An Influence Gauge to Detect and Explain Relations between Measurements and a Performance Indicator

Jérôme Lacaille

Snecma Villaroche, 77550 Moissy-Cramayel, France
jerome.lacaille@snecma.fr

ABSTRACT

What about a software tool that behaves like a gauge able to estimate the quantity of information contained in a group of measurements? Then if we have a performance indicator or a defect rate, how may we compute the maximum performance explanation contained in our dataset? The first question may be answered by entropy and the second with mutual information. The present paper recalls a simple way to use those mathematical tools in an application one wants to launch each time a new dataset has to be studied. Often the PHM team in Snecma is asked to participate in special workforces to analyze sudden crisis. This methodology helps at the very beginning of the process to identify our mathematical capability to build an explanation model. This was the case during a small engine start crisis when some spark plugs were not working. Another time we used this tool to identify the flying condition when a gearbox was heating. This methodology was first developed for industry purposes like the optimization of machine tools or process recipes. Its success is in the simplifications of the computations that enlighten the interpretability of the results. Each signal is quantified in a way that improves the mutual information with the performance indicator. This is done signal by signal, but also for any small subsets of multivariate measurements until the confidence given by the quantity and quality of the data reaches its maximum. The segmentation of the data helps and boosts the computation of the integrals. Moreover, as this methodology uses quantified data as inputs it works as well with any sort of inputs such as continuous, discrete ordered and even categorized measurements. Once a best subset of measurements is selected a simple non-linear model is built using a relaxation algorithm. This model is a set of hypercubes that classifies the input space in a very simple and interpretable way. The methodology given below is a rough approach and may be replaced by more efficient regression algorithms if one only have continuous measurements but it has some advantages like a way to search a "best rule" according to some constraints and a graphic navigation tool very efficient to correct recipes.

1. APPLICATION EXAMPLES

This section gives some application examples of the influence analysis methodology. The next section (2. Mathematic Methodology) details computations implemented for this relatively simple method. Hence it is possible to read section 1 first, references to section 2 are given anyway when necessary, or read section 2 before section 1 if more interested by the mathematical aspects.

1.1. No light up during the engine start process

Some turbofan aircraft engines do not light up on one start plug (two start plugs are positioned on opposites sides of each engine), but the event is really rare and never happens when using both plugs simultaneously (look at (Flandrois, Lacaille, Massé, & Ausloos, 2009) for analysis of the start capability of an aircraft engine). The number of such events is so small that a statistic analysis was not possible. But for aircraft engines as well as for any kind of engine, a plausible indication of the future risk of no-light-up is the increasing duration of the ignition time. So I get these durations for a set of flights from a fleet of similar engines and renormalize them according to external conditions among external temperature, pressure, altitude and oil temperature before ignition (Lacaille, 2009, 2010). Then I used the influence analysis process to find sets of parameters that may explain increase of this duration.

The potential factors were identified by engine experts and were related to engine conditions: temperatures, pressures at different measurement points, shafts' rotating speeds, variable geometries, airport, flight time, plug position, fuel system and ignition process. We collected more than 10000 starts (more than 5000 different flights) which allow computing very small confidence intervals for the influence criterion.

The first monovariate analysis (Figure 1) shows that the oil temperature was the most important factor but it does not explain more than 33% of the delay. Fuel flow regulation and adapted rotation speed increase explanation to 40%.

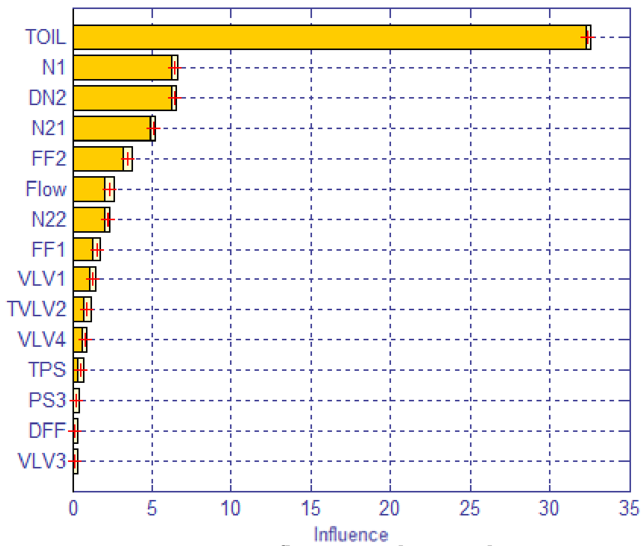

Figure 1. Monovariate influence analysis. Oil temperature shows the maximum impact on the delay but explains only a little more than 30% of this criterion.

