Data Pre-Processing and Intelligent Data Analysis

Famili, Fazel; Shen, W.-M.; Weber, R.; Simoudis, E.

International Journal on Intelligent Data Analysis, 1, 1, 1997

[52] R. Weber, Fuzzy-ID3: A Class of Models for Automatic Knowledge Acquisition, *Proceedings of the 2nd International Conference on Fuzzy Logic and Neural Networks*, Tisuka, Japan (1992), 265-268.

[53] S.M. Weiss and C.A. Kulikowski, *Computer Systems That Learn*, Morgan Kaufmann Publishers, California, (1991).

[54] J. Wnek and R.S. Michalski, Hypothesis-Driven Constructive Induction in AQ17-HCI: A Method and Experiments, *Machine Learning*, **14** (2), (1994), 139-168.

[55] A. Wu and J. Meador, Data Driven Neural-Based Measurement Discrimination for IC Parametric Faults Diagnosis, Design, Test and Application, *ASICS and Systems-on-a-Chip: Digest of Papers-IEEE VLSI Test Symposium,* Atlantic City, NJ. (1992), 194-197.

[56] S.M. Yarling, Time Series Modelling as an Approach to Automatic Feedback Control of Robotic Positioning Errors, *Proceedings of IEEE International Symposium on Electronics Manufacturing Technology,* (1993), 443-449.

[57] H.J. Zimmermann, *Fuzzy Set Theory and Its Applications*. Third Edition, Kluwer Academic Publishers, Boston, (1996).

[38] P. Riddle, R. Segal, and O. Etzioni, Representation Design and Brute-Force Induction in a Boeing Manufacturing Domain, *Applied Artificial Intelligence*, **8** (1994), 125-147.

[39] A. Rieger, Data Preparation for Inductive Learning in Robotics, *IJCAI Workshop on Data Engineering for Inductive Learning,* Montreal, Canada (1995), 70-78.

[40] P. Rubel, J. Fayn, P.W. Macfarlane, and J.L. Willems, Development of a Conceptual Reference Model for Digital ECG Data Storage, *Proceedings of Computers in Cardiology Conference*, (1991), 109-112.

[41] Y. Senol and M.P. Gouch, The Application of Transputers to a Sounding Rocket Instrumentation: On-Board Autocorrelators with Neural Network Data Analysis, *Parallel Computing and Transputer Applications (First Ed),* (1992), 798-806.

[42] J. Sjoberg, Regularization as a Substitute for Preprocessing of Data in Neural Network Training, *Artificial Intelligence in Real-Time Control (First Ed.),* (1992), 31-35.

[43] S. Smith, C. Gordon, Rapid Yield Learning Through Data Integration, *Semiconductor International*, **19**(10), (1996), 97-102.

[44] H.W. Sorenson, *Kalman Filtering: Theory and Application*, IEEE Press, New York, (1985).

[45] K. Staenz, Quality Assessment and Preprocessing of Data Acquired with the Programmable Multispectral Imager, *Canadian Journal of Remote Sensing,* **17**(3), (1991), 231-239.

[46] R. Stein, Preprocessing Data for Neural Networks, *AI Expert* (1993), 32-38.

[47] G.D. Tattersall, K. Chichlowski, and R. Limb, Preprocessing and Visualization of Decision Support Data for Enhanced Machine Classification, *Proceedings First International Conference on Intelligent Systems Engineering,* Edinburgh, England (1991), 275-280.

[48] P. Thirion, Direct Extraction of Boundaries from Computed Tomography Scans, *IEEE Transactions on Medical Imaging,* **13**(2), (1994), 322-328.

[49] P. Turney, Data Engineering for the Analysis of Semiconductor Manufacturing Data, *IJCAI Workshop on Data Engineering for Inductive Learning,* Montreal, Canada (1995), 50-59.

[50] M.R. Versaggi, Understanding Conflicting Data, *AI Expert* (April 1995), 21-25.

[51] J.T.W.E. Vogels, A.C. Tas, F. van den Berg and J. van der Greef, A New Method for Classification of Wines Based on Proton and Carbon-13 NMR Spectroscopy in Combination with Pattern Recognition Techniques, *Chemometrics and Intelligent Laboratory Systems:Laboratory Information Management*, **21** (1993), 249-258.

[24] B. Marangelli, Data Preprocessing for Adaptive Vector Quantization, *Image and Vision Computing,* **9**(6), (1991), 347-352.

[25] C. Matheus, *Feature Construction: An Analytic Framework and Application to Decision Trees,* Ph.D. Dissertation, University of Illinois, Computer Science Department, Urbana-Champaign, (1989).

[26] A.D. McAulay and J. Li, Wavelet Data Compression for Neural Network Preprocessing, Signal Processing, *Sensor Fusion and Target Recognition SPIE 1699* (1992), 356-365.

[27] W. Meier, R. Weber, and H.J. Zimmermann, Fuzzy Data Analysis - Methods and Industrial Applications, *Fuzzy Sets and Systems*, (61), (1994), 19-28.

[28] J.R. Mekemson, User Interface for FHWA's TRAF-FRESIM Model, *Proceedings of 4th International Conference on Microcomputers in Transportation*, Baltimore, Maryland (1992), 516-527.

[29] J.D. Meng and J.E. Katz, Using a Digital Signal Processor as a Data Stream Controller in Digital Subtraction Angiography, *Proceedings of IEEE Nuclear Science Symposium and Medical Imaging Conf*erence, Santa Fe, New Mexico (1991), 1839-1843.

[30] R.S. Michalski, *et al*, The Multi-Purpose Incremental Learning System AQ15 and its Testing Application to Three Medical Domains, Proceedings of AAAI-86, Philadelphia, PA. AAAI Press, (1986), 1041-1045.

[31] G. Murphy, *Similitude in Engineering,* The Ronald Press Company, New York, (1950).

[32] V. Nedeljkovic and M. Milosavljevic, On the Influence of the Training Set Data Preprocessing on Neural Networks Training, *Proceedings of 11th IAPR International Conference on Pattern Recognition* (1992), 33-36.

[33] D. Nikolayev and K. Ullemeyer, A Note on Preprocessing of Diffraction Pole-density Data, *Journal of Applied Crystallography,* **27** (1994), 517-520.

[34] G. Noriega and S. Pasupathy, Application of Kalman Filtering to Real-Time Preprocessing of Geophysical Data, *IEEE Transactions on Geoscience and Remote Sensing,* 30, **5**(1980), 897-910.

[35] M. Ohta, E. Uchinol, and O. Nagano, A New State Estimation Method with Prefixed Algorithmic Form Matched to Effective Data Processing, *Acustica,* 77(1992), 165-175.

[36] G. Pagallo, Learning DNF by Decision Trees, *Proceedings of International Joint Conference on Artificial Intelligence*, Morgan Kaufmann, Palo Alto, CA, Vol. I, (1989), 639-644.

[37] J. R. Quinlan, *C4.5: Programs for Machine Learning,* Morgan Kaufmann Publishers, San Mateo, CA, (1993).

[12] Z. Duszak and W.W. Loczkodaj, *Using Principal Component Transformation in Machine Learning*, Proceedings of International Conference on Systems Research, Informatics and Cybernetics, Baden-Baden Germany, (1994), 125-129.

