Monitoring and Data Quality Control of Financial Databases from a Process Control Perspective

(Practice-Oriented Paper)

Janusz Milek[†‡], (janusz.milek@predict.ch)

Martin Reigrotzki[†], (martin.reigrotzki@predict.ch)

Holger Bosch[†], (holger.bosch@predict.ch)

Frank Block[†], (frank.block@predict.ch)

[†]PREDICT AG, Reinach BL, Switzerland, (www.predict.ch)

[‡]Automatic Control Laboratory, ETH Zürich, Switzerland, (www.aut.ee.ethz.ch)

Abstract

The paper presents the application of several process control-related methods to the monitoring and control of data quality in financial databases. The quality control process itself can be seen as a classical control loop. Measurement of the data quality is conducted via application of quality tests, which exploit data redundancy defined by meta-information or extracted from data by statistical models. Appropriate processing and visualization of the test results enable human or automatic diagnosis of possible data quality problems. Selected model-based process monitoring methods are shown to be useful for detection, diagnosis, and, in some cases, also compensation of data quality problems. The test results are of interest not only for data quality control but also for business-relevant information extraction and monitoring. The presented methods are incorporated into our DQontrol product [1], and have been applied in the monitoring of a productive financial database at a customer site.

1. Introduction

Information quality is one of the most important factors determining quality of conclusions drawn using consolidated data. Hence, it is necessary to continuously measure and improve the information quality. Huge financial databases, containing terabytes of data and invaluable information amounts, particularly need detailed and efficient monitoring approaches to extract useful information and distinguish it from artifacts and errors, which have to be eliminated.

Usually, the databases contain data from diverse sources which are loaded on a periodic and partially manual basis. Systematic data quality monitoring of such databases requires automated quality testing, result visualization, diagnosis, and compensation of data quality problems. Due to the truly industrial scale, high relevance, and hierarchical structure, huge databases can be treated similarly to large industrial processes. This analogy can be helpful to arrive at useful monitoring

approaches. Moreover, it can be expected that modern statistical process monitoring methods can be particularly useful to monitor databases. These methods exploit spatial and temporal redundancy of the data.

Note that modern process monitoring systems are not just limited to (i) fault detection, but may also include the following further stages: (ii) fault isolation, (iii) diagnosis, and (iv) compensation, so that their application gives rise to fault-tolerant measurement and control systems, see [11]. Counterparts of the mentioned stages in database quality monitoring systems may improve not only the quality but also fault-tolerance.

2. Some Elements of the Quality Control Loop

Left picture in Figure 1 shows the information flow in an example database containing financial data, e.g., of a bank, telecommunication, or insurance company. The data which usually come from heterogeneous sources, are extracted, transformed, and loaded into the database, before being delivered to the users. Data quality can decrease at each stage if the corresponding operations are disturbed in some way. Such a disturbance will be called a *fault*. The process of ensuring high data quality can be treated as a control system (Figure 1, right picture), composed of the measurement elements, controller, and actuators. The system must cover the whole information processing chain, from the data capture to the end user delivery.

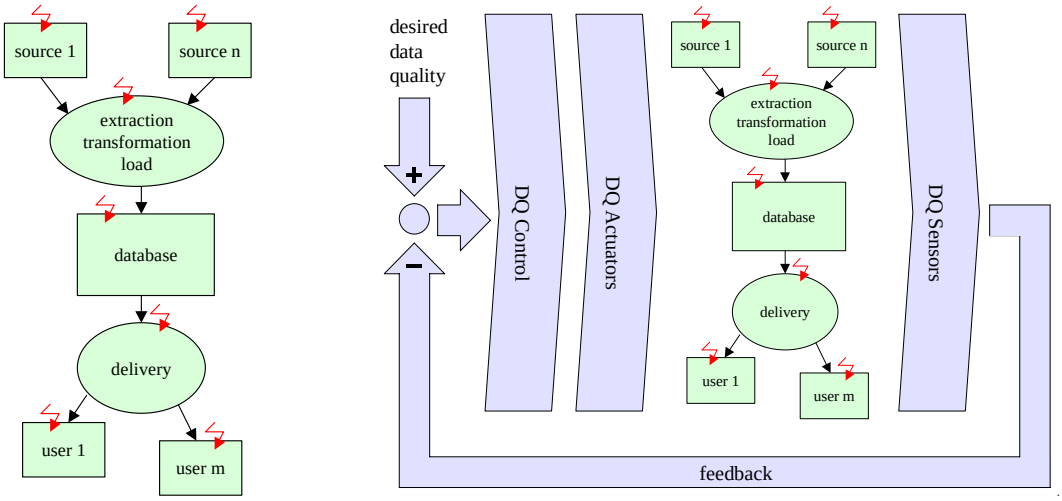

Figure 1: Example data flow (left) and information quality control system (right)

DQ Sensors measure data quality by running a number of quality tests. DQ controller analyses the test results, performs diagnosis and schedules appropriate data quality improvement actions. DQ actuators implement these actions. The goal of the overall feedback is to enforce the desired data quality. Some elements of the quality improvement process, for example data quality measurement, visualization, documentation, fault diagnosis, or compensation of simple faults can be conducted autonomously in a fully automatic way. Other, more complex elements of the process, like analysis and compensation of complex data quality problems cannot be automated and will not be considered here.

2.1. Factors Influencing Data

The data in a financial database can be influenced by the following factors: (i) individual customer behavior, (ii) market-related variations, (iii) seasonal variations, (iv) data quality issues. One goal of the monitoring can be to detect and distinguish all these types of factors. Basic data quality deficiencies (like missing values, data formats, and code tables) are most visible in the fault-related dimensions and can be easily identified using simple formal tests. Seasonal and market variations can be modeled using time series analysis, where individual customer variations can be analyzed e.g., using data mining methods [13]. The individual variations are suppressed using the later described data aggregation technique.

