A Boosting Approach for Understanding Out-of-control Signals in Multivariate Control Charts

Esteban Alfaro, Jose-Luis Alfaro, Matias Gamez, Noelia Garcia

▶ **To cite this version:**

Esteban Alfaro, Jose-Luis Alfaro, Matias Gamez, Noelia Garcia. A Boosting Approach for Understanding Out-of-control Signals in Multivariate Control Charts. International Journal of Production Research, 2009, 47 (24), pp.6821-6834. 10.1080/00207540802474003 . hal-00530219

HAL Id: hal-00530219
https://hal.science/hal-00530219

Submitted on 28 Oct 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

For Peer

A Boosting Approach for Understanding Out-of-control Signals in Multivariate Control Charts

Journal:	*International Journal of Production Research*
Manuscript ID:	TPRS-2008-IJPR-0271.R2
Manuscript Type:	Original Manuscript
Date Submitted by the Author:	02-Sep-2008
Complete List of Authors:	ALFARO, ESTEBAN; UNIVERSIDAD DE CASTILLA-LA MANCHA, Economía Política y Hacienda Pública, Estadística Económica y Empresarial y Política Económica ALFARO, JOSE-LUIS; UNIVERSIDAD DE CASTILLA-LA MANCHA, Economía Política y Hacienda Pública, Estadística Económica y Empresarial y Política Económica GAMEZ, MATIAS; UNIVERSIDAD DE CASTILLA-LA MANCHA, Economía Política y Hacienda Pública, Estadística Económica y Empresarial y Política Económica GARCIA, NOELIA; UNIVERSIDAD DE CASTILLA-LA MANCHA, Economía Política y Hacienda Pública, Estadística Económica y Empresarial y Política Económica
Keywords:	CONTROL CHARTS, PROCESS CONTROL, NEURAL NETWORK APPLICATIONS
Keywords (user):	boosting trees, Hotelling$\hat{\varepsilon}$s T2 control chart

scholarONE™
Manuscript Central

For Peer Review Only

A Boosting Approach for Understanding Out-of-control Signals in Multivariate Control Charts

E. Alfaro[a], J.L. Alfaro[a1], M. Gámez[a] and N. García[a]

[a] *Facultad de CC. Económicas y Empresariales de Albacete, Universidad de Castilla-La Mancha, Plaza de la Universidad, 1 02071 Albacete, Spain.*

(Received xx May 2008)

The most widely used tools in statistical quality control are control charts. However, the main problem of multivariate control charts, including the Hotelling's T^2 control chart, lies in that they indicate that a change in the process has happened, but do not show which variable or variables are the source of this shift. Although a number of methods have been proposed in the literature for tackling this problem, the most usual approach consists in decomposing the T^2 statistic.

In this paper, we propose an alternative method interpreting this task as a classification problem and solving it through the application of boosting with classification trees. The classifier is then used to determine which variable or variables caused the change in the process. The results prove this method to be a powerful tool for interpreting multivariate control charts.

Keywords: Statistical process control, Hotelling's T^2 control chart, boosting trees

[1] Corresponding author. Email: JoseLuis.Alfaro@uclm.es

For Peer Review Only

1. Introduction

In statistical process control (SPC), univariate techniques are designed to control the quality of the product by analyzing only one product characteristic. In most industrial production processes, however, there are several interrelated characteristics which jointly influence the quality of the final products. Although one possible solution might be to develop univariate control methods for each quality characteristic, a better alternative involves simultaneously controlling each feature using multivariate statistical techniques. Multivariate statistical process control does not only analyze the effect of each characteristic on the quality, but also considers the effect of the interactions among them.

Of all these multivariate techniques three stand out: distance-based methods (basically Hotelling's T^2 control chart) and MEWMA and MCUSUM control charts. In this article, we have used Hotelling's T^2 control chart, which is a multivariate extension of the Shewhart control chart and one of the widest used.

In most multivariate control methods, a statistic is calculated to summarize the information. In this paper we use the T^2 statistic. This depicts dispersion and position measurements of the variables being analyzed. For individual observations, the Hotelling's T^2 statistic at time i is defined by:

$$T_i^2 = (\mathbf{x}_i - \boldsymbol{\mu})' \boldsymbol{\Sigma}^{-1} (\mathbf{x}_i - \boldsymbol{\mu}) \quad (1)$$

where $\mathbf{x}_i$ represents a p-dimensional vector of measurements made on a process at time period i. Let us assume that when the process is in control, $\mathbf{x}_i$ are independent and follow a multivariate normal distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. It should be noted that if $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are known, T_i^2 follows a Chi-square distribution with p degrees of freedom (χ_p^2).