[13] W. Emde, C.U. Habel, and C.R. Rollinger, The Discovery of the Equator or Concept Driven Learning, *Proceedings of IJCAI-83,* Sydney Australia, Morgan Kaufmann (1983), 455-458.

[14] B. C. Falkenhainer and R.S. Michalski, Integrating Quantitative and Qualitative Discovery in the ABACUS System, In Y. Kodratoff and R.S. Michalski, eds., *Machine Learning: An Artificial Intelligence Approach,* III, Morgan Kaufmann, Palo Alto, CA, (1990), 153-190.

[15] A. Famili, and P. Turney, Intelligently Helping Human Planner in Industrial Process Planning, *AIEDAM*, **5**(2), (1991), 109-124.

[16] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, From Data Mining to Knowledge Discovery, In U. Fayyad *etal,* eds., Advances in Knowledge Discovery and Data Mining, AAAI/MIT Press, Menlo Park, CA, (1996), 1-34.

[17] L. Hesselink, Research Issues in Vector and Tensor Field Visualization, *Proceedings of IEEE Workshop on Visualization and Machine Vision,* Seattle, WA. (1994), 104-105.

[18] S. Iwasaki, Clustering of Experimental Data and its Applications to Nuclear Data Evaluation, *Proceedings of the Symposium on Nuclear Data*, Tokai, Japan (1992), 211-221.

[19] M. Ke and M. Ali, MLS, A Machine Learning System for Engine Fault Diagnosis, *Proceedings of the 1st International Conference on IEA/AIE,* Tullahoma, TN. (1988), 24-30.

[20] P.M. Kelly and J.W. White, Preprocessing Remotely-Sensed Data for Efficient Analysis and Classification, Applications of Artificial Intelligence, *Proceedings of SPIE-The International Society for Optical Engineering-1963,* Orlando, FL. (1993), 24-30.

[21] P. Langley, G.L. Bradshaw, and H.A. Simon, Rediscovering Chemistry with the Bacon System, In R.S. Michalski, J.G. Carbonell, and T.M. Mitchell, eds., *Machine Learning: An Artificial Intelligence Approach,* Morgan Kaufmann, Los Altos, CA, (1983), 307-326.

[22] N. Lavrac, D. Gamberger, and P. Turney, Cost-Sensitive Feature Reduction Applied to a Hybrid Genetic Algorithm, *Proceedings of the 7th International Workshop on Algorithmic Learning Theory*, Sydney, Australia (1996), 1-12.

[23] D.B. Lenat, Learning from Observation and Discovery, In R.S. Michalski, J.G. Carbonell, and T.M. Mitchell, eds., *Machine Learning: An Artificial Intelligence Approach,* Morgan Kaufmann, Palo Alto, CA, (1983), 243-306.

Acknowledgements:

The authors would like to thank Sylvain Létourneau for providing comments on an earlier version of this paper.

References

[1] M.R. Azimi-Sadjadi and S.A. Stricker, Detection and Classification of Buried Dielectric Anomalies Using Neural Networks - Further Results, *IEEE Trans. on Instrumentations and Measurement*, **43(1)** (1994), 34-39.

[2] J.C. Bezdek and S.K. Pal, *Fuzzy Models for Pattern Recognition*, IEEE Press, New York (1992).

[3] D.G. Bobrow and D.A. Norman, Some principles of Memory Schemata, in D.G. Bobrow and A. Collins, Eds., *Representation and Understanding: Studies in Cognitive Science*, Academic Press, New York (1975), 138-140.

[4] E. Bontrager, *et al*, GAIT-ER-AID: An Expert System for Analysis of Gait with Automatic Intelligent Pre-processing of Data, *4th Annual Symposium on Computer Applications in MED CARE* (1990), 625-629.

[5] R.L. Chen and C.J. Spanos, Statistical Data Pre-processing for Fuzzy Modelling of Semiconductor Manufacturing Process, *Proc. 3rd Intl. Conf. on Industrial Fuzzy Control and Intelligent Systems*, Houston, Texas (1993), 6-11.

[6] J.J. Clark, *Data Fusion for Sensory Information Processing Systems*, Kluwer Academic Publishers, (1990).

[7] C. Cortes, L.D. Jackel, and W.P. Chiang, Limits on Learning Machine Accuracy Imposed by Data Quality, *Proceedings of the First International Conference on Knowledge Discovery & Data Mining*, Montreal, Canada (1995), 57-62.

[8] V.G. Dabija. *et al*, Learning to Learn Decision Trees, *Proceedings of the Ninth American Conference on Artificial Intelligence*, San Jose, CA. (1992), 88-95.

[9] J.F. Davis, B. Bakshik, K. Kosanovich, and M. Piovoso, Process Monitoring, Data Analysis and Data Interpretation, *Proceedings of the Intelligent Systems in Process Engineering Conference,* Snowmass, CO. (1995), 1-12.

[10] O.E. de Noord, The Influence of Data Preprocessing on the Robustness and Parsimony of Multivariate Calibration Models, *Chemometrics and Intelligent Laboratory Systems*, **23** (1) (1994), 65-70.

[11] J. DeWitt, Adaptive Filtering Network for Associative Memory Data Preprocessing, *World Congress on Neural Networks,* San Diego, CA (1994), IV34-IV38.

5.0 Concluding Remarks and Challenges

In this paper we have provided an overview of problems with the real world data. We have also discussed a number of data preprocessing techniques. These are techniques that are applied in order to: (i) solve data problems, (ii) understand the nature of the data, and (iii) perform a more in-depth data analysis. Two real world applications, that contain a number of data problems, were given and the approaches taken to solve data problems were explained.

The following are some of the important issues to be considered when data has to be preprocessed for data analysis:

(i) Although data preprocessing is useful and in many applications necessary in order to perform a meaningful data analysis, if proper techniques are not selected, it may result in loss or change of useful information to be discovered during the analysis.

(ii) To perform a meaningful data preprocessing, either the domain expert should be a member of the data analysis team or the domain should be extensively studied before the data is preprocessed. Involvement of the domain expert would result in some useful feedback to verify and validate the use of particular data preprocessing techniques.

(iii) In most applications, data preprocessing may be iterative. This means that certain preprocessing techniques, such as data elimination or data selection, may be used in a number of iterations until the best data analysis results are obtained.

One of the most important problems in data preprocessing is how do we know what valuable information exists in the raw data so that we can make sure it is preserved. This may depend upon our definition of data preprocessing. Some may argue that data preprocessing is not a completely "pre" process of data analysis. It needs feedback from the main data analysis process. After all, the ultimate judgement whether one has done a good job for data preprocessing is to see if the "valuable information" has been found in the later data analysis process.

Of the most important challenges in this area of research is development of a tool box or an expert system that can provide proper advice for selection and use of the best data preprocessing technique. Given some preliminary information about the data and the domain and all the data analysis objectives, one should be able to develop:

(i) An expert system that can look at a sample of data and some additional information presented by the user to provide some advice as the best data preprocessing strategy to be applied.

(ii) A tool box containing all data preprocessing techniques with examples that can be used as required. A possible link between the expert system and the tool box can demonstrate how particular data preprocessing techniques can influence the results of data analysis.

data analysis algorithms given the data status. In addition, we had to select the data sets from the following levels of granularity:

- fleet level where we had to select the data sets that contained relevant parameters for the entire fleet of aircraft,
- aircraft level where the selection was for data sets that contained parameters for a particular aircraft,
- engine level where we had to select the data sets that contained relevant parameters for particular engine of an aircraft,
- operation level where the selection was for data sets that contained parameters representing specific duration of aircraft operation.

