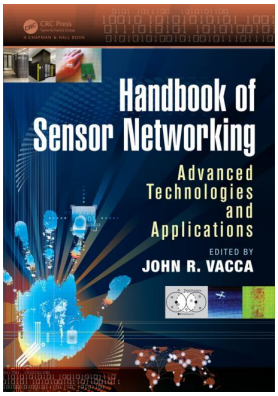

Handbook of Sensor Networking
Advanced Technologies and Applications

John R. Vacca

Data Mining in Sensor Networks

Publication details
https://test.routledgehandbooks.com/doi/10.1201/b18001-16
Sinjini Mitra, Pramod Pandya
Published online on: 13 Jan 2015

11

Data Mining in Sensor Networks

11.1 Introduction **11**-1
11.2 Data Mining: An Overview **11**-2
Core Ideas in Data Mining • Steps in Data Mining • Challenges in Data Mining
11.3 Data Mining in Sensor Networks **11**-4
Data Stream Clustering • Data Stream Classification • Frequent Pattern Mining
11.4 Time Series Sensor Streams **11**-6
Dimensionality Reduction • Compressions and Filtering • Forecasting
11.5 Social Sensing **11**-8
11.6 Distributed Algorithms **11**-9
11.7 Nonlinear Regression Using Choquet Integral **11**-9
11.8 Challenges in Sensor Data Mining **11**-11
11.9 Summary **11**-11
References **11**-12

Sinjini Mitra
California State University

Pramod Pandya
California State University

11.1 Introduction

Sensor network is a collection of addressable nodes of a data network, capable of capturing data in real time. In this sense, sensor network is a distributed computing and communication system or a resource. To our minds, the first and most obvious such a system would be the Internet, with a vast number of nodes designed for data processing, resource sharing, and communications. Hardware engineering advancement in last few years has given rise to devices with smaller and smaller footprint capable of collecting many different kinds of data. Of course, the sensor network that we all possess, and we cannot function without it, is our own brain, albeit not quite similar to the silicone-based distributed computing and communication system. In very recent years, devices with built-in global positioning satellite (GPS) have entered the arena of sensor network. We have now on our hand very smart and immensely capable sensor network with the ability to collect vast amounts of data, ready for processing, and guiding us in decision making.

Sensor data have become pervasive in recent years because of the popularization and wider availability of sensor technology that is cheap and easy to use. Sensors produce large volumes of data continuously over time, and this leads to numerous computational challenges in terms of data storage and data manipulation and analyses and retrieval. One major issue is *scalability.* The scalability challenges of sensor data have reached extraordinary proportions, with the increasing proliferation of ubiquitous and embedded sensors and mobile devices, each of which can potentially generate large streams of data. Coupled with the fact that many of these sensors are connected to the Internet, it is foreseeable that

in the near future, machine-generated data will dominate human-generated data by several orders of magnitude and this gap is only likely to increase with time [1]. In this context, the challenges associated with scalable and real-time management and mining of sensor data will potentially become even more significant in the coming years.

In the last couple years, we have a birth of a new discipline—data analytics. The pace of advancement in software technologies has not kept in pace with hardware technologies so far. So we have to address the following challenges:

- *Data management*: How do we store the data? What media do we use to store the data? Do we store all the data? How long should we store the collected data? What are the legal consequences of storing the data?
- *Sensor data mining and processing*: We have to address the need for new algorithms to process the data—of course, this requirement demands faster processors and memories. Do we process the sensor data in network or out of network?

Sensor data mining is relatively a new area of research. It involves collection, modeling, and processing of sensor data arising from several different types of sources. Often these are collectively referred to as *sensor data analytics*. The deluge of available data makes it possible to apply data mining techniques for obtaining a variety of useful analytical insights. The next section provides a brief outline of some basic data mining tools, and in the subsequent sections, we provide the application of these techniques to various types of sensor data. We conclude with a discussion of the current challenges to mining sensor networks and the emerging areas of research in this domain.

11.2 Data Mining: An Overview

The science of extracting useful information from large datasets or databases is known as *data mining*. A more elaborate definition of data mining according to Hand [2] is as follows:

> Data mining is the analysis of (often large) observational data sets to find unsuspected relationships and to summarize the data in novel ways that are both understandable and useful to the data owner.

Data mining is used in a variety of fields and applications, from the military and intelligence agencies to health-care and other business organizations. It is a relatively new discipline and is in a constant state of evolution, lying at the intersection of statistics, machine learning, data management, pattern recognition, artificial intelligence, and other areas. Several techniques for exploring and analyzing data have been around for a long time in the world of statistics such as regression methods (one of the most widely used analytical approaches even today), discriminant analysis, and analysis of variance, but many of these tools could not be implemented on today's huge datasets without adequate computational power and resources. Perhaps the most pertinent factor propelling the growth of data mining in recent times is the explosion of data, declining cost of massive data storage facilities and increasing availability of automatic data-capturing mechanisms such as sensors. Scannable bar codes, point-of-sale (POS) devices, mouse click trails, patient's health records, banking transactions, and GPS data are just a few examples. The mass retailer WalMart in 2003 captured 20 million transactions per day that was stored in a 1 TB database.