For individual multivariate observations when $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are unknown, we estimate these parameters from a reference sample (phase I). The usual parameter estimators are the sample mean vector ($\bar{\mathbf{x}}$) and the sample covariance matrix ($\mathbf{S}$). Thus, if we consider the sample $\{\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_n\}$ where $\mathbf{x}_i=(x_{i1}, x_{i2},\ldots,x_{ip})$, with x_{ij} the i-th individual observation of the j-th characteristic, the T^2 statistic for $\mathbf{x}_i$ can be constructed in the following way:

$$T_i^2 = (\mathbf{x}_i - \bar{\mathbf{x}})' \mathbf{S}^{-1} (\mathbf{x}_i - \bar{\mathbf{x}}) \quad (2)$$

http://mc.manuscriptcentral.com/tprs Email: ijpr@lboro.ac.uk
For Peer Review Only

where $\bar{\mathbf{x}} = (\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_p)$ and $\mathbf{S}=(s_{uv})_{u,v=1,2,\ldots,p}$. The elements of these estimators are defined as:

$$\bar{x}_u = \frac{1}{n}\sum_{i=1}^{n} x_{iu} \qquad \text{and} \qquad s_{uv} = \frac{1}{n-1}\sum_{i=1}^{n}(x_{iu} - \bar{x}_u)(x_{iv} - \bar{x}_v)$$

The distribution of the T^2 statistic was analyzed in Tracy et al. (1992) and Mason & Young (2002). If $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are unknown and are estimated using $\bar{\mathbf{x}}$ and $\mathbf{S}$ obtained from a historical data set, then the upper control limit at level α (UCL) of the T^2 statistic for a new individual observation vector $\mathbf{x}$ is given by:

$$\text{UCL} = \frac{p(n+1)(n-1)}{n^2 - np} F_{\alpha,p,n-p} \qquad (3)$$

where $F_{\alpha,p,n-p}$ is the 100·(1-α)% quantile of the Snedecor's distribution with *p and n-p* degrees of freedom.

On the other hand, when the observation vector $\mathbf{x}$ is included in the computation of $\bar{\mathbf{x}}$ and $\mathbf{S}$, the UCL is given by:

$$\text{UCL} = \frac{(n-1)^2}{n} \beta_{\alpha,p/2,(n-p-1)/2}$$

where $\beta_{\alpha,p/2,(n-p-1)/2}$ is the 100·(1-α)% quantile of the Beta distribution with *p/2 and (n-p-1)/2* degrees of freedom. Henceforth, we will consider the distribution given in Equation 3, as it is commonly accepted, with an overall false alarm probability (α) of 0.05.

Detecting out-of-control observations is relatively easy with the graphical techniques of multivariate control since the analysis is similar to the univariate case; determining the causes of that change, however, is more complicated. In this article, we propose the application of a classification technique to determine the variable or variables that have caused the out-of-control situation. In particular, we propose the application of boosting trees as an alternative to the neural networks that have been widely applied to the problem. Section 2 of this paper, therefore, describes various developments in terms of interpreting an out-of-control signal. Section 3 briefly presents the main points of boosting as a classification technique. Section 4 explores the examples proposed in the literature and shows the advantages of the proposed method over various other

http://mc.manuscriptcentral.com/tprs Email: ijpr@lboro.ac.uk
For Peer Review Only

alternatives. Finally, our concluding remarks and future lines of research are outlined in Section 5.

2. Review of published work

The problem of interpreting out-of-control signals in multivariate control charts is partly responsible for holding back the development of these techniques in industry. Multivariate control charts in general and Hotelling's T^2 control chart in particular, do not indicate which variable or variables have caused the out-of-control situation. This interpretation requires complex subsequent work to determine which variables have changed since an out-of-control situation might be due to one or more variables being out of control, or to a change in the relationship among the variables.

In order to interpret out-of-control signals the simplest alternative consists in analyzing univariate control charts for each quality characteristic. However, this approach has certain disadvantages. The first is that when there are many variables, this technique can prove to be tedious because of the large number of univariate control charts to be analyzed. The second, and perhaps the most important drawback, is that an out-of-control signal is not normally caused by only a single feature but rather by the relationships among them. Taking this into account, univariate control charts are not able to show all the out-of-control signals, neither to determine the causes of the multivariate out-of-control situation.

Various authors have tackled this problem by introducing methods which help to interpret out-of-control signals in multivariate control charts. Alt (1985) suggests using individual mean charts with Bonferroni-type control limits and replacing $Z_{\alpha/2}$ in the individual average control chart with $Z_{\alpha/(2p)}$ (where Z_k is the $100{\cdot}k\%$ quantile of the standard Normal distribution); Hayter & Tsui (1994) extended the idea of Bonferroni-type control limits by giving a procedure which used simulations to simultaneously obtain an exact control interval for each variable mean. These intervals are essentially substitutes for the individual average control charts and are usually more effective identifying the variable or variables which cause the out-of-control signal. This procedure can also be used in situations where the normality assumption is not valid.

Another approach for diagnosing an out-of-control signal is to decompose the T^2 statistic into components that reflect the contribution of each individual variable. If T^2

http://mc.manuscriptcentral.com/tprs Email: ijpr@lboro.ac.uk

is the current value of the statistic, and $T^2_{(i)}$ is the value of the statistic for all used variables except the *i*-th one, we can calculate an indicator of the contribution of the *i*-th feature on the set in the following way:

$$d_i = T^2 - T^2_{(i)}$$

Nowadays, the use of T^2 decomposition as proposed by Mason et al. (1995) is considered to be the standard way. The main idea behind this method is to decompose the T^2 statistic into independent parts, each one showing the influence of an individual variable. The problem with this method is that the decomposition of the T^2 statistic into p independent T^2 components is not unique. Therefore, this situation has generated a number of different articles, the most outstanding of which were published by Mason et al. (1996, 1997), Doganaksoy et al. (1991), Timm (1996) and Runger et al. (1996).

