CHANGE-POINT DETECTION WITH SUPERVISED LEARNING AND FEATURE SELECTION

Victor Eruhimov, Vladimir Martyanov, Eugene Tuv
Intel, Analysis and Control Technology, Chandler, AZ, U.S.A.

George C. Runger
Industrial Engineering, Arizona State University, Tempe, AZ, U.S.A.

Keywords: Data streams, ensembles, variable importance, multivariate control.

Abstract: Data streams with high dimensions are more and more common as data sets become wider. Time segments of stable system performance are often interrupted with change events. The change-point problem is to detect such changes and identify attributes that contribute to the change. Existing methods focus on detecting a single (or few) change-point in a univariate (or low-dimensional) process. We consider the important high-dimensional multivariate case with multiple change-points and without an assumed distribution. The problem is transformed to a supervised learning problem with time as the output response and the process variables as inputs. This opens the problem to a wide set of supervised learning tools. Feature selection methods are used to identify the subset of variables that change. An illustrative example illustrates the method in an important type of application.

1 INTRODUCTION

Data streams with high dimensions are more and more common as data sets become wider (with more measured attributes). A canonical example are numerous sensors (dozens to hundreds) with measurements generated from each over time. Many characteristics can be of interest from a system that generates such data, but one systemic question is whether the system has been stable over a time period, or whether one of more changes occurred. In a change-point problem, historical data from streams is reviewed retrospectively over a specified time period to identify a potential change, as well as the time of the change. This historical analysis differs from real-time monitoring where the goal is to detect a change as soon as it occurs.

Change points are of interest in areas as diverse as marketing, economics, medicine, biology, meteorology, and even geology (where the data streams represent data over depths rather than over time). In medicine, a change-point model can be used to detect whether the application of a stimulus affects the reaction of individual neurons (Belisle et al., 1998). In the study of earthquakes, it is of interest to distinguish one seismicity from another (Pievatolo and Rotondi, 2000).

Modern data streams often must handle high dimensions. A common approach is to use a multivariate control chart for process monitoring such as Hotelling's T^2 control chart (Hotelling, 1947). This is a widely-used multivariate control chart to monitor the mean vector of a process based on the Mahalanobis distance of the current data vector from a historical mean data vector. The distance measure used in T^2 incorporates the correlations among the attributes that are measured. However, because this distance measure is fundamentally based on a sum of squared deviations of the elements of the current vector, it loses sensitivity to change points that occur in only one or a few attributes among many (and result in small changes in Mahalanobis distance). More sensitive extensions were developed for real-time monitoring such as a multivariate exponentially weighted moving average control charts (MEWMA) (Lowry et al. 1992), and a multivariate cumulative sum control charts (MCUSUM) (Runger and Testik 2004). These extensions are still based on sums of squares with the previously mentioned, intrinsic limitations as the dimension increases.

Eruhimov V., Martyanov V., Tuv E. and C. Runger G. (2007).
CHANGE-POINT DETECTION WITH SUPERVISED LEARNING AND FEATURE SELECTION.
In *Proceedings of the Fourth International Conference on Informatics in Control, Automation and Robotics*, pages 359-363
DOI: 10.5220/0001631303590363

The objective here is to handle the high-dimensional, complex data that is common in modern sensed systems, and still detect change point that might occur in only one (or a few) variable among hundreds. Consequently, we present a two-phased approach. In the first phase we identify the attributes responsible for the change point. With a much smaller subset of attributes to work with in the second phase, simpler methods can be used to identify the time(s) at which the change(s) occur. The first phase uses a novel transformation of the problem to one of supervised learning. Such a transformation was explored by (Li et al., 2006). The work here adds a second phase, uses a much more powerful feature selection algorithm, and provides a more challenging example. In Section 2 the change-point problem is transformed to a supervised learning problem. Section 3 discusses feature selection. Section 4 provides a realistic example.