11.2.1 Core Ideas in Data Mining

In this section, we briefly outline some of the main data mining tools used today.

11.2.1.1 Classification

Classification is perhaps the most widely used data mining tool in most applications, used for categorical variables. A credit card transaction can be legitimate or fraudulent. A packet of data traveling on

a network can be benign or threatening. A common task in data mining is to examine data where the classification is unknown (say, we do not know whether a particular credit card transaction is fraudulent or not), with the goal of predicting what that classification is or will be. Similar data where the classification is known are used to develop rules, which are then applied to the data with the unknown classification. Some well-known techniques of classification include logistic regression, classification and regression trees (CARTs), *k*-nearest neighbors (*k*-NNs), naïve Bayes, neural networks, and linear discriminant analysis (LDA).

11.2.1.2 Prediction

Prediction is similar to classification, except here the goal is to predict the value of a numerical variable (e.g., selling price of a house) rather than a class (e.g., fraudulent transaction or not). Most of the aforementioned classification methods can also be suitably adapted to perform prediction although the most popular prediction tool is linear regression.

11.2.1.3 Clustering

The goal of clustering or cluster analysis methods is to segment a given set of data records into a set of homogeneous groups (called *clusters*) based on several measurements made on those records for the purpose of generating insight. It is very popular in business applications such as customized or targeted marketing and industry analysis. Common clustering approaches include hierarchical clustering and *k*-means clustering.

11.2.1.4 Association Rules

Large databases on customer transactions lend themselves naturally to the analysis of associations among items purchased or *what goes with what*. *Association rules* or *affinity analysis* can be used in a variety of ways. For example, grocery stores can use such information after a customer's purchases have all been scanned to print discount coupons and to help arrange and organize items on shelves for better chances at selling together. Online vendors such as Amazon.com and Netflix use these methods in their recommender systems that suggest new purchases to customers based on past ones.

11.2.1.5 Data Exploration and Reduction

A critical component of data mining and often the initial step is preprocessing and cleaning the data at hand. This involves reviewing and examining the data to identify important and relevant variables, detect outliers or missing and inaccurate data, and transform data (if necessary). Sensible data analysis often required distillation of complex data into simpler data. Rather than dealing with thousands of product types, a market researcher might want to aggregate them into a smaller number of groups. This process of consolidating a large number of variables (or cases) into a smaller set is termed *dimension reduction*. Such data processing and exploration also often provide insights into the types of data mining tasks required to answer the specific questions of interest.

11.2.1.6 Data Visualization

Another technique for exploring data is through graphical analysis or visualization methods. Such methods are very useful in not only understanding the behavior of the variables included in the study but forming an initial idea about relationships between several variables. Some common traditional visualization tools are histograms, bar and pie charts, scatterplots, boxplots, and line plots (for time series data). Some more novel methods include heat maps; interactive plots involving zooming, panning, and filtering; treemaps; network plots; and map charts, among others.

Data mining techniques fall broadly in two categories: (1) *supervised learning algorithms* and (2) *unsupervised learning algorithms*. In supervised methods, an outcome or response variable is available and learning occurs based on training data (where the outcome values are known). Once the algorithm is trained, it is applied to another set of data, called validation data, where the outcome is

unknown and needs to be determined. Classification and prediction methods fall in the domain of supervised learning tools. Unsupervised learning algorithms, on the other hand, are those used when there is no outcome variable to predict or classify. Hence, there is no learning or training from cases where such outcomes are known; instead, the task here is to unravel and study the underlying patterns in the dataset. Association rules, cluster analysis, visualization methods, and dimension reduction techniques are all unsupervised learning methods.

11.2.2 Steps in Data Mining

Here is a list of steps to be followed in a typical data mining effort:

1. Develop an understanding of the purpose of the data mining project.
2. Obtain the dataset to be used in the analysis (often sampling techniques and database retrieval methods are employed for this).
3. Explore, clean, and preprocess the data (includes data visualization as well).
4. Reduce the data (if necessary).
5. Determine the data mining task.
6. Choose the data mining techniques to be used.
7. Use algorithms to perform the task.
8. Interpret the results.
9. Deploy the model.

11.2.3 Challenges in Data Mining

The main challenges in the area of data mining include data management and computational efficiency. With the availability of more and more data, it is imperative to have adequate data management and storage facilities. Often real data are unstructured and complex in nature (e.g., health-care data), and unless these data can be processed in such a way that they are amenable to the existing data mining algorithms, the task at hand cannot be accomplished. Therefore, the rapid and continuing improvement in computing capacity and data handling mechanisms are essential enablers of the growth of the field of data mining in the current years. Another challenge is the choice of the correct method or ensemble of methods to use in a particular scenario. There are numerous data mining methods available today, so an in-depth understanding of the problem at hand and correct formulation of the research questions are extremely important to help select the appropriate tool. Often multiple techniques are employed and then results are compared to assess the best one suitable for the given data. Caution should also be exercised in accurately interpreting the results so that people with no statistical and computing background are also able to develop an understanding of the insights gained from the data mining activity.

