UDC 519.814
519.876.2

Special Guest Issue

CDQM, Volume 8, Number 3, 2005, pp. 55-61

COMMUNICATIONS
IN
DEPENDABILITY AND
QUALITY
MANAGEMENT
An International Journal

Statistical process control of the stochastic complexity of discrete processes

***Armin Shmilovici*[1*] *and* *Irad Ben-Gal*[2]**

[1]*Department of Information Systems Engineering, Ben-Gurion University of the Negev, Beer-Sheva, 84105, Israel, E-Mail: armin@bgumail.bgu.ac.il*
[2]*Department of Industrial Engineering, Tel-Aviv University, Ramat-Aviv, Tel-Aviv 69978, Israel, E-Mail: bengal@eng.tau.ac.il*

accepted October 24, 2005

Summary

Changes in stochastic processes often affect their description length, and reflected by their stochastic complexity measures. Monitoring the stochastic complexity of a sequence (or, equivalently, its code length) can detect process changes that may be undetectable by traditional SPC methods. The context tree is proposed here as a universal compression algorithm for measuring the stochastic complexity of a state-dependent discrete process. The advantage of the proposed method is in the reduced number of samples that are needed for reliable monitoring.

Key words: *Process control; Control charts; Stochastic complexity; Context tree algorithm.*
Mathematics Subject Classification: *62M02, 93E35, 62M10, 62P30.*

1. INTRODUCTION

Most statistical process control (SPC) methods use a Shewhart type control chart to monitor attributes or variables (e.g., mean and standard deviation) of the controlled process. The observed statistics are often assumed to be independent and identically distributed (i.i.d.) and, in many cases, also normally distributed. As evidenced by the wide spread implementation of the Shewhart control charts, this tool has proved to be very useful in practice.

Nonetheless, there are practical situations where the i.i.d. and the normality assumptions are grossly inaccurate, and can lead to false alarms or to late detection of faults. For example, many industrial processes are being controlled by a feedback controller that takes an action whenever the process deviates from a pre-specified set-point. These actions introduce dependency among consecutive observations and deviation from the normality assumption.

[*]corresponding author

The need to find substitutes for the traditional Shewhart control charts has been recognized in the literature. Model specific methods, such as the EWMA, the *exponential weighted moving average*, assume a known underlying time series model and need relatively large amounts of data to produce accurate monitoring. These methods often can not capture more complex process' dynamics, such as the ones that are modeled by Hidden Markov Models (HMM). Model generic methods, such as the ITPC (Alwan et al., 1998) and the CSPC (Ben-Gal et al., 2003), rely on asymptotic properties based on information theory as a replacement for explicit distribution assumption on the process characteristics. In these methods, as well, large amounts of data are needed either for deriving an empirically based control limit, or for using an analytically derived control limits that are based on asymptotic considerations.

The *stochastic complexity* of a sequence is a measure of the number of bits needed to represent and reproduce the information in a sequence. This statistic is commonly used as a yardstick for choosing between alternative models for an unknown time series.

Universal coding methods have been developed to compress a data sequence without relying on prior assumptions regarding the properties of the data generating mechanism. Universal coding algorithms – typically used for data compression – model the data for coding in a less redundant representation. The size of the compressed data is a practical measure of the stochastic complexity of the process. Some universal algorithms are known to have asymptotic performance as good as the optimal non-universal algorithms. This means that for long sequences, the model provided by the universal source behaves like the “true” unknown model for all tasks one wishes to use it for, such as coding, prediction, and decision making in general. Here we extend this list of tasks also to statistical process control.

In this paper we use Rissanen (1983) context tree algorithm, which is a well known universal coding model, for measuring the stochastic complexity of a time series. The advantage of this specific algorithm, in comparison to many other universal algorithms, is that it has been known to have a quick non-asymptotic convergence rate (Rissanen, 1983, Ziv 2002). Thus, it can be used to compress even relatively short sequences, like the ones often available from industrial processes.

The key idea in this paper is to monitor the stochastic complexity (or, equivalently, the code length) of a data sequence. The context tree model is used as the universal model for measuring the stochastic complexity of a state dependent discrete process. The advantages of this method are in the relatively small number of samples that are needed for reliable monitoring and in its computational tractability.

2. THE CONTEXT TREE METHOD

Following the notation in Ziv (2000, 2002), let us consider a discrete sequence emitted by a stationary source with N+1 symbols $X_{-N}^{0} \equiv X_{-N}, X_{-N+1}, \ldots, X_0$, where each symbol x_i belongs to an alphabet A of cardinality $|A|$. In the *estimation problem*, one has to estimate $P(X_1 \mid X_{-N}^{0})$, i.e., the unknown conditional probability distribution of X_1 given the sequence X_{-N}^{0}.