2 CHANGE POINTS WITH SUPERVISED LEARNING

A supervised learning model requires a response or target variable for the learning. However, no obvious target is present in a change-point problem. Still, a key element of a data stream is the time attribute that provides an ordering for the measured vectors. In a stationary data stream without any change points, no relationship is expected between time and the measured attributes. Conversely, if the distribution changes over time, such change should allow for a relationship to be detected between the measured attributes and time (Li et al., 2006). Consequently, our approach is to attempt to learn a model to predict time from the measurements in the data stream

$$t = g(x_1, ..., x_p) \tag{1}$$

where t is the time of an observation vector and $g()$ is our learned model. If the time attribute can be predicted, a change in the measurement vectors is available to predict. Attributes that are scored to be important to this model are the subset of important variables. Consequently, phase one of our analysis can be completed from this model and its interrogation. Any number of change points can occur in this framework.

A more direct approach might attempt to model each attribute as a function of time such as $x_j = g(t)$ for $j = 1, 2, \ldots, p$. However, separate models do not use the relationships among the variables. A change might break the relationships between variables within a significance difference in each variable individually. Common examples in data streams depict points that are not unusual for any attribute individually, but jointly depict an important change.

Any monotonic function of time can be used as the target attribute for the learner. The identify function used here is a simple choice and other functions can be used to highlight or degrade the detection of change points in different time periods. Also, any one of many supervised learners can be applied. Our goal is to detect a subset of important variables and this is the primary purpose for our following selection.

Because we are most interested in an abrupt change in the mean of one or more attributes in the data stream it is sensible to use a supervised learner that can take advantage of such an event in the system. Furthermore, the phase one objective is to identify the important variables. Consequently, decision trees are used as the base learners because they can effectively use a mean change in only one or few predictor attributes. They also have intrinsic measures of variable importance. Ensembles of decision trees are used to improve the measure of variable importance for the phase one objective.

3 FEATURE SELECTION

If an attribute changes over time, it should be more useful to predict time than an attribute that is statistically stable. Consequently, the phase to identify changed attributes is based on a feature selection method for a supervised learner. There are several approaches such as filter, wrapper, and embedded methods. An overview of feature selection was provided by (Guyon and Elisseeff, 2003) and other other publications in the same issue. Also see (Liu and Yu, 2005). The feature selection phase needs to process hundreds of attributes and potentially detect a contribution of a few to the model to predict time. Furthermore, in the type of applications of interest here, the attributes are often related (redundant). Consequently, the effect of one attribute on the model can be masked by another. Moderate to strong interactive effects are also expected among the attributes. Consequently, a feature selection methods need high sensitivity and the ability to handle masking and interactive effects. We use a feature selection methods based on ensembles of decision trees.

Tree learner are fast, scalable, and able to handle complex interactive effects and dirty data. However, the greedy algorithm in a single tree generates an unstable model. A modest change to the input data can make a large change to the model. Supervised ensemble methods construct a set of simple models (called base learners) and use their vote to predict new data. Numerous empirical studies confirm that ensemble methods often outperform any single base learner (Freund and Schapire, 1996), (Dietterich, 2000). Ensembles can be constructed as parallel or serial collections of base learners. A parallel ensemble combines independently constructed base learners. Because different errors can cancel each other, an ensemble of such base learners can outperform any single one of its components (Hansen and Salamon, 1990), (Amit and Geman, 1997). Parallel ensembles are often applied to high-variance base learners (such as trees). (Valentini and Dietterich, 2003) showed that ensembles of low-bias support vector machines (SVMs) often outperformed a single, best-tuned, canonical SVM (Boser et al., 1992).

A well-known example of a parallel ensemble is a random forest (RF) (Breiman, 2001). It uses subsampling and to build a collection of trees and injects additional randomness through a random selection of variable candidates of each node of each tree. The forest can be considered a more sophisticated bagging method (Breiman, 1996). It is related to random subspace method of (Ho, 1998). A forest of random decision trees are grown on bagged samples with peroformance comparable to the best known classifiers. Given M predictors a RF can be summarized as follows: (1) Grow each tree on a bootstrap sample of the training set to maximum depth, (2) Select at random $m < M$ predictors at each node, and (3) Use the best split selected from the possible splits on these m variables. Note that for every tree grown in RF, about one-third of the cases are out-of-bag (out of the bootstrap sample). The out-of-bag (OOB) samples can serve as a test set for the tree grown on the non-OOB data.

