Testing of Sensor Condition Using Gaussian Mixture Model

Ladislav Jirsa and Lenka Pavelková
Institute of Information Theory and Automation, Czech Academy of Sciences
Pod Vodárenskou věží 4, Prague, Czech republic
{*jirsa, pavelkov*}*@utia.cas.cz*

Keywords: Sensor Faults, Bayesian Statistics, Gaussian Mixture, Dynamic Weights.

Abstract: The paper describes a method of sensor condition testing based on processing of data measured by the sensor using a Gaussian mixture model with dynamic weights. The procedure is composed of two steps, off-line and on-line. In off-line stage, fault-free learning data are processed and described by a probabilistic mixture of regressive models (mixture components) including a transition table between active components. It is assumed that each component characterises one property of data dynamics and just one component is active in each time instant. In on-line stage, tested data are used for transition table estimation compared with the fault-free transition table. The crossing of given level of difference announces a possible fault.

1 INTRODUCTION

Fault detection plays an important role in today's industry. An industrial plant has many possible fault sources, e.g., sensors, actuators, hardware components, communication lines. There exist a large amount of approaches and solutions, mainly tailored for a particular system, see e.g. (Isermann, 2011).

Sensors belong to basic units of an industrial plant. Their faults may be critical for correct control of a system or decision of proper operational state.

The analytical redundancy is a frequently used method in fault detection. It is based on indirect measurement of the variable of interest and requires a model of the concerned physical system. The information on the variable is obtained by other available quantities as inputs of the system model, giving the required variable as output. For example, in (Walambe et al., 2010), air-breathing combustion system of an aircraft engine is modelled, the model is fed by a set of signals from various sensors and the residual signal is processed by a bank of extended Kalman filters, each one corresponding to one type of a sensor fault. State of aircraft engine is similarly diagnosed in (Wei and Yingqing, 2009), whereas in (Lu et al., 2012), *hardware redundancy* is added, when signals from two physical sensors are compared against the output of the engine model and tested for two types of faults.

Concerning particular classes of system model, analysis of existence and inference of explicit relations for state estimation are elaborated e.g. a discrete-time linear systems with state delays for probabilistic sensor gain faults (He et al., 2008) or continuous Lipschitz nonlinear systems of three or more sensors using linear matrix inequalities (Rajaman and Ganguli, 2004).

Absence of the explicit system and fault model can be substituted by methods based on learning, e.g. support vector machines used for classification of faults into multiple classes (Wang et al., 2014b) in metal cutting industry or, in the same application area, (auto)regressive model of multisensory information processed by cointegration method, used for prediction of tool wear (Wang et al., 2014a). Application of Gaussian processes and Rényi entropy is shown in (Boškoski et al., 2013) for bearing fault prognostics and estimation of faultless lifetime. To classify the states, both faultless and faulty data must be processed.

A question of extracting features contained in correct data has been considered. This paper proposes an alternative approach to a sensor condition monitoring. A generic data-based probabilistic methodology not requesting a system and fault model is developed, detecting situations when a single sensor in question provides data that are not in accordance with the historical experience. The applicability of the approach is intended, initially, just for a sensor monitoring. Also, only one sensor (data source, quantity) is explicitly considered without dependencies on other quantities.

The work is included in application of probabilistic methods in cold metal rolling industry, see (Dedecius and Jirsa, 2010), (Ettler et al., 2011), (Ettler et al., 2013), (Jirsa et al., 2013), (Ettler and Dedecius, 2013). There, an industrial plant is described by a hierarchical structure where a *sensor* is one of low-level units. Data generated by a sensor can be distorted in two ways: (a) faulty sensor under correct process condition, (b) correct sensor under faulty process condition. We do not distinguish between these two cases and we focus on output data properties, despite of cause or localization of eventual fault.

As a data description method, probabilistic mixtures of regressive models (components) with normal noise are used. It is assumed that in each time instant only one component (model) describes the current data. If the data behaviour changes and another component becomes suitable for its description, a transition between the components occurs, in other words, data description switches from one model to another. Transition probability between components in the mixture, i.e. transition table, represents dynamic weights of particular component in the mixture.

The mixture components describe particular properties of processed data. A component can be interpreted in one of two ways:

- Each component represents a particular mode of the system generating the data. These modes are defined in advance by expert. For example, frequency of engine rotation increases, frequency is constant, frequency decreases. Road is empty, jammed or closed. Patient is male or female. Number of components corresponds to the number of modes and each component has a clearly defined meaning.
- Each component represents a particular property of the data in some sense. This property has not been known in advance and it has been detected during the mixture estimation, according to the model structure, prior information and other conditions. Meaning of such a component may not be clearly interpreted as in the previous case.