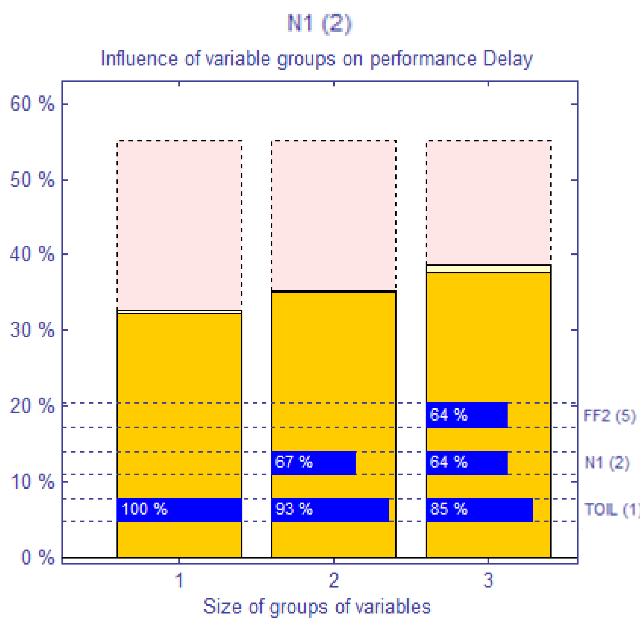

Figure 2. Computation of the best sets of 1, 2 and 3 parameters which jointly influence the ignition duration. The bar sizes give the influence criterion and the light part on top is the confidence interval. The small blue horizontal bars (and values) are the importance of each parameter (monovariate influence of each single parameter over the set classification). The 55% limit on the top of the graph is the maximum influence that may be obtained using all the available data.

Finally expert impact analysis identifies atmospheric conditions and fuel density as main causes of variations of the ignition delay.

1.2. A gearbox is heating abnormally during flights

Abnormal heating in a gearbox was detected during flights tests on rare occasions. Data from 51 maneuvers were collected to isolate the flight conditions when this phenomenon arises.

The quality criterion was a difference between observed temperature and the expected temperature of the gearbox estimated according to a physical model. The factor of the analysis described

- flight conditions: aircraft speed, altitude, different attitude angles;
- engine conditions: rotation speeds, engine temperatures;
- measurement conditions: maneuver type, temperature stabilization time.

A first monovariate analysis (Figure 3) identifies the altitude (Alt) as linked to almost 80% of the temperature increase on our set of observations.

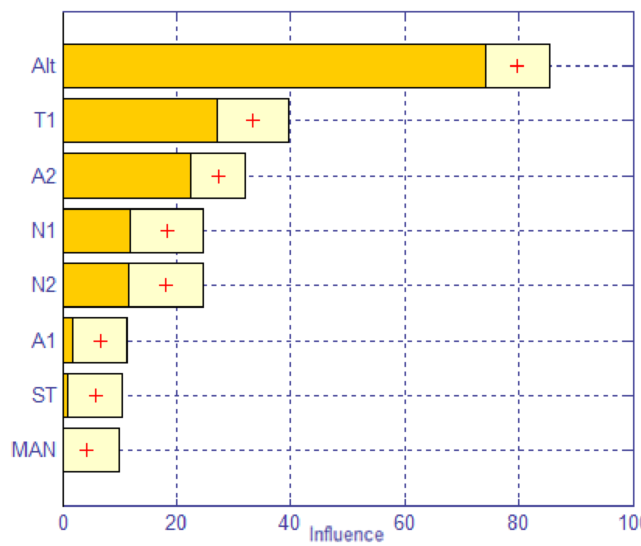

Figure 3. Monovariate analysis on 51 experiments for identification of conditions related to increase of a gearbox temperature. The confidences intervals are the light boxes at the end of each bar.

The multivariate analysis (Figure 4) shows that on those 51 experiments an explanation may be improved using aircraft attitude (A2, ~5%), engine speed (N1) and temperature stabilization time (ST).

The number of observations was low but the confidence intervals were not so big and the analysis confirmed the intuition of the engine experts which were able to redesign the gearbox and eliminate this event.

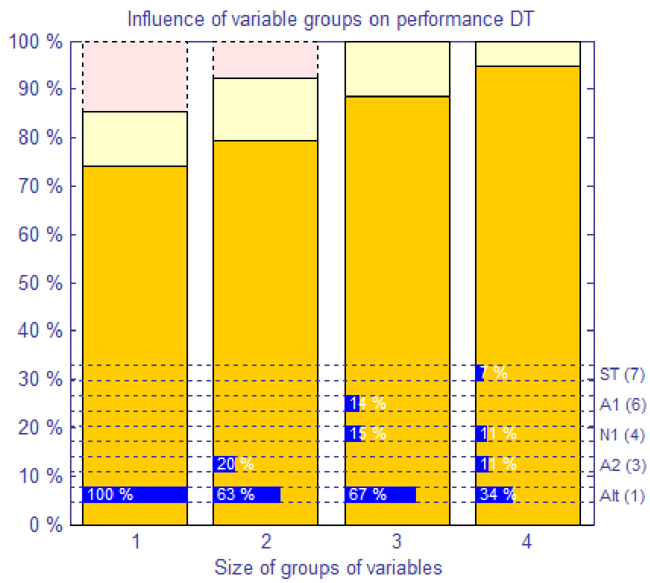

Figure 4. Multivariate analysis on 51 experiments for identification of conditions related to increase of a gearbox temperature.

