical Information in Sensor Networks. In SIGMOD (2004) 527-538
3. A. Deligiannakis, Y. Kotidis, and N. Roussopoulos.: Hierarchical in-Network Data Aggregation with Quality Guarantees. In EDBT (2004) 658-675
4. M. J. Franklin and S. R. Jeffery et al.: Design Considerations for High Fan-In Systems: The HiFi Approach. In CIDR (2005) 290-304
5. A. Mainwaring and J. Polastre et al.: Wireless Sensor Networks for habitat monitoring. In WSNA (2002) 88-97
6. B. X. and O. Wolfson.: Time-Series Prediction with Applications to Traffic and Moving Objects Databases. In MobiDE (2003) 56-60
7. S. Guha, C. Kim and K. Shim.: XWAVE: Approximate Extended Wavelets for Stream Data. In VLDB (2004) 288-299
8. A. Deshpande and C. Guestrin et al.: Model-Driven Data Acquisition in Sensor Networks. In VLDB (2004) 588-599
9. R. C. Oliver and K. Smettem et al.: Field Testing a Wireless Sensor Network for Reactive Environmental Monitoring. In ISSNIP (2004) 7-12
10. 10 J. Han and M. Kamber.: Data Mining Concepts and Techniques. Morgan Kaufmann Publishers (2000)
11. M. Garofalakis, and P. B. Gibbons.: Approximate Query Processing: Taming the Terabytes! In VLDB Tutorial (2001)
12. G Strang, Introduction to Linear Algebra, 3rd Edition, Wellesley-Cambridge Press (1998)
13. F. Korn, H. V. Jagadish and C. Faloutsos.: Efficient Supporting Ad Hoc Queries in Large Datasets of Time Sequences. In ACM-SIGMOD (1997) 289-300
14. L. M. Camarinha-Matos, L. S. Lopes, and J. Barata.: Assembly Execution Supervision with Learning Capabilities. In ICRA (1994) 272-279
15. S. Guha and N. Mishara et al.: Clustering Data Streams. In FOCS (2000) 359-366
16. S. Hettich, and S. D. Bay.: The UCI KDD Archive (Synthetic Control Chart Time Series, Robot Execution Failures) [http://kdd.ics.uci.edu]. Irvine, CA: University of California, Department of Information and Computer Science (1999)
17. JAMA: A Java Matrix Package: http://math.nist.gov
18. Multivariate Data Analysis Software: http://astro.u-strasbg.fr/~fmurtagh/mda-sw/
19. FFT Spectrum Analyzer: http://www.dsptutor.freeuk.com/analyser/SA102.html