However, these aproaches can be combined, i.e. a part of components can be defined in advance and a part can be detected *ad hoc*. In this work, the second approach is adopted: nothing is *a priori* known about data and resulting components describe a particular property, e.g. mean and variance in case of static model or a particular dynamic feature in case of dynamic model of the given order.

The proposed procedure is composed of two steps, *off-line* and *on-line*.

In off-line stage, historical fault-free learning data are processed to estimate an off-line transition table between the mixture components.

In on-line stage, the tested data are matched against the mixture components and on-line transition table is estimated. Compared to the off-line one, the tested data are declared either correct or faulty.

The paper is organized as follows: Section 2 describes methodology of data description and estimation of mixture parameters, Section 3 contains experiments with data matching and faults simulation.

2 METHODOLOGY

Throughout the paper, this notation is used: vectors are represented by columns, their elements are in text enumerated in brackets as $x = [x_1, x_2, \ldots, x_{\ell_x}]'$, where $'$ denotes transposition and ℓ_x represents length of the vector x. Symbol x^* denotes set of x-values, x_t is the value of x at discrete-time instant $t \in t^* \equiv \{1, 2, \ldots, T\}, T < +\infty$. Set of time labelled quantities x_t up to time t is denoted as $x(t) \equiv \{x_t, x_{t-1}, \ldots, x_1, x_0\}$, where x_0 represents prior value or expert knowledge, possibly empty, $f(x|s)$ is a probability (density) function (pdf) of x conditioned by s, random variable is not formally distinguished from its value. The point estimate (mean value) of x is denoted by $\hat{x}$.

2.1 Model of the Data

Data processing focuses on single *scalar* variable (data channel) without any consideration of explicit dependencies to other variables. We use autoregressive (AR) model, generally of m-th order, with normal noise, optionally with offset (absolute term).

The data sequence $\{y_1, y_2, \ldots, y_T\}$ (or simply data) is modelled by

$$y_t = \vartheta' \psi_t + e_t \tag{1}$$

with vector of constant but unknown regression coefficients ϑ and regression vector $\psi_t = [y_{t-1}, y_{t-2}, \ldots, y_{t-m}, 1]'$. If the trailing 1 is present, then offset is added to the model, if omitted, then the model is purely autoregressive. In case of $\psi_t = [1]'$, the model is static.

The term e_t represents a white normal noise. It is described by Gaussian pdf with zero mean and unknown but constant variance r, i.e. $f(e_t) = \mathcal{N}_{e_t}(0, r)$. It is related both to a noise on y_t and to imperfect model, with the assumptions mentioned above.

2.2 Gaussian Mixture

Data distribution is approximated by a Gaussian mixture (Kárný et al., 2005) with dynamic weights of components (Nagy et al., 2011). For its construction, it is assumed that each component represents one of n operating modes of the data generating system. The component, that is *active* in time instant t, is labelled by $c_t \in c^* \equiv \{1,\ldots,n\}$. In other words, the c_t-th component generates the data vector $\Psi_{c;t} \equiv \left\{y_t, \psi'_{c;t}\right\}'$ at time t.

Unlike in (1), the data vector $\psi_{c;t}$ is labelled by c because each component can be generally described by a model with different structure.

2.2.1 Component Model

Mixture model describes n different operating modes, c-th component describes behaviour in c-th mode, $c \in c^* = \{1,\ldots,n\}$. Its form is

$$f(y_t|c,y(t-1),\Theta_c) \equiv f(y_t|c,\psi_{c;t},\Theta_c) = = \mathcal{N}_{y_t}\left(\vartheta'_c\psi_{c;t}, r_c\right). \quad (2)$$

The parameter $\Theta = (\vartheta, r)$, where r is the noise variance.

The conjugate prior is the Gauss-inverse-Wishart pdf

$$f(\vartheta,r|y(t)) \equiv f(\vartheta,r|V,\nu) = = I(V,\nu)^{-1}\, r_c^{-\frac{1}{2}(\nu+\ell_\psi+2)} \times \times \exp\left\{-\tfrac{1}{2r}\,\mathrm{tr}\left([-1,\vartheta']\,V\,[-1,\vartheta']'\right)\right\}, \quad (3)$$

where $I(V,\nu)$ is normalization integral, V and ν are finite sufficient statistics, tr is a matrix trace, V is an extended information matrix—symmetric and positive definite matrix having the size of the extended regression vector Ψ. The statistic ν is a data counter—positive scalar. The term -1 appears as a unit coefficient by y_t of opposite sign to ϑ when expressing e_t using (1).