In serial ensembles, every new learner is based on the prediction errors from previously built learners so that the weighted combination forms an accurate model. A serial ensemble results in an additive model built by a forward-stagewise algorithm and *Adaboost* introduced by (Freund and Schapire, 1996) is the best-known example.

Neither parallel nor serial ensembles alone are sufficient to generate an adequate best subset model that accounts for masking, and detects more subtle effects. A simple example by (Tuv, 2006) illustrated this. In some cases, weak but independent predictors are incorrectly promoted in the presence of strong, but related predictors. In other cases the weak predictors are not detected. An integrated solution is expected to provide advantages and several concepts described previously were integrated into a best subset selection algorithm by (Tuv et al., 2007). Only a brief summary is provided here. The best-subset algorithm contains the following steps:

1. Variable importance scores are computed from a parallel RF ensemble. Each tree uses a fixed depth of 3-6 levels. There are some modified calculations based on OOB sample that are described in more detail by (Tuv et al., 2007).
2. Noise variables are created through a random permutation of each column of the actual data. Because of this random permutation, the noise variables are known to not be associated with the target. The noise variables are used to set a threshold for statistically significant variable importance scores to select important (relevant) variables
3. Within decision trees, surrogate scores can be calculated from the association between the primary splitter at a node and other potential splitters. The details were originally provided by (**?**). These surrogate scores describe how closely an alternative splitter can mach the primary. This is turn provides a measure of masking between these variables. When such scores are combined from all nodes in a tree and all trees in an ensemble, a robust metric for variable masking can be obtained. A masking matrix is computed and noise variables are again used to determine significance thresholds. A set of short serial ensemble is used.
4. Masked variables are removed from the list of important variables
5. The target is adjusted for the currently identified important variables, and the algorithm is repeated. The adjustment calculates generalized residuals that apply to either regression of classification problems. Less important variables can be more clearly identified once the dominant contributors are eliminated. Trees-based models are not well-suited for additive models and the iteration substantially improves the performance in these cases.

4 ILLUSTRATIVE EXAMPLE

Because change-point detection is an unsupervised learning task, simulated data is used with known change points inserted. A data set to mimic a real manufacturing environment includes 10 sensors that each generate time series (with 100 time data points) from given distributions. Each time series could be represented as a trapezoid with added curvatures, an oscillation with random phase in the center, and Gaussian noise on the order of 10% of the signal. Curvatures and the center oscillation phase are sampled from fixed uniform distributions. This set of time series provides the results for one batch and the objective is to detect changes in a series of batches. The dataset consists of 10000 batches and each 1000 samples there is a change induced by a shift in one of internal parameters used to generate time series by its standard deviation.

Such high-dimensional data can be analyzed directly, or a different representation can be used to extract features that might be of interest. For example, Fourier transforms, discrete wavelet transforms, and orthogonal polynomials are only a few of the methods to represent high-dimensional data. Without a priori information of features affected by a change point, the set of features is extracted from these methods is still often quite large.

Chebyshev polynomials are used here to represent this high-dimensional data. The representation is $y(x) = T_n(x)$ where $T_n(x)$ by definition is a polynomial solution of degree n of the equation

$$(1-x^2)\frac{d^2y}{dx^2} - x\frac{dy}{dx} + n^2 y = 0, \qquad (2)$$

where $|x| \leq 1$ and n is a non-negative integer. For $n = 0$ $T_0(x) = 1$. Chebyshev polynomials can also be calculated using one of useful properties: $T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x)$ and $T_n(x) = \cos(n \cdot \cos^{-1}(x))$.

A set of Chebyshev polynomials $\{T_n(x)\}_{n=0,1,...}$ is orthogonal with respect to the weighting function $(1-x^2)^{-1/2}$:

$$\int_{-1}^{1} \frac{T_m(x)T_n(x)dx}{\sqrt{1-x^2}} = \left\{ \begin{array}{l} \frac{1}{2}\pi\delta_{nm}, n>0, m>0 \\ \pi, n=0, m=0 \end{array} \right. , \qquad (3)$$

where δ_{mn} is the Kronecker delta.

