Copyright © 2002 IFAC
15th Triennial World Congress, Barcelona, Spain

FUZZY CLUSTERING AND CLASSIFICATION FOR AUTOMATED LEAK DETECTION SYSTEMS

Nathalie Taillefond and Olaf Wolkenhauer

Control Systems Centre, UMIST,
PO BOX 88, Manchester M60 1QD, UK
e-mail:n.taillefond@student.umist.ac.uk
e-mail:o.wolkenhauer@umist.ac.uk

Abstract: A methodology for pipeline integrity monitoring systems using a mixture of clustering and classification tools for fault detection is presented here. The approach is used to classify more readily faults or changes in the context of on-line leak detection with initially off-line training. The methodology is applied to a small-scale pipeline monitoring case where portability, robustness and reliability are amongst the most important criteria. The results are encouraging as relatively low levels of false alarms and increased fault detection are obtained. *Copyright © 2002 IFAC*

Keywords: Classifiers, Fault detection, Fault diagnosis, Methodology, Pattern recognition

1. INTRODUCTION

This paper introduces a methodology enabling fault detection for on-line applications using clustering and classification techniques. The concern is supervision with fault-diagnosis as described by Patton *et al*, 1989 and Isermann, 1997. Supervision is based on measured variables for which features are calculated, symptoms are generated via change detection, a fault diagnosis is performed and decisions for counteractions are made.

This system is essentially a man-machine interface; the operator takes corrective actions as advised by his or her own internal procedures. The data provided are time series of a physical system collected at regular intervals. The various data sets used are those of non-linear, time-varying processes. The aim of the resulting application is to automate the fault detection process using more intuitive and non-statistical techniques to resolve labour intensive issues linked to the development of any bespoke data processing system. The aim is the design of a generic classifier for the system through clustering.

The outline of the rest of the paper is as follows: section 2 highlights the basis of the methodology requirements within the application are outlined, section 3 gives details of how the methodology is built and section 4 shows an industrial application on a small scale water network.

2. DESIGN CRITERIA

The user should notice no difference whilst running the added software. The requirements for designing this module are solely stemming from the developer who requires a system that will reduce development time and eventually remove the need for pre-processing. This also implies that the inherent qualities of statistical systems - amongst which robustness - will need to be reproduced within the new module. Given that the system will work on less than predictable applications, where multi phase flow, changes in batches, changes in setpoints or in external conditions affecting the readings are commonplace, this will lead to a strict list of requirements.

2.1 Requirements

The design of a classifier involves two stages: the construction of the membership function and the definition of decision rules (Denœux, 1997). However as this application is developed in view of integration, it needs to be flexible and straightforward. The following criteria were selected as most desirable:

Portability: As the application is to be used for all types of fluids in all types of environments with varying properties and operating conditions, it is best when fewer arbitrary boundaries are used. Hence no thresholds other than irrefutable ones - such as positive and negative - for known behaviour can be used.

Ease of use aims at addressing the need for minimum developer interaction and inputs in bespoke applications. It is to be developed for different applications with the same processing basis; it cannot have boundaries or thresholds besides greater or less than zero in many instances.

Reliability: Similar measurements and conditions made repeatedly under identical circumstances are expected to yield concordant results. This particular property is required for the diagnostic and in a way linked to robustness described below.

Robustness is the essential design criterion translating the need to withstand all types of changes in the data and minimise false alarm levels. Techniques that have the ability to tolerate noise and outliers in the data are in high demand in industrial applications with variable inputs. As quoted in Davé (1997) a robust procedure can be characterised by the following:

- Have a reasonably good efficiency (accuracy) for the assumed model
- Small deviations from the model assumptions should impair the performance only by a small amount
- Larger deviations from the model assumptions should not cause a catastrophe.

For engineering applications, only the second requirement needs to be looked at. Besides noise and outliers, the notion of getting the right number of clusters, indicating the right partition for the space, is closely linked to robustness. Some techniques currently available to solve this issue have their share of limitations and the overall solution remains elusive (Davé, 1997).

The classical approach based on variations of k-means is not robust. The alternative formulations based on noise clustering or possibilistic clustering are robust in that they are based on robust statistics (Davé, 1997). In this application, the Gustafson-Kessel, a member of the fuzzy k-means family, is used. A major drawback of this algorithm family is the partition initialisation (Babuška, 1998) leading to reduced robustness of systems using them without adjustment.
As mentioned earlier, robustness is one of the essential criteria of the methodology. Robustness is generally defined as a statistical characteristic. Davé (1997) argues that to fit robustness into engineering applications is only realistic where cases considered are those that may really occur within real life experiments. Extreme cases may not be encountered and within the clustering as a starting point of the methodology accounting for them may be totally irrelevant.

