Detection of Abnormal Flights Using Fickle Instances in SOM Maps

Marie Cottrell, Cynthia Faure, Jérôme Lacaille, Madalina Olteanu

Detection of abnormal flights using fickle instances in SOM maps

Marie Cottrell [1], Cynthia Faure [1,3], Jérôme Lacaille [2], and Madalina Olteanu [1,4]

1 - SAMM, EA 4543
Panthéon-Sorbonne University
90 rue de Tolbiac, 75013 Paris, France
http://samm.univ-paris1.fr
&
2 - Safran Aircraft Engines,
Rond Point René Ravaud, Réau, 77550 Moissy Cramayel, France
https://www.safran-aircraft-engines.com
&
3 - Aosis Consulting,
20 impasse Camille Langlade, 31100 Toulouse, France
http://www.aosis.net/
&
4 - MaIAGE, INRA
Paris-Saclay University
Domaine de Vilvert, 78352 Jouy en Josas, France
http://maiage.jouy.inra.fr

Abstract. For aircraft engineers, detecting abnormalities in a large dataset of recorded flights and understanding the reasons for these are crucial development and monitoring issues. The main difficulty comes from the fact that flights have unequal lengths, and data is usually high dimensional, with a variety of recorded signals. This question is addressed here by introducing a new methodology, combining time series partitioning, relational clustering and the stochasticity of the online self-organizing maps (SOM) algorithm. Our method allows to compress long and high-frequency bivariate time series corresponding to real flights into a sequence of categorical labels, which are next clustered using relational SOM. Eventually, by training SOM with a large number of initial configurations and by taking advantage of the stability of the clusters, we are able to isolate the most atypical flights, and, thanks to discussions with experts, understand what makes a flight an "abnormal" data.

1 Introduction

This present paper is a part of joint work with the Health Monitoring Department of Safran Aircraft Engines Company. The Pronostic Health Monitoring consists in a set of methods to proactively detect any abnormal behavior with the goal of optimizing and planning the maintenance operations.

In an aircraft, sensors are installed on board to record multivariate time series which describe the behavior of the engines. However analyzing this important amount of data is a difficult task impossible to achieve manually. Even if the experts have a very thorough knowledge about the engine operation data, they need some help from algorithmic methods, as mentioned in [1], [2], [3].

In this paper, the flights are considered as a whole and represented by a sequence of labels. Clustering these sequences leads to highlight some groups of similar flights whilst putting to evidence some unclassifiable flights which are very interesting to study and are good candidates for "abnormality".

The data are initially constituted by 549 flights with 8 different engines, with a mean duration of 2.8 hours per flight. The acquisition frequency is 8Hz. No assumption is done about the observed time series, but one of the component is supposed to be a *key variable*, which strongly influences the behavior of the rest.

For the sake of simplicity, the bivariate case only is presented here, but the method can be easily extended to higher dimensional data.

The paper is organized as follows: Section 2 presents the data, Sections 3 and 4 define the two-levels clustering which leads to represent each flight by a sequence of labels. In Section 5, the dissimilarity matrix of all the flights is defined and computed. Section 6 shows how to use the relational SOM to cluster the flights and identify the abnormal ones. Section 7 is a short conclusion.

2 The data

One flight F is represented by a multivariate time series Z_t, with $1 \leq t \leq T$ and $Z_t \in \mathbb{R}^d$. The components of Z_t are the variables recorded by the on board sensors, for example the fan speed, the temperature inside the motor, the plane speed, the oil temperature, etc.

As we take $d = 2$ in this contribution, we only consider $Z_t = (X_t, Y_t), 1 \leq t \leq T$, where the key variable X_t is the fan speed, and Y_t is the temperature inside the engine.

V flights of different lengths are recorded. For each v, $1 \leq v \leq V$, the flight F_v is thus denoted by $Z_t^v = (X_t^v, Y_t^v), 1 \leq t \leq T_v$.

For each flight v, the time series X_t^v is split into phases which can be increasing transient, decreasing transient or stables, by using a rupture detection algorithm such as the PELT algorithm ([4]). The methodology is described in [5] and [6]. These phases have different lengths and the number of phases per flight varies.

3 First level of labeling

To overcome the difficulty of dealing with phases of different lengths, each increasing or decreasing X-phase is substituted by a fixed-length vector composed of its relevant numerical features, as lengthy, midpoint value, median, variance, variances of the two halves, means of the two halves, ...).