2.2. Quality Measurement and Classification of Tests

Data quality can be measured by performing appropriate tests. The tested piece of information is compared to the reference information. The generic test process is shown in Figure 2.

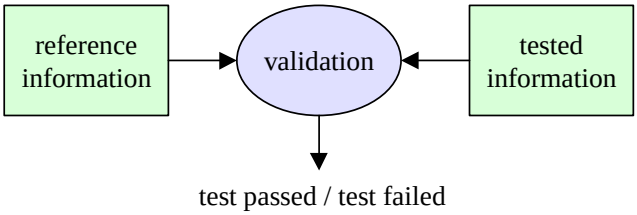

Figure 2: Principle of testing

There are two basic types of reference information:

- Meta-information, independent from the data and having the form of *strict* relations. Examples of reference information are: field validity (missing), field formats, code tables, keys, reference relations, strict business relations (like $a = b + c$, or *a stays constant*). The test related to meta-information can be called *technical*.
- Statistical models, obtained from reference fault-free data and having the form of *approximate* relations. Tests using statistical information are called *statistical* or *model-based*. Examples of models include mean, histogram, correlation, time series analysis model, as well as approximate business relations (*a similar to b, a changes slowly*).

The tests can be also classified according to the number of involved (1) variables (univariate/multivariate tests), (2) records (single/multi-record tests), and (3) tables (single/multi-table tests).

2.3. Fault Signatures

Fault signature depicts the *absolute* or *relative* number of cases when a given test failed (due to some data quality problem), aggregated in the *aggregation dimensions* and presented in the *presentation dimensions*. The latter should be discriminative with respect to faults and informative to business/user-related applications. Example presentation dimensions may include time, partition (if, for technical reasons, the data are partitioned), customer segment, product, and

subsidiary. Figure 3 shows a color-coded example of absolute test signature in customer segment/time coordinates.

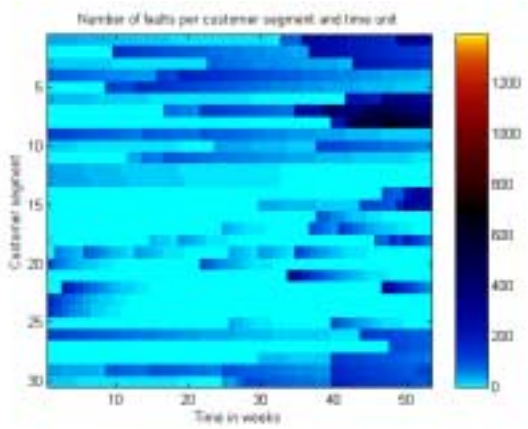

Figure 3: Example fault signature

Fault signatures enable simple fault visualization, assessment of data quality, and as such can be useful for fault elimination/correction, since it is simple to isolate data subsets having low data quality, or analyze data quality as a function of time.

2.4. Classification of Meta-Information and Technical Tests

Meta-information describes strict relations which must be satisfied by the data. Examples of such relations are given in Table 1.

Level	**Test type**
Field level	missing value, format, code table, ranges
Referential integrity	keys, duplicates
Business-related	univariate and multivariate strict business-related dependencies for one customer or account, for example: variable is equal (less than) to sum (concatenation) of other variables, a variable is equal to number of corresponding entries in another table, a variable is constant in time, a variable equals another variable shifted in time, etc.

Table 1: Meta-information and the related tests

Unfortunately, obtaining complete meta-information can be difficult and time consuming, especially if data are loaded from several sources. Related data quality problems are common in such a case; the only solution is to continuously collect and update the meta-information.

2.5. Classification of Statistical Models and Statistical Tests

Statistical monitoring methods comprise ideas belonging to econometrics, process monitoring, and data mining [12]. The main assumption of the statistical monitoring is that certain statistical data properties do not depend on time (*continuity assumption* [8]). Usually, the monitoring procedure comprises two steps. First, the selected statistical properties are estimated from available reference (fault-free) data. These properties constitute the model. Then, the model is used to validate new data and reject those data samples, which are not model-conform.

The most general statistical data description can be given in the form of multivariate probability density functions (pdf). A priori known multivariate pdf can be used to classify data samples as correct (probable) or incorrect (improbable). However, estimation of the pdf for raw record-level data is very difficult due to the *curse of dimensionality* and weak relations amongst the raw data samples. (See [4] for a recently proposed algorithm.) The aforementioned problems can be avoided using the following "sub-optimal" methods.

- The first approach is to decrease the number of variables in the estimated pdf. The dimensionality reduction makes pdf estimation task more feasible but is offset by an accuracy loss, since variable correlation cannot be fully exploited. Simple pdf-related statistical tests for univariate pdf may involve its estimation, analysis of peaks in the estimated distributions (see Figure 4), or outlier detection. Advanced tests may utilize logistic regression-type tests [13], hidden Markov models [5], clustering techniques, or modeling of multivariate histograms as slowly changing time functions.
- The second approach is to use aggregated (summed) data. Such data usually exhibit stronger redundancy than the original data, even if the individual customer-related variations are suppressed. Two types of redundancy exist: spatial (between variables, segments, partitions, etc.) and temporal (in time). Redundancy is extracted by statistical models which can be identified directly from the data [7] and used for monitoring purposes. See [6], [8], and [3] for application examples.