Indices c and t were omitted at ϑ, r, V, ν and ψ.

2.2.2 Pointer Model

The pointers c_t are assumed to evolve according to the model

$$f(c_t|c_{t-1},y(t-1),\alpha,\Theta) \equiv f(c_t|c_{t-1},\alpha) = \alpha_{c_t|c_{t-1}}, \quad (4)$$

where $\alpha_{c_t|c_{t-1}}$ are transition probabilities that the system will be in mode c_t in time t, if it was in mode c_{t-1} in time $t-1$. It holds for the stochastic matrix $\alpha_{ij} \equiv \alpha_{j|i}$

$$\alpha_{j|i} \geq 0,\ \sum_{j\in c^*}\alpha_{j|i} = 1,\ \forall i,j \in c^*. \quad (5)$$

The conjugate prior is the Dirichlet pdf

$$f(\alpha|y(t)) = B(\varkappa_t)^{-1}\prod_{c_t\in c^*}\prod_{c_{t-1}\in c^*}\alpha_{c_t|c_{t-1}}^{\varkappa_{c_t|c_{t-1};t}}, \quad (6)$$

where $\varkappa_t$ is $n\times n$ matrix of sufficient statistics and $B(\varkappa_t)$ is the normalization integral.

2.2.3 Mixture Model

Assuming known parameters Θ and α, the mixture model is a marginal pdf of the joint pdf of y_t, c_t and c_{t-1}

$$\begin{aligned} f(y_t|y(t-1),\alpha,\Theta) &= \\ &= \sum_{c_t\in c^*}\sum_{c_{t-1}\in c^*} f(y_t,c_t,c_{t-1}|y(t-1),\alpha,\Theta) = \\ &= \sum_{c_t\in c^*}\sum_{c_{t-1}\in c^*} f(y_t|c_t,y(t-1),\Theta_c)\times \\ &\quad \times f(c_t|c_{t-1},\alpha)\, f(c_{t-1}|y(t-1)). \end{aligned} \quad (7)$$

The chain rule $f(a,b|c) = f(a|b,c)f(b|c)$ and the marginalization rule for a discrete variable $f(a|c) = \sum_b f(a,b|c)$ were used here.

2.3 Mixture Estimation

In case of known parameters Θ and α, the formula (7) can be used directly.

In case of unknown parameters Θ and α, as a technical approximation used in this paper, the relation (7) can be used with substituted point estimates of the unknown parameters Θ and α. This is equivalent to $f(\Theta|y(t-1)) \approx \delta(\Theta-\hat{\Theta})$, where δ is Dirac distribution and $\hat{\Theta}$ is the point estimate of Θ (the same holds for α). This approach makes algorithms simpler and faster but artificially increases precision of the model by neglecting parameters' uncertainty (Nagy, 2014).

The fully Bayesian approach requests unknown parameters to be included into the joint pdf (7) as random variables and treated consistently. The theory and procedure is described in detail in (Nagy et al., 2011). However, it exhibits higher computational complexity and posterior pdfs of unknown parameters must be approximated anyway to preserve the prior forms (3) and (6).

2.3.1 Estimation of Regression Coeffcients

Let us assume availability of the statistics $V_{c;t-1}$, $\nu_{c;t-1}$ and $\varkappa_{t-1}$ from the previous time step. The extended information matrix V can be decomposed as $V = L'DL$ with

$$L = \begin{bmatrix} 1 & 0 \\ L_{y\psi} & L_\psi \end{bmatrix},\quad D = \begin{bmatrix} D_y & 0 \\ 0 & D_\psi \end{bmatrix}, \quad (8)$$

where L is a positive definite lower triangular matrix with unit diagonal and D is a diagonal matrix with positive entries.

