AN INSTANCE BASED LEARNING APPROACH FOR BEARING REMAINING USEFUL LIFE PREDICTION WITH RECURRENT NEURAL NETWORK

Jun Sun, Qiao Sun

University of Calgary, Department of Mechanical and Manufacturing Engineering, Calgary, AB, T2N 1N4, Canada

emails: jun.sun@ucalgary.ca (J.S.), qsun@ucalgary.ca (Q.S.)

An important task of the prognostics and health management (PHM) is to predict remaining useful life (RUL), which is the lifetime of the monitored system or component before failure occurs. In our proposed instance based learning (IBL) approach, based on the spectrograms of vibration signals from the monitored bearing, the principal component analysis (PCA) technique is employed to generate the feature sequences representing bearing degradation process trends. Comparing learning instance with test instance, their corresponding feature sequences with similar trends are identified using a dynamic time wrapping (DTW) based similarity evaluation method. From the identified similar feature sequence of learning instance, we synthetically populate multiple feature sequences to train a recurrent neural network (RNN) model with long short-term memory (LSTM). The RNN-LSTM model then can be used to make RUL prediction for the test instance. A bootstrap aggregating method is used to improve the prediction accuracy. The proposed IBL approach exhibits promising performances on the experimental bearing datasets provided for IEEE PHM 2012 Prognostic Challenge.

Keywords: remaining useful life, prognostics, instance based learning, recurrent neural network, similarity evaluation

1. Introduction

An important task of the prognostics and health management (PHM) is to predict remaining useful life (RUL), which is the lifetime of the monitored system or component before failure occurs. RUL prediction has become an attractive research area with many emerging physics-based and data-driven approaches. Recently, with significant advances in the machine learning and deep learning technologies, the data-driven approaches have become more popular and attracted more attentions. The majority of the existing approaches attempt to develop and train a generalized prediction model for handling a wide range of system or component degradation cases. However, to deal with real world problems, those approaches often encounter two challenges: one is the lack of knowledge of the degradation physics and another is the insufficiency of historic run-to-failure examples.

As one of data-driven approaches, the instance-based learning (IBL) approach utilizes the experience gained particularly from similar historic instances to solve new problem instances, instead of building a generalized prediction model with all of historic instances [1]. The existing IBL research work mainly focuses on extracting degradation information from monitoring data to generate so called “health index” features as a means to evaluate instance similarity for historic instance retrieval. Since the retrieved instances have similar “health index” trends or patterns but not identical to new problem instance, the final RUL estimate is usually obtained by aggregating the multiple RUL estimates based on the multiple similar instances to improve prediction accuracy. However, the re-
al-world systems, such as wind turbines and aircraft engines, may consist of a large number of components, and have different degradation behaviours even under same operating conditions. Thus, it is natural that only few similar instances are available in the historic datasets. Especially, the existing IBL approaches have a difficulty to make accurate prediction when there is only one similar instance that can be found in historic datasets.

To address the challenges aforementioned, we proposed a novel IBL approach for RUL prediction. The remainder of this paper is organized as follows. Section 2 describes and formulates the bearing RUL prediction problem. Section 3 introduces the proposed IBL approach. Section 4 presents the experimental evaluation results. Finally, Section 5 gives a conclusion of this paper.

2. Problem Description

In this paper, the experimental bearing data was provided by FEMTO-ST Institute for the IEEE PHM 2012 Prognostic Challenge [2]. In a bearing test platform, two accelerometers were mounted on the bearing housing to measure vibration in the vertical and horizontal directions. Data sampling was conducted at 10 seconds intervals with 25.6 kHz sampling rate and 1/10 second duration. Hence, the observation contained 2560 sample points at each time cycle.

Under each operating condition (w.r.t. rotation speed and load force), the competition participants were provided with only two run-to-failure experimental datasets as learning instances in order to build prediction models, and were asked to estimate the RULs of remaining bearings as test instances. No information on the type of failure (e.g., ball, inner or outer race, cages, etc.) to be occurred was given. The spread of the life duration of all test instances was very wide, from one to seven hours. In this paper, we use the bearing datasets under one particular operating condition (w.r.t. 1800 rpm and 4000 N) and the horizontal vibration signals to evaluate our proposed IBL approach. As illustrated in Fig. 1, the two learning instances are denoted as Bearing1_1 and Bearing1_2, while the five test instances are Bearing1_3, 1_4, 1_5, 1_6, and 1_7.