Following other approaches, Jackson (1980) and Fuchs & Benjamini (1994) proposed the use of control charts based on the p principal components. These components are linear combinations of the original variables and sometimes they have not a clear interpretation, being this the main drawback of this approach. Murphy (1987) developed procedures based on discriminant analysis, the classical statistical procedure for classifying observations into predetermined groups. Detecting the cause (variable or variables) of an out-of-control signal can be considered as a classification problem where the output is the variable or variables causing that signal and the inputs are the values of the variables and the T^2 statistic.

Intensive research has recently been conducted into the use of artificial neural networks as an effective tool for interpreting out-of-control signals in multivariate control charts. The application of this technique has been developed by: Cheng (1995, 1997), Chang (1996), Zorriassatine (1998), Guh & Hsieh (1999), Guh & Tannock (1999a, 1999b), Ho & Chang (1999), Cook & Chiu (1998), Cook et al. (2001), Guh (2003), Noorosana et.al. (2003), Niaki & Abassi (2005), Aparisi et al. (2006) and Guh (2007).This approach allows the process, which identifies those variables that have changed during the production process, to be automated using a neural network. The procedure is as follows: samples are taken from the process to control it using a multivariate control chart. When the chart indicates an out-of-control case, the neural network is used to recognize the variables which have shifted. It therefore seems advisable to use Hotelling's T^2 control chart not in an isolated way but with some of the

For Peer Review Only

analysis techniques for out-of-control signals described above. This approach allows a clearer interpretation of the results.

In this research paper, we propose the application of boosting trees to determine the variable or variables which have caused the out-of-control signal. The Boosting method constitutes a powerful classification technique alternative to neural networks. In the empirical application, we will see how well they behave when interpreting out-of-control signals. In the following section, we will briefly describe the boosting algorithm which will be used.

3. AdaBoost-SAMME

A classifier system builds a model which is able to predict the class of a new observation given a data set. The accuracy of the classifier will depend on the quality of the method used and the difficulty of the specific application. When the obtained classifier achieves a better accuracy than the default rule it is due to the classification method has found some structure in the data. The AdaBoost method (Freund & Schapire 1996) uses a single classifier as a subroutine making the most of it in terms of accuracy.

AdaBoost applies the classification system repeatedly to the training data, but at each epoch the learning attention is focused on different examples of this set using adaptive weights ($\omega_b(i)$). Once the training process has finished, the single classifiers obtained are combined into a final, highly accurate, one. The final classifier therefore usually achieves a high degree of accuracy in the test set as several authors have shown both theoretically and empirically (Freund et al. 1998, Breiman 1998, Bauer & Kohavi 1999, Dietterich 2000, and Banfield et al. 2004).

Since the AdaBoost method is mainly intended for dichotomous problems, various modifications of this algorithm have been proposed for multi-class problems (Friedman, Hastie & Tibshirani 2000). In this study, we have chosen the Stagewise Additive Modeling using a Multi-class Exponential loss function (SAMME) algorithm (Zhu et al. 2006). There are three reasons for our decision: SAMME is a natural extension of Adaboost for more than two classes, it is very easy to be programmed and it has shown encouraging results in Zhu's paper. SAMME can be summarized as follows:

SAMME Algorithm (Zhu et al. 2006)

---

http://mc.manuscriptcentral.com/tprs Email: ijpr@lboro.ac.uk
For Peer Review Only

1. Start with $\omega_b(i) = 1/n$, i=1, 2, ..., n.

2. Repeat for b=1, 2, ...,B

   a) Fit the classifier $C_b(\mathbf{x})\in\{1,2,\ldots k\}$ using weights $\omega_b(i)$ on TS^b.

   b) Compute: $\varepsilon_b = \sum_{i=1}^{n} \omega_b(i) I(C_b(\mathbf{x}_i) \neq y_i)$ and $\alpha_b = ln((1 - \varepsilon_b)/\varepsilon_b)+ln(K\text{-}1)$

   c) Update the weights $\omega_{b+1}(i) = \omega_b(i) \cdot \exp(\alpha_b I(C_b(\mathbf{x}_i) \neq y_i)$ and normalize them.

3. Output the final classifier $C(\mathbf{x}) = \arg\max_k \sum_{b=1}^{B} \alpha_b I(C_b(\mathbf{x}) = k)$

A training set is given by $TS_n = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..,(\mathbf{x}_n, y_n)\}$ where y takes values of $\{1,2,..,K\}$. The weight $\omega_b(i)$ is assigned to each observation $\mathbf{x}_i$ and is initially set to $1/n$. This value will be updated after each step. A basic classifier denoted as $C_b(\mathbf{x}_i)$ is built on the new training set, TS^b, and is applied to each training sample.

Then, the error of the classifier in the b-th iteration, represented by ε_b, is calculated as in step 2b. From this error, a constant α_b is calculated and will be used to update the weights. It is worth mentioning that the only difference between this and the AdaBoost algorithm is the way in which the alpha constant is calculated, since in AdaBoost it is defined as $\alpha_b = ln((1 - \varepsilon_b)/ \varepsilon_b)$. Due to this modification, in the SAMME algorithm it is only necessary that $1- \varepsilon_b > 1/K$ in order for the alpha constant to be positive and the weight updating follows the right direction. That is to say, the accuracy of each weak classifier should be better than the random guess instead of 1/2, which would be an appropriate requirement for the two class case but very demanding for the multi-class one.