Some of the emerging research areas of application in data mining include data from GPS devices, social networks, clickstreams (on the Internet), and surveillance videos (at airports, say), among others. More and more algorithms and software packages are also emerging in the market that help carry out data mining tasks, such as SAS, SPSS (owned by IBM now), Tableau (visualization software), and XLMiner (Microsoft Excel Add-In). All of the data mining tools outlined in this section are elaborated in great detail in [2].

11.3 Data Mining in Sensor Networks

In recent years, there has been an explosive growth in the amount of data generated by sensor networks in different arenas. Hence, data mining and analytical tools are also constantly evolving to be able to deal with these massive datasets. One area of growth has been in the area of mathematical and statistical

model-based techniques, such as time series models and Markov models. Particularly, when the volume of data is very large, it leads to a number of computational and mining challenges:

- As the volume of the data increases, it becomes increasingly more difficult to process them efficiently with multiple passes. Thus, one data item has to be processed at one time. This leads to implantation problems for the existing algorithms, and they need to be redesigned.
- In most cases, there is an inherent temporal component in mining data streams arising from sensor networks, which tend to evolve over time. Data mining techniques thus need to be designed carefully so as to be able to handle the temporal variations in the underlying data.
- Data collected from sensors are often noisy and error prone; hence, it calls for tools to reduce the degree of uncertainty in the mining tasks. Much of the errors occur in the transmission stage and can often have incompleteness (e.g., battery of the GPS system runs out).
- Data from sensor networks often need to be analyzed in a distributed fashion; hence, data mining methods such as clustering and classification also need to be adapted to meet these requirements.

In the next sections, we outline some data mining tools for sensor data and the associated issues and challenges.

11.3.1 Data Stream Clustering

Clustering is a popular data mining technique that helps in learning patterns in a dataset in an unsupervised manner. However, it is difficult to adapt these traditional clustering methods to data streams from sensor data because of one-pass constraints discussed earlier.

An interesting adaptation of the *k*-means algorithm was discussed in [3], which uses a partitioning-based approach to create clusters over the entire data stream. However, in certain applications in practice, it might be necessary to be able to examine clusters over specified time intervals. For example, an analyst may wish to study the behavior of clusters in the data stream over the past week or the past month in order to fully understand underlying data behavior and perform comparative analysis. One such technique is microclustering [4] in which first-order and second-order moments of the data are tracked via feature vectors. These in turn help in calculating important cluster characteristics such as centroids in real time. Other authors [4] clearly demonstrated that this method is more effective than the partitioning-based approach. A couple of examples of data applications for this technique are (1) high-dimensional data [5], (2) data with uncertainty [6], (3) text data [7], and (4) categorical data [7]. For both text- and categorical-type data, counts of frequencies of discrete attributes as well as correlations are stored instead of the moments as in the case of quantitative data. A number of density-based clustering approaches are also available for stream clustering [8,9].

Another type of sensor data arises in a distributed setting where large volumes of data are collected separately at different sensors. In such a case, the natural approach is to transmit all the data to a centralized server, a phenomenon that significantly raises costs. Moreover, computation becomes harder too. A method proposed in [10] performs local clustering at each node and then merges these different clusters into one single big cluster. A second method [10] is also mentioned for distributed clustering, which is called the *parallel guessing algorithm*. Another method for distributed sensor stream clustering that reduces the dimensionality and cost by maintaining an online discretization may be found in [11].

11.3.2 Data Stream Classification

Classification is one of the most widely applied data mining techniques applied in case of sensor data streams. Owing to the temporal component of stream data, some adjustments are required

for traditional classification methods. The concept of stream evolution is sometimes referred to as *concept drift* in the stream classification literature [12]. Of all available methods, the following are most popular:

1. *Very fast decision tree (VFDT) method*: This method is adapted from the classic decision tree method with the use of sampling-based approximations. These are so designed as to be able to handle evolving data streams by using sliding windows to update the classifier at each step. Furthermore, the VFDT algorithm [13] has been extended to process numerical attributes and reduce the sample size.
2. *On-demand classification*: This method focuses on the case when both the training and the test streams evolve over time and work by creating class-specific microclusters from the underlying data [14]. For an incoming record in the test stream, the class label of the closest microcluster is used to determine the class label of the test case.
3. *Ensemble-based classification*: This technique [15] uses an ensemble or combination of classification methods such as C4.5 (decision trees) and naïve Bayes in order to enhance the classification accuracy. This method works best if the data behave differently over time and a different method produces the optimal results in each instance. Thus, the use of multiple methods together provides the required robustness as well as increased accuracy.
4. *Compression-based methods*: Such methods typically work by the use of compression techniques applied to real-time classification of streaming sensor data, wherein time series bitmaps are updated in constant time [16]. This makes these classifiers very efficient in practice.