Using the last property we can represent any piecewise continuous function $f(x)$ in the interval $-1 \leq x \leq 1$ as a linear combination of Chebyshev polynomials:

$$\sum_{0}^{\infty} C_n T_n(x) = \left\{ \begin{array}{l} f(x), \text{where } f(x) \text{ is continuous} \\ \frac{f(x-0)+f(x+0)}{2} \text{ in discontinuity points} \end{array} \right. \qquad (4)$$

Here

$$C_n = \frac{A}{\pi} \int_{-1}^{1} \frac{f(x)T_n(x)dx}{\sqrt{1-x^2}}, \qquad A = \left\{ \begin{array}{l} 1, n = 0 \\ 2, n > 0 \end{array} \right. . \qquad (5)$$

For a function $\{f_i\}_{i=1,...,P}$ defined on a discrete domain we calculate the coefficients of the Chebyshev decomposition using a straightforward formula:

$$C_n = \frac{A}{\pi} \sum_{i=1}^{T} \frac{f_i T_n(x_i)}{\sqrt{1-x_i^2}}, \qquad (6)$$

where $x_i = -1 + \frac{2}{P}(i - \frac{1}{2})$. Therefore, the coefficients $\{C_n\}$ become the features for the change-point detection. We use first 25 coefficients for each time series resulting in 250 features for each sample.

In the first phase of the analysis the feature selection algorithm simply uses a sequential batch index as the target. The polynomial coefficients provide the inputs. The feature selection module identifies the distribution change and a set of features responsible for the change.

In the second phase, moving T^2 statistics are calculated using only the selected features between n_1/n_2-samples prior/after the current data point, correspondingly, to detect the change points:

$$T^2 = \frac{n_1 n_2 (n_1 + n_2 - 2)}{n_1 + n_2} (\overline{\mathbf{y}}_1 - \overline{\mathbf{y}}_2)' \mathbf{W}^{-1} (\overline{\mathbf{y}}_1 - \overline{\mathbf{y}}_2), \qquad (7)$$

where

$$\begin{aligned} \mathbf{W} &= \sum_{j=1}^{n_1} (\mathbf{y}_{1j} - \overline{\mathbf{y}}_1)(\mathbf{y}_{1j} - \overline{\mathbf{y}}_1)' \qquad (8) \\ &+ \sum_{j=1}^{n_2} (\mathbf{y}_{2j} - \overline{\mathbf{y}}_2)(\mathbf{y}_{2j} - \overline{\mathbf{y}}_2)'. \end{aligned}$$

In the second phase, moving T^2 statistics are calculated using only the selected features between n_1/n_2-samples prior/after the current data point, correspondingly, to detect the change points:

$$T^2 = \frac{n_1 n_2 (n_1 + n_2 - 2)}{n_1 + n_2} (\overline{\mathbf{y}}_1 - \overline{\mathbf{y}}_2)' \mathbf{W}^{-1} (\overline{\mathbf{y}}_1 - \overline{\mathbf{y}}_2), \qquad (9)$$

where

$$\begin{aligned} \mathbf{W} &= \sum_{j=1}^{n_1} (\mathbf{y}_{1j} - \overline{\mathbf{y}}_1)(\mathbf{y}_{1j} - \overline{\mathbf{y}}_1)' \qquad (10) \\ &+ \sum_{j=1}^{n_2} (\mathbf{y}_{2j} - \overline{\mathbf{y}}_2)(\mathbf{y}_{2j} - \overline{\mathbf{y}}_2)'. \end{aligned}$$

We retrain a model with each 200 samples using all samples from the previous detected change point until the current sample. We do not make predictions on the first 100 samples in the beginning and after a change point. The results are shown in Figure 1. Here T^2 is shown with feature selection in the top figure and without feature selection in the bottom figure. Notice that changes are not detected before feature selection improves the sensitivity of the control chart. After feature selection the changes are apparent.