Then it holds for parameter estimates

$$\hat{\vartheta} = L_{\psi}^{-1} L_{y\psi} \text{ and } \hat{r} = \frac{D_y}{\nu - 2}. \quad (9)$$

2.3.2 Estimation of Transition Probabilities

Point estimate of α (see (4)) equals

$$\hat{\alpha}_{c_t|c_{t-1}} = \frac{\varkappa_{c_t|c_{t-1};t-1}}{\sum_{c_t \in c^*} \varkappa_{c_t|c_{t-1};t-1}}. \quad (10)$$

The probability $w_{c_t|c_{t-1}}$ that the system was at time t in the mode c_t and at time $t-1$ in the mode c_{t-1}, with known (measured) output y_t, is

$$w_{c_t|c_{t-1}} = f(y_t|c_t, \psi_{c;t}, \hat{\Theta}_c)\ \hat{\alpha}_{c_t|c_{t-1};t}\ w_{c_{t-1}}, \quad (11)$$

where $f(y_t|c_t, \psi_{c;t}, \hat{\Theta}_c)$ is likelihood function for the estimation step (parametrised model of the data as a function of parameters with fixed data) and $w_{c_{t-1}}$ is unconditional probability that the system was at time $t-1$ in the mode c_{t-1}.

The probability w_{c_t} of mode c_t at time t is

$$w_{c_t} = \sum_{c_{t-1} \in c^*} w_{c_t|c_{t-1}} \equiv f(c_t|y(t)). \quad (12)$$

2.3.3 Update of Statistics

For models of components, the statistics of the c-th component are updated according to these formulae:

time update

$$V_{c;t|t-1} = \lambda_m V_{c;t-1} + (1-\lambda_m) V_A \quad (13)$$
$$\nu_{c;t|t-1} = \lambda_m \nu_{c;t-1} + (1-\lambda_m) \nu_A \quad (14)$$

data update

$$V_{c;t} = V_{c;t|t-1} + w_{c_t} \Psi_{c;t} \Psi'_{c;t}, \quad (15)$$
$$\nu_{c;t} = \nu_{c;t|t-1} + w_{c_t} \quad (16)$$

where λ_m is forgetting factor for components, $0 < \lambda_m \leq 1$, and V_A, ν_A are optional alternative statistics defined by the user to stabilize the update. The higher λ_m is, the more information from the previous data is kept in the statistics and the less shift of Θ in time is allowed with new data.

Prior matrix $V_{c;0}$ is chosen as diagonal with small positive values to guarantee regularity, prior $\nu_{c;0}$ should be small positive, too.

For model of pointers, the statistic $\varkappa$ is updated in the following way

time update

$$\varkappa_{c_t|c_{t-1};t|t-1} = \lambda_w \varkappa_{c_t|c_{t-1};t-1} + (1-\lambda_w) \varkappa_A \quad (17)$$

data update

$$\varkappa_{c_t|c_{t-1};t} = \varkappa_{c_t|c_{t-1};t|t-1} + w_{c_t|c_{t-1}} \quad (18)$$

where λ_w is forgetting factor for transition probabilities, $0 < \lambda_w \leq 1$, and $\varkappa_A$ stabilizing alternative matrix defined by the user.

All elements of prior matrix $\varkappa_{c_t|c_{t-1};0}$ can be chosen as a small positive constant.

2.3.4 Notes on Approximation

Except of the adopted approach of substituting point estimates of unknown parameters into conditions of pfds, there is one more issue to be pointed out.

The data update (18) is actually approximation in the situation when we are uncertain about the pointers c_{t-1} and c_t, which are substituted by their conditional probabilities $w_{c_t|c_{t-1}}$. This approach is called *quasi-Bayes* (QB) and consists in approximation of the Kronecker $\delta(c_t, c)$ by its mean value $E_c[\delta(c_t, c)] = w_{c_t}$ if the true c_t is unknown (Kárný et al., 2005).

Another possible approximation is to substitute the unknown pointers c_{t-1} and c_t by indices of components with maximum value of likelihood (ML) within the mixture, i.e.

$$c_t = \arg\max_{c \in c^*} f(y_t|c, \psi_{c;t}, \hat{\Theta}_c). \quad (19)$$

Then, only the corresponding element of $\varkappa_{c_t|c_{t-1};t}$ is incremented by 1 in (18). We tried both these approaches.

2.3.5 Notes on Estimation Algorithm

The computation starts with prior statistics $V_{c;0}$, $\nu_{c;0}$ and $\varkappa_{c_t|c_{t-1};0}$. Probabilities w_{c0} are chosen as uniform.

The time step begins with point estimation of parametes $\hat{\Theta}_c$ and $\hat{\alpha}_{c_t|c_{t-1}}$ using (9) and (10). Then, matrix $w_{c_t|c_{t-1}}$ is obtained by (11) and it is used to get unconditional probablilities w_{c_t} by (12) as weights for update of components' statistics in (15) and (16). Finally, the transition probability statistic is updated using conditional probability $w_{c_t|c_{t-1}}$ according to (17) and (18), t is incremented by 1 and new time step with a new data vector is started.