Using the vibration signal, for the learning instance, the run-to-failure degradation process can be denoted as

\[
\left\{ \left( \mathbf{x}_i^{(l)}, y_i^{(l)} \right) \right\} \quad i = 1, 2, ..., M \tag{1}
\]

For a given test instance with unknown RUL, the degradation process can be denoted as

\[
\left\{ \left( \mathbf{x}_j^{(p)}, y_j^{(p)} \right) \right\} \quad j = 1, 2, ..., N \tag{2}
\]

In Eq. (1) and Eq. (2), \( \mathbf{x}_i^{(l)} \) and \( \mathbf{x}_j^{(p)} \) are feature vectors representing the bearing degradation states at each recording time cycle indexed of \( i \) and \( j \), from the beginning of bearing usage, for the learning and test instances, respectively. Each feature vector consists of the number \( K \) of features extracted from the raw vibration signal, within a time duration \( T \) (e.g., \( T = 1/10 \) s). The interval \( \Delta t \)
(e.g., $\Delta t = 10$ s) is set between two consecutive time cycles.

$$x_i^{(l)} \text{ or } x_j^{(p)} = (x_1, x_2, ..., x_K)^{i (l) \text{ or } (p)}$$

Also in Eq. (1) and Eq. (2), $y_i^{(l)}$ and $y_j^{(p)}$ are the RUL at the time cycle $i$ and $j$ of learning instance and test instance, respectively. The $y_i^{(l)}$ is known while $y_j^{(p)}$ unknown. The value of $y_i^{(l)}$ can be calculated by

$$y_i^{(l)} = M - i$$

where $M$ is the index of time cycle when the bearing usage of the learning instance is terminated due to failure.

Using the IBL approach, it is critical to identify the learning instance from historical datasets, whose degradation process trend is similar to the give test instance. Based on the similar learning instance, the RUL prediction problem can be solved by estimating $y_j^{(p)}$, which is the RULs of the given test instance, especially at its last time cycle $j = N$.

3. Methodology

The proposed IBL approach consists of three major steps including feature extraction, similarity evaluation, and degradation prediction, as illustrated in Fig. 2. The three major steps are elaborated in this section.

![Figure 2: The proposed IBL approach for RUL prediction.](image)

3.1 Feature Extraction

In the feature extraction step, the fast Fourier transform (FFT) is applied on the raw vibration signals to generate spectrograms for both learning instance and test instance, as shown in Fig. 3 (a) and (b). In order to reduce noise influence, the spectrograms are smoothed using the moving averaging filters in both frequency direction and time direction. In this paper, the window lengths of moving average filters in frequency and time directions are set to 40 and 80, respectively.

It is not efficient and effective to use the larger number of frequency components to build a high-dimensional RUL prediction model. Thus, the principal component analysis (PCA) technique is utilized for dimension reduction. PCA can reduce a larger number of features to the representative principal components using a linear transformation while maintaining most of the variability of the dataset. In this paper, PCA is first applied on the spectrogram of leaning instance to generate a set of eigenvectors. From the eigenvectors, the first five principal components (PCs) can be selected to represent the bearing degradation states. In the example of learning instance Bearing1_1, the first five PCs are selected to reflect 99.86% of data variability. They are accounted for the data variability 92.70%, 5.27%, 1.48%, 0.25%, 0.16%, respectively.

By projecting the spectrograms of learning and test instances on the five major PCs, the corresponding PC coefficients (i.e., eigenvalues) can be obtained. These coefficients are included in the feature vector $(x_i^{(l)} \text{ or } x_j^{(p)}$ in Eq. (1) and Eq. (2)) to represent each degradation state. As a result, five PC feature sequences can be obtained as illustrated in Fig. 3 (c) and (d). In this paper, PCA is applied on the spectrogram in the range from 0 to 8,000 Hz, where the main bearing and structure resonance modes and frequencies excited by bearing defects in the studied experimental datasets.

The procedure for feature extraction can be summarized as follow:

1) Spectrogram Generation

a) Generate spectrograms from raw vibration signals using FFT for learning and test instances.
b) Smooth the spectrograms generated from 1.a) using a moving average filter in the frequency direction.

c) Smooth the spectrograms obtained in 1.b) using a moving average filter in the time direction.

2) Dimension Reduction

a) Apply PCA on the spectrogram of learning instance obtained in 1.c).

b) Obtain PC coefficients and the corresponding PC feature sequences by projecting the spectrograms of learning and test instances on the PCs generated in 2.a).