11.3.3 Frequent Pattern Mining

The problem of frequent pattern mining in data streams consists of finding the frequent item sets either over a sliding window or the entire data stream [17]. For the *entire data stream model*, the frequent patterns are mined over the entire data stream. The main difference from a conventional pattern mining algorithm is that the patterns need to be mined in one pass over the entire data stream. In the *sliding window model*, on the other hand, the evolution of data over time is accounted for by determining frequent patterns over a particular sliding window. A method for determining the frequent patterns over a sliding window is included in [18]. The primary focus of the algorithm is to detect the closed frequent item sets over the entire data stream. The proposed algorithm is called *moment*, and the primary underlying idea of this technique is based on the fact that the boundary between the closed frequent item sets and frequent item sets moves slowly. The reader is referred to [19] for a more in-depth review of the various clustering, classification, and pattern mining techniques for mining sensor data.

11.4 Time Series Sensor Streams

The main component of time series sensor streams is correlation; hence, analytical techniques for this purpose have gained a lot of popularity. The problem consists of capturing correlations both across multiple streams (e.g., prices in the same market) and in single streams across time (autocorrelations). The latter help capture periodic trends in time series streams as well since values at different times are usually not independent. Both these problems are inherently related although the results are interpreted slightly differently in each case.

The main data mining techniques applicable in case of time series sensor streams are dimensionality reduction, filtering, and forecasting. These are all closely related, and all three areas are too extensive to cover within this chapter. So we discuss each of these tools in brief details in the next few subsections and provide appropriate references.

11.4.1 Dimensionality Reduction

Since data arising from sensor networks are typically of large dimensions, it is necessary to perform some form of dimension reduction in order to consolidate the set of variables to be analyzed. Dimension reduction is thus an important aspect of data mining for all applications and is often the first step in the process following data cleaning and preprocessing. There are several methods for dimension reduction such as subset selection and floating search, but the most popular one is *principal component analysis* (PCA). PCA works particularly well in case we have subsets of measurements (predictors in case of predictive models) that are highly correlated. In that case, it provides few variables that are weighted linear combinations of the original variables that retain the explanatory power of the full set. One drawback of this tool is that it is applicable only for numerical or quantitative variables and does not work with categorical or qualitative variables. For a detailed overview of the PCA technique, the reader is referred to [2].

11.4.2 Compressions and Filtering

Initial work on time series representation used Fourier transforms [20,21]. More recent work focuses on fixed, predetermined bases or approximating functions. A PCA [22] approximates the time series with piecewise constant or linear functions. DAWA [23] combines the discrete cosine transform (DCT) and discrete wavelet transform (DWT). However, all these techniques enable compression of the time series for indexing purposes and not for knowledge discovery. The seminal work found in [24], for rule discovery in time series, is based on sequential patterns extracted after discretization. More recently, vector quantization has been used for time series compression [25]. The author in [26] has presented ways to efficiently store time series as well as facilitate easy computation of correlations and application of other diagnostic tools (like graphical techniques, for instance) after compression. This is accomplished via multiscale analysis that yields sparse time and frequency representations of the series.

The recently developed theory of compressed sensing [27,28] studies the problem of signal summarization and reconstruction based on a subset of observed values. Specifically, the framework helps to estimate projections of a signal into a set of given basis functions from a small sample of its values.

11.4.3 Forecasting

Forecasting is probably the most important component of time series analysis, whereby the future value is estimated based on past ones from historical data. Probably the most popular and widely used forecasting method is based on *autoregressive* (*AR*) models. The information found in [29] contains details about all these time models. The main idea is to express the observation at time t, namely, x_t as a function of its previous values plus noise:

$$\mathbf{x}_t = \phi_1 \mathbf{x}_{t-1} + \phi_2 \mathbf{x}_{t-2} + \cdots + \phi_W \mathbf{x}_{t-W} + \epsilon_t,$$

where

W is the forecasting window size
ϕ_i's are parameters of the model
ϵ represents the noise

The parameters are estimated from the data, and then past values of the time series sensor are used to forecast a future value using this equation. Some common estimation methods include ordinary least squares, method of moments (Yule–Walker equations), or Markov chain Monte Carlo (MCMC) methods.

Another variant of the AR model is the *autoregressive moving average* (ARMA) model that is also used widely for forecasting in time series data. The model consists of two parts: an AR part and a moving average (MA) part. The model is usually then referred to as the ARMA(p, q) model where p is the order of the AR part and q is the order of the MA part. MAs are used for smoothing a time series and work

by calculating the average of successive values of the series using a moving window. ARMA models in general can, after choosing p and q, be fitted by least squares regression to find the values of the parameters, which minimize the error term. It is generally considered good practice to find the smallest values of p and q that provide an acceptable fit to the data. For a pure AR model, the Yule–Walker equations may be used to provide a fit. Finding appropriate values of p and q in the ARMA(p, q) model can be facilitated by plotting the partial autocorrelation functions for an estimate of p and likewise using the autocorrelation functions for an estimate of q. Further information can be gleaned by considering the same functions for the residuals of a model fitted with an initial selection of p and q. Many statistical software packages like R, MATLAB®, SAS, and STATA have built-in functions for fitting these models.