Intuitive diagnostic ability: The diagnostic resulting from the application of the methodology must be very analogous to any expert diagnostic whilst being obtained by fuzzy clustering.

2.2 Tools

Besides the use of fuzzy techniques, fault detection can be obtained by gathering more information through mathematical models (Isermann, 1997). One of the drawbacks of such techniques is the prerequisite knowledge of the physical system. As several models might be necessary, the use of such techniques may be best avoided, as they will meet earlier criterion of portability.

The tools available are essentially the k-means and fuzzy k-means family of clustering algorithms. The application demands unsupervised clustering considering that the data is unknown until it arrives.

Training data can be obtained with the high likelihood that no large faults will be available, reducing the field of the classifier training. Failing this the data will have to be simulated or adapted from pipelines similar in configuration. This is for safety reasons.

The properties of the algorithms exploited here are the search for spherical clusters by FKM and the sensitivity of density and shape of GK. Other features also prove interesting. In their initial form, both FKM and GK look for clusters of equal volume.

For FKM, this will mean that for each neighbouring cluster the eigenvalues of its covariance matrix will be expected to be similar. For the GK algorithm, this may not be so, as the density - through the covariance - is taken into account and may alter the eigenvalues of each cluster covariance matrix. To illustrate the differences between the algorithms details of the FKM algorithm are given below and differences with the GK algorithm are highlighted.

FKM and GK algorithms

Let Z be a data matrix size r x s such that

$$Z = \begin{array}{c} \xrightarrow{\text{Variables}} \\ \begin{bmatrix} z_{11} & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & z_{rs} \end{bmatrix} \end{array} \Bigg\downarrow \text{Observations}$$

Given the data set *Z*, the number of cluster 1<*k*<*r*, the fuzzy weight *m* determines the overlap of clusters, the stopping criterion *ε* and the norm-inducing matrix *A* are used in the following algorithm:

Repeat for l =1,2…

Step 1: Compute the cluster means:

$$v_i^{(l)} = \frac{\sum_{t=1}^{r} (\mu_{it}^{(l-1)})^m z_t}{\sum_{t=1}^{r} (\mu_{it}^{(l-1)})^m}, 1 \le i \le k$$

where μ_{it} is the cluster membership value for t^{th} the observation in cluster *i* such that $\mu_{it}^{(l)} \in [0,1]$

Step 2: Compute the distances:

$$D^2_{itA_i} = (z_t - v_i^{(l)})^T A_i (z_t - v_i^{(l)}), 1 \le i \le k,$$

$1 \le t \le r$

where A_i is an *s*x*s* norm inducing matrix for cluster *i*.

Step 3: Update the partition matrix:

If $D_{itA_i} > 0$ for $1 \le i \le k, 1 \le t \le r$

$$\mu_{it}^{(l)} = \frac{1}{\sum_{j=1}^{k} \left(D_{itAi} / D_{jtAi}\right)^{2/(m-1)}},$$

otherwise

$\mu_{it}^{(l)} = 0$ if D_{itA_i} >0, and $\mu_{it}^{(l)} \in [0,1]$ with

$$\sum_{i=1}^{k} \mu_{it}^{(l)} = 1$$

Until $\left\| U^{(l-1)} - U^{(l)} \right\| < \varepsilon$

The main difference between GK and FKM is the norm-inducing matrix, A_i (Step 2) as it impacts on the evaluation of the membership matrix. For FKM it would be the identity matrix I, whilst for GK it would be determined as

$$A_i = \det(F_i)^{1/s} F_i^{-1}$$

where F_i is the fuzzy covariance matrix defined as

$$F_i = \frac{\sum_{t=1}^{r} (\mu_{it}^{(l-1)})^m (z_t - v_i^{(l)})(z_t - v_i^{(l)})^T}{\sum_{t=1}^{r} (\mu_{it}^{(l-1)})^m}, 1 \le i \le k$$

The weighting factor *m* is set equal to two in the simulation below. For both algorithms, the closeness of cluster centres (or prototypes) may lead to membership functions being very similar. When merging, overestimating the cluster number in the first round of clustering increases this possibility.
There is no assumption that the data is normalised, scaled or noise-free.