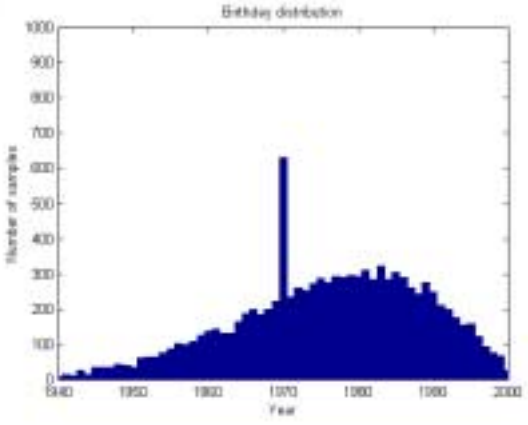

Figure 4: Univariate histogram of birthdays reveals clear outlier (default value)

In this paper only the second approach is exploited, the one common in the process monitoring methods. The following table summarizes redundancy types and the corresponding models:

Redundancy type	Redundancy meaning	Appropriate models
Temporal	relates values of one variable for different time instants	time series or lagged-variable models like AR, ARIMA
Spatial	relates values of several variables for one time instant	multivariate static models like linear regression, PCA, nonlinear models like NNPCA, hypersurface
Spatial and temporal	relates values of several variables for different time instants	multivariate time series models like VAR, VARMA, transfer function models like ARX, ARMAX, BJ

Table 2: Redundancy types and the corresponding models

2.6. Data Aggregation

Analysis of the aggregated data (like total amounts of assets, customers counts, and service sales) may give a valuable insight into the contents and data quality of a financial database. These time series have a generally understood meaning and can be compared to the usually available reference controlling data. Additionally, most of them can be treated as global indicators, useful for decision makers (e.g., to perform market monitoring and prediction). Moreover, it is often fair to suppose that such time series are slow changing and the relations between particular variables are almost constant over time (consider average assets per customer group, which also have clear intuitive meaning).

As previously noted, data aggregation suppresses individual customer variations. Hence, minor faults related to single customers may go unnoticed. The aggregated data are influenced by seasonality and business conditions as well as data quality issues. The aggregation dimensions should, if possible, coincide with the already mentioned fault-related dimensions.

The aggregated data form compact multivariate time series and can be stored for reference for long periods. Examples of aggregated data are: (1) number of accounts per customer segment, product type, partition, and time unit, and (2) sum of transactions per customer segment, product, partition, and time unit. Sensible aggregation operations include sum, count, mean, variance, minimum, maximum, histograms; such aggregates can be further aggregated in other dimensions without need for re-computations [4].

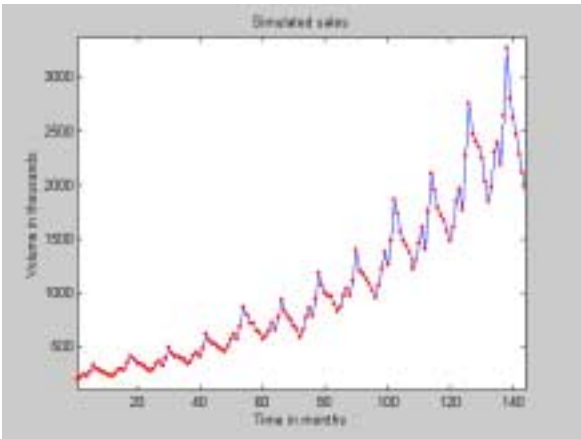

Figure 5: Example aggregate (simulated sales) forms a time series and exhibits visible temporal redundancy. The example follows [2].

3. Selected Process Control-Related Tests

This section demonstrates the application of selected process monitoring methods with respect to financial databases. Generally, such methods may include the following stages [9]-[10]:

- *Fault detection,* i.e., obtaining evidence that a group of variables/data samples is influenced by a fault. The detection principle is to test if data satisfy the model equation.
- *Fault diagnosis,* i.e., determination of which variable(s) is/are influenced by the fault. The isolation is performed via model-based variable elimination and by testing the consistent data subsets (via application of the so called structured residuals, which are described later).
- *Fault compensation,* i.e., reconstruction of the proper value of given variable. The reconstruction is possible using the model and fault-free variables and requires a high degree of redundancy.

There exist two basic process monitoring approaches [6]. In the first approach (called *parity space* method) a constant residual generator is estimated from fault-free data. Then, the generator is used to test new data. An alarm is raised if the residuals exceed given thresholds. In the second approach there is no fixed model. Instead, a parametric model is constantly estimated from the data. The monitoring is performed by testing variations of the model parameters. Block diagrams of both approaches are shown in Figure 6.

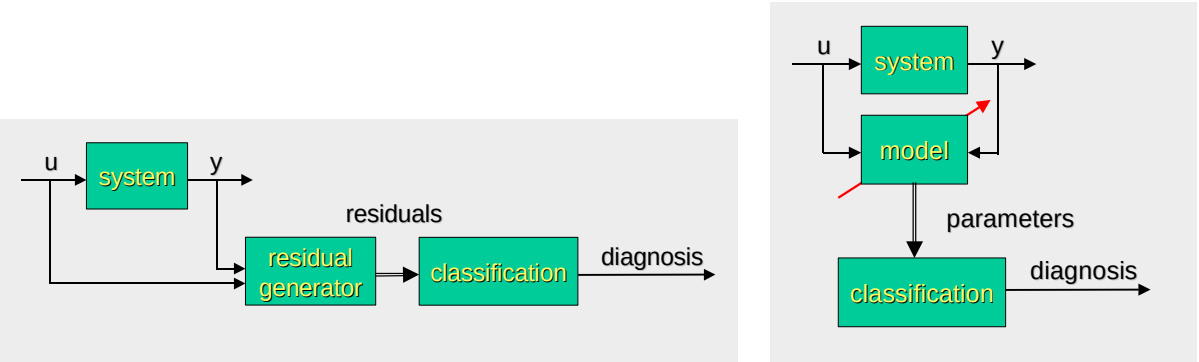

Figure 6: Model-based monitoring: by testing residuals (left) and model parameters (right)

Table 3 contains examples of model-based tests, which belong to general model classes from Table 2, and are described in detail in the forthcoming sections.