Then any clustering algorithm may be used on these vectors. Here the procedure consists in a SOM map training, combined with a hierarchical agglomerative clustering (HAC) applied to the code-vectors computed by SOM ([7]). We group the increasing phases of all the series X_t^v into clusters, denoted by $CA_1, CA_2, \ldots, CA_I$. The same holds for the decreasing phases grouped into clusters denoted by $CD_1, CD_2, \ldots, CD_J$. The set of the stable phases is denoted by CS. These clusters are called "level-1 clusters".

At this step, each flight is labeled by a sequence of labels, which are elements of the set $\{A_1, A_2, \ldots, A_I, D_1, D_2, \ldots, D_J, S\}$, according to the nature of the successive phases of time series X_t^v: A_i if the phase belongs to cluster CA_i, D_j if the phase belongs to cluster CD_j, S for the stable phases of CS.

4 Two-levels clustering and resulting labels

To take into account the second variable Y, we define an embedded level 2 clustering: each cluster CA_i or CD_j is split into a partition formed by the clusters $CA_{i,k}, k = 1, \ldots, K(i)$ or $CD_{j,l}, l = 1, \ldots, L(j)$, which are built according to the second variable Y_t^v. In the same way as for the level-1 clustering, each Y-phase is summarized by its numerical features to make possible the use of classical clustering algorithms.

This two-levels allows us to assign a two-indexes label $A_{i,k}, D_{j,l}$ or label S, to any bi-dimensional phase of any flight, so that all of the data is now summarized into V label sequences denoted by F_v - for the sake of simplicity we use the same notation for a flight and for its sequence of labels.

Table 1 presents the computed values of I, J, $K(i), i = 1, \ldots, I$, $L(j), j = 1, \ldots, J$.

We obtain $I = 7$ level-1 clusters of the increasing phases and $J = 8$ level-1 clusters of the decreasing phases.

I	1	2	3	4	5	6	7
Number of level-2 clusters	4	6	6	6	6	10	4

J	1	2	3	4	5	6	7	8
Number of level-2 clusters	6	8	10	6	10	6	6	8

Table 1: Number of clusters.

The number of labels of the 549 labeled sequences resulting from the two-levels clustering, is 22,5 on average, with a minimum of 10 and a maximum of 35. Figure 1 shows the distribution of the 20 most frequent labels (outside the S label).

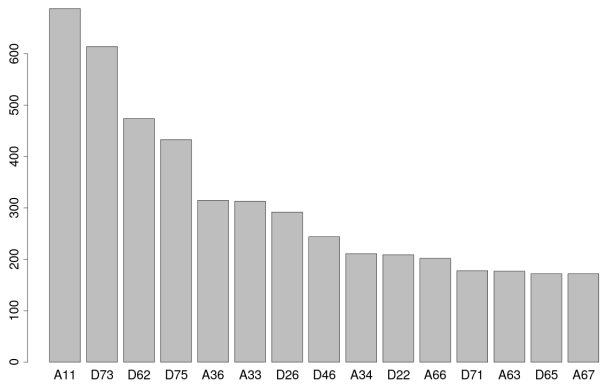

Fig. 1: Distribution of the 20 most frequent labels (outside the S label)

The more frequent labels are the label $A_{1,1}$ (684 occurrences) which is a taxi-phase label during which the plane rolls on the tarmac, followed by other taxi-phases labels ($A_{1,1}, A_{1,4}, D_{73}, A_{3,6}, A_{3,3}$) and some descent phases ($D_{6,2}, D_{2,6}$).

5 Dissimilarity Matrix and Relational SOM

As these labeled sequences are not known by numerical features, we have to use the Relational SOM defined in [8], which is a generalization of the original SOM algorithm defined for numerical data. It only requires as input a dissimilarity matrix between the data. Hence, this method may be applied to any complex data (time series, graphs, texts, etc...) as long as a dissimilarity matrix can be computed.

The dissimilarities are defined according to the Optimal Matching method [9], borrowed to biology and to genetic algorithms and which is based on the computation of transition costs from a label to another one.

Several cases have to be distinguished:

- Substitution costs: two labels are exchanged
- Deletion costs: a label is deleted
- Adding costs: a label is added

5.1 Substitutions costs

Let us define the substitution costs for which we have to consider several cases:

- *Between increasing phases labels substitution costs*

Let A_{ik} and $A_{i'k'}$ be two different labels of i phases.
If $i = i'$, the two level-2 clusters belong to the same level-1 cluster A_i, and we define the cost function c by:

$$c(A_{ik}, A_{ik'}) = \frac{\|\overline{CA_{ik}} - \overline{CA_{ik'}}\|}{\max_{s,s'} \|\overline{CA_{is'}} - \overline{CA_{is'}}\|}$$

where $\overline{CA_{xy}}$ is the bidimensional mean vector of the cluster CA_{xy} and $\|\cdot\|$ is the Euclidean distance in the numerical features space.