1.3. Adjustment of a carbon deposition process

The quality of an electric anode used in the extraction process of aluminum depends on the anode carbon density. In the anode making operation one wants to find how to adjust the chemical bath in order to optimize the density. In this application one does not try to find any potential causes of degradation but to isolate a good set of parameters to adjust.

The first step was to quantify the anode density, fixing some important levels observed during a batch of experiments. Then classification trees where learnt to build a clusterisation of each parameter set adapted to the quantified density (section 2.4). This transformation of the inputs allows a simple computation of entropies and influence criterion (section 2.5). But the quantification process obtained from trees also define bounds for each parameters and once a parameter detected as influent on the process these bounds are used to build an optimized recipe.

Each tree input is a set of parameters, as this is a combinatory problem; a relaxation chain was implemented by a genetic algorithm which converges to a good enough set of parameters. The algorithm maintains a population of agents, each one corresponding to the selection of a subset of parameters. A mutation step managed random changes, suppressions or adjunctions of one parameter, and a cross-over step combined pairs of sets from random binary partitions.

In this application we had 1681 experiments with 17 adaptable control parameters and an anode density measurement as performance criterion. A cross validation procedure ensured robustness of the analysis with a k-fold methodology with batches of 10% of random experiments kept for the test phase. Four levels of density were retained by the process experts: high, medium high, medium low and low.

Figure 5 displays results obtained by selection of parameters using progressively increasing set sizes. It is not usual that a same parameter maintains its influence when the set size increases. It was however the case here.

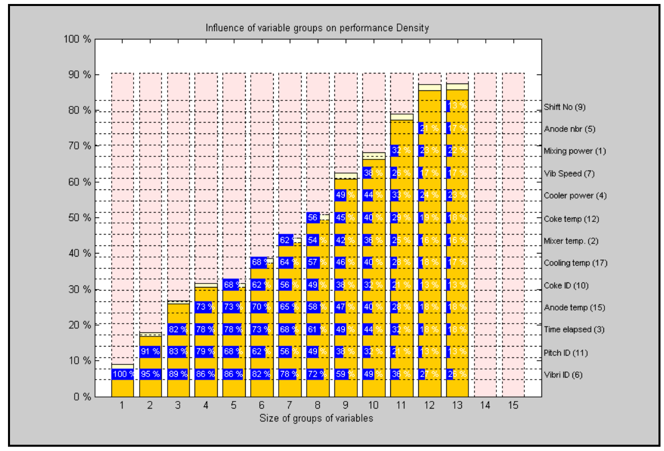

Figure 5. Progressive selection of the best parameters to adjust.

The analysis results show that one cannot expect much improvement of the fabrication process by just changing one of the parameters. As said before the classification trees helps to build recipes that optimize some constraints, one of them is proposed on Figure 6.

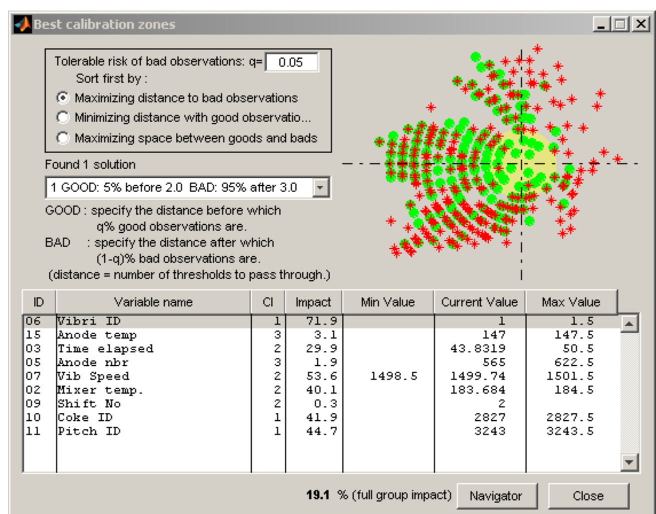

Figure 6. A recipe for concurrent adjustment of 9 control parameters that limit the number of low density anode production. Top left of the interface is the constraint selection, top right is a flatten representation of the local dispersion of good and bad production around the recipe.

This application concludes by a prototype helping adjustment of parameters in real time when the process was drifting.

1.4. Optimization of a crankshaft production chain

The crankshaft production chain for automobiles engines is made of a succession of machine tools. Each tool has a small list of adjustable parameters. Almost at the end of the production chain was an expansive finishing equipment. Some weeks before a scheduled maintenance operation of this equipment the production yield decreased dramatically but the production experts were not affirmative about the cause of this degradation. The team extracts some measurements, a set of data per operation for three of them which were identified as probable causes: the draft production (20 variables), an intermediate polishing operation (10 variables) and the finish process (30 variables). They also had some process recipe data (25 variables) and context measurements and origin of the input material (13 variables). Each of these factors was a set of measurements identified by the production experts as influent on the quality of the crankshaft.