Out-of-range data is the most difficult problem to detect. This is for cases in which the data for particular parameters does not contain a meaningful value, whether it is a sensor measurement or the value of a parameter that is automatically generated by a software under certain conditions. In addition, some technical background (domain knowledge) may be required to understand the proper range of some parameters, as for certain parameters a range may be acceptable under certain conditions and not acceptable under others. Two methods that we applied to identify some of the out-of-range data were data visualization and preliminary analysis of statistical information of numeric parameters.

(iv) Incomplete records

In parametric data, we encountered cases (data records) in which a substantial number of parameters were not available. For example, in engine divergence reports, that are generated when certain engine parameters exceed a threshold, some records did not contain all snap shots of engine parameters. This caused a problem so that in small time windows that only 50 records were available, if 20% of them had incomplete records, the remaining ones were not sufficient for the data analysis.

4.2.2 Use of Data Preprocessing Techniques

There were two objectives for using data preprocessing techniques in the aerospace application: (i) to solve problems in the data, and (ii) to learn more about the nature of the data.

For missing parameters, after identifying the percent of missing attributes in each record, records containing more than 20% were eliminated and the ones with 20% or less, were kept for data analysis. Of the two analysis engines [15, 36] used for decision tree induction, one handled missing attributes by replacing them with values derived from the existing ones.

For improper data types, we treated this problem in two ways: (i) we eliminated all the records that contained improper data types. This was in the cases where a large number of parameters (> 20%) were of improper types. This obviously caused loss of some useful data, (ii) we replaced improper data types by N/A's that were treated similar to missing attributes.

For range checking, ideally there should be a data range file set up so that all the data are filtered for out-of-range values before they are used. After identifying data sets and records with out-of-range data, we treated the out-of-range attribute values the same way as corrupt ones.

Handling incomplete records required some additional work. This was for particular data sets that contained up to 8 pairs of snap shots. Data subsets representing records with up to 4, 6 and 8 snapshots were created and analysed separately.

This application required an in-depth understanding of the domain and handling data from different levels of granularity. We had to know the domain knowledge for the purpose of: (i) filtering data for corrupt and out-of-range attributes, (ii) selecting proper

4.2 Aerospace Domain

This application is related to the analysis of operation and maintenance data collected by a commercial airline that operates a number of highly sophisticated aircraft. The goal of the analysis was to generate decision trees that can explain either performance or component failures in the operation of the aircraft. Performance failures are particular cases of aircraft operation in which a performance parameter is below or above an acceptable level (e.g. engine-exhaust-temperature > 700). Component failures are cases in which a particular component fails, it is replaced and the problem is rectified (e.g. temperature sensor failure).

The data is generated from the aircraft when it is either on the ground or in a flight. The data consists of several groups, three of which are: (i) logs of aircraft operation problems or snags (e.g. replaced #2 engine LPTCC), (ii) failure/warning messages (descriptive text) generated by on-board computers when certain parameters exceed a threshold (e.g. engine 2 exhaust gas temperature over limit), (iii) parametric data of various sensor measurements collected during different phases of aircraft operation (e.g. engine exhaust gas temperature, shaft speed). Parametric data comes in several groups, generated at different stages of aircraft operation. Examples are engine cruise and engine divergence data sets that are generated at different frequencies. Each group consists of 100-300 parameters of numeric and symbolic attributes. For certain groups, such as engine divergence data, each record may contain up to 16 snap shots of divergence related parameters, each representing data measured at x number of seconds before or after occurrence of the incident.

4.2.1 Problems with the Data

Data from the real world is never perfect. The aerospace application discussed in this paper was not an exception. The extracted Snags and warning/failure messages seemed relatively clean and complete. However, the parametric group of data, that represents various conditions of the aircraft and its operation (such as engine parameters) contained several forms of improper and incomplete data. Following are classes of problems that we observed:

(i) Missing attributes and missing attribute values

Each record in parametric group of data consists of several numeric and non-numeric parameters. We noticed that, within each record, a number of parameters had not been measured, recorded, or transferred properly. As a result, the final converted data had records with missing attributes or missing attribute values.

(ii) Improper types (numeric/symbolic data)

For an efficient and meaningful data analysis, the parameter values in each field have to be of the same type (numeric or non-numeric). In this application, it was noticed in several cases, that the parameter types were not consistent.

(iii) Out-of-range data

Sensors within a plasma etcher measure several hundred parameters. A subset of these parameters are periodically captured (approximately once a second) and made available as in-process data. This constitutes the time sequence data. In a typical scenario, a wafer can spend more than 200 seconds in an etcher. A hundred parameters measured once a second yield a maximum of 20,000 measurements for a single wafer. The large amount of time sequenced data make this a particularly interesting application of machine learning techniques. Many commercial products exist for this process [43]

Technically, the task we face is that given a set of positive and negative wafer etching operation examples, each being represented as about 100 data sequences (one sequence per parameter) with more than 200 time steps, to induce a set of probabilistic rules that can detect defective wafers during their manufacture process in real-time. Such detection may occur even before a manufacturing operation is completed so that timely corrections can be made to the process to minimize the loss of productivity.

4.1.1 Problems with the Data

Data problems in semiconductor manufacturing are due to many reasons. Examples are: (i) sensor related failures at the time of parameter measurements, conversion and transmission, (ii) operator related errors at various stages, (iii) software related errors, such as data acquisition and others. Following are descriptions of some of these errors:

(i) Incomplete Records with missing information for certain parameters of a lot (a lot consists of a number of bins each with 20-24 wafers).

(ii) Out-of-range data, especially parametric data where probers measure certain parameters on the wafer.

(iii) Corrupt data due to various reasons in the data acquisition, software or hardware.

4.1.2 Use of Data Preprocessing Techniques

The data collected from any semiconductor manufacturing does not normally show problems at the first glance. However, when large amounts of data is collected, there is always a chance of having some problems with the data. Use of proper techniques to solve data problems is very important. This is due to the information contents of the remaining fields.

Use of a data filtering mechanism is the most common method in dealing with semiconductor manufacturing data. Most companies develop their own “range file” which depends on the product class, product design, and production process. In other words, data within certain range is only accepted for certain classes of products or processes and rejected for others.

Use of data visualization and some statistical techniques are also common. These techniques only allow the process engineers and data analysts to learn about the process (e.g. sensor failure conditions). It would also help in a more accurate analysis of data for yield analysis [43] or process optimizations (e.g. for proper thresholds for dependent parameters and problem definitions).

4.0 Real World Applications

In this section we provide two examples that include data problems from real world applications and the use of some of the techniques discussed in previous section. The techniques discussed are related to data transformation and information gathering techniques that have been incorporated into a data analysis tool [15] or have been used as part of a data analysis activity.

4.1 Semiconductor Manufacturing

This application is related to the problem of plasma process faulty detection. The data to be analysed are time sequences from the manufacture of semiconductor wafers. Like many other industrial processes, semiconductor wafer manufacturing requires very tight process control, yet contains some element of "black art". The extremely high costs of the manufacturing equipment and infrastructure, as well as the nature of the industry provide strong motivation for improving the efficiency of the process, the quality of the products, and yield.