If $i \neq i'$, one has to take into account the distance between the level-1 clusters CA_i and CA'_i and also the distance between the level-2 clusters CA_{ik} and $CA_{i'k'}$. So the substitution cost is defined by:

$$c(A_{ik}, A_{i'k'}) = \frac{\|\overline{CA_i} - \overline{CA_{i'}}\|}{\max_{s,s'} \|\overline{CA_s} - \overline{CA_{s'}}\|} + \frac{\|\overline{CA_{ik}} - \overline{CA_{i'k'}}\|}{\max_{s,s'} \|\overline{CA_{is'}} - \overline{CA_{i's'}}\|}$$

– *Between decreasing phases labels substitution costs*

The substitution cost between decreasing phase labels D_{jl} and $D_{j'j'}$ is defined in the same way by:

$$c(Djl, D_{jl'}) = \frac{\|\overline{CD_{jl}} - \overline{CD_{jl'}}\|}{\max_{s,s'} \|\overline{CD_{js'}} - \overline{CD_{js'}}\|}$$

if $j = j'$,
and

$$c(D_{jl}, D_{j'l'}) = \frac{\|\overline{CD_j} - \overline{CD_{j'}}\|}{\max_{s,s'} \|\overline{CD_s} - \overline{CD_{s'}}\|} + \frac{\|\overline{CD_{jl}} - \overline{CD_{j'l'}}\|}{\max_{s,s'} \|\overline{CD_{js'}} - \overline{CD_{j's'}}\|}$$

if $j \neq j'$.

– *Between increasing and decreasing phases labels substitution costs*

According to the definition of increasing and decreasing phases, these substitution costs have to take large values. We take all these costs equal to

$$\alpha \max\left(\max_{i,k,i',k'} c(A_{ik}, A_{i'k'}), \max_{j,l,j',l'} c(D_{jl}, D_{j'l'})\right)$$

where α is a positive number chosen by the user.

– *Between increasing or decreasing phases labels and S labels substitution costs*

These costs are defined in such a way that they are larger than all the substitution costs between increasing phases or those between decreasing phases, therefore

$$c(A_{ik}, S) = \max_{s,u,s',u'} c(A_{su}, A_{s'u'})$$

and

$$c(D_{jl}, S) = \max_{s,u,s',u'} c(D_{su}, D_{s'u'})$$

5.2 Adding costs and deletion costs

These costs are also defined to be very high equal to

$$\beta \max\left(\max_{i,k,i',k'} c(A_{ik}, A_{i'k'}), \max_{j,l,j',l'} c(D_{jl}, D_{j'l'})\right)$$

where β is a positive number chosen by the user.

Table 2 shows a part of the substitution costs matrix. We observe that the substitution costs between two labels beginning by A_1 are smaller than those between one label beginning by A_1 and other one beginning by A_2, as desired.

	$A11$	$A12$	$A13$	$A14$	$A21$	$A22$	$A23$	$A24$	$A25$
$A11$	**0,00**	**0,14**	**0,62**	**0,92**	1,29	1,95	1,97	1,78	1,20
$A12$	**0,14**	**0,00**	**0,65**	**1,00**	1,28	1,65	1,59	1,78	1,42
$A13$	**0,62**	**0,65**	**0,00**	**1,00**	1,77	1,78	1,76	1,01	1,59
$A14$	**0,92**	**1,00**	**1,00**	**0,00**	1,79	1,00	1,70	1,14	1,59
$A21$	1,29	1,28	1,77	1,79	**0,00**	**0,02**	**0,18**	**0,03**	**0,29**
$A22$	1,95	1,65	1,78	1,00	**0,02**	**0,00**	**0,16**	**0,05**	**0,27**
$A23$	1,97	1,59	1,76	1,70	**0,18**	**0,16**	**0,00**	**0,21**	**0,24**
$A24$	1,78	1,78	1,01	1,14	**0,03**	**0,05**	**0,21**	**0,00**	**0,30**
$A25$	1,20	1,42	1,59	1,59	**0,29**	**0,27**	**0,24**	**0,30**	**0,00**

Table 2: Partial representation of the substitution cost matrix.

Let us denote by Δ the dissimilarity matrix, where $\Delta(v, v')$ is the dissimilarity between the labeled sequences F_v and $F_{v'}$, defined as the minimal value of the sum of the required changes costs to exchange F_v and $F_{v'}$.