In step 2c the weights for the $b+1$-th iteration are calculated and normalized so that they add up to one. Therefore, the weights of the wrongly classified observations are increased and the weights of the correctly classified ones are decreased, forcing the single classifier built in the following iteration to focus on the hardest examples. Furthermore, the differences when the weights are updated are greater when the error of the single classifier is small since more importance is given to the few mistakes made when the classifier achieves a high level of accuracy.

For Peer Review Only

This process is repeated at every step for b=1, 2, 3, …, B. Finally, the ensemble classifier is built as a linear combination of the single classifiers weighted by the corresponding constant α_b, giving more importance to those classifiers with smaller errors. It is worth noting that the alpha constant can therefore be interpreted as an adaptive learning rate which is calculated every epoch as a function of the error.

4. Experimental results

Our proposal is a combined two-step approach: firstly, we detect the out-of-control signal using Hotelling's well-known T^2 control chart and then we apply the boosting classifier to determine which variable or variables have changed. In the phase I of the control process, the classification method is trained to detect those variables and then, the obtained model is used when the system is working (phase II), detecting the variable or variables involved in an out-of-control situation.

In order to show how our approach works, we considered it relevant to use examples which have previously proved useful for this task. Specifically, we compare our results with those provided by Niaki & Abbasi (2005) and Alfaro et al. (2008) which are based on three examples that cover different levels of difficulty since they work with two, three and four variables, respectively.

In order to apply the SAMME algorithm we have used a slightly modified version of the Adaboost function that belongs to the adabag package (Alfaro et al. 2006) which runs under the R program (http://www.R-project.org). This program is a free statistical package which has been widely developed in recent years and constitutes a strong tool for spreading the results of research worldwide. The Adaboost-SAMME function has two parameters which need to be set; the size of the single trees, and the number of these trees. There are several ways to determine the size of each tree, and here we use the *maxdepth* parameter which limits the distance between the leaf nodes and the root node. For instance, if *maxdepth*=1, only one split is developed and the tree has only two leaf nodes, when *maxdepth*= 2 there are 4 leaf nodes, and so on. Regarding the number of trees, this parameter is fixed as a result of a trade-off between complexity and accuracy. The higher these parameters are, the more complex the classifier is. Hence, we need to increase the value of these parameters as the difficulty of the problem increases, i.e. when there are more classes (example 2) or the classes are more difficult to be separated (example 3).

For Peer Review Only

4.1 *Example 1: the case of two variables*

Let us start with the simple case of two variables, covering the measures of stiffness (X_1) and bending strength (X_2) for a particular grade of lumber. The specifications (which are either known or estimated from a large amount of previous data) for the mean vector and covariance matrix are:

$$\boldsymbol{\mu} = \begin{pmatrix} 265 \\ 470 \end{pmatrix}, \quad \Sigma = \begin{pmatrix} 100 & 66 \\ 66 & 121 \end{pmatrix}$$

The out-of-control observations are detected by applying Hotelling's T^2 control chart with $\alpha = 0.05$. The subsequent question is whether the out-of-control situation was caused by a change in the X_1 variable, the X_2 variable, or both, and this can be set as a classification problem with three classes: change in X_1, change in X_2, and change in both. If we consider a 1.5 standard deviation shift, the mean vectors are, respectively, (265+1.5*10, 470), (265, 470+1.5*11), and (265+1.5*10, 470+1.5*11). We therefore generate 500 repetitions, collecting each value of the T^2 and the value of the variables which mark the previously described situation. These 1500 observations (500 of each class) comprise our training set. The boosting classifier is built with 100 single trees, each one of them with a maximum depth of 2.

In order to test the trained model, we generate sets of observations with shifts of 2, 2.5, and 3 standard deviations by following the same procedure. We apply Hotelling's T^2 control chart to detect the out-of-control signals for each class where the Type I error (α) is 5%. The procedure is repeated until 500 observations of each class have been achieved, and the test set obtained.

Once we have trained the boosting model on the set generated with a 1.5 standard deviation shift, we test its generalization capability when confronted by new cases, and more specifically, we have considered shifts of 2, 2.5 and 3 standard deviations, respectively. The results can be seen in Table 1. In the three cases, the boosting model clearly manages to predict the correct class for most of the examples. It can also be seen that when the size of the shift increases, it is easier for the classifier to detect the true reason for the out-of-control signal and it makes fewer errors. The improvement is more gradual for the class "change in X_1", while in the other two classes there are minor differences between shifts of 2.5 and 3 standard deviations, although in both cases the number of errors is significantly lower than for a 2-deviations shift.

http://mc.manuscriptcentral.com/tprs Email: ijpr@lboro.ac.uk
For Peer Review Only

4.2 *Example 2: the case of four variables*

This example was firstly used at the beginning of the nineties by Doganaksoy et al. (1991). It works with four variables connected with ballistic missile testing. The mean vector and the covariance matrix are:

$$\boldsymbol{\mu} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad \Sigma = \begin{pmatrix} 102.74 & 88.34 & 67.03 & 54.06 \\ & 142.74 & 86.55 & 80.02 \\ & & 64.57 & 69.42 \\ & & & 99.06 \end{pmatrix}$$

In this case, we detect the out-of-control signal as in the previous example although because it uses four variables, the complexity of interpreting the out-of-control signals increases considerably since there are 2^4-1=15 possible classes. In this example, the size of the shift in the training set is 2σ (7500 observations) and 2σ, 2.5σ, and 3σ for the test sets. The boosting classifier is built with 300 single trees with a maximum depth of 3.