Semiconductor wafer manufacture consists of four main operations performed several times over. These operations are: growth or deposition, patterning or photolithography, etching, and diffusion or implantation. Each operation consists of multiple steps during which the wafer is subject to specific physical and chemical conditions according to a recipe. Testing the unfinished product between manufacturing steps is expensive and difficult. Reworking a bad product is almost impossible. This leads to two problems. First, when a problem occurs at a particular step, it may go undetected till final test is performed, thereby tying up downstream processing on a product that has already been doomed to be scraped. Second, when final test indicates that a product is of bad quality, it is usually difficult to determine which single step in the manufacturing process is the source of the problem.

Both of these problems would be solved if it were possible to collect the physical and chemical conditions (called in-process data) of wafer processing, and to automatically determine the success or failure of each manufacturing step by inspecting this data. Until recently, it was difficult to access the in-process data for most semiconductor manufacturing operations. Recent efforts by semiconductor equipment manufacturers and semiconductor wafer manufacturers have resulted in the establishment of a common interface (SECS: Semiconductor Equipment Communications Standard) through which different manufacturing tools can make their in-process data available.

The thrust of this work is to specifically study metal etch using reactive ion etch techniques in plasma etchers.

A reactive ion etching operation is a process in which reactive gas plasma ions are accelerated to the wafer surface where both chemical reaction and sputtering take place in a controlled manner to produce the desired etch profile. Typically, etch follows a photolithography operation. In the case of metal etch, a wafer is covered with metal in a metalization step. Then the desired patterns are drawn using photolithography. Finally, etching is used to remove the excess metal, leaving behind the required patterns of metal.

laps with Knowledge Driven Constructive Induction [54] in which domain knowledge is used to construct a new representation space.

3.3.5 Dimensional Analysis

The principal objectives of the theory of similitude and dimensional analysis are to establish those relationships necessary to permit reliable predictions to be made from data collected on processes or models, and to establish the type of relationship existing among features involved in the associated physical phenomenon in order that the most pertinent data may be collected and analysed systematically. Dimensional analysis is based on the dimensions in which each of the pertinent quantities involved in a phenomenon is expressed. The goal of using dimensional analysis is therefore to transform the existing measurement space into a series of dimensionless terms that can be used for data analysis. The most important advantage of dimensional analysis is that it generates qualitative rather than quantitative relationships. When dimensional analysis is combined with experimentation and data collection, it may be made to supply quantitative results and accurate prediction equations. Dimensional analysis is based on the Buckingham π–Theorem [31] to transform any dimensionally invariant variables $A_1, \ldots, A_m, B_1, \ldots, B_r$ of the following relation

$$T = F(A_1, \ldots, A_m, B_1, \ldots, B_r) \tag{3}$$

into a (simpler) form of r dimensionless variables $\pi_1, \ldots, \pi_s$ where:

$$\pi_1 = f(\pi_2, \ldots, \pi_s)$$

and each of the π terms is represented as:

$$\pi_s = \frac{B_j}{A_1^{a_{j1}}, A_2^{a_{j2}}, A_m^{a_{jm}}} \tag{4}$$

where $j = 1, \ldots r$, and $i = 1, \ldots m$

In the above equation T is the controlled parameter. According to the Buckingham π theorem, the number of dimensionless and independent quantities s required to express a relationship among the variables in any phenomenon is equal to the number of quantities involved n , minus the number of dimensions in which those quantities may be measured b.

Other forms of creating new attributes is in the form of building high level attributes [36]. For example, for machine learning applications, appropriate high level attributes may be defined a priori or during the data analysis process. A priori attributes are defined as combinations of the primary attributes of a fixed type and size.

3.3.2 Time Series Analysis

When data exhibits too much variation or non-stationary behaviour, the use of time series models from the data may provide a more reliable approach than some other techniques such as traditional control charting techniques. Variation is present in data from most application domains. For example, in industrial processes, this variation could be due to: (i) process equipment, (ii) raw material used in the process, (iii) process environment, (iv) human operating procedures and (v) individual decisions or process plans [56]. In most process monitoring applications, time series analysis means transforming data into a static collection of features that represent a view of the operation at some time. This is done during the data interpretation phase in which labels are assigned based on some discriminant relative to the extracted features. Time series analysis may also be applied to transform temporal data into a new form that temporally related events (e.g. records in the data set) that contain trends (e.g. increasing/decreasing) be represented into a single record.

3.3.3 Data Fusion

Many types of sensors may be used to gather information on the surrounding environments. Examples are: visual, thermal (infrared), proximity, and tactile sensors, ultrasonic and laser range finders. In these cases, different sensors are designed based on different physical principles, operate on a wide range of spectrum and possess distinct characteristics. Individual parameters measured by each of these sensors operating alone provide limited sensing range and are inherently unreliable due to the operational errors. However a synergistic operation of many sensors provides a rich body of information on the measured parameters and makes the data analysis more reliable and meaningful. This means that the measurements form different sensors is fused to provide single parameters that are more meaningful and create accurate results [6]. Some of these data fusion techniques may only be due to the corrections that have to be made on certain measurements.

3.3.4 Data Simulation

Data simulation deals with the problem of immeasurable or unavailable parameters in large measurement spaces. When the model of a process is known but all the parameters are not available, it may be possible to simulate those parameters and incorporate them into the entire measurement space so that the effects of those parameters can be investigated. An example is recording some parameters from ambient conditions in a complex manufacturing process (e.g. semi-conductor manufacturing) that may be either difficult or expensive to measure. However, the effects of these parameters on the process are known, it may be justifiable to measure and control these parameters as required. When the goal of induction is to induce decision trees, data simulation over-

- Data-Driven in which a data analysis tool analyzes and explores the data focusing on interrelationships among parameters in a data set and on that basis suggests changes to the representation space in the form of new attributes. BACON [21] and ABACUS [14] are examples of systems built based on this approach.
- Knowledge-Driven where the system applies expert provided domain knowledge to construct and/or verify new representation space. Use of process knowledge for data correction and creation of new features (e.g. use of thermodynamic properties for correction of jet engine parameters) are examples of knowledge driven operation. AM [23] and AQ15 [30] are examples of systems based on this approach.
- Hypothesis-Driven where a system incrementally transforms the representation space by analysing generated results in one iteration of data analysis and using the detected patterns in the results as attributes for the next iterations. BLIP [13] and CITRE [25] are two systems that incorporate Hypothesis-Driven induction capability.

If rule-based fuzzy systems are used for data analysis, several parameters have to be established which do not exist in conventional rule-based systems. Fuzzy rules are most often formulated as "**If-Then statements**" where the **If-part** is called the premise whereas the **Then-part** builds the conclusion [57]. The **If-part** consists of an aggregation of vaguely described expressions which are defined by membership functions. Their shape has to be established before such a rule-based fuzzy system can be applied. Hence, this determination of membership functions based on given data sets belongs to the step of preprocessing. There are several ways how this task can be done. Here we show how fuzzy clustering approaches can be used for this determination.

In fuzzy clustering objects are grouped into several classes where each object is described by its feature vector [2]. The assignment of objects to classes, however, is done in a fuzzy sense rather than in a crisp one, i.e. for each object a degree of class membership is computed. One algorithm which is widely used for fuzzy clustering is Fuzzy C-Means [27]. This algorithm is described in more detail in [2].