The distribution of the dissimilarities is illustrated at Figure 2.

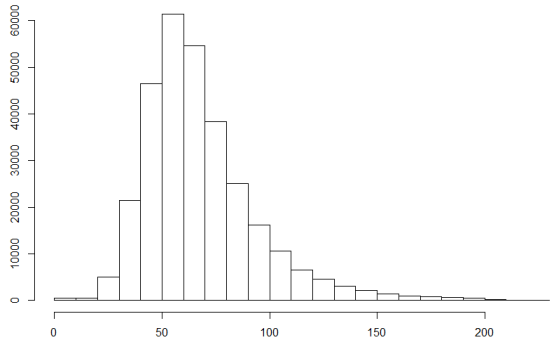

Fig. 2: Dissimilarity distribution

Figure 3 presents the most representative flight, determined as that one which minimizes the sum of all the dissimilarities between it and all the others. It has a "normal" behavior (taxi, take-off, climb, cruise, descent, landing) !

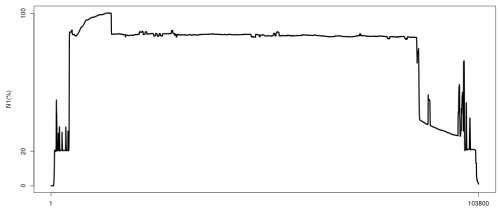

Fig. 3: Variable fan speed of the representative flight

6 Clustering the labeled sequences and identifying fickle flights

We use a 10×10 Kohonen map and the relational SOM algorithm trained on the dissimilarity matrix Δ to get a clustering of the flights, that is of the labeled sequences.

If we consider several runs (at least 50) of the SOM algorithm, for a given size of the map and for a given data set, we observe that most of the pairs of flights are almost always or almost never in the same cluster. But there are also pairs of flights whose associations look random. These pairs of flights are called *fickle* pairs. This question was addressed by [10] in a bootstrap framework and used for text mining in [11] and [12].

After having identified the fickle pairs, we define the fickle flights as being those which belong to an important number of fickle pairs (greater than a certain threshold).

The most fickle flights are then identified and are good candidates for expertise in order to detect anomalies.

After 100 runs of the relational SOM algorithm, the percentages of attractive, repulsive, fickle pairs are computed (see Table 3). Figure 4 shows the labeled

Attractive pairs	Repulsive pairs	Fickle pairs
30.02	58.09	11.80

Table 3: Computed percentages of attractive, repulsive, fickle pairs

sequences re-ordered according to their fickleness, i.e. the number of fickle pairs they belong to.

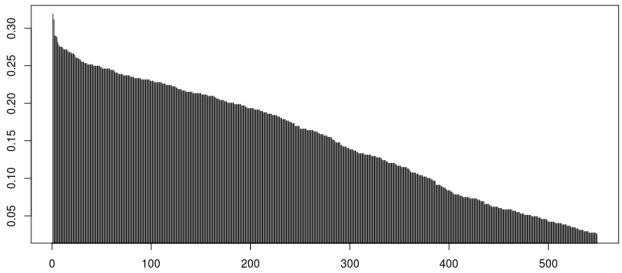

Fig. 4: Fickleness

Then it is possible to seek the most fickle labeled sequences that is the most fickle flights. They are mainly on the edges of the Kohonen maps. The following figures represent 4 fickle flights. The abscissa is the time, the left ordinate is the value of the X variable which is the fan speed, the right ordinate in red is the altitude represented to facilitate the interpretation.

The first fickle flight (Figure 5) looks like the representative flight of Figure 3, however the climb phase is very long and there is an unusual variation during the climb.

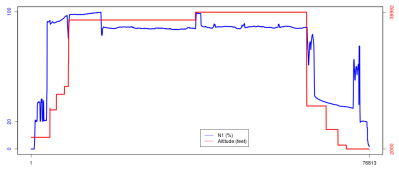

Fig. 5: Fickle flight (Example 1)

Next example (Figure 6) is a very short flight with an atypical behavior of the fan speed variable.

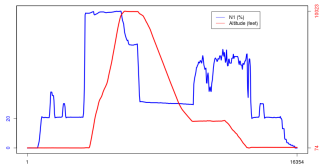

Fig. 6: Fickle flight (Example 2)

In Figure 7, there is an inconsistency between the fan speed and the altitude: it can be a measurement error of the altitude, that has to be confirmed by the experts.

