An improved segmentation-based HMM learning method for Condition-based Maintenance

T Liu[1,2]**, J Lemeire**[1,2]**, F Cartella**[1,2] **and S Meganck**[1,3]
[1] ETRO Dept., Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussels, Belgium
[2] FMI Dept., Institute for Broadband Technology - IBBT, Gaston Crommenlaan 8 (box 102), B-9050 Ghent, Belgium
[3] COMO Lab., Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussels, Belgium

E-mail: tingting.liu@vub.ac.be, jan.lemeire@vub.ac.be, francesco.cartella@vub.ac.be, stijn.meganck@vub.ac.be

Abstract. In the domain of condition-based maintenance (CBM), persistence of machine states is a valid assumption. Based on this assumption, we present an improved Hidden Markov Model (HMM) learning algorithm for the assessment of equipment states. By a good estimation of initial parameters, more accurate learning can be achieved than by regular HMM learning methods which start with randomly chosen initial parameters. It is also better in avoiding getting trapped in local maxima. The data is segmented with a change-point analysis method which uses a combination of cumulative sum charts (CUSUM) and bootstrapping techniques. The method determines a confidence level that a state change happens. After the data is segmented, in order to label and combine the segments corresponding to the same states, a clustering technique is used based on a low-pass filter or root mean square (RMS) values of the features. The segments with their labelled hidden state are taken as 'evidence' to estimate the parameters of an HMM. Then, the estimated parameters are served as initial parameters for the traditional Baum-Welch (BW) learning algorithms, which are used to improve the parameters and train the model. Experiments on simulated and real data demonstrate that both performance and convergence speed is improved.

1. Introduction

In modern industry, the demands of low cost, high reliability and human safety are highly increasing, therefore effective maintenance strategies to increase profitability and competitiveness play an important role in industry. Condition Based-Maintenance (CBM) is a decision-making strategy based on real-time diagnosis of impending failures and prognosis of future equipment health [1]. Diagnostics is an assessment of the current status of a system. It detects errors and faults based on the observed abnormality of the system. Prognostics deal with the fault and degradation prediction before they occur. Diagnostics and prognostics can be performed separately, however a combination of both tasks can decrease costs and improve efficiency and accuracy of the results, therefore diagnostics can be used as a basis and pre-processing step for prognostics in hybrid approaches.

In this paper, we focus on data driven diagnostics which aim at transferring the data collected by the sensors into relevant models. A statistical approach - Hidden Markov Model (HMM) has been successfully applied to several applications in both academic and engineering fields [2]. Most often, an HMM is built and trained with randomly initialized parameters, which increases the risks of inaccuracy by getting trapped into local maxima. The proposed method estimates initial parameters of HMM which are assumed close to the optimal ones.

By the nature of industrial machines and equipment, the health states can be safely assumed to remain stable (i.e. persistent) during the whole life time, whereas in other application domains of HMM, states can fluctuate quite heavily. In this paper, we focus on persistent states. Based on this assumption, the signal can be split into several large segments by a segmentation algorithm, where

each segment corresponds to a regime with relatively stationary behaviour. In general, the states are differentiated by the differences of their observations, i.e., $P(O|S_i) \neq P(O|S_j)$, i and j represent different states. In order to identify the different regimes based on their different behaviours, i.e., determine changes of regimes, change detection technique is required. The Student t-test [3] is a simple parametric technique to detect a signal change, but it only addresses the changes of the mean value. Regression analysis [4] can also be applied for change detection but it is difficult to deal with small deviations. Bayesian analysis [5] is an option as well however it requires the mathematical model of the data known in advance. All the assumptions made by the parametric techniques cannot be always met in real applications. Therefore, the non-parametric techniques such as Mann-Kendall [6] and cumulative sum charts (CUSUM) [7] are preferable and more suitable in sequential analysis [8]. Mann-Kendall technique evaluates the sign of all pairwise differences of observed values and is widely used in climate change analysis, while CUSUM detects systematic changes over time and has the advantages of its simplicity, the graphical interpretation of results and is able to detect abnormal patterns, etc [8]. CUSUM has been successfully used in fault detection and change detection in mechanical systems [7]. Here we use change-point analysis method proposed by Taylor [9] which uses a combination of CUSUM and bootstrapping techniques. It detects change-points when a change occurs in a signal, where the change-points can be seen as state transition points. Based on this method, the observations can be segmented at the detected change-points. Moreover, in order to combine the homogenous segments, clustering algorithms are used on the low-pass filtering or root mean square (RMS) values of the feature data. As a result, each segment is clustered and labelled, representing a different hidden state. These learned hidden states can be used to compute the initial parameters of an HMM model.

The remainder of the paper is as follows. Section 2 gives an overview of the basic theories of HMM and the problems with the learning method. Section 3 shows the basic ideas of the proposed method and summarizes the approach methodology of this paper. Section 4 shows the details of the algorithm of change-point analysis and the estimation of initial parameters. Simulation and experimental results are illustrated and analysed in Section 5 and Section 6. In the end, conclusions are given in Section 7.