3.2 Similarity Evaluation

The purpose of the similarity evaluation step is to identify a pair of feature sequences between the learning and test instances that have similar trends or patterns. Dynamic time warping (DTW) algorithm is used in this step, as it can provide a robust similarity measure for comparing two time series. DTW algorithm allows similar curve shapes to match even if two time series have different lengths.

With respect to one of the extracted PC features $k$ (e.g., $k = 1$ or $2$, denoting the 1st PC or 2nd PC, and so on) obtained in 3.1, the two corresponding feature sequences of the learning instance and test instance can be described respectively, as follows:

$$L_k = \left( x_{k,1}^{(l)}, x_{k,2}^{(l)}, \ldots, x_{k,M}^{(l)} \right)$$  \hspace{1cm} (5)

$$P_k = \left( x_{k,1}^{(p)}, x_{k,2}^{(p)}, \ldots, x_{k,N}^{(p)} \right)$$  \hspace{1cm} (6)

Aligning the two sequences using DTW, there are many warping paths available and each of these paths indicates the matching relationship between the two sequences, as denoted by:

$$W = \left[ w_1, w_2, \ldots, w_H \right]$$  \hspace{1cm} \[\text{max}(N, M) \leq H < N + M\]  \hspace{1cm} (7)

where the path element $w_h$ represents one alignment and $H$ is the number of alignments. To establish the warping path, three constraints are required by the DTW algorithm, including boundary constraints, continuity constraints and monotonicity constraints [3]. One of these warping paths that has the minimal cumulative cost is considered the DTW distance:

$$D(P_k, L_k) = \min_w \{ \sum_{h=1}^{H} d(w_h) \}$$  \hspace{1cm} (8)

Considering the number of alignments $H$ in the path, the average DTW distance is computed by

$$D(P_k, L_k) / H$$  \hspace{1cm} (9)

Using the average DTW distance to evaluate the similarity of sequences, the less distance implies that the two compared sequences are more similar to each other.

The optimal wrapping path is determined using dynamic programming in the DTW algorithm. In this paper, instead of aligning two sequences globally, the subsequence DTW method [4] is used to identify the segment (i.e., subsequence) within the feature sequence of learning instance that is similar to the full-length feature sequence of test instance.

Utilizing the DTW method, the similarity evaluation step can be summarized as follows:

1) Data Normalization

a) Compute the mean and standard deviation of feature sequence $L_k$.

b) Normalize $L_k$ and $P_k$ using the mean and standard deviation in 1.a)
2) DTW Evaluation
   a) Perform the DTW method on the two normalized sequences \((P_k, L_k)\) obtained 1.b).
   b) Compute the average DTW distance between \((P_k, L_k)\), using Eq. (9).

   In the example of comparing Bearing1_3 and Bearing1_1 with respect to the PC feature sequences \((k = 1, 2, 3, 4, \text{ and } 5)\), the average DTW distances are obtained as shown in Table 1. \(P_1\) and \(L_1\) (i.e., 1st PC feature sequences) are considered the most similar sequences in the five sequence pairs compared. Particularly, the subsequence in the time cycle range of \([150, 2060]\) on \(L_1\) is identified similar to the entire sequence of \(P_1\) in the time cycle range of \([0, 1800]\).

   **Table 1: DTW based similarity evaluation of feature sequences \((P_k, L_k)\) of Bearing1_1 and Bearing1_3.**

<table>
<thead>
<tr>
<th>k</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTW Distance</td>
<td>20.98</td>
<td>43.24</td>
<td>872.9</td>
<td>297.9</td>
<td>100.0</td>
</tr>
<tr>
<td>(H) (Number of Alignments)</td>
<td>2912</td>
<td>3110</td>
<td>2154</td>
<td>2964</td>
<td>1868</td>
</tr>
<tr>
<td>Average DTW Distance</td>
<td>0.0072</td>
<td>0.0139</td>
<td>0.4122</td>
<td>0.1005</td>
<td>0.0535</td>
</tr>
</tbody>
</table>

3.3 Degradation Prediction
   As a machine learning technique, the recurrent neural network (RNN) can effectively learn sequential pattern or trend from the time series data. Thus, the RNN technique is used to build the prediction model and provide RUL estimates in the degradation prediction step.