From the point of view of classification problems, this is the most complex case owing to the high number of classes involved (15). Nevertheless, the boosting classifier also obtains the most correct answers not only when the shift occurs in a single variable but also when several of them change simultaneously. In general terms, as we expected, the larger the shift in the mean vector, the lower the error. Tables 2, 3, and 4 show the results of boosting in this example.

4.3 *Example 3: the case of three variables*

This example focuses on a company which produces washing powder where three variables are controlled: colour, free oil percentage, and acidity percentage. The corresponding classification problem has 2^3-1= 7 classes. The estimated mean vector and covariance matrix are:

$$\bar{\mathbf{x}} = \begin{pmatrix} 67.5 \\ 12.0 \\ 97.0 \end{pmatrix}, \quad \mathbf{S} = \begin{pmatrix} 0.68 & 0.36 & -0.07 \\ & 1.00 & -0.12 \\ & & 0.03 \end{pmatrix}$$

In this case the shift size is 3σ for the training set (700 observations) and 2σ, 3σ, and 4σ for the test set. The boosting classifier is built with 300 single trees with a maximum depth of 5.

For Peer Review Only

It is worth mentioning that we use estimations of the true population values of the parameters which are unknown since this is a real case. The closeness of this case to reality does, in fact, make it even more interesting. As in the previous examples, the boosting approach is able to find the true nature of the change that occurred in most cases for the three shift sizes (2, 3 and 4 standard deviations). The worst result is for the lower 2 standard deviation shift where the classifier finds it difficult to distinguish the classes when the shift affects more than one variable; the results when the three variables change simultaneously are particularly bad. On the other hand, accuracy is much better in the other two test sets and very good when the shift is 3σ. Table 5 shows these results.

4.4 *Comparison of boosting results with previous studies*

The aim of our research is to present the boosting method as a useful and extremely powerful tool for interpreting out-of-control signals in multivariate process quality control. Nevertheless, since the same examples used for data generation were previously employed, we do believe that it is worth comparing the results of our proposal with those of previous works. We will therefore analyze the differences between our paper and that of Niaki & Abbasi (2005), who applied neural networks (MLP, Multilayer Perceptron) and multivariate Shewart (MSCH), and Alfaro et al. (2008), who used a classification tree (CART, Breiman et al. 1984) for the same task. The comparison is made using only the test set errors for conciseness purposes. It is worth noting that the data have been generated from the same mean vectors and covariance matrices, but they are not the same.

Table 6 presents the error percentages for the boosting method proposed here (SAMME) and the CART, MLP and MSCH methods mentioned above. The results show how SAMME performs better than any of the methods in each test set, with the exception of MSCH in the case of 2 variables for a 2 standard deviation shift. SAMME outperforms both the MLP and the CART tree in all scenarios. It is also worth highlighting the successful results for 2.5 and 3σ with two variables, for 3σ with three variables where the error is below 5% and for 3σ with four variables with an error of 5.73%. These results are extremely promising and encourage us to continue working on the application of this type of classifier.

For Peer Review Only

5. Conclusions

The main contribution of this research paper is to propose the use of boosting trees using the SAMME algorithm to interpret the out-of-control signals that occur in multivariate process quality control. These ensemble trees have proved to be a very powerful tool when classifier accuracy is a key factor. It is worth mentioning that our proposal is a combined two-step approach: we first detect the out-of-control signal using Hotelling's well-known T^2 control chart, and we then apply the boosting classifier to determine which variable or variables have changed.

We have developed an empirical application which confirms the usefulness of this boosting method for this particular task with very encouraging results in all the examples and under all scenarios envisaged. Moreover, the separate use of training and test sets guarantees the generalization capability of these classifiers. This means that the classifier has not learned only the characteristics of a particular set, but it has been able to understand the intrinsic nature of the problem, and will therefore be able to correctly classify new observations.

Furthermore, comparison with the results of previous studies proves to be entirely satisfactory since our proposed method achieves the best results, outperforming the classification tree (CART), the multilayer perceptron (MLP), and the multivariate Shewart chart (MSCH). There are many aspects that we have not covered in this paper but which we would like to explore in future research, such as the use of the boosting method not just as a tool for interpreting out-of-control signals but also for detecting whether the process is in control or not, or even for determining the size of the shift when it occurs. It would also be interesting to apply some pruning methods to reduce the number of trees in boosting techniques. The goal being to reduce significantly the complexity without any important loss of accuracy, which obviously would be advantageous

For Peer Review Only

6. References

Alfaro, E., Alfaro, J.L, Gámez, M. & García, N., 2008. Classification trees to interpret out-of-control signals in multivariate control charts. In: XVI International Symposium on Mathematical Methods Applied to the Sciences, 19-22 February 2008, San José, Costa Rica, 29.

Alfaro, E., Gámez, M. & García, N., 2006. *Adabag: implements AdaBoost.M1 and bagging* [online]. R package version 1.0 (2006). Available from: http://www.R-project.org [Accessed 11 March 2008].

Alt, F.B., 1985. Multivariate quality control. In: N.L. Johnson and S. Kotz. *Encyclopedia of Statistical Sciences 6*: John Wiley and Sons.

Aparisi, F., Avendaño, G. & Sanz, J., 2006. Techniques to interpret T^2 control chart signals. *IIE Transactions*, 38, 647–657.