The reader is referred to [29] for details about the implementation of these forecasting models. Furthermore, the information found in [30] contains a detailed overview of all the data mining tools applicable to time series sensor streams along with associated issues and challenges.

11.5 Social Sensing

A number of sensor applications today collect data that can be directly associated with human interactions. This is driven in part by the explosive growth of social networking sites in the recent years, such as Facebook, Twitter, LinkedIn, and Google+ to name a few. Facebook has 1.2 billion monthly users as of January 2014 [31], Twitter reports 200 million active users as of February 2013 [32], and this has given users the ability to easily share information online by connecting individuals and groups. Moreover, all these technologies are available on mobile devices (smartphones, tablets, etc.) today in the form of mobile apps, which makes them easily accessible on the go. These networks are data rich and contain a lot of structure that can be mined effectively to gain valuable insights into behavioral patterns [33,34]. A natural way to enhance the power of such social applications is to embed sensors within these platforms to continuously collect large amounts of data for prediction and other mining purposes. A few applications are as follows:

1. *Vehicle tracking applications*: Much data are available from real-time tracking of locations of automobiles today via GPS. This can provide other drivers with important information as to how to avoid points of traffic congestion in the city. This is also important for military vehicles, which often require location coordination. One example of vehicular application can be found in [35], which proposed the *GreenGPS* system.
2. *Health-care applications*: In recent years, numerous medical sensor devices are being used to track the personal health of individuals or make recommendations regarding their lifestyle. This is highly useful for deploying emergency responses; for making long-term predictions about chronic diseases such as high blood pressure, cholesterol, asthma, and diabetes; and for providing suggestions about healthy living habits like diet control and exercises [36].

Social sensing applications provide numerous research challenges from the perspective of analysis, some of which are listed below:

- The main concern in social networks is regarding privacy since a lot of personal data are shared over these. Hence, suitable privacy-control mechanisms need to be embedded in these systems that gather the data.
- Most sensors operate on batteries, which often have limited life. Certain types of sensor data collection can drain the battery life more quickly than others (e.g., GPS, mobile phones). Therefore, it is critical to design the applications with the underlying trade-offs, so that the battery life is maximized without significantly compromising the goals of the application.
- The volume of data can be very large, especially those that arise from real-time continuous tracking such as GPS and social media sites. Moreover, these data are often unstructured and do not conform to any known standards that can be analyzed using traditional software. Thus, appropriate techniques and software packages are required to store and process these efficiently. There are many advances made in this area today with the advent of cloud computing capabilities

and platforms for handling *big data* (e.g., HADOOP). Besides, sometimes it is necessary to mine sensor data in a dynamic fashion, providing real-time output (e.g., credit monitoring to trigger alerts in the event of fraudulent activities), which increases associated challenges significantly.

- Sensor data are often error prone and hence there are several challenges about dealing with *trustworthiness* of the collected data.

Once the data are collected and stored properly in databases and data warehouses, common data mining techniques can be applied, such as clustering, association rules, prediction, and classification techniques to make inferences. If predictions are required over time, time series forecasting methods described in the earlier section can be employed. One tool that is increasingly becoming popular is the use of *text analytics* and *sentiment analysis* [37] for mining all the social interactions and opinions expressed in social media sites to understand consumer behavior and preference about specific brand products—an extremely important application of these outcomes being in marketing.

11.6 Distributed Algorithms

As we have stated earlier, the sensor network is an assembly of processing nodes with small footprint. The primary reason for small footprint is so that the demand for power requirement is low enough. There is also need for power requirement for transmission of data from the sensor nodes to central data storage. Do we require sensor nodes to transmit the data as they collect them, or some processing should take place at the node, so as to eliminate and discard the data that would be redundant due to being corrupt as a consequence of presence of noise? Of course, if no processing takes place at nodes, then large bandwidth would be required to transmit all the collected data in real time. Apart from this, there too would be duplication of data collected from each of the sensor as well. Hence, we need some sort of filtering mechanism to eliminate duplicated and redundant data. We could summarize the sensor data network as distributed communication system—which necessitates a distributed algorithm [38]. Sensor nodes can be configured to act either in a synchronous or an asynchronous mode. In a synchronous mode, sensor nodes would have a global view; therefore, nodes would exchange messages with one another as they collect data and route the data after some local processing. So we need a routing algorithm to this end. In an asynchronous mode, each sensor node would operate independently of one another, collect data, and route the data to a central depository.