Even though all data are processed, the resulting mixture is usually not estimated satisfactorily after one pass (iteration) of the data sequence. It is recommended to perform multiple iterations in this way:

- store prior statistics $\nu_{c;0}$ and $\varkappa_{c_t|c_{t-1};0}$,
- estimate the mixture using all the data, obtain statistics $V_{c;T}$, $\nu_{c;T}$ and $\varkappa_{c_t|c_{t-1};T}$,
- calculate $\lambda_M = $ $= (\sum_{c \in c^*} \nu_{c;0} - n\nu_A)/(\sum_{c \in c^*} \nu_{c;T} - n\nu_A)$, where n is number of components,

- calculate $\lambda_W = \\ = \sum_{i,j\in c^*}\left(\varkappa_{j|i;0} - \varkappa_{A,ij}\right) / \sum_{i,j\in c^*}\left(\varkappa_{j|i;T} - \varkappa_{A,ij}\right)$,
- flatten the mixture using time updates (13), (14) and (17) using λ_M and λ_W,
- stabilizing terms (those with the alternative statistics) in the time updates are necessary for numerical reasons, e.g. $0 < \nu_A < \nu_0$, V_A low positive diagonal, $\varkappa_A$ a matrix filled with a positive constant, according to the estimation performance,
- replace prior statistics by the flattened posterior statistics $V_{c;0|T}$, $\nu_{c;0|T}$ and $\varkappa_{c_t|c_{t-1};0|T}$ and start a new iteration.

This procedure improves prior information for the new iteration which results in a better posterior estimate of the mixture. Iterations can be repeated until the model converges according to a chosen criterion (e.g. (20), for $\hat{\alpha}$s of two subsequent iterations, or other), usually $7\times$–$10\times$, according to the order of the model and number of estimated relevant components.

The number of components n is most often unknown in advance. For mixture estimation, we adopted this approach:

- choose initial number of components which is much higher than the expected final number (e.g. $n = 35$),
- place the components uniformly into the domain of $f(y_t|c, y(t-1), \Theta_c)$, choose small variance r_c (e.g. $(\max(y) - \min(y))/(100n)$), see (2),
- estimate the mixture using the description given above,
- optionally cancel (remove) insignificant components with lower weights, according to a chosen criterion, to achieve compromise between low number of components and satisfactory descriptiveness of the mixture model.

Numerically stable and fast square root algorithms, performing updates (13) and (15) directly on matrices L and D, where $V = L'DL$ (8), are available e.g. in (Kárný et al., 2005). They guarantee positive definiteness of the extended information matrix V and numerical manipulation with L and D is simpler and faster. Direct decomposition of V to L and D may be numerically unstable in some cases.

2.4 Comparison of Transition Tables

Having two matrices of statistics with the same dimensions, $\varkappa$ and $\tilde{\varkappa}$, we perform a simple comparison. First, we get the point estimates of transition tables $\hat{\alpha}$ and $\hat{\tilde{\alpha}}$ using (10), then calculate a value of the criterion

$$\rho = \frac{1}{n} \sum_{i,j\in c^*} \left|\hat{\alpha}_{j|i} - \hat{\tilde{\alpha}}_{j|i}\right| \tag{20}$$

and compare it against a chosen value $\bar{\rho} > 0$. If $\rho < \bar{\rho}$, the matrices match. Criterion ρ can reach values from 0 (exact match) to 2 (total mismatch).

3 EXPERIMENTS

Experiments were performed on industrial data from several cold rolling mills. As no faulty data were available at the moment, several types of faults were simulated by distortion of the data recorded in normal operating conditions.

The initial purpose was to demontrate influence of various situations on the criterion value ρ in (20). Therefore, any critical value $\bar{\rho}$ has not been proposed.

3.1 Estimation Procedure

Probabilistic mixture was estimated using data chosen as the reference data describing faultless operation. Then the components were fixed, i.e. parameters $\hat{\Theta}$ were kept constant. Using the same reference data, the algorithm was run once more (with several iterations), except that updates of the components' statistics V_c and ν_c (13), (14), (15) and (16) were disabled and the statistics were left intact. Only α was estimated. This parameter was denoted as the reference transition table α^r. Using the same procedure with the fixed components' parameters for a different data sequence, another transition table α was obtained, i.e. each data sequence was characterized by its own transition table generated by a particular mixture model.

As mentioned in part 2.3.4, quasi-Bayes (QB) or maximum likelihood (ML) approximation can be used in (18). With QB, convergence of $\hat{\alpha}$ was rather slow, whereas ML converged very rapidly within less than 10 iterations. Therefore, the ML approximation was used in computations.