2. Hidden Markov Model

2.1. Theory

A Hidden Markov Model (HMM) is a doubly stochastic process. The underlying process is characterized by a Markov chain and unobservable (hidden) but can be observed through another stochastic process which emits the sequence of observations. An example of an HMM is shown in shown in figure 1.

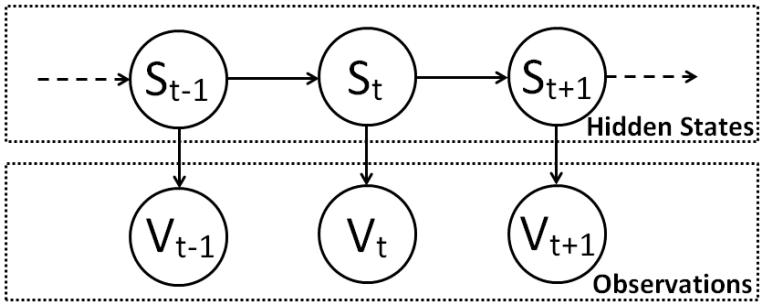

Figure 1. A Hidden Markov Model.

An HMM is described by the following parameters [2]:

- N, the number of states in the model. The individual states are denoted as $S = \{S_1, S_2, \ldots, S_N\}$;

- M , the number of distinct observation symbols per states. The individual symbols are denoted as $V = \{V_1, V_2, \ldots, V_M\}$;
- $A = \{a_{ij}\}$ the state transition probability distribution, where

$$a_{ij} = P(q_{t+1} = S_j \mid q_t = S_i), 1 \le i, j \le N. \tag{1}$$

- $B = \{b_j(V_k)\}$, the observation symbol probability distribution in state S_j, where

$$b_j(V_k) = P(o_t = V_k \mid q_t = S_j), 1 \le j \le N, 1 \le k \le M. \tag{2}$$

- $\pi = \{\pi_i\}$, the initial state distribution, where

$$\pi_i = P(q_1 = S_i), 1 \le i \le N. \tag{3}$$

where o_t and q_t are the observation and state at time t respectively. Let $\{o_1, o_2, \ldots, o_T\}$ denote a sequence of all observation symbols up to time T where an observed point o_t is taken at time t and $o_t \in V (1 \le t \le T)$. The actual state sequence up to time T is denoted by $\{q_1, q_2, \ldots, q_T\}$, where a state $q_t \in S (1 \le t \le T)$. For convenience, a compact notation $\lambda = (A, B, \pi)$ is used to represent an HMM model. Various types of HMMs exist, and in this paper, only ergodic HMMs with discrete symbol observations are considered.

In real-world applications, three basic problems related to HMMs have been identified and solved [2].

- *Evaluation*: Given an HMM λ and a sequence of observations $O = \{o_1, o_2, \ldots, o_T\}$, what is the probability $P(O \mid \lambda)$ that the observations are generated by the model?
- *Decoding*: Given an HMM λ and a sequence of observations $O = \{o_1, o_2, \ldots, o_T\}$, what is the most likely sequence of states $Q = \{q_1, q_2, \ldots, q_T\}$, in the model that produced the observations?
- *Learning*: Given a model and a sequence of observations, how to adjust the parameters $\lambda = (A, B, \pi)$ in order to maximize the probability of the observations given this model $P(O \mid \lambda)$?

For the learning problem, there is no analytical solution; however, a locally maximized parameter can be achieved with an iterative procedure such as the Baum-Welch (BW) method (i.e. Expectation-Maximization algorithm). The probability of being in state S_i at time t and the state S_j at time $t+1$ given the model and observation sequence is defined by [2]:

$$\begin{aligned} \xi_t(i, j) &= P(q_t = S_i, q_{t+1} = S_j \mid O, \lambda) \\ &= \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{P(O \mid \lambda)} \\ &= \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)} \end{aligned} \tag{4}$$

Where

$$\alpha_t(j) = \sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} b_j(o_t) \tag{5}$$

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(o_{t+1}) \beta_{t+1}(j) \tag{6}$$

Thus, the probability of being in state S_i at time t is:

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i, j) \tag{7}$$

The expected number of transitions from S_i is $\sum_{t=1}^{T-1} \gamma_t(i)$; the expected number of transitions from S_i to S_j is $\sum_{t=1}^{T-1} \xi_t(i, j)$. The re-estimated parameters of an HMM $\bar{\lambda}$ can be calculated by formulas as below:

$$\bar{\pi}_i = \gamma_1(i) \tag{8}$$

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \tag{9}$$

$$\bar{b}_j(V_k) = \frac{\sum_{t=1}^{T} \gamma_t(i) \delta(o_t, V_k)}{\sum_{t=1}^{T} \gamma_t(i)} \tag{10}$$

where

$$\xi_t(j) = \max_{1 \le i \le N} [\delta_{t-1}(i) a_{ij}] b_j(o_t), 2 \le t \le T, 1 \le j \le N \tag{11}$$

2.2. Failure of learning method

The idea of the BW learning method is that the initial parameters are randomly generated at first and then the parameters are iteratively re-estimated as long as the new model has a better log-likelihood than the previous one, i.e., $P(O \mid \bar{\lambda}) \ge P(O \mid \lambda)$, until it converges. Rabiner [2] pointed out that it has the risk that the log-likelihood converges to local maxima, which is shown in figure 2. The learning is repeated 10 times, and each time random initial parameters are chosen.