We could model the sensor data network using graph theory, $G = (V, E)$ with vertices $V = \{v_1, v_2, v_3, \ldots\}$ and edges $E = \{e_1, e_2, e_3, \ldots\}$, where each of the vertex of the graph is the sensor node and the edges of the graph are the communication links. We could break up the sensor data network, into groups of smaller data networks, or we could build larger sensor data network from groups of smaller sensor data networks. Then we could apply techniques from sheaf theory to build consistency between pieces of local information (i.e., groups of smaller sensor data networks) to arrive at global inference (corresponding to larger sensor data network in question) [39].

11.7 Nonlinear Regression Using Choquet Integral

The Choquet integral (CI) is a tool for the information fusion that is very effective in the case where fuzzy measures associated with it are well chosen [40,41]. A new approach for calculating fuzzy measures associated with the CI was proposed in a context of data fusion in multimodal biometrics. CI is a tool for the information fusion, which can generalize many operators such as the ordered weighted averaging, the arithmetic sum, the minimum, and the maximum.

Information collected from diverse sources is aggregated in a standard way by

$$y = w_1 f(x_1) + w_2 f(x_2) + \cdots + w_n f(x_n),$$

where y represents the weighted sum and that $\sum_{i=1}^{i=n} w_i = 1$.

In databases, the information sources $x_1, x_2, \ldots, x_n$ are to be regarded as attributes and $f(x_1), f(x_2), \ldots, f(x_n)$ are their observed values. Thus, the weighted sum represents the Lebesgue integral on the set of information sources and represents a linear aggregation model. Such a linear model is applied in multi-objective decision modeling and classification. In using the linear model, a fundamental assumption is made that there is no interaction among the contributions from individual attributes towards a certain target.

Regression is one of the most often used tool in statistical data analysis. Regression helps to determine if a relationship exists between observational data (predictive attributes) and a target attribute. This relationship could be either linear or nonlinear, defined by a set of unknown parameters. Once this relationship is determined, then one predicts the value of the target variable if a new set of predictive variables has been collected.

Given a set of $n + 1$ attributes $x_1, x_2, x_3, \ldots, x_n$ and y is the target attribute, then we want to determine the relationship between the x's and the y. We regard y as a random variable and express a linear relationship as noted in the following:

$$y = a_0 + a_1x_1 + a_2x_2 + a_3x_3 + \cdots + a_nx_n + N\left(0,\sigma^2\right),$$

where

$a_0, a_1, a_2, a_3, \ldots, a_n$ are the unknown regression coefficients
$N(0, \sigma^2)$ is a random variable with mean zero and unknown variance σ^2

In such a model, the underlying assumption is that there are no interactions among the $x_1, x_2, x_3, \ldots, x_n$ towards the target y. The unknown parameters $a_0, a_1, a_2, a_3, \ldots, a_n$ are determined through the usual least square method that minimizes the total squared error.

In the earlier discussion, the symbol x_i is used to denote an attribute; if it were to represent an observation, then we may use $f(x_i)$. In such a case, the multiregression model can be written as

$$a_1f\left(x_1\right) + a_2f\left(x_2\right) + a_3f\left(x_3\right) + \cdots + a_nf\left(x_n\right) = \int f d\mu,$$

which represents a weighted sum of values of function f on a set X and μ represents a classical measure. Hence, the multiregression model is expressed as

$$y = \int f d\mu + N\left(0,\sigma^2\right).$$

If interactions among the attributes towards a certain target are to be included, then the weighted sum cannot be the correct representation. Instead we need to use the nonadditive set function, such as the CI, which is a generalization of Lebesgue integral and coincides with the Lebesgue integral when the nonadditive measure is replaced by the additive measure. The CI is considered to be a type of nonlinear integral.

Definition 11.1

Let f be a nonnegative measurable function on $(X, \mathcal{F})$ and $E \in \mathcal{F}$. The CI of f on E with respect to monotone measure μ, denoted by $(C)\int_E f d\mu$, is defined as

$$(C)\int_E f d\mu = \int_0^\infty \mu\left(F_\alpha \cap E\right) d\alpha,$$

where $F_\alpha = \{x | f(x) \geq \alpha\}$, called the α-level set of f, for $\alpha \in [0, \infty)$.

If the set function is б-additive, the preceding definition is equivalent to the definition of Lebesgue integral of *f* with respect to μ. For a nonlinear multiregression model:

$$y = c + (C)\int_E (a + bf)d\mu + N\left(0, \sigma^2\right),$$

where

c represents a constant
a and *b* represent a real-valued function on *X*
f represents an observation of $x_1, x_2, x_3, \ldots, x_n$
μ is a measure on *X*

11.8 Challenges in Sensor Data Mining

As clearly evident in the earlier sections, data mining in sensor network has several technical challenges in the form of data processing, communication, and sensor management. First of all, there are huge volumes of data available from these networks, most often on a continuous basis. Secondly, there is significant noise frequently present in such data. Although much advance has occurred today in data storage and computational efficiency, there are still challenges in the form of handling so much data in a dynamic fashion that is required in case of several sensor network applications such as social networks, GPS systems and surveillance cameras at airports, and ATMs that are used for security purposes. This is sometimes referred to as the *big data problem* in the context of analytic applications. Because of potentially harsh uncertain and dynamic environments, along with energy and bandwidth constraints, wireless ad hoc networks pose additional technical challenges in network discovery, network control and routing, collaborative information processing, querying, and tasking [42].