The performance indicator was a binary output identifying good shafts. It was the output of an abnormality detection test I built from geometric measurements and that gives 87% of good detection and only 9% of false alarms according to manual verification by the experts.

A quantization step transforms the sets of process measurements vectors into 5 categorical indicators using robust classification trees (see section 2.4 below).

The influence analysis on 377 observations (Table 1) gives an influence value on the performance for each factor and identifies the polish operation as the most plausible cause with 45% of the influence on production.

Table 1. Influence analysis results for shaft production.

Factor	Influence	Precision
Context	3%	4%
Process recipe	1%	3%
Draft	0%	33%
Polish	45%	9%
Finish	11%	3%

However the draft operation may also be a potential cause because of the poor precision of the computation.

An optimal tuning was found for this polishing tool with a measurable impact on the quality of the fabrication process. This recipe is a direct application of the classification tree bounds found for data quantification.

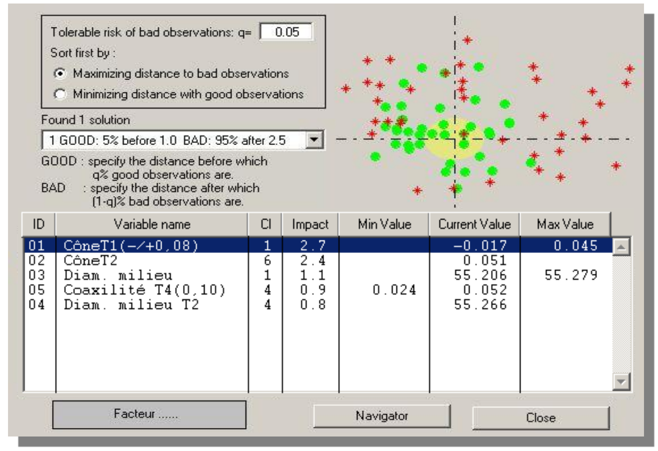

Figure 7. Identification of a recipe for the polishing tool that maximizes the number of good shafts.

Equivalent recipes may be found for each operation on the process.

1.5. Causes of bubble production in glass fabrication

This analysis was at the origin of the development of the influence analysis methodology. It was about glass-making process for automobile windshield production.

The process of glass production is complex, we schematize: raw material enter the oven, gas burners melt this material in a fluid that undergoes two successive vortexes, finally some of this fluid exits the oven on a mercury mat and is shaped according to specifications during the annealing phase.

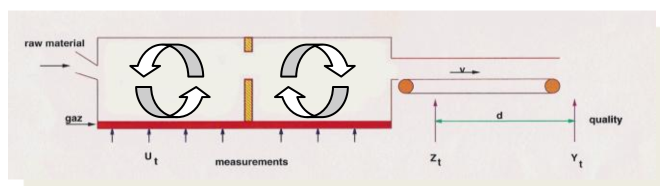

Figure 8. A schematic view of a glass fabrication oven.

Lots of factors may be source of bubbles production:

- RESUR: displacement of the resurgence point (a position around the middle of the oven where the two vortexes exchange material).
- PROFIL: change in the energy distribution profile given by the gas flow for each burner.
- PRESSURE: the pressure in the oven.
- TIREE: The speed of the glass exiting the oven.
- COMRED: Reduction combustion.
- And lot more.

This application was clearly a case where a temporal filter should be applied on the data. If you input colored material, then the produced glass leaves are progressively colored.

Figure 9 is a schematic view of a classical transfer function for the color.

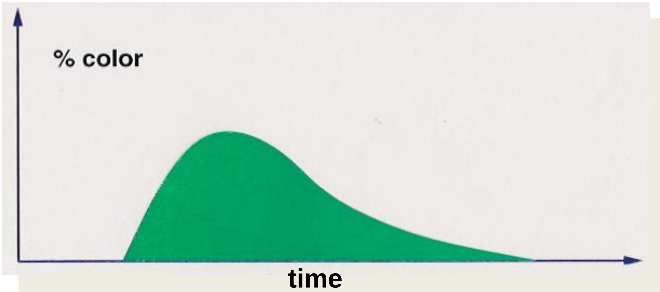

Figure 9. Progressive apparition of the color in the output glass.

A temporal filter was then applied on the measurements corresponding to each of the potential causes (as described section 2.2 below). Then a clusterisation helps computing the different influence values.

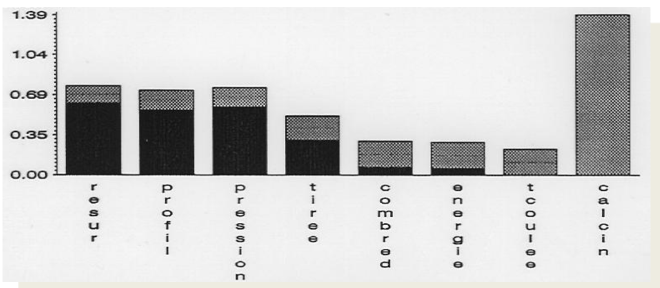

Figure 10. Influence analysis on the causes of bubbles production during glass fabrication process. Resurgence point position appears to be an important cause as well as energy profile and pressure.