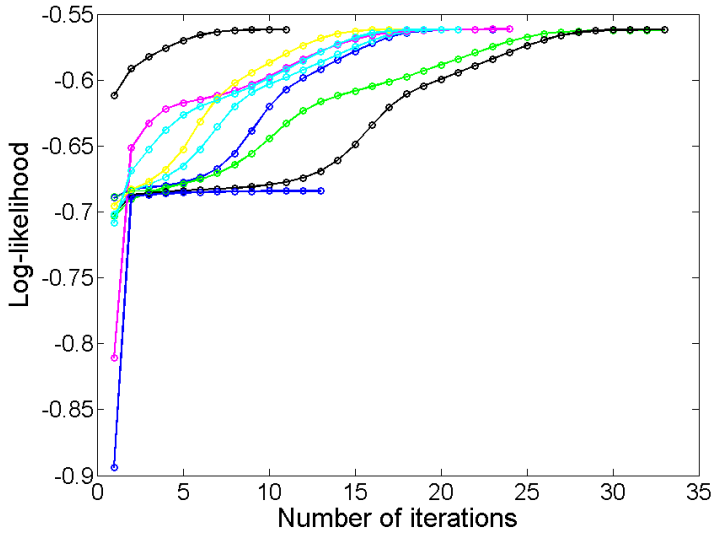

Figure 2. Convergence of log-likelihood values.

In this experiment, the training data consisted of 10 sequences of 1000 observations generated by a randomly generated HMM. The parameters of the generated HMM model are shown in table 1.

Table 1. Parameters of the generated HMM model.

Initial probabilities	Transition probabilities	Observation probabilities
$\pi = \begin{bmatrix} 0.33 & 0.33 & 0.33 \end{bmatrix}$	$A = \begin{bmatrix} 0.990 & 0.010 & 0 \\ 0.005 & 0.990 & 0.005 \\ 0.001 & 0.001 & 0.998 \end{bmatrix}$	$B = \begin{bmatrix} 0.950 & 0.050 \\ 0.478 & 0.522 \\ 0.244 & 0.756 \end{bmatrix}$

The data set is always the same one for each run, however due to the randomness of the initial parameters generated by BW learning algorithm; the results are different each time. Some runs start with better initial parameters and end earlier, and some start with bad parameters which end later. In figure 2, there is one run which falls into local maxima with a log-likelihood around -0.68, while other 9 runs reach a better result around -0.55.

3. Overview of Methodology

3.1. Assumptions

The assumptions made by the proposed method are given below:

- The HMM model is assumed to be an ergodic HMM, meaning that the hidden states are fully-connected, in other words, each state can stay at the same state or transfer to any other state at the next time stamp. The ergodic HMM allows revisiting previous states. This is a proper assumption when dealing with machines or equipment which can be repaired during their lifetime.
- The observations data are discrete but non-categorical.
- States are assumed to be persistent (i.e., $P(S_t = i \mid S_{t+1} = i)$ is relatively high), instead of flipping frequently. This assumption is reasonable for real industry machines or equipment which is generally consistent and keeps at one state for a while. This property will be exploited to identify the regimes that correspond to hidden states.
- The number of states is assumed to be known, for our test we use 3 states, which stands for "normal", "bad" and "serious damage" conditions of machines. The proposed method works for an arbitrary number of states.

3.2. Basic idea: recognizing regimes

As mentioned in the previous section, each hidden state of an HMM model is assumed to be persistent (i.e. self-transition probability $P(S_t = i \mid S_{t+1} = i)$ is higher than a threshold, for example, 0.9) in this paper, therefore, the signal can be split into multiple large regimes. In order to recognize various regimes, one way is to detect where the changes occur (i.e., $P(O \mid S_i) - P(O \mid S_j) > \beta$), where i and j represent different states and β is a threshold, meaning the observation distributions change when moving from one state to another). Segmentation techniques are used to detect such changes in behaviour, where each segment is supposed to belong to one state. Moreover, in order to combine the regimes that belong to the same states, a clustering method is applied in order to group the regimes that belong to the same state. Finally, the learned path of regimes can be seen as a hidden state path, which can be further used to learn the parameters of an HMM model. Figure 2 already shows that better initial parameters will converge faster to the optimal result, therefore, our method which learns

better initial parameters will improve the convergence speed of the method and likely also avoids local maxima.