Consider the class of universal conditional probability measures that count the recurrence of the longest suffix of X_1 in X_{-N}^{0}. The suffix – $X_{-K_0(X_{-N}^{0})}^{0}$ – is a subsequence of X_{-N}^{0} called also as the *context*, where K_0 is itself a function of the observed symbols often satisfying $K_0(X_{-N}^{0}) << N$ (Ziv, 2000).

In the context tree, $\mathrm{T} \equiv T(S, \Theta_S)$ with a structure S and parameters Θ_S, a context is represented by the path of arcs starting at the tree root until reaching a leaf (an external node). A context is represented in the tree in reversed order with respect to the order of observance. Thus, deeper nodes in the tree correspond to previously observed symbol sequences in the process. The lengths (depth) of various contexts (branches in the tree) are not necessarily equal. The conditional probabilities for symbols are estimated given every context in the tree and are given in the tree nodes, as demonstrated in *Figure 2*. Given a context tree, compression can be obtained as a result of the recurring patterns in the data. Each node in the tree is related to a specific recurring context (sub-sequence). The original sequence can be coded by the sub-sequences in the context tree. A sequence that does not belong to the same class of sequences from which the context tree was generated (trained) is expected to obtain a lower compression rate with respect to sequences that belong to that class (for further details on the context trees see also Rissanen, 1983, Ziv, 2000, 2002 and Ben-Gal et al., 2003).

3. SPC FOR THE STOCHASTIC COMPLEXITY BASED ON THE CONTEXT TREE MODEL

The proposed SPC procedure has two stages. First, in the training stage a "reference" tree is trained from "in-control" data, Y_{-N+1}^{0}. Then, in the monitoring stage sequences are being compressed based on the reference tree and their compression rate is plotted on a control chart against predefined control limits.

Three parameters have to be set for developing an SPC procedure that monitors the stochastic complexity of a sequence: N, the sequence length on which the reference context tree is trained; $K_{\max}$, the maximal depth of the initial context tree that is later trimmed to obtain the reference context tree; and $\hat{N}$, the sequence length for the monitored sequences for which the compression statistic is computed based on the reference context tree. According to Ziv (2002), one can set the depth of the context tree such that $P(X_1 \mid X_{-K_{\max}}^{0}) \geq \frac{1}{|A|L}$, where $(|A|L)^{-1}$ is assumed to be the smallest probability of a symbol to occur at *any* context in the tree. The value of L effectively determines the resolution of the probabilities in the tree, and the minimal probability change that is detectable by the tree. The other parameters satisfy:

$$\hat{N} \geq K_{\max}^{3} \text{ and } N > \hat{N}|A|L^{3}. \tag{1}$$

Thus, the order for defining the parameters is $L \rightarrow K_{\max} \rightarrow \hat{N} \rightarrow N$.

The stochastic complexity of any monitored sequence $X_1^{\hat{N}}$ that is prefixed by a context X_{-D+1}^{0}, and generated by the same source that generated the training sequence Y_{-N+1}^{0} can be measured by the universal algorithm based on the reference context tree. There exists a recursive method to calculate the stochastic complexity measure (Ziv, 2000, 2002):

$$-\log_2\left(\Pr\left(X_1^{\hat{N}} \mid X_{-D+1}^{0}, \mathrm{T}\right)\right) = -\log_2\left(\prod_{j=1}^{\hat{N}} \Pr\left(X_j \mid X_{-D}^{j-1}, \mathrm{T}\right)\right) = -\sum_{j=1}^{\hat{N}} \log_2\left(\Pr\left(X_j \mid X_{j-K_1(j)}^{j-1}\right)\right), \tag{2}$$

where $D < \hat{N}$ is chosen to comply with the initial conditions and $K_1(j) \equiv K_1\left(X_j, X_{j-D}^{j-1}, \mathrm{T}\right)$. The first equality relies on the probability multiplication chain rule and the second equality relies on the dependences found by the tree structure. The stochastic complexity of the monitored sub-sequence is measured by the sum of the negative log probabilities of symbols given their context in that specific sub-sequence of size $\hat{N}$. The specific context $K_1(j)$ of the *j*-th symbol depends on the previous symbols (including those in the prefix X_{-D+1}^{0} to that sub-sequence) and the parameters of the tree (obtained from the training sub-sequence).

The context tree T with $|S|$ leafs can be represented by a multinomial distribution $\Pr\left(X_j \mid X_{j-K_1(j)}^{j-1}\right)$. The conditional probabilities of symbols, $\Pr\left(X_j \mid X_{j-K_1(j)}^{j-1}\right)$, are independent when the context tree is trained on a sufficiently long sequence Y_{-N+1}^{0}, $N >> |S|$. In such case, the expression is approximately the sum of i.i.d. random variables. The value of the stochastic complexity of the sequence $X_1^{\hat{N}}$ prefixed by the sequence X_{-D+1}^{0} is a random variable with $|S|^N$ possible discrete values. For a sufficiently large $\hat{N}$, the distribution of the stochastic complexity can be approximated with the Central Limit Theorem (Ziv, 2000, 2002).