Actually, after processing enough data from the sequence into sufficient statistics V, ν and $\varkappa$, both QB and ML approximations performed practically equivalently, because the likelihood function was narrow enough to single out one component, as in case of ML. However, the initial phase of the iteration, when likelihood assigns similar values to different components, slows the convergence down, even, in some cases, leads to infinite loops.

The forgetting factors were set as $\lambda_m = \lambda_w = 1$, i.e. estimation without forgetting, assuming constant parameters.

For number of components n, prior statistics were set to $V_{c,0} = 0.3n/\text{data\_range}$, $\nu_{c,0} = 25/n$, $\varkappa_{j|i;0} = 1/(2n^2)$. Alternative statistics were set to $V_\text{A} = \text{diag}(10^{-7})$, $\nu_\text{A} = 1/(4n)$, $\varkappa_{\text{A},j|i} = 1/(20n^2)$.

For experiments, we used data from metal cold rolling mill. Total rolling force was chosen for a particular material and pass as learning data of sample size 5 458. Data values were in the range $\langle 0,5\rangle$. We identified static component model ($m = 0$ + offset) and AR component model of 2nd order ($m = 2$ + offset). Static model converged with 14 and AR model with 7 significant components.

Histogram of the data is shown in Figure 1, the mixture identified with static component model is shown in Figure 2. Note that dynamic pointers are used even in case of static component model.

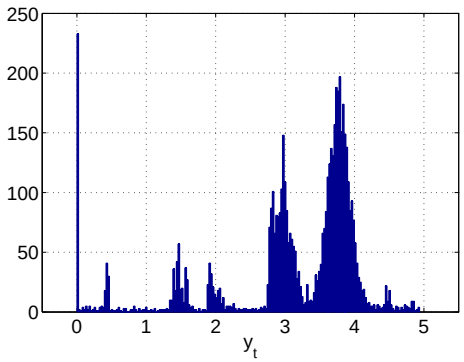

Figure 1: Histogram of learning data.

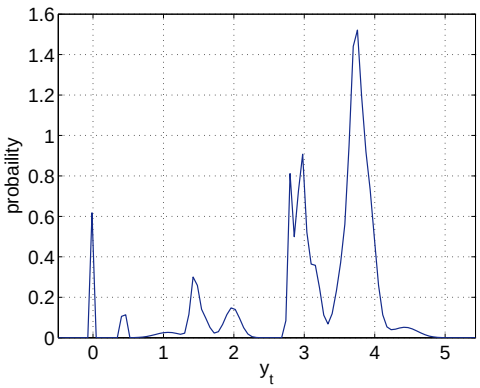

Figure 2: Static mixture describing the learning data.

3.2 Data Matching

First, learning data were, through their transition tables, matched against data describing the same quantity, total rolling force, on the same rolling mill and the same material during different metal strip passes (data 1). Then, similar quantity, upper rolling force, which is approximately half of the total force, was tested in the same conditions as above (data 2). Last, total rolling force measured on a different rolling mill with a different material was processed (data 3).

The data were compared using mixtures with both static and dynamic component models. The results for static mixture model are shown in Table 1, for dynamic mixture model in Table 2.

Table 1: Data matching with static model (see text for explanation).

	ρ_min	ρ_mean	ρ_max
data 1	0.064	0.198	0.331
data 2	0.709	0.993	1.110
data 3	0.814	1.086	1.223

Table 2: Data matching with dynamic model (see text for explanation).

	ρ_min	ρ_mean	ρ_max
data 1	0.122	0.165	0.243
data 2	0.312	0.341	0.362
data 3	1.025	1.118	1.254

3.3 Simulation of Faults

As no faulty data were available at the moment, the sensor faults had to be simulated. In all cases below, learning data y_t were modified to y_t^fault, $t \in t^*$, in a specified way. Such a pair of the data arrays, y_t and y_t^fault, was used to generate a pair of transition tables. This was done both for static model and for dynamic model. The pair of transition tables for each model was compared using (20). The criteria ρ_s (static model) and ρ_d (dynamic model) are shown in the tables.

3.3.1 Additive Noise

Data y_t were modified by additive Gaussian noise e_t,

$$\begin{aligned} y_t^\text{fault} &= y_t + e_t \\ e_t &\sim \mathcal{N}(0, r_\text{f}), \end{aligned}$$

where r_f is noise variance. Table 3 shows influence of r_f on data matching.