3.3. Scheme

With the above idea, the learned parameters can be used as initial parameters of BW learning algorithm to avoid the faults caused by starting with random parameters. The intention of this pre-processing step is to optimize the BW learning and lower the probability of getting stuck in local maxima. The whole assessment scheme is shown in figure 3.

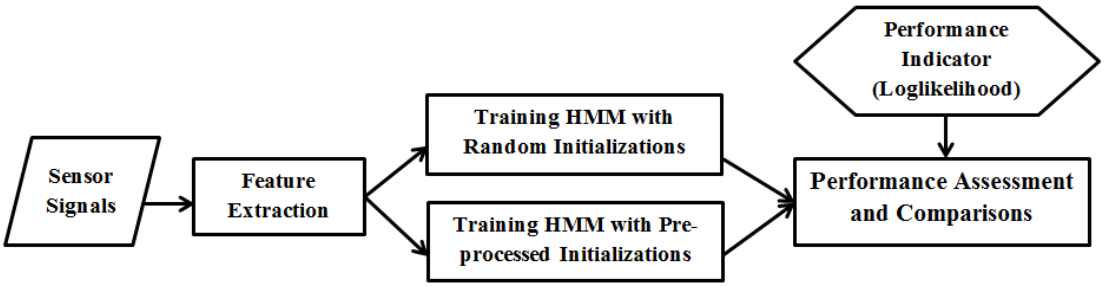

Figure 3. Scheme of performance assessment.

Firstly, useful features of the sensors' signals are extracted. These features are not only used as training and testing data, but also applied for the parameters estimation.

The selection of the initial parameters follows the steps as below:

- **Segmentation**: uses Taylor [9] change point analysis to determine whether a change has taken place, then segment the observation data where there is a change.
- **Feature extraction**: mean values of the low-pass filtered values or the RMS values inside each segment are used as the features to base the clustering on.
- **Clustering**: segments with K-means, thus each cluster represents a hidden state.
- **Learn parameters**: learn the parameters of the HMM defined by the number of states resulted from the clustering algorithm.

Finally, a comparison between HMM models with and without random initializations are tested and compared with log-likelihood as an indicator of the performance.

4. Algorithm details

4.1. Change-point detection

Taylor [9] proposed a change-point analysis method which is capable of detecting multiple changes. It iteratively uses a combination of cumulative sum charts (CUSUM) and bootstrapping techniques to detect the changes.

CUSUM is defined as: let $X_1, X_2, \ldots, X_N$ represent the N data points and $S_0, S_1, \ldots, S_N$ be the cumulative sums. Let $\overline{X}$ be the average values of the whole data set, and the start of the cumulative sum at zero be S_0. The cumulative sums are calculated by adding the difference between the current value and the average of the previous sum, which is $S_i = S_{i-1} + (X_i - \overline{X})$, where $i = 1, 2, \ldots, N$. A sudden turn (i.e. peak) in CUSUM chart indicates a change in the data sets, and bootstrap analysis is used to determine the change (or inflexion) with a confidence level.

An estimator of the magnitude of the change is defined by the difference between the maximum and minimum values of the CUSUM results, i.e., $S_{diff} = S_{\max} - S_{\min}$. Let $X_1', X_2', \ldots, X_N'$ be a bootstrap sample which is generated by reordering the original values. Let $S_0', S_1', \ldots, S_N'$ represent the

CUSUM of the bootstrap. The difference of the bootstrap CUSUM is denoted by $S_{diff}^{'} = S_{\max}^{'} - S_{\min}^{'}$. A large number of bootstraps are performed and let X denote the number of bootstraps for which $S_{diff}^{'} < S_{diff}$. The confidence level is calculated by $100 * \frac{X}{N}\%$, and typically 90% and 95% confidence is required.

To determine when the change occurred quantitively, two estimators proposed by Taylor [9], which are:

- **CUSUM estimator:**

$$|S_m| = \max_{i=0,\ldots N} |S_i| \tag{12}$$

where S_m is the point furthest from zero in the CUSUM chart.

- **Mean square error (MSE) estimator:**

$$MSE(m) = \sum_{i=1}^{m} (X_i - \overline{X_1})^2 + \sum_{i=m+1}^{N} (X_i - \overline{X_2})^2 \tag{13}$$

After a change is detected, the data set is split into two segments where on both sides, the same procedure is repeated. As a result, multiple changes are detected and the data set is divided into several segments. An example is shown in figure 4.