Method	**Redundancy**	**Model**	**Objects**	**Tested values**
Differencing in time	temporal	$x(t)=x(t-1)+\varepsilon(t)$	record numbers, aggregated assets	residuals
RLS	spatial	$x_t^i = \hat{\alpha}_t^i \bar{x}_t$	record numbers, aggregated assets	parameters
Ellipsoidal bounding	spatial	$x(t)^T F^{-1} x(t) < \alpha$	aggregated transactions	residuals
PCA	spatial	$\Theta^T x(t) = 0$	aggregated transactions	residuals

Table 3: Example model-based tests

3.1. Monitoring Almost Constant Variables via Differencing in Time

This method is appropriate for exploiting temporal redundancy and testing if the aggregated variables change slowly, e.g., in time/partition or time/customer segment coordinates. Examples of such variables are record counts, asset sums, number of customers, accounts, etc.. The underlying model is $x(t)=x(t-1)+\varepsilon(t)$, where $\varepsilon(t)$ is small compared to $x(t)$.

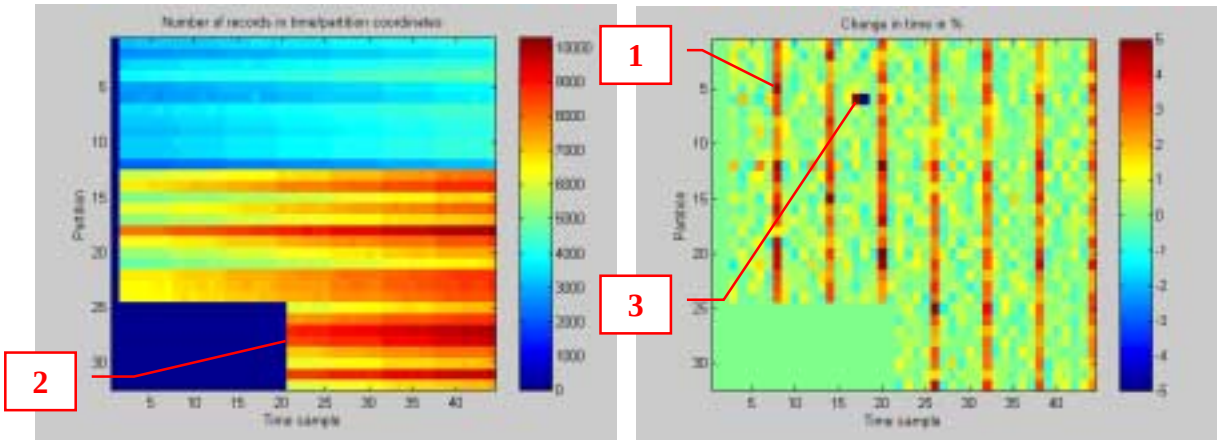

Figure 7: Example of data load monitoring via differencing number of records: left – original number of records, right – relative difference in %

Simulated results shown in Figure 7 depict the original, color-coded record count (left picture) and its relative increase in percent (right picture), generated via the following residual generator: *e(t) = 100*(x(t)-x(t-1))/x(t).*

The monitoring principle is to raise the alarm if bounds on *e(t)* are violated. (Note that monitoring complex seasonal variations like the one shown in Figure 5 requires more advanced time series models like ARIMA [3], but the monitoring principle is still the same.) The residuals *e(t)* are color-coded and graphically presented in partition/time coordinates. The following effects are visible in Figure 7: (1) increase in number of records every 6 time units, (2) the appearance of new partitions #25-32, and (3) small single outlier, not visible in the record counts.

The relative increase in the number of customers/assets can be also graphically presented for the most recent month in customer segment/product coordinates. Such an analysis (called *BusinessBarometer* in DQontrol [1]) is probably even more interesting for database users than for data quality specialists, since it may deliver direct business-related benefits.

3.2. Monitoring Slowly-Varying Relations via Recursive Least Squares (RLS) Method

This method can test if, for a given time instant, the record count for a given partition and time sample is proportional to the total number of records in the table. Periodically, e.g., quarterly loaded tables for which the previously discussed differencing method is not appropriate, have the aforementioned property; an example is shown in Figure 8.

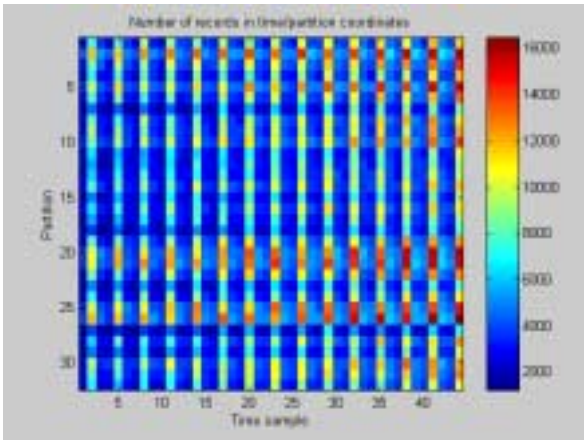

Figure 8: Number of records in an example table with periodic loads

The number of records in each partition x_t^i is modeled as a function $x_t^i = \hat{\alpha}_t^i \bar{x}_t$ of mean for all partitions, denoted $\bar{x}_t$. The coefficient $\hat{\alpha}_t^i$ is estimated using RLS algorithm [7]. Note that the model utilizes spatial redundancy, i.e., proportions between the record counts in all partitions. The algorithm is defined by two equations: the parameter update and covariance update

