An efficient framework for data stream analysis[1]

Gianfranco Cellarosi, Claudio Sartori

DEIS - University of Bologna, IEIIT - CNR
Viale Risorgimento, 2, 40136 Bologna, Italy
gcellarosi@deis.unibo.it, claudio.sartori@unibo.it

Abstract

A data stream is commonly intended as a flow of data that has to be analyzed where the flow rate and the amount of data are so high, are so high that we are not able to analyze them. In the context of data stream, we solve the problem of time series analysis and alarm detection. This paper, rather than focusing on the specific analysis problem, introduces a framework devised to make technically possible an efficient and effective application of the various analysis tools, given the data stream constraints.

Keywords: Forecasting, temporal datamining, sliding windows, data stream.

1. Introduction

In this paper we discuss the architectural choices and the performances of a framework to survey the behaviour of time series and to detect events that take place when "something" changes in the real world process (represented by the time series). Some theoretical aspects have been studied from the authors in microbiological environment [1] [2].
We take the opportunity to introduce in a few words the present framework. Stock prices, customer transactions, health care and total quality management are examples of applications originating time series, and for which an analysis of the evolution in time could be useful. In these areas time series analysis is typically used to forecast trends, but there is interest also for raising alerts when there is reason to suspect that something is changing in the world, and that some corrective action should be taken. In this perspective, timeliness and accuracy of the alarm are the major requirements, and of course they can be contrasting, since the computing time required by a more accurate analysis in some cases could not allow to deal with the new data arrivals [1].

A time series is a typical form of time dependent data: a time series is defined as a set of observation ordered by time. Statisticians use to deal with time series by building "models" whose parameters are derived by observation of the past behaviour. For example, we can study behaviour of price stock and we can evaluate a forecast to buy or to sell. Large amounts of data can be a problem, since the forecast cost can be very high: too many data, too many models to be evaluated in a short time.

This is the problem: how deal with the analysis of large DataStream? Can we find problem instances where the computational costs are low enough to keep up with the arrival stream? Can we simplify time series analysis? As a particular instance of the time series problem, we consider the output of a microbiological laboratory. Everyday, patients' specimens are examined to determine the presence of the infecting agents and possibly the sensibility of such agents to various pharmacological principles. The problem is therefore to identify a threshold, above which the number of positive detections is to be considered abnormal, taking into account random variations. In typical settings, this implies to analyze continuously four/five hundreds of time series with thousands of values everyday.

One feasible solution seems to be the reduction of the amount of data to be considered, by introducing the concept of "sliding window", i.e. consider only a window with the "last n data". In this way part of the history, the oldest one, is disregarded, and results with some degree of approximation are obtained. Most of the proposals following this guideline chooses (with some criteria) the window size and then goes on with model evaluation. But at the best of our knowledge, nobody uses the window size as one of the parameters to be estimated.

In summary, we iterate the following activities, until a satisfactory solution is reached: a) select one of the predefined mathematical models, b) select model parameters that best fit the observed data c) choose the

1 The work is partly supported by the Italian MIUR Project "European Citizen in E-Governance"

smallest sliding window size allowed without affecting the results d) evaluate the confidence interval.

In section 2 we briefly recall the works done in the area of time series analysis; in section 3 we set up the definitions necessary to introduce our methods; in section 4 we describe our framework and its architectural aspects and, finally, in section 5 we report the experimental results and in section 6 we discuss the performances.

2. Related Work

Paper [6] explores a parametric approach using ARMA, which is a classical statistical tool. Our work in a sense includes this toll, since ARMA is one of the model we evaluate, to obtain the best fit to the real data.

Paper [8] performs time series anomaly detection by generating states and rules. Our framework also sets a rule to generate a forecast, but, on the contrary of this paper, we don't allow the user to easy modify the rules found by the system, because we trust in the mathematical apparatus used to generate the rules.

Paper [3] applies the clustering methods to a set of continuously sliding windows for grouping similar temporal patterns disperse along the time series.