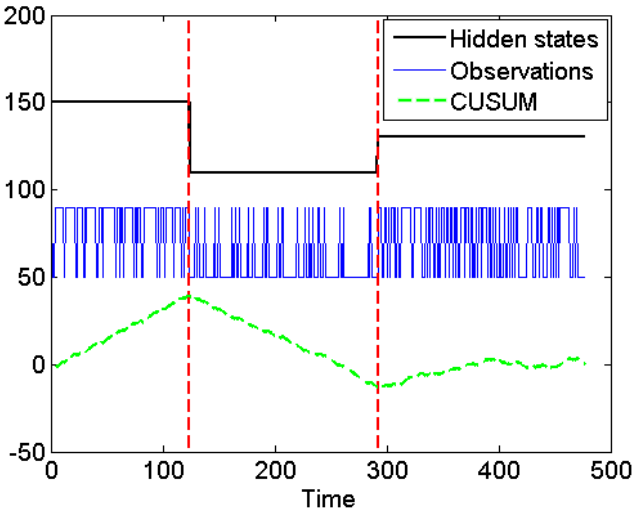

Figure 4. CUSUM based change-point detection

In figure 4, the solid line on the top represents the hidden states. The solid line in the middle represents the binary observations which have different probabilities within various regimes. In this case, the values are weighted values in order to show more clearly in the graph. The dotted line at the bottom is the CUSUM chart of the observations. A sudden turn of the CUSUM means that a change of regimes occurs. The red dotted vertical lines are the changes detected by change-point analysis method.

4.2. Clustering

After the observations are segmented, k-means clustering is applied to combine and label each segment. In the test, function $kmeans(X, k)$ of Matlab is used, where X is the input matrix and k is the number of pre-defined clusters. These segments are assumed to be the hidden states of an HMM, based on which the initial parameters can be calculated by simple counting method. K-means

clustering technique is used here to label the states to identify similar states. An example of applying a k-means clustering with 3 states is shown in figure 5.

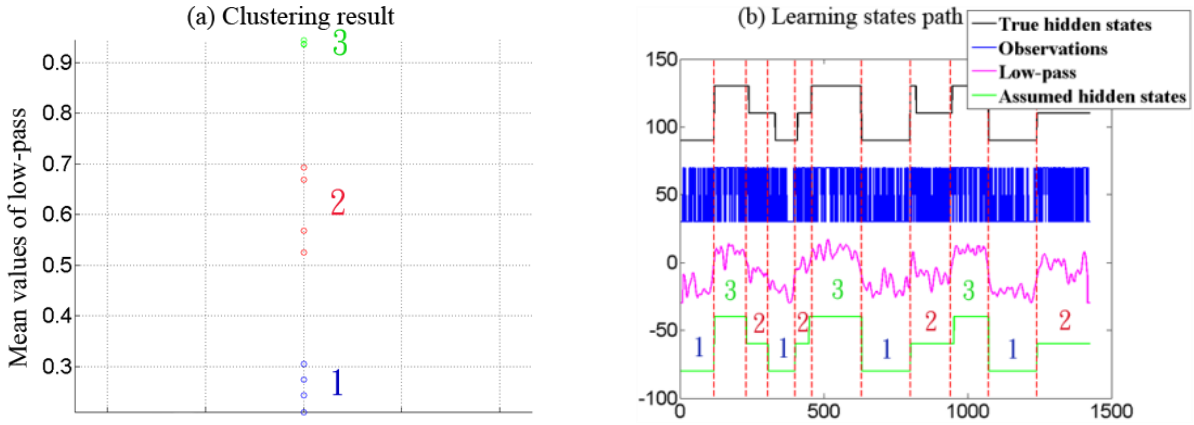

Figure 5. K-means clustering with 3 states.

The data set is generated with one sequence of 1426 data points, with 3 states and binary observations. Here in figure 5(b) the data was plotted with weighted values in order to be seen clearly. Based on the mean values of the low-pass filtering of each segment, K-means clustering is applied. The resultant 3 clusters are shown in figure 5(a). This clustering result is used to label each segment, resulting a learned (assumed) path of the hidden states shown at the bottom of figure 5(b).

Note that with general K-means clustering the initial parameters are randomly generated which might lead to incorrect clustering and gives completely wrong clusters. For example, the top two nodes in the cluster 2 in the middle are sometimes wrongly clustered into cluster 3 on the top. In the proposed method, one extra step is added in the K-means clustering to check the results. If the maximum distance inside each cluster is larger than twice the minimum distance between clusters, then we redo the K-means clustering from the beginning. This check step is repeated a maximum number of times, here we used maximum 5 repetitions. Experiments show that each time the maximum distance inside each cluster is smaller than the minimum distance between clusters; therefore, using twice the minimum distance for checking is not critical in our test.

4.3. Estimation of initial parameters

HMM parameters (i.e. probability matrices) can be calculated by simple counting. Let S_i represents the current state, S_j represents the next state. Let S represent the set of all states and O the set of observations. The symbol $\#(*)$ represents the number of $*$. The initial parameters can be calculated as below:

Initial state distribution:

$$\pi = \left[\frac{\#(S_1 = 1)}{\#(S_1)} \quad \frac{\#(S_1 = 2)}{\#(S_1)} \quad \frac{\#(S_1 = 3)}{\#(S_1)} \right] \tag{14}$$