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \frac{P(t-1)\varphi(t)}{1+\varphi^T(t)P(t-1)\varphi(t)}(y(t) - \varphi^T(t)\hat{\theta}(t)),$$

$$P(t) = \left(P(t-1) - \frac{P(t-1)\varphi(t)\varphi^T(t)P(t-1)}{1+\varphi^T(t)P(t-1)\varphi(t)}\right)\frac{1}{\lambda},$$

where $\theta(t)$ denotes the parameter vector $(\hat{\theta}(t) \equiv \hat{\alpha}_t^i)$, $P(t)$ covariance matrix, $\varphi(t)$ regression vector $(\varphi(t) \equiv \bar{x}_t)$, $y(t)$ modeled variable $(y(t) \equiv x_t^i)$, and λ forgetting factor.

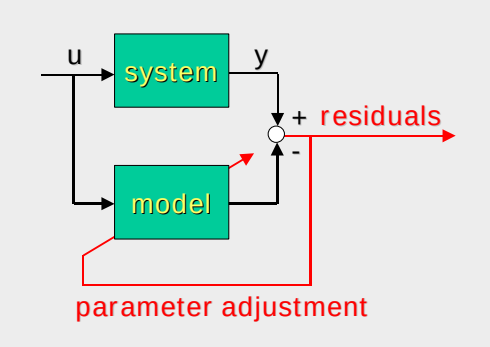

Figure 9: Principle of adaptive modeling used by RLS algorithm

The results, delivered by the RLS algorithm for the fault-free data from Figure 8 are shown in Figure 10. The plots include the modeled number of records in the partition #1, RLS model output, residuals $y(t) - \varphi^T(t)\hat{\theta}(t)$ and model parameter $\hat{\alpha}_t^i$.

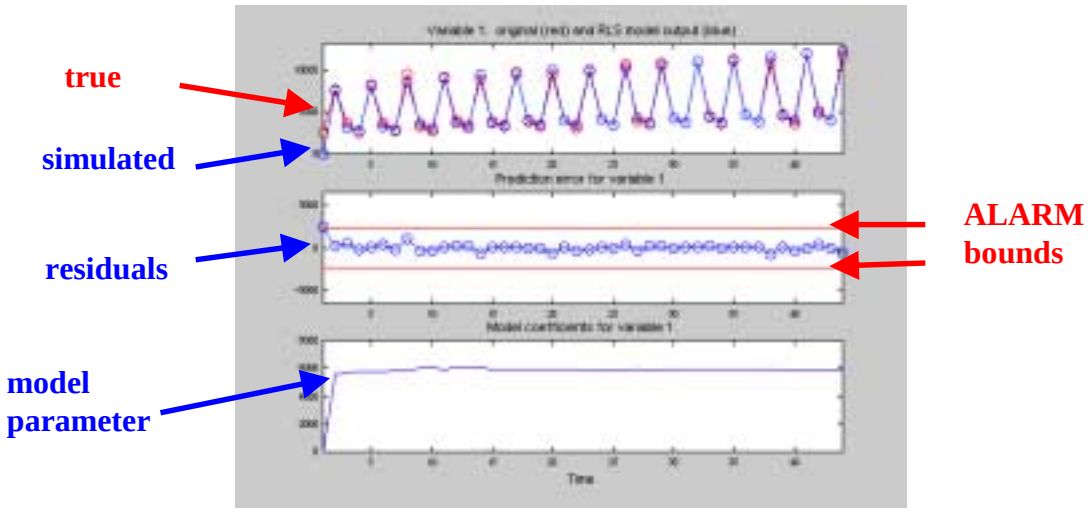

Figure 10: RLS monitoring of the record counts in the first partition

The model parameter remains almost constant, and the residuals stay within the bounds.

3.3. Monitoring Correlation via Ellipsoidal Bounding

This method is appropriate for the monitoring of selected aggregated variables for specified product and customer segment with deterministic relations between the aggregated variables. The data related to different partitions and time instants are assumed to stay within hyperellipsoidal bounds. The hyperellipsoid origin is at $\bar{x}$, and it is defined by the quadratic form $(x-\bar{x})^T F^{-1}(x-\bar{x}) = \alpha$, where

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i , \qquad F = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})(x_i - \bar{x})'$$

The data can be validated using the following condition $(x-\bar{x})^T F^{-1}(x-\bar{x}) < \alpha$ and visualized together with the bounding hyperellipsoids for varying values of the parameter α.