Paper [7] discusses the handling of the sliding deals with the problem of basic counting. Our pursuit is directed to define the width of sliding window.

Paper [4] searches an effective and efficient solution in Piecewise Linear Representation. We don't use PLR in our experiments, but the idea of our framework is to find better solution to modelling time series. Besides, this paper introduce a sliding window algorithm for PLR. But we determine sliding windows with another system because we don't use graphical solution.

3. Definitions

- Data Stream: *set of continuously incoming information about transactions.*

With such information it is always possible to form Xt, an ordered sequence of real numbers which are the measure of the phenomenon of interest (Xt,t=1,2,n) where t indicates a point in time.

- Adherence: *means measure of distance, which occurs between two series. We use Euclidean distance in this section.*

The verification of adherence is therefore one major aim our framework. Time series analysis is normally carried out in order to both post-evaluate a given phenomenon and to provide support for decision-making. The aim of our work is to provide an instrument helpful to decision making. To obtain it we compare the evolution of our numerical succession with its expected value: the more our model fits to reality, the more it is possible to talk about the model's adherence to reality. Adherence between model and reality is verified in terms of probability. This means that some differences between the model(s) and reality is considered normal. We usually want to reduce time series from interval $]-\infty, t]$ into interval $]k, t]$. Reduction is powered by a 'k' where $-\infty < k < t$. Time always has an assigned and fixed resolution. We accept to reduce efficacy in order to increase efficiency.

- Efficacy: *is the ability to obtain expected results.*
- Efficiency: *is the ability to obtain results with the minimum cost.*

We pursue a reduction of the evaluation and I/O time. Adherence is calculated considering the following factors:

- Synopsis: *a data structure used to represent a set of data with a smaller space occupation, and some degree of approximation.*

Synopses are commonly used as a data reduction method to obtain approximate answers in data stream queries [9].

- Temporal window: *let two instants t and k the temporal window is the ordered series as regards time included in between [t-k,t] where k<t.*
- Sliding window: *is a temporal window of constant width which moves together with time.*

For each model it is therefore possible to make hypotheses about the creation of a function expressing an efficacy and efficiency function (efficacy loss in favour of efficiency and vice-versa). shows the advancing of the sliding window with width k w.r.t. timeline.

4. The framework

The main problem we are facing is the continuous flow of incoming information and the subsequent need verify to continuously the adherence between forecasting and reality. A data-stream collects information regarding a large set of classes (generally thousands). For example a continual flow of information on negotiations is normally observed in a price stock data-stream. Yet another complexity factor is the random inflow of information concerning a certain object class; within the price stock framework, for example, we do not know beforehand what object class will be our next incoming information. As we aim at providing real-time decision support, our system must come into action at the arrival of each object in order to verify adherence. The higher the number of classes, the more demanding data-stream

monitoring will be. What happens, though, if the elaboration time necessary to compute the adherence of the actual model to the data is higher than the data inflow time? That is to say that our system cannot support the speed of data inflow. Our decision support cannot therefore be provided in real-time, due to a systematic delay for each object coming into the data-stream (a queue is formed as the in-flowing data enters the data stream. What's more, each forecast is the result of a model, which can be subject to a different evaluation time. The framework's task is to keep adherence evaluation time below the objects detection time. Therefore, an estimation of the top-limit time for verification of adherence, must be function both of the single time series' characteristic (for example variance, selected model) and of the number of time series, (which must undergo adherence verification), and of the average time for information inflow in the data-stream. Although there exists a number of studies and research on various methodologies for the evaluation of adherence, the effectiveness of such methodologies comes at a high price in terms of efficiency.

The framework is basically made up of two sets of software applications, one for the creation of a synopsis of the object classes time series and a second one for the real-time verification of. The first set of applications produces:

- Synopsis: a compact vision of the original data-stream content; it is basically an aid to navigation and elaboration on data-stream.
- Adherence: it means to obtain a tradeoff between efficiency and effectiveness.