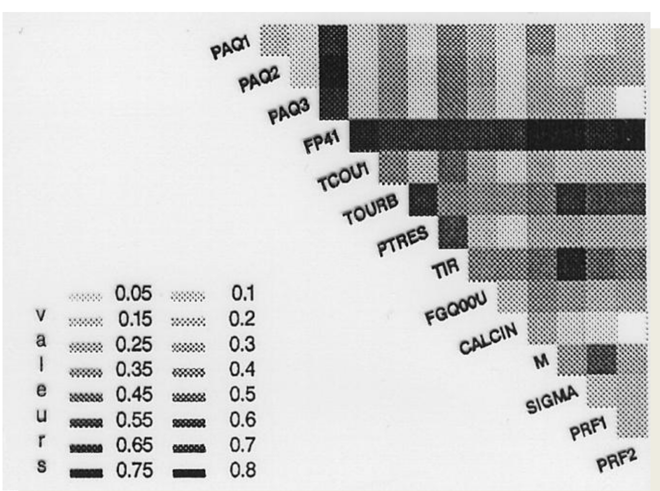

Figure 11. Comparison with correlation analysis by pairs of measurements for the glass-making process. Each pair is cross-correlated with the quality of the product.

Figure 10 shows the result of the influence analysis of bubble production in a plant north of Paris during a small production crisis in 1993. I was able to identify the main causes of defectivity.

Figure 11 shows a correlation analysis of each couple of measurements with the quality of the product (after individual temporal filtering for each measurement). This graph identifies pressure (FP41) but not clearly the main causes of degradation.

2. MATHEMATIC METHODOLOGY

2.1. Performance indicator and potential causes

Influence analysis is the computation of an absolute value that quantify a relation between a set of observations linked to some sort of physical phenomenon and a performance indicator. The performance indicator may be a fab yield, the quality of production or a defectivity rate. The potential causes of degradation or amelioration are more difficult to master. They should be modeled by sets of physical factors given by experts. Sometimes, one factor is enough to produce a clear understanding of the risk linked to the monitored system. Often such risk measurement is computed as a score of a statistic model (a log-likelihood for example). But most of the time a statistic model is difficult to build and a variation of production or a change in the behavior of a system should be extracted from multiple measurements simultaneously. This is the most frequent case when potential causes are not independent and interact in a complex system. In that case the complex system should be identified by the list of all its potential causes of defectivity.

2.2. Time delay and data synchronization

A cause of degradation may also be detected as a change in the temporal behavior of a set of signals. I usually simulate that behavior with a rough model like an autoregressive linear model in the simplest case or a recurrent network for non linear behaviors. The parameters of the model are then taken as state factors for the system to monitor. It is the case when some delay exists between the immediate effect of degradation (eventually an action) and the corresponding result on the performance indicator.

In (Lacaille, 1998) and (Lacaille, 1997) I roughly model time dependency by a rational filter. The resulting estimation is not important but the state of the system exhibits its internal dimensionality and a set of intermediate (computed) factors.

Let for example x_t be the set of measurements or computations, identified by system experts, collected at time t and relevant to explain parts of the performance. A delay δ may exist between observation x_t and result performance $y_{t+\delta}$. This delay is unknown and must even be more a combination of past observations than just a time laps. This temporal combination of past data may be approximated by an autoregressive filter. Equation (1) gives a markovian representation of such a model (Akaike, 1975) where the intermediate stochastic process z_t is the state of this system.

$$\begin{cases} y_t &= C_{t+1} z_t \\ z_{t+1} &= A_{t+1} z_t + B_{t+1} x_t \end{cases} \tag{1}$$

The dimension of vector z_t gives the rank of the system (and matrix A eigenvalues an idea of the delay). Initial values for rank and matrices A_0, B_0 and C_0 are computed by least square regression and minimization of the AIC criterion. Even evolution of the matrices may be obtained by a recurrent tracking method (2) or Kalman filter (Kalman, 1960; Welch & Bishop, 2006).

$$\begin{cases} \Delta A_t &= \eta C_t'(\hat{y}_t - y_t) z_{t-1}' \\ \Delta B_t &= \eta C_t'(\hat{y}_t - y_t) x_{t-1}' \\ \Delta C_t &= \eta(\hat{y}_t - y_t) z_t' \end{cases} \tag{2}$$

This is a rough linear approximation limited to rational filters but (Lacaille, 1994) gives clues of how to replace the linear model by a three layer recurrent perceptron.

2.3. Influence criterion

Once we have a performance indicator and a set of potential causes factors our goal is to sort these factors in order of influence on the performance. A classic solution in statistic is to use linear models and compute correlations between each factor regression and the performance indicator (adjusted R^2 coefficient to take the different indicators dimensions into account). When the relation is not clearly linear, if we do not have just monovariate indicators or even non-numeric values, but a set of measurements representing in their whole the state of a system we want to use something more generic than just a linear correlation.