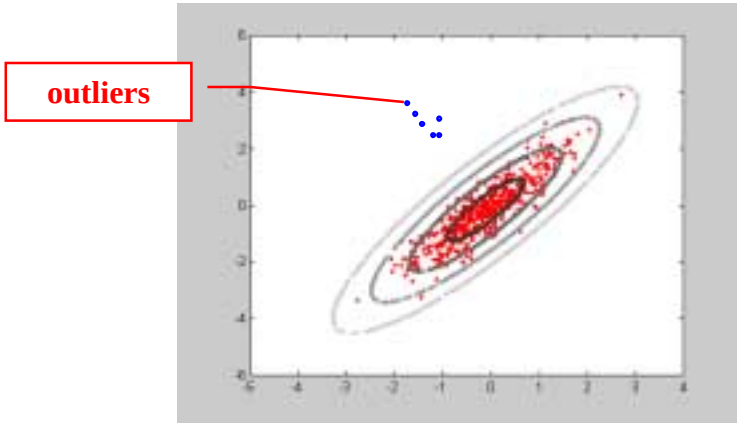

Figure 11: Example dependence between the centered number of accounts and sum of assets

3.4. Monitoring of Aggregated Variables via Principal Component Analysis (PCA) Algorithm

This method can be used for monitoring an arbitrary number of variables, e.g., aggregated for specified product and customer segments. The aggregated data related to different partitions and time instants are assumed to be approximately located in some hyperspace, i.e., to exhibit a high degree of redundancy.

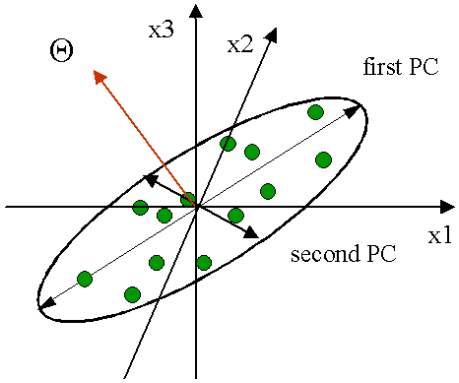

Figure 12: Principle of PCA modeling: data define a hyperellipsoid, the least hyperellipsoid axes Θ are almost orthogonal to the data

The Principal Component Analysis (PCA) Model

The aggregated, centered, and normalized data are stored in the matrix $X \in \Re^{K\times N}$, such that its columns correspond to N variables and rows $x(k)^T \in \Re^N$ to K time samples: $X := [x(0) \quad x(1) \quad \cdots \quad x(K-1)]^T$. The Singular Value Decomposition $X = U\Sigma V^T$, where $U \in \Re^{K\times K}$ is an orthogonal matrix, $\Sigma \in \Re^{K\times N}$ is a diagonal matrix $\Sigma = [diag(\sigma_i) \,|\, 0_{(K-N)\times N}]$, $i = 1..K$ with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_K \geq 0$, and $V \in \Re^{N\times N}$ is an orthogonal matrix, enables determination of the model. First, the elbow at the plot of singular values (Figure 13) has to be found to fix the model order L. The coefficient matrix Θ is given as the last N-L column vectors v of V $\Theta = [v_{L+1} \quad v_{L+2} \quad \cdots \quad v_N]$, while the orthonormal vectors spanning the model hyperspace are the first L column vectors of V, $P = [v_1 \quad v_2 \quad \cdots \quad v_L]$. Note that the model utilizes spatial redundancy (multivariate proportions between the aggregated variables).

PCA Fault Diagnosis Algorithm

Fault diagnosis is performed in the following steps

- Fault detection via evaluation and assessment of the norm of the primary residuals

$$res = \left\| \Theta^T x \right\|_2 \tag{1}$$

- Fault isolation via evaluation and assessment of the norm of the structured residuals

$$res_i := \sqrt{x^T \Pi_i [I - P(P^T \Pi_i P)^{-1} P^T] \Pi_i x} \tag{2}$$

where Π_i is a diagonal matrix with ones for retained variables and zeros for eliminated variables. Note that structured residuals are sensitive to faults in the retained variables and completely insensitive to faults in the eliminated variables.

- Fault-free reconstruction of the data

$$\breve{x}_i = -(\breve{\Theta}\breve{\Theta}^T)^{-1}\breve{\Theta}\hat{\Theta}^T \cdot \hat{x}_{-i}, \qquad (3)$$

where $\breve{x}_i$ is a vector containing the reconstructed variable(s). $\hat{x}_{-i}$ is a vector containing all variables except the variable(s) *i* to be reconstructed. $\breve{\Theta}$ contains only the column(s) *i* of Θ and $\hat{\Theta}$ contains the remaining columns.

Application Example

The presented example utilizes simulated data but follows a real financial application. The processed data contain various customer transactions for a given product and customer segment. The data are aggregated within partitions and time periods. In the test example there are 19 variables in 5 partitions, collected during 30 time samples. An additional 5 variables contain the numbers of summed entries in each partition. (Altogether there are 100 variables.)

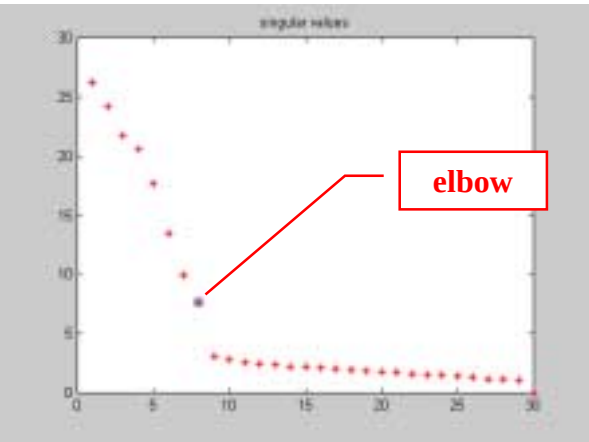

Figure 13: Singular values of the data matrix suggest model with 8 degrees of freedom

Figure 13 depicts singular values of the normalized and centered fault-free reference data matrix. The suggested model order L = 30-8 = 22.

Following possible real situations it is assumed that a simulated fault may corrupt: (i) single variable in one partition, or (ii) single variable in all partitions, or (iii) all variables in one partition. Note that for each case the Π_i matrices in (2) are different. Figure 14 shows the primary residuals $\Theta^T x$ and the norm of the structured residuals (2) for an example fault corrupting all variables from the partition 2 (#21-39) for the time sample 9.