If for the construction of rule-based fuzzy systems membership functions are needed to define the vagueness in terms of specific linguistic variables, like high temperature, such fuzzy clustering algorithms can be employed in the following way. The objects to be clustered are the objects given for the main step of data analysis. As one task of preprocessing their feature values for the respective linguistic variable (e.g. temperature) are taken as the only feature values for clustering. The number of terms of the linguistic variable determines the number of classes to be found by clustering. By classifying objects in this one-dimensional feature space membership values are given as cluster results. Based on these membership values, the entire membership function can be derived from the given data set. In this way fuzzy clustering supports the task of building a rule-based fuzzy system by automatically generating membership functions.

Generation of membership functions is done as part of the definition of the rule base when a rule-based fuzzy system is used for data analysis. Establishing such a rule base consists of defining linguistic variables and their terms which are modelled by membership functions [2]. Techniques to select proper linguistic variables for rule-based fuzzy systems have been suggested in machine learning applications [52].

retically, selecting X attributes (from Y) is equivalent to selecting X basis vectors, spanning the subspace on these X vectors and projecting the database onto this space. Therefore, identifying principal components allows us to reduce the dimensionality of a database in which there are large number of interrelated variables, while retaining as much as possible of the variation present in the database. This reduction is achieved by transforming to a new set of variables, called *principal components*, which are highly uncorrelated, and which are ordered so that the first few retain most of the variation present in all of the original variables.

Identifying principal components involves checking the linear dependency among independent variables in a set of data attributes. Whether this is done automatically as part of the data analysis or separate from the analysis process, use of principal components requires domain knowledge. The reason being the importance of parameters reported in data analysis process. For example, if principal components analysis is combined with decision tree induction for analyzing data from a particular industrial process in which all process parameters are not controlled and adjusted at the same expense, the results of principal components analysis should be carefully reviewed by process engineers so that parameters are properly selected for decision tree induction.

3.2.5 Data Sampling

Of particular importance to data sampling are cases in which the algorithm used for data analysis requires a subset of the entire data, either for splitting the data for training/testing or evaluating the performance of the data analysis algorithm through iterative process of varying the sample size such as neural networks applications. The important issue here is the correct choice of samples (e.g. training/testing) in order to obtain and preserve the best possible performance for the algorithm in use. For example, in neural networks applications, one usually has only a small set of correctly classified patterns, known as the training set. The main question is: does the given training set fairly represent the underlying class conditional probability density functions? A number of sampling techniques have been proposed [32, 53].

3.3 Generation of New Information

Most data analysis applications involve solving problems that are common in day-to-day operation of an enterprise (e.g. a semiconductor wafer fabrication operation). However, within the same enterprise, there are always goals for in-depth analysis of data (e.g. research and development and process optimization). In this case, the goal is to put some additional effort for an in-depth data analysis and discovery of all valuable information that may exist in the data.

3.3.1 Adding New Features

Adding new features overlaps with a number of areas such as constructive induction [53] and definition of membership functions (fuzzy clustering). The main operation in constructive induction is to manually or automatically derive new features in terms of the existing ones. The three common forms of constructive induction are:

3.2.1 Data Visualization

Visualization of data have progressively evolved from techniques that mimic experimental methods to more abstract depictions of the data. There are at least two reasons for this evolution. First, many important quantitative parameters are not directly measurable [17]. Second, proper representation of highly multivariate data contained in vector fields, while at the same time avoiding visual clutter, requires simplification of the display by extracting and rendering only the relevant features of the data.

3.2.2 Data Elimination

In preprocessing the data through data elimination, sometimes two objectives are achieved:

- the volume of the data is reduced substantially. Examples are image data analysis [20].
- the data is partially classified. Examples are associating similar image pixels with one another.

Other form of data elimination is through Univariate Limit Checking methods such as "absolute value check":

$$\text{if } (X_{i,min} < X_i(t) < X_{i,max}) \text{ then class } w_j$$

where, $X(t)$ is the value of the measured variable at any time t and X_{min} and X_{max} are the lower and upper mapping limits, respectively.

3.2.3 Data Selection

To solve the problem of large amounts of data, several researchers have developed methods for accurately analyzing and categorizing data on much smaller data sets. By preprocessing large data sets by a technique known as vector quantization or clustering, computational requirements necessary for data analysis and manipulation are greatly reduced. Advantages to data selection on large data sets are numerous. Many times working with multispectral data, our goal is grouping together sets of similar data - something that clustering algorithms do automatically. Kelly and White [20] developed a clustering technique to analyze large amounts of image data. The basic principle of clustering in this work is to take an original image and represent the same image using only a small number of unique pixel values. In some cases, this resulted in reduction of the data by a factor of seven.

Other forms of data selection for digital data is DSP preprocessing of the raw data to reduce the raw data volume by a large factor and potentially producing real-time subtracted images for immediate display.

3.2.4 Principal Components Analysis

The use of principal components has been extensively studied [12]. The main goal of identifying principal components is to select proper attributes for data analysis. Theo-

either entered by the people who fill the questionnaires or the staff at a statistics centre. Data editing requires extensive domain knowledge as any incorrect editing of data may result in loss of useful information. Proper data editing is also important for natural language processing and applications in which data is extracted from on-line text in order to create assertions for a data base.

3.1.4 Noise Modelling

Noise in the data can be attributed to several sources, noise added by amplifiers and signal conditioning circuitry, aperture error and jittering in the sampling device, nonlinearities and quantization noise in the analog-to-digital (A/D) converter, extraneous noise picked up from the environment [40], and data transmissions between channels and sensor thresholds (upper/lower).

Fourier transform is among the most common methods of noise modelling for data preprocessing. Classic Fourier transform analyzes signals in terms of frequency components among the whole spatial domain, which loses time localization. Fourier transform is therefore appropriate for long time periodic signals. Short time window Fourier transform uses a set of window functions to restrict transform length and can be used to provide better time localization. Window size is normally determined by the lowest frequency.

Several adaptive schemes have been proposed for noise estimation. These methods are classified into Bayesian, maximum likelihood, correlation, and covariance matching. The first two assume time-invariance of noise statistics and are computationally demanding. In correlation methods some linear processing of the output is autocorrelated, and a set of equations is derived that relates these functions to the unknown parameters. Covariance matching techniques attempt to make the filter residuals consistent with their theoretical covariances.

Other forms of noise modelling and smoothing, are obtained by data compression, through omitting low frequency components of the data. Data compression can enhance and improve interpolation which results in better classifications on the testing data sets [26]. Smoothing the data, which is quite sensitive to data reliability, may allow the reduction of measuring time in experiments such as diffraction [33].

One of the most important strengths of noise modelling is that it can help in selection of relevant data and proper set up of thresholds in data classifications.

3.2 Information Gathering

We can picture a data analysis tool as a system that can unknowingly analyse data that are clean and sufficient for a given data analysis task. Limited or incomplete results are obtained when all data characteristics are not known, data analysis is not properly guided, or different internal parameters within a data analysis tool are not properly set. Our emphasis in this section is to discuss interactive techniques that are applied to the data so that we can: (i) better understand the nature of the data and (ii) use a given data analysis tool more efficiently.

proper techniques to rectify the problems. Following are a number of techniques that have been developed and applied to transform data from various domains.