3.3.2 Additive Bias

Data y_t were modified by additive bias $b > 0$,

$$y_t^\text{fault} = y_t + b.$$

Table 4 shows influence of b on data matching.

Table 3: Additive noise.

r_f	ρ_s	ρ_d
10^{-5}	0.005	0.006
10^{-4}	0.034	0.032
10^{-3}	0.254	0.130
10^{-2}	0.738	0.724

Table 4: Additive bias.

b	ρ_s	ρ_d
0.01	0.012	0.009
0.05	0.223	0.022
0.1	0.358	0.318
1	0.557	0.641

3.3.3 Additive Drift

Data y_t were modified by additive drift d_t,

$$\begin{aligned} y_t^{\text{fault}} &= y_t + d_t, \\ d_t &= \frac{t}{T}\ d_{\max}. \end{aligned}$$

Table 5 shows influence of $d_{\max}$ on data matching.

Table 5: Additive drift.

$d_{\max}$	ρ_s	ρ_d
0.001	0.001	0.004
0.01	0.003	0.005
0.1	0.040	0.017
1	0.201	0.101
2	0.302	0.146

3.3.4 Block Dropout

Data y_t were modified by block dropout. The dropout occurs in the interval $\langle t_d, t_d + h\rangle$, where $h = T\frac{p}{100}$. Value p represents percentage of dropout related to the sample size. Value t_d was chosen randomly but kept constant. The data values within the dropout interval were set to zero.

Table 6 shows influence of p on data matching.

Table 6: Block dropout.

$p[\%]$	ρ_s	ρ_d
0.1	0.001	0.006
1	0.002	0.006
5	0.006	0.010
10	0.018	0.017
20	0.021	0.020

4 CONCLUSION

The paper presents a method for comparison of scalar data arrays. The data are described by a probabilistic Gaussian mixture with dynamic component pointers and represented as a corresponding transition table between the components, according to the given mixture model. It is a case of unsupervised transformation of data into a multidimensional feature applied for classification. Two component models were used: static component model (offset only) and dynamic component model (AR model of 2nd order with offset).

Matching of data sequences was quantified to demonstrate the sensitivity of the method to specific faults and situations. No classification of correct/faulty data was performed yet due to unconvincing results.

To represent faults, additive noise, bias and drift were simulated. As they demonstrated behaviour of the method, in our point of view, sufficiently, multiplicative faults were not included in the study although they occur in practice as well.

The data values were in the interval $\langle 0, 5\rangle$, which is obvious from Figures 1 and 2.

4.1 Performance of the Models

Static model is naturally more sensitive to data difference because it ignores data dynamics, which results in higher values of the criterion ρ and higher dispersion when processing multiple data sequences. This property can be seen in Tables 1 and 2, data 1 and 2. On the other hand, if the data dynamics is completely different from the one of the learning data, the dynamic model is more sensitive (with lower dispersion), see the same tables, data 3.

Sensitivity of static model to simulated faults is generally higher as well. The exception is the lowest values of noise and drift. In case of block dropouts, sensitivity depends in a more complex way on the droput size, which, however, both static and dynamic models reflect very mildly.

The sensitivity of the method is limited by lack of external information on data added to its construction. For specific data, which may be suitable for the method application using a specific model, this autonomous property can be of advantage. Dynamic component model of corresponding order would focus on dynamic properties rather than absolute scale. On the other hand, estimation success of dynamic component model is increased if transitions are rare.

It was observed that for a particular data sequence, usually one or two structures of dynamic models yielded a sufficient number of significant components. This property indicates that the method might "choose" the suitable dynamic features for the given data. However, it requires further testing.

4.2 Open Problems

Generic aiming and flexibility of the described method is balanced by its limited performance. Practical applicability of the method may be enhanced by considering several topics, among others

- assuming bounded noise rather than Gaussian,
- reconsidering advantages and disadvantages of dynamic component models,
- validating of the mixture by other means than convergence of its transition table,
- trying a better criterion for comparison (e.g. Kullback-Leibler divergence of Dirichlet pdfs, dynamic clustering in parameter space etc.)
- employing components of different dynamic order,

The QB approximation is more consistent with the Bayesian methodology used in this work than the ML one. Therefore, fixing the convergence issue in case of QB approximation is desirable.

As the method is unsupervised in the described phase, no critical value of $\bar{\rho}$ was set up. The question is how to make this set up methodically and generally. The method is based on occurrence of specific features with a specific set of data sequences and adequate model. Stability of these features within the set indicates data similarity. The question is whether these features can be extracted by the method or added externally by an expert.