Transition probability distribution:

$$A = \begin{bmatrix} \frac{\#(S_j = 1 \mid S_i = 1)}{\#(S = 1)} & \frac{\#(S_j = 2 \mid S_i = 1)}{\#(S = 1)} & \frac{\#(S_j = 3 \mid S_i = 1)}{\#(S = 1)} \\ \frac{\#(S_j = 1 \mid S_i = 2)}{\#(S = 2)} & \frac{\#(S_j = 2 \mid S_i = 2)}{\#(S = 2)} & \frac{\#(S_j = 3 \mid S_i = 2)}{\#(S = 2)} \\ \frac{\#(S_j = 1 \mid S_i = 3)}{\#(S = 3)} & \frac{\#(S_j = 2 \mid S_i = 3)}{\#(S = 3)} & \frac{\#(S_j = 3 \mid S_i = 3)}{\#(S = 3)} \end{bmatrix} \tag{15}$$

Observation probability distribution:

$$B=\begin{bmatrix} \frac{\#(O=1\mid S_i=1)}{\#(S=1)} & \frac{\#(O=2\mid S_i=1)}{\#(S=1)} & \frac{\#(O=3\mid S_i=1)}{\#(S=1)} \\ \frac{\#(O=1\mid S_i=2)}{\#(S=2)} & \frac{\#(O=2\mid S_i=2)}{\#(S=2)} & \frac{\#(O=3\mid S_i=2)}{\#(S=2)} \\ \frac{\#(O=1\mid S_i=3)}{\#(S=3)} & \frac{\#(O=2\mid S_i=3)}{\#(S=3)} & \frac{\#(O=3\mid S_i=3)}{\#(S=3)} \end{bmatrix} \tag{16}$$

5. Simulation study

Simulated signals are generated by randomly-generated HMM models and used to evaluate and compare the assessment performances of HMM models with random and non-random initial parameters. For simplicity, the number of states are set to be three and the observation data are generated with binary data (i.e., only 0s and 1s). To make sure the states are stable, we predefine the range of self-transition probabilities are above 0.95. Twenty sequences of 500 observations are generated as training data by sampling the HMM. The parameters of the generated HMM model is shown in table 2:

Table 2. Parameters of the generated HMM model.

Initial probabilities	Transition probabilities	Observation probabilities
$\pi=\begin{bmatrix}0.33 & 0.33 & 0.33\end{bmatrix}$	$A=\begin{bmatrix}0.97 & 0.02 & 0.01\\ 0.00 & 0.99 & 0.01\\ 0.03 & 0.01 & 0.96\end{bmatrix}$	$B=\begin{bmatrix}0.76 & 0.24\\ 0.46 & 0.54\\ 0.43 & 0.57\end{bmatrix}$

To learn the data set and compare the results with both the traditional HMM learning and segmentation-based HMM method, the test was repeated 5 times, the results are shown in figure 6:

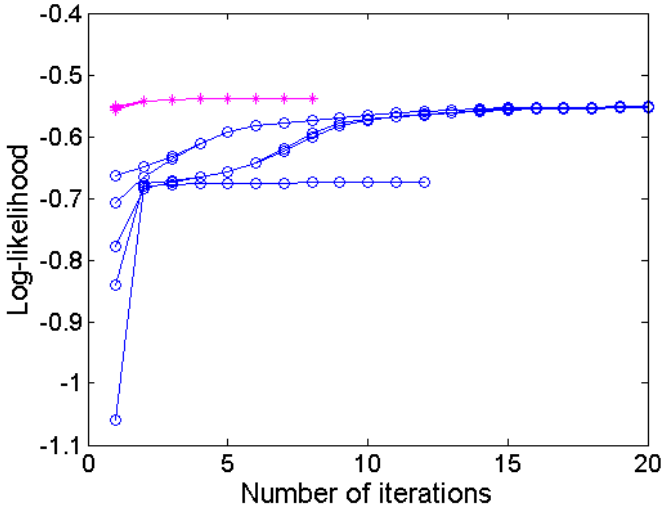

Figure 6. Comparison of the learning methods.

In figure 6, the log-likelihood of the 5 runs using segmentation-based initialization are plotted with asterisks, while the traditional HMM learning are plotted with circles. The number of repetitions and log-likelihood values are shown in table 3.

The results show that in general segmentation-based HMM learning has a better log-likelihood and needs fewer iteration steps than randomly initialized one. Moreover, there is one within the randomly initialized HMM learning which is trapped into a local maxima of -0.6746 within 5 steps.

Table 3. Comparisons of log-likelihood and number of repetitions.