3.1.1 Data Filtering

Data filtering is broad. At one end of the spectrum, data filtering deals with simple problems such as corrupt data. At the other end, it deals with noisy data. A number of data preprocessing techniques are based on data filtering to remove undesirable data in the time domain, frequency domain or time-frequency domain. The ideal filtering technique should remove irrelevant features with minimal distortion of the relevant signal features. Of the most common filtering techniques are: (i) time domain filtering, where the mean or median of the measured data in a window of predetermined size is taken, (ii) frequency domain filtering, where data is transformed via Fourier analysis and high frequency contributions are eliminated from the data, (iii) time-frequency domain filtering, where the measured data is transformed simultaneously in the time and frequency domain and provides the ability to capture a wide variety of signal features in a computationally efficient manner. The basic assumption in data filtering is that sufficient amount of domain knowledge is available so that useful information is not lost.

The most common technique is Kalman filtering, which provides the optimum linear recursive estimator (in the mean-squared form) of the state X(j) at some time t_j, given measurements Z(1) through Z(k), where $k > j$. Traditional approaches are documented in [44]. The use of Kalman filtering requires the complete knowledge of noise statistics, which in some real world applications may not be possible. In addition, in Kalman filtering, the linearity of the system, the Gaussian distribution properties of the system noise and the observation (background) noise are all assumed at the very beginning of the analysis [35].

3.1.2 Data Ordering

The most common forms of data ordering are in applications where data are stored in relational or network database management systems. The main objective here is to organize data in proper locations (tables) for further retrieval and analysis. A conceptual data model (e.g. entity relationship) is usually prepared first. Entities and relationships are identified. Attributes within the entities are listed and the type of relationships (1-1, 1-n or n-1) are labelled. An example of data ordering for real world application [39] is automatic preprocessing of patient data from ECG (electrocardiogram) process where large amounts of data have to be properly ordered for: (i) short-term prior to be reviewed by a physician, (ii) long-term with the perspective of its future use, particularly for comparison and other forms of data analysis. Data ordering requires a model of the process or system from which the data is acquired and may overlap with data warehousing.

3.1.3 Data Editing

Data editing is applied in preprocessing text or symbolic data types where the data elements consist of one or more string of characters representing unique information for a particular attribute. Examples are census related data where data elements have been

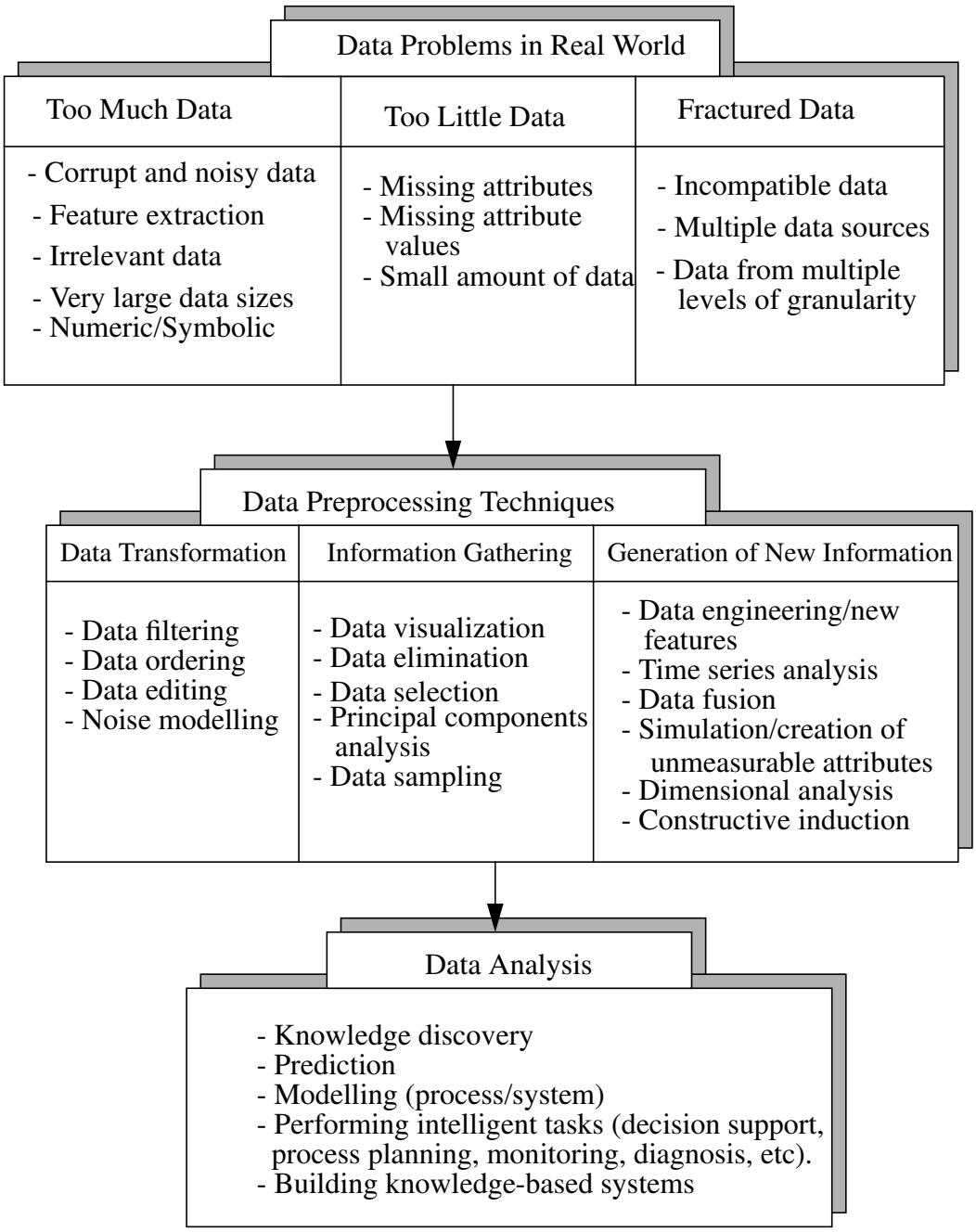

Figure 1: Problems with the data and data preprocessing techniques

3.1 Data Transformation

The basic limitations in data collection and data analysis are due to the quality and completeness of the data [7]. Inaccuracies in the measurements of input or incorrect feeding of the data to a data analysis tool (e.g. a classifier) could cause various problems. It is therefore the primary task in data analysis to identify these insufficiencies and select

applications, such as neural networks, is the accuracy versus simplicity of the results [26]. In most real world applications, there is a need for at least one form of data preprocessing. Almost all applications of inductive techniques require a well planned use of data preprocessing techniques. Fayyad, Piatetsky-Shapiro, and Smyth [16] emphasize use of data preprocessing techniques as an essential part of any knowledge discovery from data base project.

To provide a context for presenting a comprehensive listing of data preprocessing techniques, we will first look at the definitions and requirements of these techniques. We will then present a framework for the existing techniques that have been reported in the literature. Each technique may have several strengths and weaknesses. In addition, one must also be aware of the assumptions to properly apply each of these techniques. For example, of the techniques related to **too much data**, data filtering and data elimination throw away data while others like data sampling, help to select the important data sets, yet others like noise modelling and principal component analysis assist in summarizing/ compressing the data.

2.2 Preparation for Data Analysis

When problems with the data are solved and the data is prepared, there are still a number of steps that can be followed before the actual data analysis starts. All actions taken to understand the nature of data and advanced techniques to perform in-depth data analysis are in this category.