Mutual information $I(X,Y)$ measures the quantity of information shared between two stochastic variables X and Y. It comes from the computation of entropy $H(X)$ which defines the total quantity of information contained by a stochastic variable. (Note that the stochastic variables used here are not necessary of dimension 1 or event numeric.) For example, a qualitative variable with values in a set of k labels $\{x_0 \ldots x_k\}$ has an entropy value between 0 if the variable is constant and its maximum $\log(k)$ if it has a uniform distribution (when all labels are random).

The following equations give the integral formulation of the entropy and mutual information:

$$\mathrm{H(X)} = -\int_{\mathrm{x}} \mathrm{dP(X=x)} \log \mathrm{P(X=x)} \tag{3}$$

$$\mathrm{I(X,Y)} = \int_{\mathrm{x,y}} \mathrm{dP(X=x,Y=y)} \log \frac{\mathrm{P(X=x,Y=y)}}{\mathrm{P(X=x)P(Y=y)}} \tag{4}$$

Equation (4) shows that the mutual information may be interpreted as the Kullback-Leibler divergence between the distribution of the couple of variables (X,Y) and the product of their distributions *as if* they were independent. Hence it measures some sort of distance from the couple's law to stochastic independence.

The two formulations are linked by the following set of equalities:

$$\begin{aligned} I(X,Y) &= H(Y) - H(Y|X) \\ &= H(X) - H(X|Y) \\ &= H(X) + H(Y) - H(X,Y) \end{aligned} \tag{5}$$

Hence mutual information is less than each of the entropies and clearly corresponds to the entropy of one of the variable from which one suppresses the information *given* by the second.

The selected influence criterion is defined by the proportion of information contained by factor X that explains the performance indicator Y.

$$\lambda_Y(X) = I(Y|X) = \frac{I(X,Y)}{H(X)} \in [0,1] \tag{6}$$

We do not use $H(Y)$ for denominator because we do not want to favor factors with too much entropy. Such factors may be too complex and will not have enough robustness in a data analysis process.

2.4. Data quantification and measurement selection

A generic computation of entropy is difficult to implement. A well known solution is to use a nearest neighbor based approximation known as the Kraskov method described in (Kraskov, St, & Grassberger, 2004). But this method implies that a distance between measurements exists. This may be tricky when observations come from different sources and are not really comparables. It is always possible to define individual comparisons for each factor, but we still have a problem to build a multivariate solution.

A simplest approach consists in the quantization or categorization of performance indicator and factors. Moreover it is easier to convince experts with logical rules like "when X_1 is low and X_2 is medium then the performance is low" instead of a complex analytical representation.

The quantization of our data is achieved in two phases. First we define thresholds for the performance indicator. This is an easy task and very clear to experts. Then for each factor, which can be multivariate, we implement a clusterisation driven by the quantified performance factor. Usually we train a regression tree with cross validation procedure to limit tree depth and increase the robustness. Then we associate a different label to each of the leaves. Other clusterisation methods may be used like SVM (Burges, 1998) or Bayesian networks (Pearl, 1988) and much more. Classification trees (Breiman, Friedman, Olshen, & Stone, 1984) keep an advantage in interpretability of the clusters and helps building good recipes for the analyzed process.

2.5. Integrals computation and confidence intervals

Once the input data quantified, the computation of the entropies are straightforward: it is just a matter of counting. Let define the following notations for the quantified factor $\hat{X}$:

$$\hat{X} \in \{x_1 \dots x_k\} \text{ and } P(\hat{X} = x_i) = p_i \tag{7}$$

Each p_i is estimated by the proportion $\widehat{p_i}$ of label x_i. Then the estimated entropy of quantified factor $\hat{X}$ is given by

$$\widehat{H} = -\sum_{i=1}^{k} \hat{p}_i \log \hat{p}_i \tag{8}$$

$\widehat{p_i}$ is a statistics, then a stochastic variable of mean p_i and variance $\sigma_i^2 = \frac{p_i\,(1-p_i)}{N}$, N being the number of observations.

To estimate a confidence interval for this approximate entropy we use the Neymann-Pearson approximation:

$$\chi^2 = 2N \sum_{i=1}^{k} \hat{p}_i \log \frac{\hat{p}_i}{p_i} \tag{9}$$

Where this value follows a χ^2_{k-1} distribution (Chi^2 with $k-1$ freedom degrees). Hence we obtain

$$\sum_{i=1}^{k} \hat{p}_i \log \hat{p}_i - \sum_{i=1}^{k} p_i \log p_i - \sum_{i=1}^{k} (\hat{p}_i - p_i) \log p_i = \frac{1}{2N}\chi^2 \tag{10}$$

$$H - \widehat{H} = \sum_{i=1}^{k} \frac{(\hat{p}_i - p_i)}{\sqrt{p_i}} (-2\sqrt{p_i} \log \sqrt{p_i}) + \frac{1}{2N}\chi^2 \tag{11}$$