The synopsis is necessary to determine which measures are to be included, if the objective is the building of a time series model. The measures useful will be those summarising time series characters.

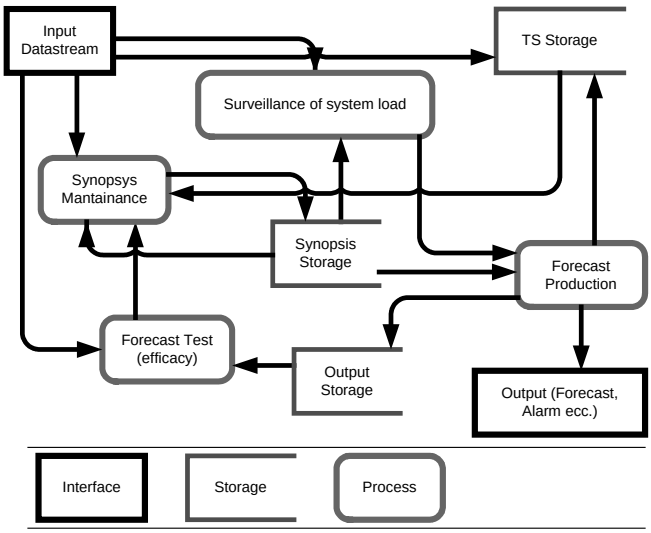
In literature we find several experiments giving prominence to the possibility of using DataStream or time series in order to identify particular events.

A DataStream may collect data both regarding a sample and population. During the analysis of data the following statistical questions have to be worked out

- which is the optimal size k of the temporal sliding window?
- how can the control of adherence be used in order to reduce parameter evaluation errors?

The temporal windows, which are identified on the same class of objects, are presumably the empirical manifestation of a certain phenomenon. The study of the time series involves spotting a model, which would fit the series with a negligible error E. It is presumed that this phenomenon will not incur a model variation or parameters variations. It is definitely possible to predict values for the series for a whole number of successive moments (instants). The following conditions could take place:

Picture 1 Data Flow Diagram

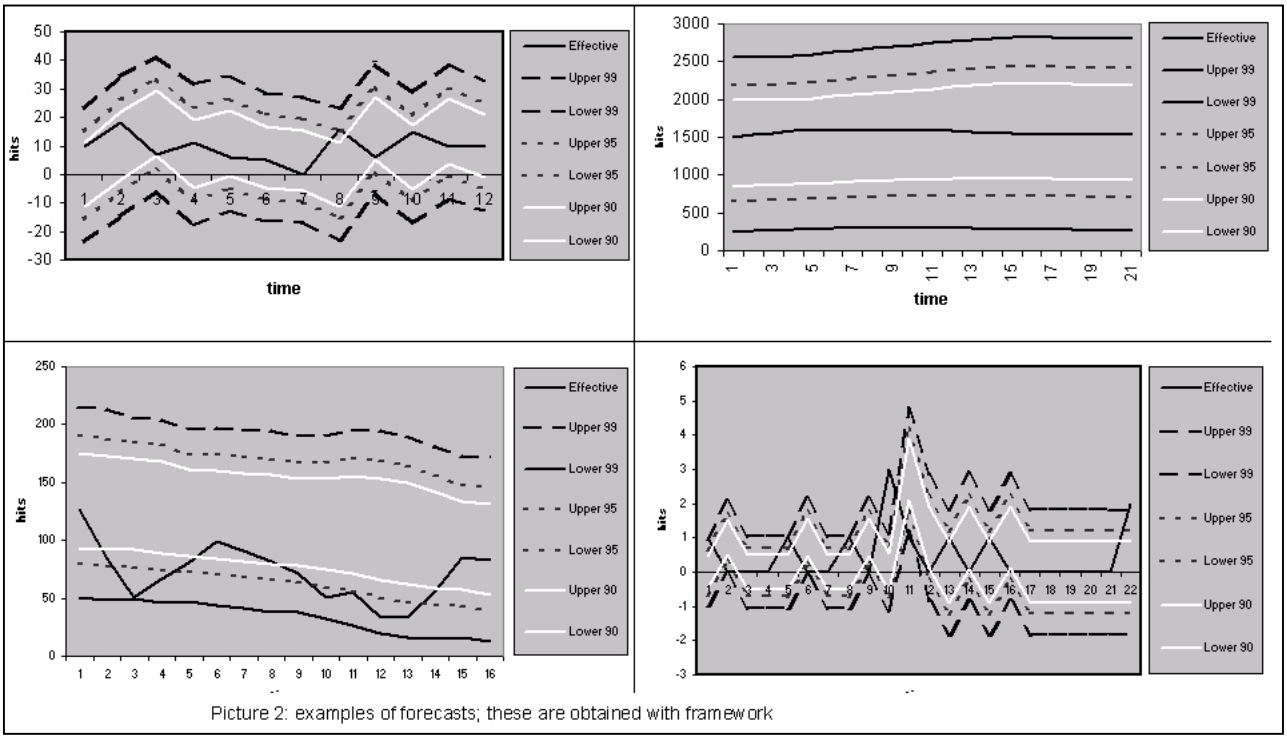


The data queued into input stack are useful both for Data Stream construction and for control activity. For example, for synopsis maintenance or for evaluation of system load.

- The event determines an observation which is far from the model but with no changes in the parameters
- The event determines a variation in the parameters

The framework, supported by a system of rules, must take a decision. In order to take this decision the system must/can use behavioural crossroads. On the basis of studies carried out within the frame of other scientific works, a possible variation in the model or in its parameters may indicate the manifestation of an uncommon event, which can be used for detecting alarms. To perform forecast, the composition (generally with a product) of two functions is used: one depending on time f(t) and the other depending on the Distribution law g(d). The forecast P is therefore made up of P(f(t),g(d)). We can try to reduce the window by searching for a substantial invariance in both the parameters and the model. I can try to reduce the window by searching for a substantial invariance of the distributing properties such as the mean and variation of the model (g(d)). In this framework we will operate in a way, which is different from the canonical statistic approach, which involves a deep preliminary study before being able to select an appropriate model series and a window width that is able to satisfy the user's expectations.

From the point of view of methodology in the process of data collection, the time series typology suffers from a great lack of homogeneity. It is preferable not to use far too deep theoretical studies. The framework is therefore destined to contain a mechanism studying functions f(t) and g(d), on the basis of which it adopts certain strategies.



Picture 2: examples of forecasts; these are obtained with framework

5. Experiments

In this section we show some experimental results in order to demonstrate the efficacy and efficiency of our framework.

Picture 2 shows four examples of forecast provided with confidence interval. The main issue is the difference of forecasts depending on the confidence interval. As it was expected, highly irregular time series require a higher confidence interval (e.g. 99), corresponding to the wider allowed band. Variation could be considered normal, if it doesn't exceed the forecast. On the converse, if it exceeded an alarm is lead off.

In some appliances we have on one hand limit enforced by nature of time series (e.g. negative temperature in Kelvin Scale is not permitted), on the other hand we have mathematical forecast which can indicate logical/illogical value.

In our experiment we consider a standard time unit, called 'day'; out of this experiment, obviously, time scale depends from environment of application (day, hour, week…). To prevent these errors it is useful to establish a set of rules. This problem is a side effect because we search a 'vertical' solution in our framework for all environment of time series. We deploy in our framework an application of statistical analysis of time series.

We have tested the forecasting capabilities with different widths of sliding windows, starting from the same end day. If we enlarge the width of the sliding window we obtain a convergence of the forecasting value.

We have selected MSE[2], to empathize variance and forecasting in function of an enlargement of the sliding window from the same end day. In our example we use MSE to select the model which best fits the real data. The selected model minimizes the MSE.