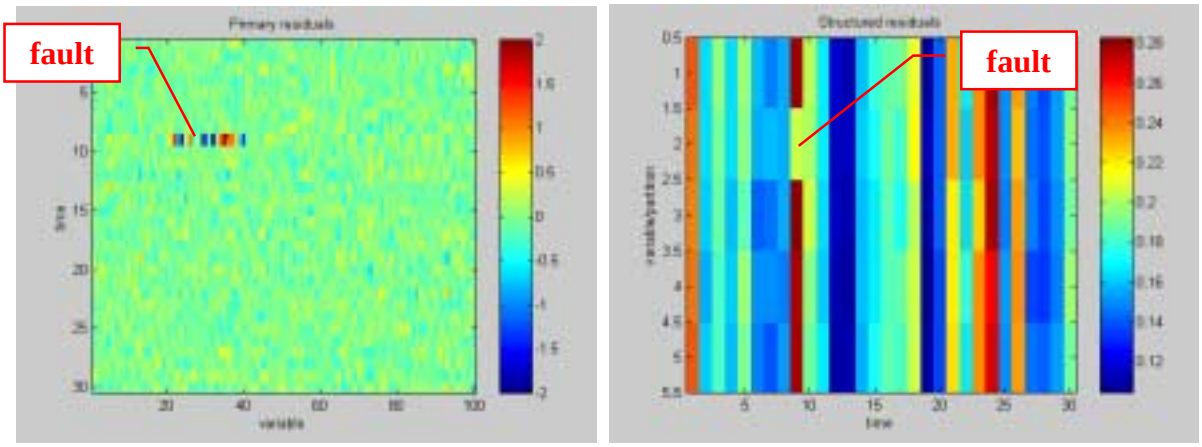

Figure 14: Primary residuals (left) and structured residuals (right)

The fault causes an increase of the norm of the primary residuals $\Theta^T x$ (left picture), and decrease of the structured residuals (2) related to partition 2 (right picture), and is correctly isolated. Hence, the fault-free reconstruction of the corrupted data via (3) is possible. Figure 15 shows such a reconstruction of the variable #22 from partition 2, which enables an assessment of the fault's magnitude and sign, at least in the terms of the aggregated quantities.

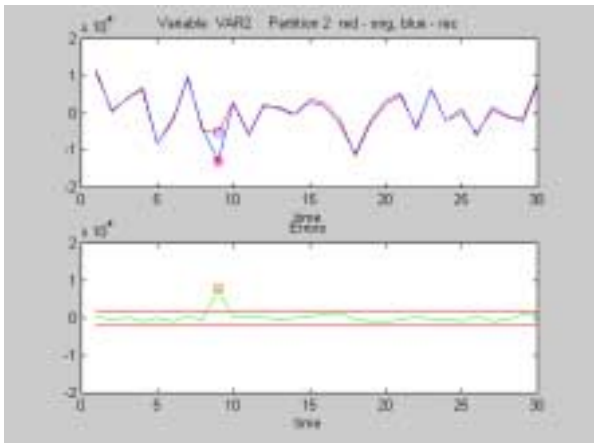

Figure 15: Fault-free reconstruction (3) of the corrupted data (above) and the residuals (below)

4. Fault Visualization, Diagnosis, and Compensation

Automated measurement of data quality produces large amounts of findings within a possibly short time period. In an example database with 2000 variables and 5 tests per variable, a total of 10000 aggregated test results must be evaluated. In order to cope with such amounts of findings, the subsequent processing must be highly automated and exploit appropriate visualization methods.

4.1. Drill-Down Visualization and Alarming

Aggregated test results, presented in the form of signatures, can be further compressed to enable quick assessment of the overall database quality. Figure 16 shows an example of a drill-down traffic-light like assessment of variables and tables. Depending on number and distribution of the test failures their result can be assessed as satisfactory (green), warning (yellow), or alarm (red). The results of all quality tests for one variable can be combined into a joint quality assessment. At the next stage quality assessments of all variables in one table are used to build overall quality assessment of the table.

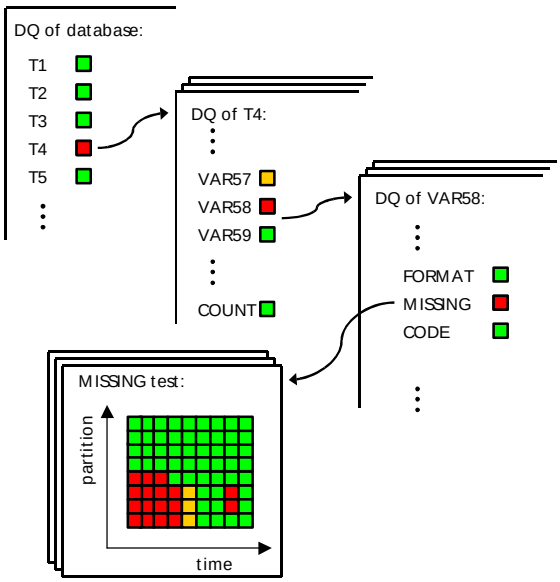

Figure 16: Example of a hierarchically structured data quality assessment

Additional alarms can be associated with a decrease of the quality at any stage, using the previously described differencing of the fault signatures. A good practice is to require that each major alarm is acknowledged by data quality personnel. Note that color-coded signaling, drill-down visualization of the monitored system's state, and alarm acknowledgements are state-of-the-art tools used in modern process monitoring systems.