Another daunting challenge facing sensor data mining is regarding where the processing of the gathered data takes place. If sensor data have to be processed out of network, then a fast communication link is required to transmit the data from the sensor network to a central storage media. Of course, in such a scenario, duplicate and even corrupt data could get transmitted, leading to a waste in bandwidth usage. Solution to this problem would be a local processing at the sight of the network—filter out duplicate and corrupt data and transmit well-behaved data. We need efficient and fast algorithms to this end.

Moreover, the application of data mining to sensor data requires choice of the appropriate methods, choice of appropriate parameters, and interpretation of results that often are not straightforward and require the expert skills. It is imperative to understand the importance of applying the right technique to each type of data in order to make the correct business decisions. Ad hoc and heuristic methods should be avoided and caution should be exercised in mining sensor network data. Furthermore, cost and privacy issues, as mentioned in context of some of the aforementioned specific mining tasks, are present as well and need to be taken into account.

11.9 Summary

Sensor networks and hence sensor data are ubiquitous now, and so are data analytics or data mining methods. The marriage of these two leads to empowering insights about the phenomenon at hand. Sensor data are varied and often arrive dynamically in large volumes, so an understanding of the underlying issues and challenges is crucial to processing and analyzing them. As technology become more and more advanced, with several platforms available today for handing complex data forms, application of data mining tools to sensor data is becoming more and more popular and widespread in different areas. The future thus holds tremendous promise in terms of the power of information gained from

analyzing such data. Google has sold a limited number of wearable computers with an optical head-mounted display. Users can interact with the device through a touch pad or voice-activated mechanism, to access the e-mail, browse the web, do navigation, record the video, and use the webcam. Google glass has catapulted sensor data technology to a new dimension. The next generation of information technology would be carbon-based sensor data network, processing and transmitting sensor data in real time! The presently available information technologies sooner would be antiquated, and we would have to continue to spin the wheels of technologies endlessly, and such will be our fate for eternity!

References

1. Lohr, S. (2012). The age of Big Data. *The New York Times*, Sunday Review, Appeared on February 12, 2012. http://www.nytimes.com/2012/02/12/sunday-review/big-datas-impact-in-the-world.html?pagewanted=all. Accessed February 10, 2014.
2. Shmueli, G., Patel, N.R., Bruce, P.C. (2010). *Data Mining for Business Intelligence: Concepts, Techniques, and Applications in Microsoft Office Excel with XLMiner*. Wiley, Hoboken, New Jersey.
3. Guha, S., Mishra, N., Motwani, R., O'Callaghan, L. (2000). Clustering data streams. *IEEE FOCS Conference*, Redondo Beach, CA.
4. Aggarwal, C.C., Han, J., Wang, J., Yu, P. (2003). A framework for clustering evolving data streams. *VLDB Conference*, Berlin, Germany.
5. Aggarwal, C.C., Han, J., Wang, J., Yu, P. (2004). A framework for high dimensional projected clustering of data streams. *VLDB Conference*, Toronto, Canada.
6. Aggarwal, C.C., Yu, P. (2008). A framework for clustering uncertain data streams. *ICDE Conference*, Cancun, Mexico.
7. Aggarwal, C.C., Yi, P. (2006). A framework for clustering massive text and categorical data streams. *SIAM Data Mining Conference*, Bethesda, MD.
8. Cao, F., Ester, M, Qian, W., Zhou, W. (2006). Density-based clustering of an evolving data stream with noise. *SIAM Data Mining Conference*, Bethesda, MD.
9. Chen, Y., Tu, L. (2008). Density-based clustering for real-time stream data. *ACM KDD Conference*, Las Vegas, NV.
10. Cormode, G., Muthukrishnan, S., Zhuang, W. (2007). Conquering the divide: Continuous clustering of distributed data streams. *ICDE Conference*, Istanbul, Turkey.
11. Rodrigues, P., Gama, J., Lopes, L. (2008). Clustering distributed sensor data streams. *PKDD Conference*, Antwerp, Belgium.
12. Domingos, P., Hulten, G. (2000). Mining high-speed data streams. *Proceedings of ACM KDD Conference*, Boston, MA.
13. Jin, R., Aggarwal, C.C. (2003). Efficient decision tree construction on streaming data. *ACM KDD Conference*, Washington, DC.
14. Aggarwal, C.C., Han, J., Wang, J., Yu, P. (2004). On-demand classification of data streams. *ACM KDD Conference*, Seattle, WA.
15. Wang, H., Fan, W., Yu, P., Han, J. (2003). Mining concept-drifting data streams using ensemble classifiers. *ACM KDD Conference*, Washington, DC.
16. Kasetty, S., Stafford, C., Walker, G., Wang, X., Keogh, E. (2008). Real-time classification of streaming sensor data. *ICTAI Conference*, Sacramento, CA.
17. Giannella, C., Han, J., Pei, J., Yan, X., Yu, P. (2002). Mining frequent patterns in data streams at multiple time granularities. *Proceedings of NSF Workshop on Next Generation Data Mining*, Baltimore, MD.
18. Chi, Y., Wang, H., Yu, P., Muntz, R. (2004). Moment: Maintaining closed frequent item sets over a stream sliding window. *ICDM Conference*, Brighton, UK.
19. Aggarwal, C.C. (2013). Mining sensor data streams. In C. Aggarwal (ed.), *Managing and Mining Sensor Data*. Springer, New York.