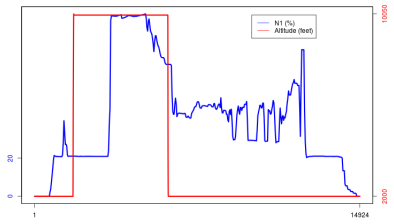

Fig. 7: Fickle flight (Example 3)

For the last example (Figure 8), the altitude has several levels but seems to be normal, whilst the fan speed is chaotic!

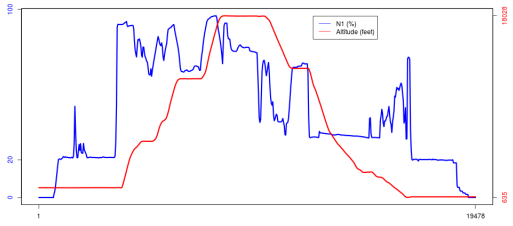

Fig. 8: Fickle flight (Example 4)

All these examples are illustrations of quite atypical flights, which have to be analyzed and characterized by the specialized experts.

7 Conclusion

The transformation of the flights represented by bidimensional time series into sequences of labels makes possible their clustering, in order to identify groups of similar flights, but overall to highlight some atypical flights which are the fickle flights computed after repeated runs of SOM.

This methodology is an interesting tool to mine very complex data and discover abnormal or atypical individuals. The generalization to multidimensional data is straightforward, although the computing time could be increasing with the number of the embedded clustering which are necessary to define the labels.

References

1. A. Bellas, C. Bouveyron, M. Cottrell, and J. Lacaille. Anomaly detection based on confidence intervals using som with an application to health monitoring. In T. Villmann, F.-M. Schleif, M. Kaden, and M. Lange, editors, *Advances in Self-Organizing Maps and Learning Vector Quantization, Proceedings of the 10th International Workshop, (WSOM 2014)*, AISC, pages 145–155, Mittweida, Germany, July 2014. Springer-Verlag.
2. T. Rabenoro, J. Lacaille, M. Cottrell, and F. Rossi. Anomaly detection based on indicators aggregation. In *International Joint Conference on Neural Networks (IJCNN 2014)*, pages 2548–2555, Beijing, China, July 2014.
3. J. Lacaille and V. Gerez. Online abnormality diagnosis for real-time implementation on turbofan engines and test cells. In *Annual Conference of the Prognostics and Health Management Society 2011*, volume 2, 2011.
4. R. Killick and I. Eckley. Optimal detection of changepoints with a linear computational cost. *JASA*, 107(500):1590–1598, 2012.
5. J-M. Bardet, C. Faure, J. Lacaille, and M. Olteanu. Comparison of three algorithms for parametric change-point detection. In Verleysen M., editor, *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2016)*, pages 2–7, Bruges, Belgium, 2016.
6. Cynthia Faure, Jérôme Lacaille, Jean-Marc Bardet, and Madalina Olteanu. Indexation of bench test and flight data. In *Third European Conference of the Prognostics and Health Management Society 2016*. PHM Society, 2016.
7. Cynthia Faure, Jean-Marc Bardet, Madalina Olteanu, and Jerome Lacaille. Design aircraft engine bivariate data phases using change-point detection method and self-organizing maps. In *Conference: ITISE - International work-conference on Time Series*, Granada, Spain, September 2017. University of Granada.
8. M. Olteanu and N. Villa-Vialaneix. On-line relational and multiple relational som. *Neurocomputing*, 147(1):15–30, 2015.
9. A. Abbott and J. Forrest. Optimal Matching Methods for Historical Sequences, Journal of Interdisciplinary History. *Neural Networks*, 16, 3:471–494, 1986.
10. Eric de Bodt, Marie Cottrell, and Michel Verleysen. Statistical tools to assess the reliability of self-organizing maps. *Neural Networks*, 15, 8-9:967–978, 2002.
11. Nicolas Bourgeois, Marie Cottrell, Benjamin Deruelle, Stéphane Lamassé, and Patrick Letrémy. How to improve robustness in kohonen maps and display additional information in factorial analysis: Application to text mining. *Neurocomputing*, 147:120–135, 2015.
12. Nicolas Bourgeois, Marie Cottrell, Stéphane Lamasse, and Madalina Olteanu. Search for Meaning Through the Study of Co-occurrences in Texts. In I. Rojas, G. Joya, and A. Catala, editors, *Advances in Computational Intelligence, IWANN 2015, Part II*, volume 9095 of *Lecture Notes in Computer Science*, pages 578–591, Palma de Mallorca, Spain, June 2015. Springer-Verlag.