2.2.1 Understanding the Nature of Data

When all known problems with the data are solved, understanding the nature of data would be useful in many ways:

- Proper use of most data analysis tools and interpretation of large and complex data sets are beyond the capability of human brain. It is therefore useful to perform some form of data preprocessing for better understanding of the data. Examples are data visualization and principal component analysis.
- Most data analysis tools have some limitations related to data characteristics. It would therefore be useful to know these characteristics for proper selection and set up of data analysis process. An example is percent of missing attribute values in the entire data set.
- Sensor problems cannot be identified if some form of data preprocessing is not performed. Sensor/measurement problems cause unusual data distribution and inaccurate data representations. Examples are Analog-to-Digital and Digital-to-Analog conversion processes used in measurements of Analog and Digital Signals.

2.2.2 Data Preprocessing for In-depth Data Analysis

Ordinary data analysis tools and techniques provide means of analyzing data up to a level that may not be sufficient in all applications. In-depth data analysis requires additional support facilities for data preprocessing, that have to be properly used before the actual data analysis starts. For example, if the data is analyzed for inducing rules and the data is taken as records representing single events, temporal and other forms of trends in the data would not be properly recognized through the induction process. However, if the data is transformed so that records represent trends rather than single events, the results of data analysis would be more meaningful.

Other forms of data preprocessing for in-depth data analysis are: (i) manual or automatic addition of new features that are derived from the existing ones, which overlaps with constructive induction, (ii) data simulation for creating parameters that are not normally measured, (iii) data fusion which is performed for integrating data from multiple sources, and (iv) dimensional analysis which provides support for creation and use of qualitative rather than quantitative relationships.

3.0 Data Preprocessing Techniques

Data preprocessing is beneficial in many ways. In classifications with neural networks, one can eliminate irrelevant data to produce faster learning due to smaller data sets and due to reduction of confusion caused by irrelevant data. The common trade-off in many

contain very useful information. Traditionally if more than 20% of attribute values are missing, the entire record is eliminated.

iii Small amount of data

In this case, although all data attributes are available, the main problem is that the total amount of data is not sufficient for all kinds of data analysis. For example, most data analysis algorithms require around 100 examples of training data to be properly trained to classify future examples. The reliability of the concepts learned or rules generated may not be sufficient if enough examples are not available.

2.1.3 Fractured data

i. Incompatible data

Data compatibility becomes important when data is collected by several groups. This is specially true in domains where sensor data are collected and analyzed. Sensor data consists of a lot of text and symbolic attributes where groups of data have to be combined. The incompatibility problems could be due to human way of representing the text or even use of natural language understanding/processing capabilities in the data collection process.

ii. Multiple sources of data

In large enterprises, data could be scattered in a number of departments and on different platforms. In most cases, the data is even acquired and maintained using different software systems. The goal, depth and standard of data collection may vary across the enterprise. As a result, when data from more than one group is required for data analysis, problems related to the use of data from multiple sources may arise.

iii. Data from multiple levels of granularity

In some real world applications data comes from more than one level of granularity. Examples are semiconductor manufacturing and aerospace. In semiconductor manufacturing, data is collected at all stages of production. At the lowest level, data may be collected from each integrated circuit on a wafer. These represent measurements for each unit of production (a typical wafer may contain over 200 integrated circuits). These represent all the measurements that have to be done on all wafer production units. At the next level, data could come from particular sites on a wafer that are called test sites. This data is collected to estimate similar properties of all the sites on a wafer. Other levels are wafer level and bin (batch) level. At the wafer level, the data represent an entire wafer, whether before the production stage such as raw properties of a wafer or after the production stage, such as overall thickness [48]. The bin level is the highest level where the data represent parameters for a group of wafers in a container (bin).

Similarly, in aerospace domain where data for a large fleet of aircraft are analysed, levels of granularity may consist of: (i) fleet level (e.g. all aircraft of a particular type), (ii) aircraft level (e.g. a particular fin-number), (iii) system level (e.g. a particular engine of an aircraft), and (iv) system operation level (e.g. operation of the engine for a particular duration or cycle).

In many domains such as space (e.g. image data) and telecommunications (e.g. large network operations), the volume of data and the rate at which data are produced may be a limiting factor on performing on-time data analysis. The amount of data is sometimes beyond the capability of the available hardware and software used for data analysis. For example, interpreting remotely-sensed data requires expensive, specialized computing machinery capable of storing and manipulating large amounts of data quickly.

vi. Numeric/Symbolic data

When data is organized for analysis, it generally consists of two types:

- numerical data that result from measuring parameters that can be represented by a number. Numeric data may be either discrete or continuous.
- symbolic or categorical data that result from measuring process or system characteristics. This class of data is usually qualitative.

Analysing data involving both numeric and symbolic parameters is a complex task that requires attention during data preprocessing and proper use of the data analysis tool.

2.1.2 Too little data

i. Missing attributes

Missing or insufficient attributes are examples of data problems that may complicate data analysis tasks such as learning and hinder accurate performance of most data analysis systems. For example in the case of learning, these data insufficiencies limit the performance on any learning algorithm or a statistical tool applied to the collected data - no matter how complex the algorithm is or how much data is used. On the other hand, the data could look perfect at a glance but may be out-of-range, due to improper sensor measurements, data conversion/transmission problems. Most factory management software systems, have access to data range tables which are used to filter out-of-range data.

Corrupt and missing attributes create several problems. Following are two examples focusing on induction as the data analysis process:

(i) In decision tree induction, missing attributes cause vectors to be of unequal length. This results in a bias when either the information value of the two vectors representing two attributes is compared or a test is to be performed on the values of an attribute.

(ii) Many data analysis applications involve splitting the data into training and testing sets. Although the splitting process may be iterated several times, missing attributes may cause inaccurate evaluation of the results.

ii. Missing attribute values

In this case, the data records are not all complete, some contain missing attribute values. These data records cannot be eliminated because on one hand the total amount of data may not be sufficient and on the other hand the remaining values in the data record may

2.1.1 Too much data

i. Corrupt and noisy data

Corrupt data could be due to reasons such as sensor failure, data transmission or improper data entry, many of them may be unknown at the time of data collection. Noise in the data can be attributed to several reasons:

- data measurement or transmission errors,
- inherent reasons such as characteristics of processes or systems from which data is collected.

Regardless of the reason, corruptness and noise in the data have to be correctly identified and proper solutions have be found to deal with the problem. In general, noise in the data would weaken the predictive capability of the features. For any given application, data sets may be completely noisy, to somewhat noisy, to completely free of noise. On the other hand, data sets that may look noisy on their own and through data visualization, may be highly predictive and noise free.

ii. Feature extraction

In complex on-line data analysis applications, such as chemical processes or pulp and paper applications, although there may be hundreds of measurements, relatively few events may be occurring. The data from these measurements must therefore be mapped into meaningful descriptions of event(s). This is a difficult task without proper data preprocessing facilities. Preprocessing the data for proper interpretation is a form feature extraction that conditions the input data to allow easier subsequent feature extraction and increased resolution [9]. An example of feature extraction is numeric-symbolic interpretation where numeric data from a process are mapped into useful labels. The problem boundary is defined backwards from the label of interest so that feature extraction is associated only with the input requirements to generate the label.

iii. Irrelevant data

Many data analysis applications require extraction of meaningful data from large data sets. When human beings are in the loop, they select the relevant data by focusing on key pieces of information and sometimes using the rest of the data only for confirmation or to clear up ambiguities. On-line expert systems used for data analysis are examples in which one has to be able to extract relevant information from raw data [4]. The main goal of eliminating irrelevant data is to narrow the search space in data analysis.