Banfield, R.E., Hall, L.O., Bowyer, K.W., Bhadoria, D., Kegelmeyer, W.P. & Eschrich S., 2004. A Comparison of Ensemble Creation Techniques. In: F. Roli, J. Kittler and T. Windeatt, ed. *Multiple Classifier Systems*, Lecture Notes in Computer Science, 3077. Cagliari, Italy: Springer, 223-232.

Bauer, E. & Kohavi, R., 1999. An empirical comparison of voting classification algorithm: Bagging, boosting and variants. *Machine Learning*, 36, 105-142.

Breiman, L., 1998. Arcing classifiers. *The Annals of Statistics*, 26(3), 801-849.

Breiman, L., Friedman, J.H., Olshen, R. & Stone, C.J., 1984. *Classification and regression trees*. Belmont, Wadsworth International Group.

Chang S.I. & Aw C.A., 1996. A neural fuzzy control chart for detecting and classifying process mean shifts. *International Journal of Production Research*, 34(8), 2265–2278.

Cheng C.S., 1995. A multi-layer neural network model for detecting changes in the process mean. *Computers and Industrial Engineering*, 28(1), 51–61.

Cheng C.S., 1997. A neural network approach for the analysis of control chart patterns. *International Journal of Production Research*, 35(3), 667–697.

Cook D.F. & Chiu C.C., 1998. Using radial basis function neural networks to recognize shifts in correlated manufacturing process parameters. *IIE Transactions*, 30(3), 227–234.

For Peer Review Only

Cook, D.F., Zobel, C.W. & Nottingham, Q.J., 2001. Utilization of neural networks for the recognition of variance shifts in correlated manufacturing process parameters. *International Journal of production Research*, 39(17), 3881–3887.

Dietterich, T.G., 2000. Ensemble methods in machine learning. In J. Kittler and F. Roli, ed. *Multiple Classifier Systems*, Lecture Notes in Computer Science, 1857. Cagliari, Italy: Springer, 1-15

Doganaksoy, N., Faltin, F.W. & Tucker, W.T., 1991. Identification of out of control quality characteristics in a multivariate manufacturing environment, *Communications in Statistics - Theory and Methods* , 20, 2775-2790.

Freund, Y., Schapire, R.E., Bartlett P. & Lee W.S., 1998. Boosting the margin: A new explanation for the effectiveness of voting methods. *The Annals of Statistics*, 26(5), 1651-1686

Friedman, J., Hastie, T. & Tibshirani, R., 2000. Additive logistic regression: a statistical view of boosting. *The Annals of Statistics*, 38(2), 291-293.

Fuchs, C. & Benjamini, Y., 1994. Multivariate profile charts for statistical process control. *Technometrics*, 36(2),182–195.

Guh R.S., 2003. Integrating Artificial Intelligence into On-line Statistical Process Control. *Quality and Reliability Engineering International*, 19, 1-20.

Guh R.S., 2007. On-line Identification and Quantification of Mean Shifts in Bivariate Processes using a Neural Network-based Approach. *Quality and Reliability Engineering International*, 23, 367-385.

Guh R.S. & Hsieh Y.C., 1999. A neural network based model for abnormal pattern recognition of control charts. *Computers and Industrial Engineering*, 36, 97-108

Guh R.S., Tannock J.D.T., 1999a. A neural network approach to characterize pattern parameters in process control charts. *Journal of Intelligent Manufacturing*, 10(5), 449–462.

Guh R.S., Tannock J.D.T., 1999b. Recognition of control chart concurrent patterns using a neural network approach. *International Journal of Production Research*, 37(8), 1743–1765.

Hayter, A.J. & Tsui, K.L., 1994. Identification and Quantification In Multivariate Quality Control Problems. *Journal of Quality Technology*, 26, 197-208.

For Peer Review Only

Ho E.S. & Chang S.I., 1999. An integrated neural network approach for simultaneous monitoring of process mean and variance shifts—a comparative study. *International Journal of Production Research*, 37(8), 1881–1901.

Jackson, J.E., 1980: Principal components and factor analysis: Part I - Principal Components. *Journal of Quality Technology*, 12, 201-213.

Mason, R.L. & Young, J.C., 2002. *Multivariate Statistical Process Control with Industrial Applications*. Philadelphia: American Statistical Association and the Society for Industrial and Applied Mathematics (ASA-SIAM).

Mason, R.L., Tracy, N.D. & Young, J.C., 1995. Decomposition of T^2 for multivariate control chart interpretation. *Journal of Quality Technology*, 27, 109-119.

Mason, R.L., Tracy, N.D. & Young, J.C., 1996. Monitoring a multivariate step process. *Journal of Quality Technology*, 28, 39-50.

Mason, R.L., Tracy, N.D. & Young, J.C., 1997. A practical approach for interpreting multivariate T^2 control chart signals. *Journal of Quality Technology*, 29, 396-406.

Murphy, B.J., 1987. Selecting out of control variables with the T^2 multivariate quality control procedure. *Journal of the Royal Statistical Society Serie D (The Statistician)*, 36, 571-583.

Niaki, S.T.A. & Abassi, B., 2005. Fault Diagnosis in Multivariate Control Charts Using Artificial Neural Networks. *Quality and Reliability Engineering International*, 21, 825-840.

Noorossana, R., Farrokhi, M. & Saghaei, A., 2003. Using neural networks to detect and classify out-of-control signals in autocorrelated processes. *Quality and Reliability Engineering International*, 19, 1–12.