20. Agarwal, R., Faloutsos, C., Swami, N.A. (1993). Efficient similarity search in sequence databases. *FODO*, Chicago, IL.
21. Faloutsos, C., Raghunathan, M., Manolopoulos, Y. (1994). Fast subsequence matching in time-series databases. *SIGMOD*, Minneapolis, MN.
22. Chakrabarti, K., Keogh, E., Mehrotra, S., Pazzani, M. (2002). Locally adaptive dimensionality reduction for indexing large time series databases. *ACM Transactions on Database Systems (TODS)*, 27(2), 188–228.
23. Hsieh, M.J., Chen, M.S., Yu, P.S. (2005). Integrating DCT and DWT for approximating cube streams. *CIKM*, Bremen, Germany.
24. Das, G., Lin, K.I., Mannila, H., Raghunathan, G., Smyth, P. (1998). Rule discovery from time series. *KDD*, New York, NY.
25. Lin, S., Gunopulos, D., Kalogeraki, V., Lonardi, S. (2005). A data compression technique for sensor networks with dynamic bandwidth allocation. *TIME*, Burlington, VT.
26. Reeves, G., Liu, J., Nath, S., Zhao, F. (2009). Managing massive time series streams with multi-scale compressed trickles. *VLDB Conference*, Lyon, France.
27. Donoho, D. (2006) Compressed sensing. *IEEE TOIT*, 52, 1289–1306.
28. Haupt, J., Nowak, R. (2006). Signal reconstruction from noisy random projections. *IEEE TOIT*, 26, 4036–4048.
29. Borckwell, P.J., Davis, R.A. (1991). *Time Series: Theory and Methods*. Springer Series in Statistics, 2nd edn. Springer-Verlag, New York.
30. Papadimitriou, S., Sun, J. (2013). Dimensionality reduction and filtering on time series sensor streams. In C. Aggarwal (ed.), *Managing and Mining Sensor Data*. Springer, New York.
31. Wikipedia page on Facebook. http://en.wikipedia.org/wiki/Facebook.
32. Wikipedia page on Twitter. http://en.wikipedia.org/wiki/Twitter.
33. Clauset, A., Newman, M.E.J., Moore, C. (2004). Finding community structure in very large networks. *Physical Review E*, 70, 066111.
34. Kumar, R., Raghavan, P., Rajagopalan, S., Tomkins, A. (1999). Trawling the web for emerging cybercommunities. *WWW Conference*, Toronto, Canada.
35. Ganti, R.K., Pham, N., Ahmadi, H., Nangia, S., Abdelzaher, T. (2010). *GreenGPS: A Participatory Sensing Fuel-Efficient Maps Application*, Mobisys, San Francisco, CA, June 2010.
36. Ganti, R.K., Srinivasan, S., Gacic, A. (2010). Multi-sensor fusion in smartphones for lifestyle monitoring. *International Conference on Body Sensor Networks*, Singapore, June 7–9, 2010.
37. Barker, M., Barker, D.I., Bormann, N.F., Neher, K.E. (2008). *Social Media Marketing: A Strategic Approach*. Cengage, South Western, Cengage Learning (Publisher), Mason OH.
38. Lenzen, C., Wattenhofer, R. (2008). Leveraging linial's locality limit. In *Proceedings of the 22nd Symposium on Distributed Computing* (*DISC*), Arcachon, France, pp. 394–407.
39. Kashiwara, M., Schapira, P. (1990). *Sheaves on Manifolds*. Springer-Verlag, Berlin, Germany.
40. Khalifa, A.B., Gazzah, S., Benamara, N.E. (2013). Multimodal biometric authentication using Choquet integral and genetic algorithm. World Academy of Science, Engineering and Technology, *International Journal of Computer, Information Science and Engineering*, 7(3), 27–36.
41. Su, K.-L., Jau, Y.-M., Jeng, J.-T. (2011). Modeling of nonlinear aggregation for information fusion systems with outliers based on the Choquet integral. *Sensors*, 11, 2426–2446.
42. Chong, C., Kumar, S.P. (2003). Sensor networks: Evolution, opportunities and challenges. *Proceedings of the IEEE*, 9(8), 1247–1256.