Following the above results, we propose the following scheme for an SPC based on the stochastic complexity of the monitored process as a measure for its stability:

- Use a sequence of observation Y_{-N+1}^{0} from the *in-control* process to construct a reference context tree $\mathrm{T} \equiv T(S, \Theta_S)$.
- From the multinomial distribution. $\Pr\left(X_j \mid X_{j-K_1(j)}^{j-1}\right)$, of symbols represented in T calculate the first two moments of the distribution of the stochastic complexity. These will be used later for computing the control limits either empirically or by methods explained in Ben-Gal et al., (2003).
- Denote by q_1 and q_2 the required False Alarm Rates (FARs) with respect to the upper control limit (UCL) and the lower control limit (LCL). Use a control charts with $q_1 = q_2 = 0.00135$ to comply with traditional charts with a type-I probability error of 0.27%. The density of the stochastic complexity is typically a-symmetric, so a correction may be needed for the control bounds.
- For every monitored sequence of length $\hat{N}$ use the reference context tree to compute its stochastic complexity. Insert a point in the control charts and monitor it with respect to the UCL and LCL.

4. EXAMPLE: SPC FOR A SINGLE STAGE PRODUCTION SYSTEM

Following the example in Ben-Gal et al. (2003), consider a system of two machines: M_1, M_2 separated by a finite-capacity buffer B of size 3. Machine M_2 attempts to process a part whenever it is not starved, i.e., whenever there are parts in its input buffer. Machine M_1 attempts to process a part whenever it is not blocked. *Figure 1* presents the state transition diagram for the buffer queue where the in-control production probabilities of the machines are, respectively, $p_1 = 0.9$ (the transition probability from an empty buffer to a buffer with one part) and $p_2 = 0.8$ (the transition probability from a full buffer to a buffer with two parts) respectively. Monitoring the buffer size at regular intervals results in a discrete sequence of buffer states. For illustration purposes we simulated the queues' state machine to generate a sequence of length 1,000,000:

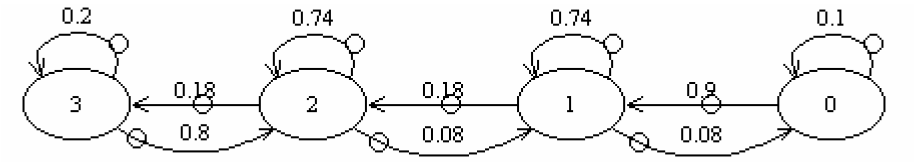

Figure 1. The state transition diagram of the queue in the "in-control" production system

Figure 2 presents the "in-control" distribution of buffer levels in the referenced process, $\Pr_a(X_j \mid X_{j-1})$, in the form of an analytical context tree T_a. Note that it is a single-level tree with $|S| = 4$ contexts and a symbol set of size $|A| = 4$. The root node presents the system steady-state probabilities and the leaves presents the transition probabilities given the current state:

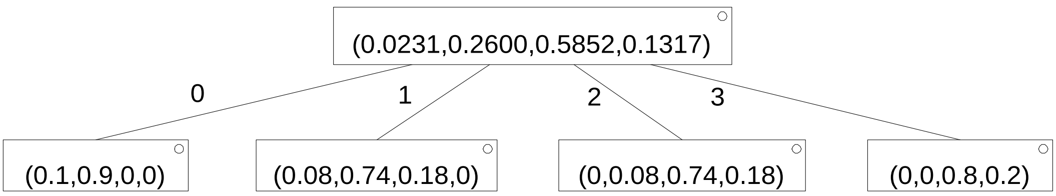

Figure 2. In-control context tree, T_a, based on the "in-control" process with $p_1 = 0.9, p_2 = 0.8$

Figure 3 presents the modified distribution of buffer levels $\Pr_b(X_j \mid X_{j-1})$ by a second context tree, T_b. The modified process was generated by selecting new production probabilities: $p_1 = 0.7, p_2 = 0.9$. The tree is used for simulating the "out of control" sequences. As may be well-recognized, the tree shows the probability differences. Note that in this case the presented distributions in both trees, T_a and T_b, can be computed analytically by the first-order Markov process in *Figure 1*. For validation purposes, we also estimated the parameters of the context tree from a sequence of observations generated by the modified process, which turned out to be accurate within two decimal digits to those represented in T_b.