For the likelihood of high scale differences in the data and the need to retain original, albeit filtered, data and it sensitivity to variable data density GK appeared the best likely candidate for the application.

3. METHODOLOGY

The application considered is an adjunction to real-time software used to monitor pipelines. Its aim is to warn operators when malfunction occurs due to fluid loss. Rather than develop and estimate levels of variable correlation for each project and adequately alter the software, what is proposed here is a generic classifier with minimal developer interaction.

Aiming to be generic implies that a series of restrictions will apply. The first to come to mind is that for the model to be effective, no arbitrary formatting is possible to normalise the data. Others involve ease of application from one system to another, reliability and robustness.

A decision model can be decomposed in two steps: training and application. The training is classically off-line, although on-line training can also be added and considered, whilst the application is a real-time process.

Dubuisson (1996) rightly hints at the most essential criteria of an automatic diagnosis scheme:

- Running mode recognition
- Possibility to give several running modes
- Non-identification of the running mode when unlike any other known one
- Help to the operator.

These fundamental points are considered in the creation of the methodology for this application.

Training (off-line):

- Get training variables in raw form or filtered
- Define data space
- Cluster/partition data space though GK algorithm obtained in (Babuška, 1998). A combination of agglomerative competitive clustering (Frigui and Krishnapuram, 1997,1999) is applied off-line
- Identify conditions for each cluster.

Classification:

- Get on-line variables and use a distance-based algorithm to estimate membership values in training clusters.

Diagnostic:

- Use the max-rule to determine the membership of the output for the on-line estimation
- Apply exclusion rules (as a reject option) as shown in Table 1 to correct the output membership for increased reliability (Boudouad, 1998 and Ménard, 2000). At this stage an automatic reject option and redefinition is not available but is being considered for further work.

To increase accuracy in the model, training and classification are done for two separate sets of temporally related data requiring multiple classifier fusion to reach a decision. On another note, data can be created to accommodate most cases based on one example for the system in question. Data can be made to be representative and efficient. An option to update the classifier on-line is also available. The advantage of such classifier design is that it will detect new conditions as they appear.

Unfortunately, leaks of very different sizes will appear as different conditions and labelling will still have to take place off-line and within this application, labelling is required. The disadvantage of such technique over the off-line training is that a period of stability is required before new characterisation, thus slowing down the first detection of a new condition.

4. EXAMPLE

The following example is one where different types of faults occur. Figure 1 illustrates one of the two sets of variables used. The conditions present in this sample are: steady state, leak in steady state and transient. Although called steady state, the data is really effectively within acceptable change boundaries.

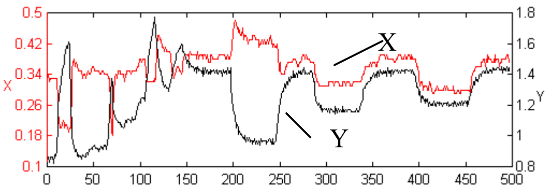

Figure1 - Test time series: time series are filtered but not totally de-noised. The lower trace axes are on the left and that of the upper trace are on the right. The data space is defined as the plotting of variables against each other. A variation of Gustafson-Kessel (GK) is used to define the partition.

In this experiment the data is used raw, as little noise is present in the system. Normally however, the data would be processed for outliers and other faulty -in quality- data. The difference in scale is no issue in the clustering, however the closeness of the clusters initially led to unreliable partition. This was solved through the implementation of a merging technique complementary to GK algorithm.

When put through the classifier, the results are very encouraging as shown in the top graph in Figure 2 compared to the expected output (bottom). Besides a simple rule engine based on Table 1 and the identification of the clusters for ease of use by the user, no other interaction has taken place from the selection of the data to the results. The rules are needed to create an overall classifier output. The identification is essential to the user.

Table 1- Excerpt of rule table for Diagnostic

Rule #	Class 1	Class2	Diagnostic
1	TR	SS	TR
2	TR	LSS	TR
3	SS	LTR	TR
4	SS	TR	SS

It can be noted in Figure 2 that a transient is detected as a leak in the top half. There may be a need to verify the rules or the data to ascertain where the fusion went wrong (see Figure 3).