4.2. Fault Diagnosis via Signature Clustering

Correlation or clustering of test signatures can be exploited to relate a detected data quality symptom to other similar symptoms, and, finally, to their original cause. All symptoms belonging to the same cluster can then be corrected simultaneously, accelerating thus the analysis and correction of the data quality findings. Furthermore, by identifying and correcting the underlying cause of the findings, future data quality problems can be eliminated. Similarity of signatures can be assessed using clustering or correlation techniques. The numerical burden can be kept low, e.g., by reducing number of dimensions by additional aggregation prior to the signature clustering.

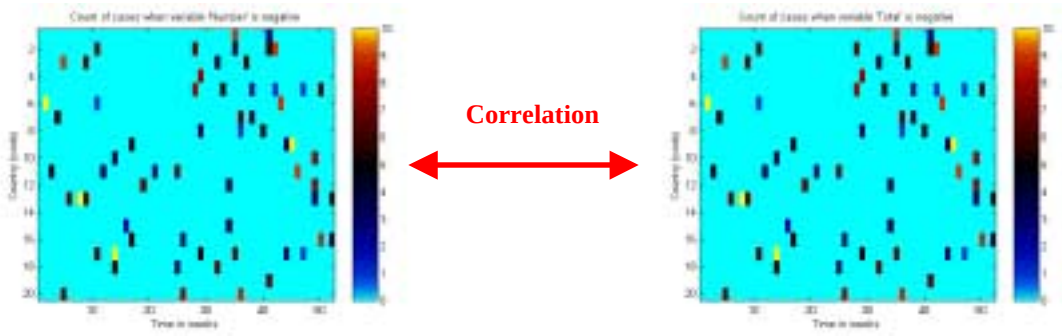

Figure 17: High correlation of the signatures suggests that both symptoms have a common cause

4.3. Data Cleansing

Automatic data correction means elimination or, at least, reduction of data quality problems by explicit modification of the original data. The correction can be performed on the variable level (modification of value) or record level (record elimination). Data correction must be implemented with care since the original data are manipulated. In certain situations like existence of an alternative data format or completely redundant variables full data correction is possible (there is no information loss). More often, however, the fault recovery information is not available, so full fault correction is not possible. Often it is better to replace the faulty (misleading) value by the missing value (what prevents numerical use of the incorrect value).

Data cleansing can be useful also when performed for the aggregated data, as in the previous PCA modeling example: comparing real aggregated values to the ones reproduced by the model enable assessment of the data loss and can be very helpful for the fault diagnosis. The discussed PCA fault isolation method can be also useful to select better information source if one of two highly correlated variables is corrupted, see [9] for a detailed algorithm.

5. Conclusions

This paper discusses several process control-related methods applied in the context of monitoring and control of data quality in financial databases. It shows that the control process can be considered a classical control feedback process, and that application of the model-based methods is useful to detect, diagnose, and eliminate data quality problems. Moreover, the model-based methods give an insight into business-related information contained in the data. The methods constitute part of DQontrol product [1], and have been applied to the data quality monitoring of a real financial database at a customer site, delivering business benefits, such as improvements of the modeling quality, a reduction in the number of the modeling cycles, and better data understanding. These benefits in turn lead to financial savings and better utilization of highly skilled data analysts.

References

[1] F. Block, J. Milek, M. Reigrotzki. (2000). Datenqualität als Basis künftiger Business Intelligence Applikationen. SAS Warehousing 2000, Zürich.

[2] G.E.P. Box, G.M. Jenkins. (1970). Time Series Analysis: Forecasting and Control. Holden-Day.

[3] R. Busatto. (2000). Using Time Series to Assess Data Quality in Telecommunications Data Warehouses. Proceedings of the 2000 Conference on Information Quality, Cambridge, MA, pp. 129-136.

[4] T. Dasu, T. Johnson, E. Koutsofios. (2000). Hunting Data Glitches in Massive Time Series Data. Proceedings of the 2000 Conference on Information Quality, Cambridge, MA, pp. 190-199.

[5] R. Elliott, L. Aggoun, J. Moore. (1995). Hidden Markov Models. Estimation and Control. Springer Verlag.

[6] J. Gertler. (1998). Fault Detection and Diagnosis in Engineering Systems, Marcel Dekker.

[7] L. Ljung. (1998). System Identification: Theory for the User. 2nd edition, Prentice Hall.

[8] S. Makridiakis, S. C. Wheelright, R. Hyndman. (1998). Forecasting: Methods and Applications. 3rd edition, John Wiley and Sons Inc..

[9] J. Milek, F. Kraus. (2000). Use of Analytic Redundancy in Fault-Tolerant Sensor Systems. IMEKO 2000 International Measurement Confederation, XVI IMEKO World Congress, Vienna, Austria, Proceedings Volume V, pp. 121-127.

[10] J. Milek, O. Hermann, F. Kraus. (2000). Use of Hypersurfaces for Fault Detection, Isolation, and Reconstruction. IFAC 4th Symposium on Fault Detection Supervision and Safety for Technical Processes. SAFEPROCESS 2000, Budapest, pp. 1199-1204.

[11] J. Milek. (2002). Diagnose in der Messtechnik. Chapter in Handbuch der industriellen Messtechnik, in preparation, 7th edition by Pfeifer, Ruhm, and Modigell, Oldenbourg Verlag.

[12] X. Z. Wang. (1999). Data Mining and Knowledge Discovery for Process Monitoring and Control. Springer Verlag.

[13] S. M. Weiss, N. Indurkhya. (1998). Predictive Data Mining. Morgan Kaufmann Publishers, Inc..