But in this last equation we note that the χ^2 value is an estimation of the sum of squared values of each term $\frac{p_i - \hat{p}_i}{\sqrt{p_i}}$, each one that may be approximated by independent normal laws of variance $\frac{\sigma_i^2}{p_i}$. Hence Cochran theorem (Cochran, 1934) states that the two terms in the sum equation (11) are independent. Thus the variance of estimated entropy $\widehat{H}$ may be estimated by the sum of variances of those two terms:

$$\mathrm{Var}\widehat{H} \approx \frac{1}{N} \sum_{i=1}^{k} p_i(1-p_i) \log^2 p_i + \frac{k-1}{2N^2} \tag{12}$$

The standard deviation σ_H of the entropy estimated by (8) is approximated by the square root of the variance (2) where each theoretical proportion is approximated by its own estimation. A confidence interval around $\widehat{H}$ is chosen as $\widehat{H} \pm \Delta H$ where ΔH is a multiple of the estimated standard deviation σ_H (usually $2\sigma_H$).

An analogous computation may be developed for the mutual information:

$$\hat{I}(X,Y) = \sum_{i=1}^{k} \sum_{j=1}^{l} p_{ij}^{xy} \log \frac{p_{ij}^{xy}}{p_i^x p_j^y} \tag{13}$$

but it is easier to use the last equality in (5) to approximate the interval bounds from twice the sum of each standard deviation. Then we finally use a logarithmic approximation for the influence coefficient.

$$\Delta I(X,Y) \le \Delta H(X) + \Delta H(Y) + \Delta H(X,Y) \tag{14}$$

$$\Delta\lambda_Y(X) \le \lambda_Y(X) \left(\frac{\Delta I(X,Y)}{I(X,Y)} + \frac{\Delta H(X)}{H(X)} \right) \tag{15}$$

This gives us a rough set of bounds we are using to draw the light bars on the example graphics section 1.

3. CONCLUSION

Influence analysis is more a methodology than a tool. To be efficient one has to prepare the data with most of process and physic knowledge as possible. But at the end it defines an application skeleton which may probably be adapted to any specific process.

The equations from section 2 applied in the examples section 1 are simple and rough approximations for the computation of mutual information but they were sufficient to solve some really interesting problems. A detailed implementation (in French) and example is given in (Lacaille, 2004) and applications in semi-conductors fabrication may be found here (Lacaille & Dubus, 2005; Lacaille, 2005, 2008). In the case of linear relations and monovariate factors one may prefer a L_1 constraint robust regression like the LASSO method (Tibshirani, 1996) to select the factors. This methodology however is entirely generic and may be applied even to non numeric categorical data and multivariate factors.

The proposed methodology to select and identify subset of influent variables may also be seen as a classification, problem. Indeed our goal is to separate datasets according to different levels of performance. In that case one may refer to the ample bibliography on the use of mutual information to help feature selection in the paradigm of classification (Doquire & Verleysen, 2011). One may even see some limitation in using mutual information for classification purposes (Frénay, Doquire, & Verleysen, 2012).

Figure 12 recall each step in our methodology. The two first steps are essentially expert driven; the synchronization is managed by linear or non-linear autoregressive models; and the data quantification is done either with decision trees, or any other classifier with an optimization loop on the mutual information computation. This optimization may be implemented with a genetic algorithm or any relaxation scheme. Finally we implement a simple model based on decision trees for inference purposes but any other estimation tool such as neural network may be used on the selected subset of measurements.

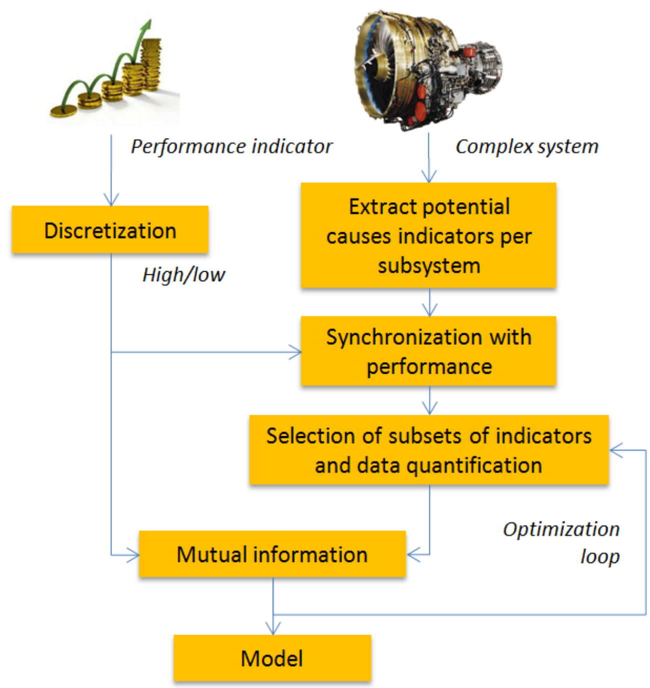

Figure 12. Methodology flowchart.

The main fact to remember is that such computation may be used at the beginning of any data mining challenge just to get some clues about the quantity of explanation one may be able to extract from a dataset.