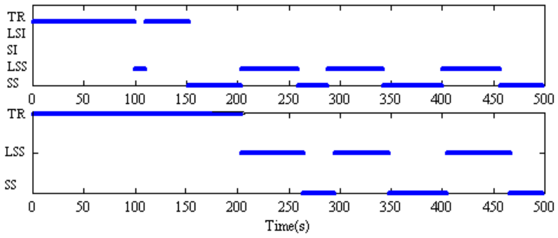

Figure 2 - Diagnostic results. Top: Classifier output, bottom: expected output. Legend: TR -transient, SI-shut-in, LSI-Leak in shut-in, SS: steady state, LSS: leak in steady state

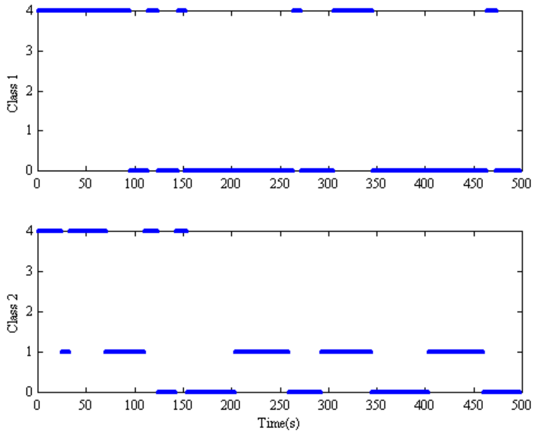

Figure 3 - Classification results.
This shows the output for each classifier. The top classifier allows determination of the regime (0:steady state, 4: transient). The lower classifier defines the type of fault. The numbers have the same meaning as the acronyms in the top graph in Figure 2.

Using this system enables a double check of the system regime. The idea is that only certain conditions, within the current realm of knowledge, occur under certain circumstances. This was achieved due to the extreme likeness of transient conditions to steady state ones, which can be noted in Figure 1.

5. CONCLUSION

The application is dealing with a series of faults and operational changes which need to be identified, although not all of them will impair the operation of the system, in particular the slow changes. Abrupt changes would either be leaks or large transient operational conditions. The faults presented here are of abrupt nature. Other applications of the methodology with incipient faults have also been run successfully.

The methodology proved successful in this context on several points. Firstly in defining the data space in a very intuitive expert-like fashion. Secondly its ease of application meant that very little information was used and in terms of inputs a minimum level of information was necessary. And finally, it provided fault diagnostics with low level of false alarms and few misdiagnoses under different conditions.

Option for on-line update clusters is being currently considered to help make the training set more adaptive. Most of the training will take place off-line and in the eventuality of a new completely unforeseen condition developing, an on-line classifier could be applied. This would help reduce model dependency and increase robustness.

6. ACKNOWLEDGEMENTS

Parts of this research are funded by the UK Engineering and Physical Sciences Research Council (EPSRC).

REFERENCES

Babuška, R. (1998), *Fuzzy Modelling for Control*, Kluwer Academic Publishers, The Netherlands

Boudaoud, A.N and Masson, M.H (1998), On-line adaptive fuzzy diagnosis system: fusion and supervision*, 13th IFAC meeting*, Hull, UK

Davé, R. and Krishnapuram, R. (1997), Robust clustering methods: a unified view, *IEEE trans. on Fuzzy Systems*, **5** (2)

Denoeux, T., Masson, M. and Dubuisson, B (1997), Advanced Pattern Recognition techniques for system monitoring and diagnosis: a survey, *RAIRO-AP II-JESA*, **31**(9-10), 1509-1539

Dubuisson, B., Masson, M. and Frelicot, C. (1996), Some topics in using Pattern Recognition for system diagnosis, *Engineering Simulation*, **13**, 863-888

Frigui, H. and Krishnapuram, R. (1997), Clustering by competitive agglomeration, *Pattern Recognition*, **30**(7), 1109-1119

Frigui, H. and Krishnapuram R. (1999), A robust competitive clustering algorithm with applications in computer vision, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **21**(5), 450-465

Isermann, R. (1997). Supervision, fault-detection and fault-diagnosis methods- an introduction*, Control Engineering Practice*, **5**(5), 639-652

Patton, R, Frank, P and Clark, R. (1989), *Fault diagnosis in dynamic systems – Theory and application*, Prentice Hall, New York

Ménard, M. (2000), Fuzzy clustering and switching regression models using ambiguity and distance rejects, *Fuzzy Sets and Systems*, **122**(3), 363-399