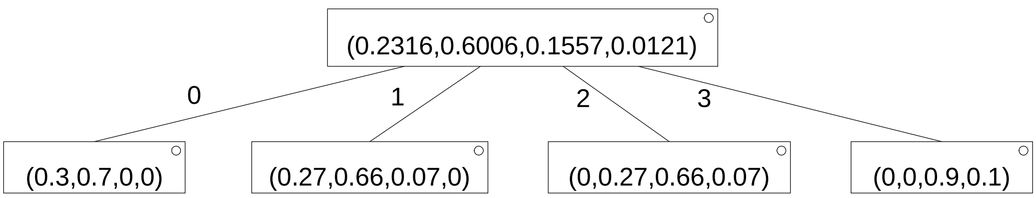

Figure 3. Context tree, T_b, for the modified process parameters with $p_1 = 0.7, p_2 = 0.9$

In this example, the (analytic and empirical) context trees are of depth 2. Following (1) it leads to monitored sequences of size $\hat{N} \geq 2^3$. Accordingly, we computed the stochastic complexity of two monitored sequence of lengths $\hat{N} = 8$ (and 16) based on (2). The control limits were determined to be the upper and lower 1.0% percentile. The stochastic complexity of the lower control limits was found to be 3.3627 (8.765) and for the upper control limit 14.8794 (25.5307), respectively, for the monitored sequence length 8 (16). Note the doubling the sequence length roughly doubles its mean stochastic complexity. The statistics from 50,000 monitored sequences of the out-of-control process resulted in detecting 16.76% (22.41%) as out of control. *Figure 4* presents a typical example for monitoring the stochastic complexity of sequence of length 16. The process is "in control" for the first 50 monitored sequences and deviates of control (via the process described by *Figure 3*) starting from monitored sequence 51. As we can see, the *average run length* for the in control process tends to infinity: $ARL_{in\text{-}control} \rightarrow \infty$. The process change is detected in this example almost immediately with an $ARL_{out\text{-}of\text{-}control} \rightarrow 4.5$. In comparison and as demonstrated in Ben-Gal et al. (2003), such a change in a state-dependent process can not be captured by traditional SPC charts, including charts of known SPC methods that are used for dependent data.

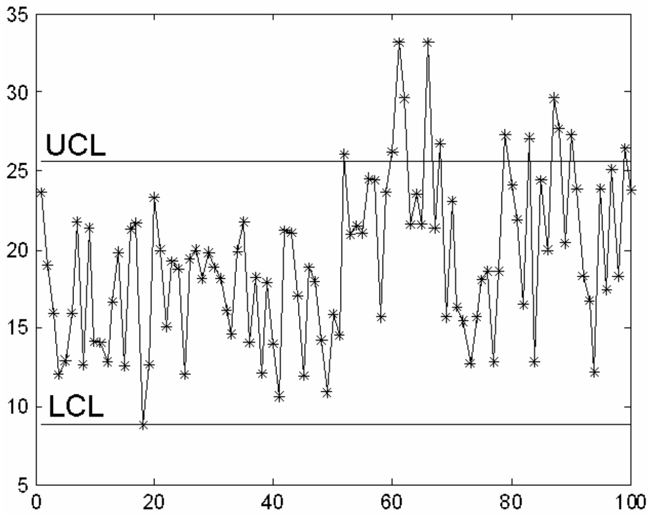

Figure 4: A SPC for the stochastic complexity of sequences of length 16; the process changed at the 51th sequence

5. CONCLUSIONS

In this paper we proposed an SPC method based on the stochastic complexity of the monitored process. The stochastic complexity measures rely on a context tree as a universal model that can capture complex dependencies in the data. In the given example, the stochastic complexity measure turned out to be sensitive to process changes.

The advantages of the proposed method are two-fold: i) it is generic and suitable for many types of discrete processes with complex and unknown dependencies; ii) it is suitable for relatively short data sequences due to the context tree's convergence properties. The viability of the proposed SPC method and its advantages were illustrated by a numerical experiment.

REFERENCES

[1] Alwan, L.C., Ebrahimi, N. and Soofi, E.S. (1998), Information theoretic framework for process control, *European Journal of Operations Research*, 111: 526-542.

[2] Ben-Gal, I., Morag G. and Shmilovici, A. (2003), CSPC: A monitoring procedure for state dependent processes, *Technometrics*, 45 (4): 293-311.

[3] Rissanen, J. (1983), A universal data compression system, *IEEE Transactions on Information Theory*, 29 (5): 656-664.

[4] Ziv, J. (2000), A universal prediction lemma and applications to universal data compression and prediction, *IEEE Transactions on Information Theory*, 47 (4): 1528-1532.

[5] Ziv, J. (2002), An efficient universal prediction algorithm for unknown sources with limited training data, from: *www.msri.org/publications/ln/msri/2002/infotheory/ ziv/1/*