5 CONCLUSIONS

As sensors continue to flourish in numerous disciplines, high-dimensional data becomes more common. Furthermore, the ability to detect changes in

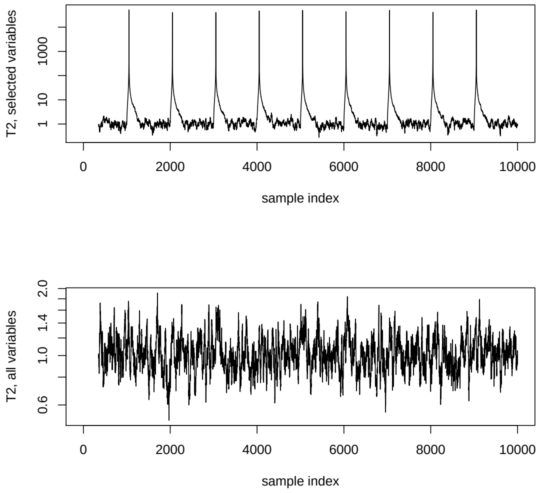

Figure 1: Time series plot of T^2 with (top) and without (bottom) feature selection.

a system or process over time remains an important need in many applications. The results here illustrate the success of a solution that integrates several important elements. The transform of the inherently unsupervised learning problem of change-point detection to one of supervised learning with a time index as the response, opens the analysis to a wide collection o tools. A sophisticated feature selection algorithm can then be applied to detect attributes that contribute to a change. In the lower-dimensional space of these attributes, the change point detection is a much simpler problem and a number of simpler tools can be applied. We uses a multivariate T^2 control chart, but other control charts, or methodologies can be considered after the important dimensional reduction. The illustrative example presents an simulation of an important practical case. One needs to summarize the information from multiple time series. Consequently, the dimensional space equals the number of series times the length of each series and the feature selection becomes critical, and the example illustrates an effective solution method for this problem.

ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 0355575.

REFERENCES

Amit, Y. and Geman, D. (1997). Shape quantization and recognition with randomized trees. *Neural Computation*, 9(7):1545–1588.

Belisle, P., Joseph, L., Macgibbon, B., Wolfson, D. B., and Berger, R. D. (1998). Change-point analysis of neuron spike train data. *Biometrics*, 54:113–123.

Boser, B., Guyon, I., and Vapnik, V. (1992). A training algorithm for optimal margin classifiers. In Haussler, D., editor, *5th Annual ACM Workshop on COLT, Pittsburgh, PA*, pages 144–152. ACM Press.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2):123–140.

Breiman, L. (2001). Statistical modeling: The two cultures. *Statistical Science*, 16(3):199–231.

Dietterich, T. G. (2000). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning*, 40(2):139–157.

Freund, Y. and Schapire, R. E. (1996). Experiments with a new boosting algorithm. In *the 13th International Conference on Machine Learning*, pages 148–156. Morgan Kaufman.

Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182.

Hansen, L. K. and Salamon, P. (1990). Neural network ensembles. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(10):993–1001.

Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(8):832–844.

Hotelling, H. (1947). Multivariate quality control-illustrated by the air testing of sample bombsights. *Techniques of Statistical Analysis*, pages 111–184.

Li, F., Runger, G. C., and Tuv, E. (2006). Supervised learning for change-point detection. *IIE Transactions*, 44(14-15):2853–2868.

Liu, H. and Yu, L. (2005). Toward integrating feature selection algorithms for classification and clustering. *IEEE Trans. Knowledge and Data Eng.*, 17(4):491–502.

Pievatolo, A. and Rotondi, R. (2000). Analysing the interevent time distribution to identify seismicity phases: a bayesian nonparametric approach to the multiple change-points problem. *Applied Statistics*, 49(4):543–562.

Tuv, E. (2006). Ensemble learning and feature selection. In Guyon, I., Gunn, S., Nikravesh, M., and Zadeh, L., editors, *Feature Extraction, Foundations and Applications*. Springer.

Tuv, E., Borisov, A., Runger, G., and Torkkola, K. (2007). Best subset feature selection with ensembles, artificial variables, and redundancy elimination. *Journal of Machine Learning Research*. submitted.

Valentini, G. and Dietterich, T. (2003). Low bias bagged support vector machines. In *ICML 2003*, pages 752–759.