Repetitions	1	2	3	4	5
Log-likelihood HMM	-0.5522	-0.5523	-0.6746	-0.5535	-0.5510
Log-likelihood Seg-HMM	-0.5389	-0.5389	-0.5389	-0.5389	-0.5389
Log-likelihood Difference	-0.0133	-0.0135	-0.1357	-0.0147	-0.0121
Num Iterations HMM	20	20	12	20	20
Num Iterations Seg-HMM	8	8	7	8	7
Num Iterations Difference	12	12	5	12	13

To further compare the results, a similar test is conducted with varying amount of data sizes and random parameters. The amounts of sequences are set from 10 to 50 with an increasing step of 10, and within each sequence, the sizes of the data points are set from 100 to 500 with an increasing step of 100. Therefore, in total 25 data sets are generated randomly with 3 states and binary observations. Figure 7 shows the comparison of the results with both methods:

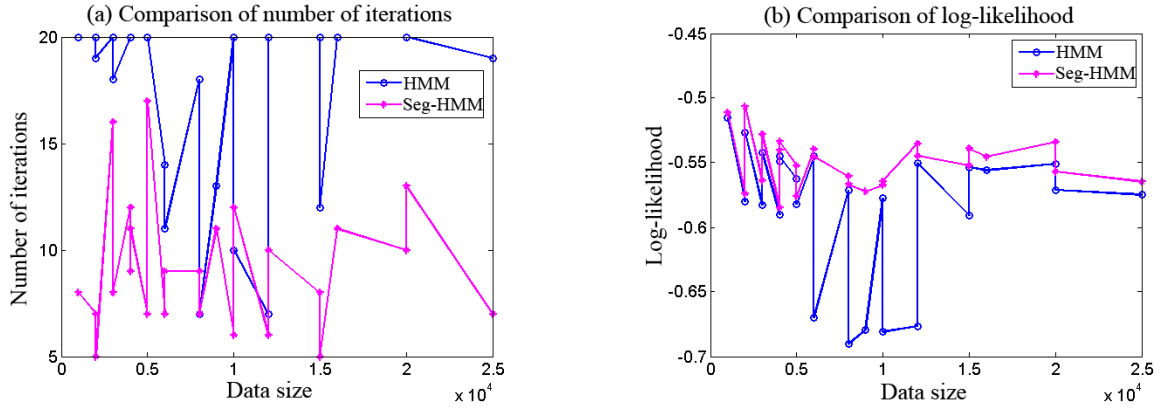

Figure 7. Comparisons with varying amount of data sizes.

From figure 7, we can see that in general segmentation-based HMM uses fewer number of iterations and has better log-likelihood values. The average values of the number of iterations and log-likelihood are calculated and compared in the table below:

Table 4. Comparisons of log-likelihood and number of repetitions.

Methods	**HMM**	**Seg-HMM**	**Differences**
Log-likelihood	-0.5846	-0.5503	-0.0343
Num of Iterations	17.12	9.24	7.88

6. Experimental study

The proposed method was trained and tested on the bearing data of NASA [10]. Four bearings were installed on one shaft. The angular velocity was kept constant at 2000 rpm and a 6000 lb radial load

was applied onto the shaft and bearings (figure 8). On each bearing two accelerometers (one horizontal X and one vertical Y) were installed for a total of 8 accelerometers to register the accelerations generated by the vibrations, where the sampling rate was fixed at 20 kHz [11].

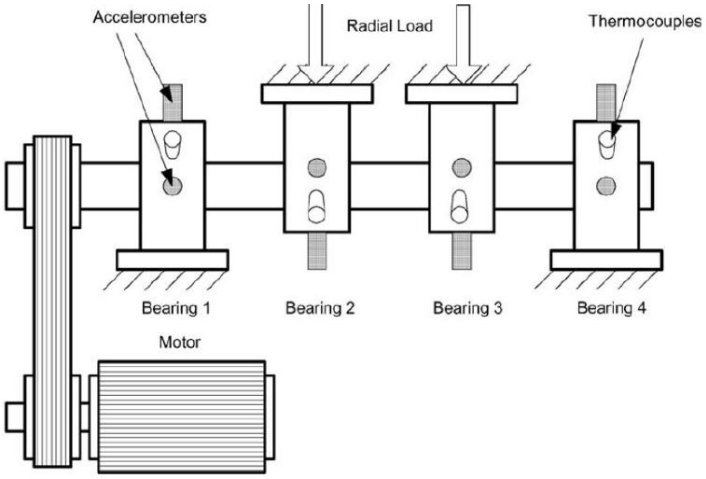

Figure 8. Bearing test rig and sensor placement illustration [12].

Bearing 3 in test 1 is considered failed at the end of its associated history. In the test, four condition monitoring data histories related to bearing 2 and 3 in test 1 are used as training set. The monitoring history related to bearing 3 in test 1 is used as test set. The test result is shown in figure 9. To transform the continuous signal into discrete one, the vibrations histories are processed with root mean square values (figure 9(a)) and quantized (i.e. rounded) to the nearest integer (figure 9(b)). Quantized data are then used as observations, based on which both traditional HMM learning and segmentation-based HMM learning methods are applied to learn the hidden states, shown in figure 9(c) and figure 9(d) respectively. Figure 9(c) shows a frequently fluctuated path while figure 9(d) shows a rather persistent path. Note that for all models the numbers of the states are assumed to be three.