We have tested the real difference between forecasts from same data but with different level of confidence.

We have seen from one side a continuous change of model selected depending from width of temporal window, on the other side a stabile model selection. This experience point out a large difference that we have among time series. Starting from synopsis construction, interesting data, which regard the model selection, come into light. On a large DataStream it can be seen that each series has fixed behavioral characteristics, just a kind of genetical behavior, written in the DNA of time series.

So we use synopsis for:

- Evaluate width of temporal window
- Evaluating confidence.
- Evaluating of variance, in order to estimate the confidence interval, the variance is a very important parameter of incidence.

2 Mean square error (MSE) is a criterion for an estimator: the choice is the one that minimizes the sum of squared errors due to bias and due to variance

6. Performances

The experiment we have carried out aims at reducing temporal windows and complexity of achieving actual time saving. Making forecasts starting from a time series is not something new. There are various methodologies that allow achieving effective results. But efficiency is what we are looking for, a system allowing us to obtain a good result with a reduction of complexity. In recursive models, reduction of width of sliding windows means a reduction of number of recursion.

We are therefore looking for a compromise between effectiveness in terms of forecast and efficiency in terms of time used to achieve that forecast.

The synopsis of each single class is not composed by just one record, but by a set of records, each one referred to a temporal window, [m,k]and generated by cycles. M decreases in a monotone way with step of 1 unit of time considered, to each step m=m-1 (it decreases of 1 unit of time). I can gather photos of time series according the variance that is the threshold evaluated and selected model related to performance.

We have tested many classes (many time series), from many environments; in particular someone of these are from UCR Dataset[3]

We have encountered various results but that is normal that is expected global result because every time series has a certain natural property inside the same series.

7. Conclusion and future work

We have demonstrated a useful of a framework to apply Statistical time series analysis of a generic DataStream. Future work will be devoted to increase performance of our framework, by improving the quality of information saved in synopsis and/or by changing the analysis methods of synopsis.

Our experiments show that our framework is effective, and guarantees the best performances, given a limited computation time. The advantage in terms of computation time, can vary from 12% to 80%, depending on the characteristics of time series. Additional work on the management of synopsis can greatly improve the performance.

8. References

[1] G. Cellarosi, S. Lodi and C. Sartori - A Practical Solution to Detect Outbreak by time series analysis - Proceedings of the 15th CBMS Ieee Symposium.

[2] G. Cellarosi, C. Sartori – Synopsis for Microbiological Data Stream Analysis – Proceedings of the CBMS 2005 Ieee Symposium.

[3] S. Poliker, A.B. Geva. - Non-Stationary Signal Analysis using Temporal Clustering - 1998 IEEE Signal Processing Society Workshop on Neural Networks for Signal Processing, Cambridge, England.

[4] E. Keogh, S. Chu, D. Hart, and M. Pazzani. - An online algorithm for segmenting time serie - Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on , 29 Nov.-2 Dec. 2001 Pages:289 - 296

[5] J. Box, G. Jenkins et R. Reinsel - Time Series Analysis : Forecasting and control - Prentince Hall-Third edition 1994 - ISBN 0-13-060774-6.

[6] K. Deng, A. W. Moore and M. C. Nechyba. - Learning to Recognize Time Series: Combining ARMA models with memory based learning. - 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation.

[7] M. Datar, A. Gionis, P. Indyk, R. Motwani. Maintaining Stream Statistics over Sliding Windows. - Proceedings of the 29th VLDB Conference 2003

[8] S. Salvador, P. Chan & J. Brodie. - Learning States and Rules for Time Series Anomaly Detection - Proceedings of 17th Intl. FLAIRS Conference 2004.

[9] P. B. Gibbons, Y. Matias - Synopsis Data Structures for Massive Data Sets, ACM-SIAM Symposium on Discrete Algorithms (SODA 99).

[3] www.cs.ucr.edu