Complexity may be significantly reduced if irrelevant data are eliminated, and only the most relevant features are used for data analysis. Reducing the dimensionality (through eliminating irrelevant data) may also improve the performance of a data analysis tool, since the number of training examples, needed to achieve a desired error rate increases with the number of measured variables or features.

iv. Very large data-sizes

2.0 Data Preprocessing

Data preprocessing consists of all the actions taken before the actual data analysis process starts. It is essentially a transformation T that transforms the raw real world data vectors X_{ik} to a set of new data vectors Y_{ij}

$$Y_{ij} = T(X_{ik}) \qquad (1)$$

such that: (i) Y_{ij} preserves the "valuable information" in X_{ik}, (ii) Y_{ij} eliminates at least one of the problems in X_{ik} and (iii) Y_{ij} is more useful than X_{ik}. In the above relation:

$i = 1, \ldots n$ where n = number of objects,

$j = 1, \ldots m$ where m = number of features after preprocessing,

$k = 1, \ldots l$ where l = number of attributes/features before preprocessing,

and in general, $m \neq l$.

Valuable information are components of knowledge that exist in the data (e.g. meaningful patterns) and it is the goal of data analysis to discover and present them in a meaningful way. Fayyad, Piatetsky-Shapiro, and Smyth [16] define four attributes for valuable information. These are: valid, novel, potentially useful, and ultimately understandable. Data problems are situations which prevent efficient use of any data analysis tool or they may result in generating unacceptable results.

Data preprocessing may be performed on the data for the following reasons:

- solving data problems that may prevent us from performing any type of analysis on the data,
- understanding the nature of the data and performing a more meaningful data analysis, and
- extracting more meaningful knowledge from a given set of data.

In most applications, there is a need for more than one form of data preprocessing. Identification of the type of data preprocessing, is therefore a crucial task.

2.1 Problems with the Data

There are always problems with the real world data. These are best shown in Figure 1 and discussed below. The nature and severity of problems depend on many reasons that are sometimes beyond the control of human operators. Our concern is due to the effects of these problems on the results of data analysis, the goal being to either rectify the data problems ahead of time or recognize the effects of data problems on the results. Data problems can be classified into three groups of: too much data, too little data, and fractured data which will be discussed in the following.

In practice, the first operation on any sets of data is preprocessing. Data preprocessing is a time consuming task, which in many cases is semi-automatic. Growing amounts of data produced by modern process monitoring and data acquisition systems has resulted in correspondingly large data processing requirements, and therefore, efficient techniques for automatic data preprocessing are important [34]. Our goal in this paper is to discuss problems that we normally encounter with the data, methods we apply to overcome these problems and how we benefit from data preprocessing using these techniques.

This paper first provides a brief overview of data preprocessing, focusing on the reasons as why it is needed. We then discuss a number of frequently encountered real world problems in data analysis. These are problems, related to data collected from the real world, that may have to be dealt with through data preprocessing. We then explain what additional forms of data preprocessing are performed to understand the nature of the data and to perform an in-depth data analysis. Various forms of data preprocessing techniques are explained in Section 3.0. Section 4.0 includes two examples of real world applications in which data has to be preprocessed. The paper ends with concluding remarks and a list of challenges specific to real world applications.

1.0 Introduction

Data analysis is the basis for investigations in many fields of knowledge, from science to engineering and from management to process control. Data on a particular topic are acquired in the form of symbolic and numeric attributes. The source of these data varies from human beings to sensors with different degrees of complexity and reliability. Analysis of these data gives a better understanding of the phenomenon of interest. The main objective of any data analysis is therefore to discover knowledge that will be used to solve problems or make decisions. However problems with the data may prevent this. In most cases, imperfections with the data are not noticed until the data analysis starts. For example, in the development of knowledge based systems, the data analysis is performed to discover and generate new knowledge for building a reliable and comprehensive knowledge base. The reliability of that portion of the knowledge base that is generated through data analysis techniques such as induction, therefore, depends on the data.

Many efforts are being made to analyse data using commercially available tools or by developing an analysis tool that meets the requirements of a particular application. Some of these efforts have ignored the fact that problems exist with the real world data and some form of data preprocessing is usually required to intelligently analyse the data. This means commercial or research tools should provide data preprocessing facilities to be used before or during the actual data analysis process. Various objectives may exist in data preprocessing. In addition to solving data problems, such as corrupt data, irrelevant or missing attributes in the data sets, one may be interested in learning more about the nature of the data, or changing the structure of data (e.g. levels of granularity) in order to prepare the data for a more efficient data analysis.

In comparing data preprocessing with human way of processing information, Bobrow and Norman [3] make a similar comparison:

> "Consider the human information processing system. Sensory data arrive through the sense organs to be processed. Low level computational structures perform the first stages of analysis and then the results are passed to other processing structures. ... The processing system can be driven either conceptually or by events. Conceptually driven processing tends to be top-down, driven by motives and goals, and fitting input into expectations: event driven processing tends to be bottom-up, finding structures in which to embed the input."

Various explanations have been given to the role and the need for data preprocessing. In the case of modelling, variations in the data which are caused by the changes in process or system conditions, as well as in data collection/transmission can be modelled in conjunction with the target information. Proper data preprocessing can eliminate these effects beforehand, which result in more parsimonious models. These models may not necessarily have better predictive abilities, but are expected to be more robust [10]. Data preprocessing would therefore lead to a smaller number of phenomena to be modelled, but it may also result in an increase in variance because of estimation errors. In the case of learning, data preprocessing would let the users to decide on how to represent the data, which concepts to learn and how to present the results of data analysis so that it's easier to interpret and apply them in real world.

Data Preprocessing and Intelligent Data Analysis

A. Famili [1], Wei-Min Shen [2], Richard Weber [3], Evangelos Simoudis[4]

(1) Institute for Information Technology, National Research Council Canada, Ottawa, Ontario, K1A 0R6 Canada famili@ai.iit.nrc.ca
(2) Information Sciences Institute, University of Southern California, 4676 Admiralty Way, Marina del Rey, CA, 90292, USA shen@ISI.EDU
(3) Management Intelligenter Technologien GmbH, Promenade 9, 52076 Aachen, Germany RW@mitgmbh.de
(4) IBM Almaden Research Center, 650 Harry Road, San Jose, CA 95120-6099, USA simoudis@almaden.ibm.com

Abstract

This paper first provides an overview of data preprocessing focusing on problems of the real world data. These are primarily problems that have to be carefully understood and solved before any data analysis process starts. The paper discusses in detail, two main reasons for performing data preprocessing: (i) problems with the data and (ii) preparation for data analysis. The paper continues with details of data preprocessing techniques to achieve each of the above mentioned objectives. A total of 14 techniques are discussed. Two examples of data preprocessing applications from two of the most data rich domains are given at the end. The applications are related to semiconductor manufacturing and aerospace domains where large amounts of data are available and they are fairly reliable. Future directions and some challenges are discussed at the end.

Keywords: data preprocessing, data analysis, data mining.

NRCC# 40166