The supervision is a crucial topic. It might improve the performance and set the critical value of $\bar{\rho}$ but, on the other hand, lead to tailored solutions with loss of intended generality.

Other models than AR or other technology could be taken into account as well.

Last but not least, the method is based on extracting information from data. Hence, a wider set of data, including those with real faults, should be studied.

ACKNOWLEDGEMENTS

This research has been supported by EUROSTARS Project 7D12004 E!7662.

REFERENCES

Boškoski, P., Gašperin, M., and Petelin, D. (2013). Bearing fault prognostics using Rényi entropy based features and Gaussian process models. *Mechanical Systems and Signal Processing*. Submitted.

Dedecius, K. and Jirsa, L. (2010). Bayesian Soft Sensing in Cold Sheet Rolling. In *Proc. of the 6th Int. Workshop on Data-Algorthms-Decision Making*. UTIA AV ČR.

Ettler, P. and Dedecius, K. (2013). Overview of bounded support distributions and methods for Bayesian treatment of industrial data. In *Proceedings of the 10th international conference on informatics in control, automation and robotics (ICINCO)*, pages 380–387, Reykjavik, Iceland.

Ettler, P., Juricic, D., Dedecius, K., and Preglej, A. (2011). Advanced soft sensor technology to be used for cold rolling mills. In *Emerging Technologies & Factory Automation (ETFA), 2011 IEEE 16th Conference on*, pages 1–4. IEEE.

Ettler, P., Puchr, I., and Dedecius, K. (2013). Bayesian model mixing for cold rolling mills: Test results. In Fikar, M. and Kvasnica, M., editors, *Proceedings of the 2013 International Conference on Process Control (PC)*, pages 359–364, Štrbské Pleso, Slovakia.

He, X., Wang, Z., and Zhou, D. H. (2008). State estimation for time-delay systems with probabilistic sensor gain reductions. *Asia-Pacific Journal of Chemical Engineering*, 3(6):712–716.

Isermann, R. (2011). *Fault Diagnosis Applications: Model Based Condition Monitoring, Actuators, Drives, Machinery, Plants, Sensors, and Fault-tolerant Systems*. Springer Verlag.

Jirsa, L., Pavelková, L., and Dedecius, K. (2013). Preliminaries of probabilistic hierarchical fault detection. In *Preprints of the 3rd International Workshop on Scalable Decision Making held in conjunction with ECML/PKDD 2013*, pages 1–14, Prague, Czech republic.

Kárný, M., Böhm, J., Guy, T. V., Jirsa, L., Nagy, I., Nedoma, P., and Tesař, L. (2005). *Optimized Bayesian Dynamic Advising: Theory and Algorithms*. Springer, London.

Lu, F., Huang, J., and Xing, Y. (2012). Fault diagnostics for turbo-shaft engine sensors based on a simplified on-board model. *Sensors*, 12(8):11061–11076.

Nagy, I. (2014). *Advanced statistical methods and their applications*. ČVUT FD, Prague. http://www.fd.cvut.cz/personal/nagyivan/PhDLoad/LecturesPhD.pdf, In Czech.

Nagy, I., Suzdaleva, E., Kárný, M., and Mlynářová, T. (2011). Bayesian estimation of dynamic finite mixtures. *Int. Journal of Adaptive Control and Signal Processing*, 25(9):765–787.

Rajaman, R. and Ganguli, A. (2004). Sensor fault diagnostics for a class of non-linear systems using linear matrix inequalities. *International Journal of Control*, 77(10):920–930.

Walambe, R., Gupta, N., Bhange, N., Ananthkrishnan, N., Choi, J., Park, I., and Yoon, H. (2010). Novel redun-

dant sensor fault detection and accommodation algorithm for an air-breathing combustion system and its real-time implementation. *Defence Science Journal*, 60(1).

Wang, G., Liu, C., Cui, Y., and Feng, X. (2014a). Tool wear monitoring based on cointegration modelling of multisensory information. *International Journal of Computer Integrated Manufacturing*, 27(5):479–487.

Wang, G., Yang, Y., Zhang, Y., and Xie, Q. (2014b). Vibration sensor based tool condition monitoring using ν support vector machine and locality preserving projection. *Sensors and Actuators A: Physical*, 209(0):24 – 32.

Wei, X. and Yingqing, G. (2009). Aircraft engine sensor fault diagnostics based on estimation of engine's health degradation. *Chinese Journal of Aeronautics*, 22(1):18 – 21.