Opitz, D. & Maclin, R., 1999. Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research*, 11, 169-198

R Development Core Team, 2004. *R: A language and environment for statistical computing* [online]. Viena: R Foundation for Statistical Computing. Available from: http://www.R-project.org [Accessed 11 March 2008].

Runger, G.C., Alt, F.B. & Montgomery, D.C., 1996. Contributors to a multivariate SPPC chart signal. *Communications in Statistics-Theory and Methods*, 25, 2203-2213.

Timm, N.H., 1996. Multivariate quality control using finite intersection tests. *Journal of Quality Technology*, 28, 233-243.

For Peer Review Only

Tracy, N.D., Young, J.C. & Mason, R.L., 1992. Multivariate Control Charts for Individual Observations. *Journal of Quality Technology*, 24, 88-95.

Zhu, J., Rosset, S., Zou, H. & Hastie, T., 2006. *Multi-class AdaBoost* [online]. Working paper. Available from: http://www-stat.stanford.edu/~hastie/Papers/samme.pdf [Accessed 11 March 2008].

Zorriassatine F. & Tannock J.D.T., 1998. A review of neural networks for statistical process control. *Journal of Intelligent Manufacturing*, 9(3), 209–224.

For Peer Review Only

Tables

Table 1. Confusion matrices for different shifts in the two variables case

		Predicted Class								
		2σ Shift			2.5σ Shift			3σ Shift		
Observed Class	Variables	X_1	X_2	(X_1, X_2)	X_1	X_2	(X_1, X_2)	X_1	X_2	(X_1, X_2)
	X_1	455	1	44	473	0	27	482	0	18
	X_2	0	477	23	0	491	9	0	490	10
	(X_1, X_2)	12	28	460	2	12	486	4	11	485

For Peer Review Only

Table 2. Results obtained for 2 standard deviation shift in the mean vector of example 2

		Predicted Class														
Observed Class		X_1	X_2	X_3	X_4	(X_1, X_2)	(X_1, X_3)	(X_1, X_4)	(X_2, X_3)	(X_2, X_4)	(X_3, X_4)	(X_1, X_2, X_3)	(X_1, X_2, X_4)	(X_1, X_3, X_4)	(X_2, X_3, X_4)	(X_1, X_2, X_3, X_4)
	X_1	454	0	0	0	4	6	20	0	0	0	1	2	2	0	11
	X_2	0	440	0	0	22	0	0	6	24	0	0	0	0	8	0
	X_3	0	0	440	0	0	25	0	11	0	18	6	0	0	0	0
	X_4	0	0	0	451	0	0	31	0	4	6	0	3	1	1	3
	(X_1, X_2)	5	16	0	0	453	0	0	0	1	0	3	11	0	0	11
	(X_1, X_3)	13	0	21	0	0	432	0	0	0	1	9	0	24	0	0
	(X_1, X_4)	31	0	0	28	0	0	424	0	0	0	0	12	5	0	0
	(X_2, X_3)	0	5	16	0	0	0	0	432	0	0	22	0	0	23	0
	(X_2, X_4)	0	11	0	8	1	0	0	0	458	0	0	15	0	2	5
	(X_3, X_4)	0	0	17	3	0	0	0	0	0	456	0	0	14	7	3
	(X_1, X_2, X_3)	1	4	4	0	7	9	0	22	0	0	444	0	0	0	9
	(X_1, X_2, X_4)	3	0	0	3	25	0	16	0	27	0	0	419	0	0	7
	(X_1, X_3, X_4)	14	0	0	2	0	14	6	0	0	19	0	0	433	0	12
	(X_2, X_3, X_4)	0	6	2	0	0	0	0	22	12	7	0	0	0	438	13
	(X_1, X_2, X_3, X_4)	2	0	0	3	11	1	4	0	2	2	6	4	8	20	437

For Peer Review Only

Table 3. Results obtained for 2.5 standard deviation shift in the mean vector of example 2

		Predicted Class														
Observed Class		X_1	X_2	X_3	X_4	(X_1, X_2)	(X_1, X_3)	(X_1, X_4)	(X_2, X_3)	(X_2, X_4)	(X_3, X_4)	(X_1, X_2, X_3)	(X_1, X_2, X_4)	(X_1, X_3, X_4)	(X_2, X_3, X_4)	(X_1, X_2, X_3, X_4)
	X_1	442	0	0	0	8	2	19	0	0	0	1	4	11	0	13
	X_2	0	444	0	0	21	0	0	4	17	0	1	0	0	9	4
	X_3	0	0	450	0	0	30	0	6	0	8	6	0	0	0	0
	X_4	0	0	0	440	0	0	31	0	6	8	0	7	2	1	5
	(X_1, X_2)	3	13	0	0	453	0	0	0	0	0	1	17	0	0	13
	(X_1, X_3)	5	0	15	0	0	461	0	0	0	0	6	0	9	0	4
	(X_1, X_4)	12	0	0	9	0	0	470	0	0	0	0	3	4	0	2
	(X_2, X_3)	0	3	9	0	0	0	0	467	0	0	11	0	0	10	0
	(X_2, X_4)	0	7	0	3	1	0	0	0	457	0	0	24	0	1	7
	(X_3, X_4)	0	0	10	0	0	1	0	0	0	466	0	0	12	9	2
	(X_1, X_2, X_3)	2	1	3	0	4	6	0	14	0	0	454	0	0	0	16
	(X_1, X_2, X_4)	0	0	0	2	14	0	7	0	10	0	0	466	0	0	1
	(X_1, X_3, X_4)	5	0	1	3	0	16	4	0	0	14	0	0	450	0	7
	(X_2, X_3, X_4)	0	3	1	1	0	0	0	18	4	14	0	0	0	447	12
	(X_1, X_2, X_3, X_4)	2	1	2	6	6	2	1	0	2	5	12	6	5	21	429