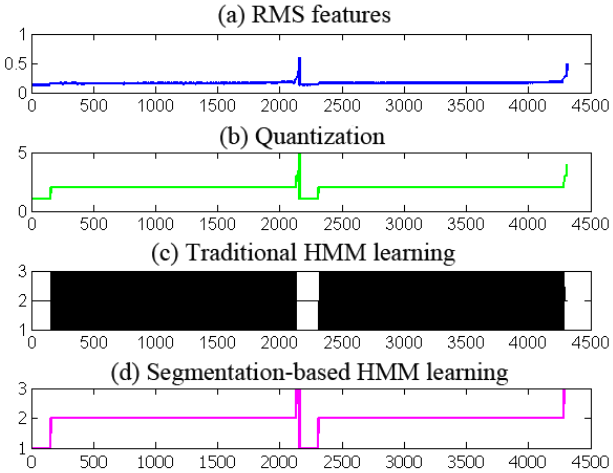

Figure 9. Bearing performance assessment.

Traditional HMM learning uses 8 iterations and achieves a log-likelihood of -0.0443, while segmentation-based HMM learning uses 4 iterations and gets a log-likelihood of -0.0135. This shows that the proposed method has a better log-likelihood and a relatively faster speed for learning.

7. Conclusion

The proposed method contains the following three steps: 1. estimation of initial parameters (with segmentation and clustering techniques), 2. learning and 3. evaluation. Firstly, to exact useful features from the data obtained from the sensors, change-point analysis method which uses a combination of cumulative sum charts (CUSUM) and bootstrapping techniques is applied for data segmentation. It determines with a confidence level that a state change happened instead of an accidental change in behavior. The resultant segments can be seen as corresponding to different states of the equipment. To label and combine segments belonging to the same state, a clustering algorithm is used on the low-pass filtering or root mean square (RMS) values of the feature data. These features related to their hidden states are taken as "evidence" to estimate the parameters of an HMM. Moreover, the estimated parameters are served as initial input model parameters for the traditional Baum-Welch (BW) learning algorithm, which is used to re-estimate parameters and train the model. In the last step, the learned model is used to assess the equipment conditions for further maintenance. The benefits of this approach is that it pre-processes the data with an intelligent but simple way to learn initial parameters, which is more accurate than regular BW learning methods with randomly generated initial parameters. Log-likelihood values are used to describe the diagnostic performances of the models and compare the models. Moreover, the extracted features, on which the learning can be based on, provide evidence for the learned model while traditional learning is only based on a global score.

We have shown that for systems with persistent states, segmentation of the observations is possible by change-point detection since systems behavior changes from one state to another. Based on segmentation, initial parameters can be estimated. The results have been empirically proven on both a test case as well as an industrial benchmark. It is shown that the proposed method to select initial parameters speeds up the learning of the HMM models with fewer iterations and better log-likelihood values in general. Better log-likelihood means better accuracy. In the future work, we would like to test on different HMMs, such as Hidden semi-Markov Model [13, 14, 15]. The results will be verified to see whether it reaches global maxima in the learning algorithm. Moreover, other performance indicators other than log-likelihood will be tested as well.

Acknowledgments

The research reported in this paper was supported in part by the PhD-VUB scholarship awarded by VUB-IRMO. This research was partly funded by the Prognostics for Optimal Maintenance (POM) project (grant nr. 100031; www.pom-sbo.org) which is financially supported by the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen).

References

[1] Peng Y, Dong M and Zuo M J 2010 *Int. J. Adv. Manuf. Technol.* **50** 297-313

[2] Rabiner, L R 1989 *Proc. IEEE.* **77** 257-286

[3] Iman R L and Conover W J 1983 *A Modern Approach to Statistics* (New York: John Wiley and Sons) p 498

[4] Montgomery D C and Peck E A 1982 *Introduction to Linear Regression Analysis* (New York: John Wiley and Sons)

[5] Perreault L, Bernier J, Bobee B and Parent E 2000 *J. Hydrology.* **235** 221-241

[6] Kendall M G 1975 *Rank correlation methods* (London: Charles Griffin & Company Limited

[7] Manly B and Mackenzie D 2000 *Environmetrics* **11** 151-166

[8] Alippi C and Roveri M 2006 *IEEE Int. Symp Circ S* (Kos: Grecia) p5752-5755

[9] Talyor W A 2000 In www.variation.com/cpa/tech/changepoint.html, online in 2011

[10] NSF-I/UCRC Center for intelligent maintenance systems prognostic data repository: Bearing data set. In http://ti.arc.nasa.gov/tech/dash/pcoe/prognostic-data-repository/, online in 2010

[12] Tobon-Mejia D A, Medjaher K and Zerhouni N 2011 *Prognostic and System Health Management Conference* (Shenzhen: China)

[13] Yu S Z and Kobayashi H 2006 *IEEE Trans Sig Proc.* **54** 1947-1951

[14] Dong M and He D 2007 *Mach. Syst. Sig. Proc.* **21** 2248-2266
[15] Yu S Z 2010 *Artif. Intell.* **174** 215-243