NOMENCLATURE

AIC	Akaike Information Criterion
LASSO	Least Absolute Schrinkage and Selection Operator
PHM	Prognostic and Health Management
SVM	Support Vector Machines

REFERENCES

Akaike, H. (1975). Markovian Representation of Stochastic Processes by Canonical Variables. *SIAM Journal of Control*, *13*(1), 162–173. doi:10.1137/0313010

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and Regression Trees*. *The Wadsworth statisticsprobability series* (Vol. 19).

Burges, C. J. C. (1998). A Tutorial on Support Vector Machines for Pattern Recognition. In L. T. Bell Laboratories (Ed.), *Data Mining and Knowledge Discovery* (Vol. 2, pp. 121–167). Kluwer.

Cochran, W. G. (1934). The distribution of quadratic forms in a normal system, with applications to the analysis of covariance. *Mathematical Proceedings of the Cambridge Philosophical Society*, *30*(2), 178–191. doi:http://dx.doi.org/10.1017/S0305004100016595

Doquire, G., & Verleysen, M. (2011). Mutual information for feature selection with missing data. In *ESANN* (pp. 27–29).

Flandrois, X., Lacaille, J., Massé, J.-R., & Ausloos, A. (2009). Expertise Transfer and Automatic Failure Classification for the Engine Start Capability System. In *AIAA InfoTech*.

Frénay, B., Doquire, G., & Verleysen, M. (2012). On the Potential Inadequacy of Mutual Information for Feature Selection. In *ESANN* (pp. 25–27). Bruges.

Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME-Journal of Basic Engineering*, *82*(Series D), 35–45. doi:10.1115/1.3662552

Kraskov, A., St, H., & Grassberger, P. (2004). Estimating Mutual Information. *Physical Review*, *69*(6).

Lacaille, J. (1994). Generalization of Stochastic and Deterministic neural Network with a Continuous State Space and a Connectivity Greater than Two. In *IEEE World Congress on Computational Intelligence* (Vol. 2). Orlando. doi:10.1109/ICNN.1994.374304

Lacaille, J. (1997). An Autoadaptative Neural Method to Synchronize Multivariate Sensors. In *Les réseaux neuro-mimétiques et leurs applications (NEURAP)*. Marseille.

Lacaille, J. (1998). Synchronization of multivariate sensors with an autoadaptive neural method. *Intelligent & Robotic Systems*, *21*(2), 155–165.

Lacaille, J. (2004). *Industrialisation d'algorithmes mathématiques*. *Université Paris 1*. Paris 1, Habilitation thesis.

Lacaille, J. (2005). Mathematical Solution to Identify the Causes of Yield Deterioration. In *International Sematech Manufacturing Initiative (ISMI)*. Austin, TX: Sematech. Retrieved from https://sites.google.com/site/jeromelacaille/semiconducteurs/ismi-2005---fab-wide-yield-analysis

Lacaille, J. (2008). Global Predictive Monitoring System for a Manufacturing Facility. US: PDF Solutions. Retrieved from http://worldwide.espacenet.com/publicationDetails/biblio?DB=EPODOC&adjacent=true&locale=en_EP&FT=D&date=20080403&CC=US&NR=2008082197A1&KC=A1

Lacaille, J. (2009). Standardized failure signature for a turbofan engine. In *IEEE Aerospace conference*. Big Sky (MT): IEEE Aerospace society. doi:10.1109/AERO.2009.4839670

Lacaille, J. (2010). Standardization of Data used for Monitoring an Aircraft Engine. USA; FR. Retrieved from http://worldwide.espacenet.com/publicationDetails/biblio?DB=EPODOC&adjacent=true&locale=en_EP&FT=D&date=20100708&CC=WO&NR=2010076468A1&KC=A1

Lacaille, J., & Dubus, H. (2005). Defectivity Analysis by a Swarm of Intelligent Distributed Agents. In *AEC-APC*. Indian Wells, CA: Sematech. Retrieved from https://sites.google.com/site/jeromelacaille/semiconducteurs/2005-09---aec-apc-indian-wells

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. In *Royal Statistical Society* (Vol. 58, pp. 267–288).

Welch, G., & Bishop, G. (2006). An Introduction to the Kalman Filter. *In Practice*, *7*(1), 1–16. doi:10.1.1.117.6808

BIOGRAPHY

Jérôme Lacaille is a Safran emeritus expert which mission for Snecma is to help in the development of mathematic algorithms used for the engine health monitoring and statistical analysis of company data. Jérôme has a PhD in Mathematics on “Neural Computation” and a HDR (habilitation à diriger des recherches) for “Algorithms Industrialization” from the Ecole Normale Supérieure (France). Jérôme has held several positions including scientific consultant and professor. He has also co-founded the Miriad Technologies Company, entered the semiconductor business taking in charge the direction of the Innovation Department for Si Automation (Montpellier - France) and PDF Solutions (San Jose - CA). He developed specific mathematic algorithms that where integrated in industrial process. Over the course of his work, Jérôme has published several papers on integrating data analysis into industry infrastructure, including neural methodologies and stochastic modeling.