For Peer Review Only

Table 4. Results obtained for 3 standard deviation shift in the mean vector of example 2

		Predicted Class														
Observed Class		X_1	X_2	X_3	X_4	(X_1, X_2)	(X_1, X_3)	(X_1, X_4)	(X_2, X_3)	(X_2, X_4)	(X_3, X_4)	(X_1, X_2, X_3)	(X_1, X_2, X_4)	(X_1, X_3, X_4)	(X_2, X_3, X_4)	(X_1, X_2, X_3, X_4)
	X_1	454	0	0	0	2	5	17	0	0	0	3	2	6	0	11
	X_2	0	462	0	0	9	0	0	4	17	0	0	0	0	6	2
	X_3	0	0	468	0	0	20	0	3	0	6	1	0	1	1	0
	X_4	0	0	0	447	0	0	37	0	5	2	0	5	0	0	4
	(X_1, X_2)	1	6	0	0	469	0	0	0	0	0	1	22	0	0	1
	(X_1, X_3)	3	0	3	0	0	485	0	0	0	0	1	0	5	0	3
	(X_1, X_4)	1	0	0	1	0	0	492	0	0	0	0	2	3	0	1
	(X_2, X_3)	0	0	3	0	0	0	0	485	0	0	5	0	0	7	0
	(X_2, X_4)	0	4	0	0	0	0	0	0	479	0	0	15	0	1	1
	(X_3, X_4)	0	0	5	0	0	0	0	0	0	498	0	0	7	0	0
	(X_1, X_2, X_3)	0	0	3	0	1	3	0	11	0	0	479	0	0	0	3
	(X_1, X_2, X_4)	0	1	0	1	8	0	6	0	8	0	0	472	0	0	4
	(X_1, X_3, X_4)	5	0	0	0	0	9	2	0	0	7	0	0	473	0	4
	(X_2, X_3, X_4)	0	2	3	0	0	0	0	13	5	11	0	0	0	463	3
	(X_1, X_2, X_3, X_4)	1	0	0	5	2	0	2	0	1	3	8	4	6	14	454

For Peer Review Only

Table 5. Results obtained for example 3

	Predicted Class (**2 standard deviation shift in the mean vector**)							
Observed Class		X_1	X_2	X_3	(X_1, X_2)	(X_1, X_3)	(X_2, X_3)	(X_1, X_2, X_3)
	X_1	89	1	6	4	0	0	0
	X_2	0	72	22	6	0	0	0
	X_3	3	5	81	10	1	0	0
	(X_1, X_2)	6	15	17	60	2	0	0
	(X_1, X_3)	14	0	29	15	42	0	0
	(X_2, X_3)	0	30	40	5	0	25	0
	(X_1, X_2, X_3)	1	0	30	42	14	4	9
	Predicted Class (**3 standard deviation shift in the mean vector**)							
Observed Class		X_1	X_2	X_3	(X_1, X_2)	(X_1, X_3)	(X_2, X_3)	(X_1, X_2, X_3)
	X_1	99	0	0	0	1	0	0
	X_2	0	98	0	2	0	0	0
	X_3	0	1	97	0	2	0	0
	(X_1, X_2)	1	1	1	96	1	0	0
	(X_1, X_3)	0	0	1	0	98	0	1
	(X_2, X_3)	0	0	0	0	0	99	1
	(X_1, X_2, X_3)	0	0	0	0	0	7	93
	Predicted Class (**4 standard deviation shift in the mean vector**)							
Observed Class		X_1	X_2	X_3	(X_1, X_2)	(X_1, X_3)	(X_2, X_3)	(X_1, X_2, X_3)
	X_1	89	0	0	2	9	0	0
	X_2	0	73	0	1	0	23	3
	X_3	0	0	82	1	10	4	3
	(X_1, X_2)	0	1	0	64	0	2	33
	(X_1, X_3)	0	0	0	0	94	0	6
	(X_2, X_3)	0	0	0	0	0	97	3
	(X_1, X_2, X_3)	0	0	0	0	0	0	100

For Peer Review Only

Table 6. Error percentage on the test sets

	2 variables			3 variables			4 variables		
Methods	2 σ	2.5 σ	3 σ	2 σ	3 σ	4 σ	2 σ	2.5 σ	3 σ
SAMME	7.2	3.33	2.87	46.00	2.86	14.43	11.85	9.39	5.73
CART (Alfaro et al. 2008)	9.07	6.53	5.47	49.86	12.71	18.71	15.28	14.08	11.19
MLP * (Niaki & Abbasi 2005)	13.93	10.73	8.73	50.00	62.14	45.29	33.03	24.87	18.69
MSCH (Niaki & Abbasi 2005)	4.60	10.93	10.93	n.a.	n.a.	n.a.	71.15	66.68	68.53

* The topologies (#units input-hidden-output) of the MLP models are: 2-9-2, 3-13-3, and 4-31-4 for the cases of 2, 3 and 4 variables respectively.

For Peer Review Only