3.3.1 RNN-LSTM Model

   RNN is the network of neurons with recurrent connections that reflects sequential information during training and prediction. As illustrated in Fig. 4 (a), in the vanilla RNN architecture, the state of hidden layer \(h_t\) is calculated from the current input \(x_t\) and the previous state of hidden layer \(h_{t-1}\), by the following formula:

   \[
   h_t = f_1(U \cdot x_t + W \cdot h_{t-1})
   \]  

   The output \(y_t\) of the RNN can be calculated as follows:

   \[
   y_t = f_2(V \cdot h_t)
   \]  

   The functions \(f_1\) and \(f_2\) are nonlinear activation functions such as the hyperbolic tangent function and sigmoid function. The \(U\), \(V\), and \(W\) are the trainable weight matrices. A back propagation algorithm, namely Back Propagation Through Time (BPTT), is used to update the network weights during training [5]. The BPTT is similar mechanism as the standard back propagation algorithm, except that it is back propagated through time rather than through layers.

   As the vanilla RNN is deep neural network in the time direction, vanishing and exploding gradients can occur during the BPTT training process. In addition, RNN can store short-term memory but is vulnerable to long-term dependency. Long Short Time Memory (LSTM) network, a special kind of RNN, is designed to overcome the vanilla RNN’s shortcomings. In the RNN-LSTM structure, a memory cell holds short-term memory for a longer period by controlling three gates, including input gate, output gate, and forget gate. The three control gates play an important role in training long-term dependency, while controlling information storage and flow in making prediction [5].
In this paper, a RNN-LSTM model is designed for the RUL prediction. The RNN-LSTM model unfolded along time direction is illustrated in Fig. 4(b). The model input sequence at each time cycle \(i\)-th, denoted as \([x_{i-d}, \ldots, x_{i}, x_i]\), contains the feature vectors at the time cycles from \((i-d)\)-th to \(i\)-th. Correspondingly, the model outputs are \([y_{i-d}, \ldots, y_{i-1}, y_i]\) and the sequence length \(d\) is the time step size of the RNN-LSTM model. The model can contain a stack of multiple layers of LSTMs, where the same number of hidden nodes is constructed for each LSTM layer. Since we use the many-to-one sequence mapping approach, for each input sequence \([x_{i-d}, \ldots, x_{i-1}, x_i]\), only at the last step, a densely connected (DC) layer accepts \(y_i\), the output from multi-layer LSTMs. The RUL prediction \(y_i\) is obtained through the DC layer as follows:

\[
y_i = f_3(O \cdot y_i)
\]

(12)

where the \(f_3\) is the activation function and \(O\) is also trainable weights. The DC layer maintains a linear activation function in the RNN-LSTM model. For example, to implement the RNN-LSTM model for a sequence mapping from 20 input steps to one output, we can construct three layers of LSTMs with 10 hidden nodes per layer, and one DC layer with one output node. This model can be denoted as RNN-LSTM \((20, 10, 10, 10, 1)\).

### 3.3.2 Model Training and RUL Prediction

In this paper, we adopt three strategies for training the RNN-LSTM model and make RUL prediction.

- **Strategy I: Modeling the degradation process based on the segment of feature sequence corresponding to the defect-propagation stage.**

  The RNN-LSTM model is trained based on the feature sequence of learning instance. For example, as illustrated the 1\(^{\text{st}}\) PC feature sequence of the learning instance in Fig. 5, there are three typical stages in the typical bearing degradation process, including i) the normal-usage stage when there are almost no degradation occurrence or trend; ii) the defect-propagation stage when sequential patterns and trends in the feature sequence can be observed, iii) the near-failure stage when the defect is developing abruptly and the bearing’s usage is near to end. As the near-failure stage usually lasts a very short time period, it can be considered same for the two similar bearing instances (i.e., the learning and test instances). Therefore, it is effective that the prediction model is trained based on the defect-propagation segment.

- **Strategy II: Populating the feature sequence of learning instance to generate multiple feature sequences for training model.**

  Because the learning and test feature sequences has similar trends but not identical as illustrated in Fig. 5 (the learning sequence in blue color, the test sequence in red color), the prediction model built on the learning sequence may not provide RUL estimates accurately for the test instance. Thus, based on the defect-population segment of the learning sequence, we synthetically generate multiple learning sequences for training model. The learning sequence population is implemented by stretching or shrinking the learning sequence in length along the time direction (e.g., length stretching 20% or shrinking -20% in the sequence length) using the linear interpolation technique. The idea behind this strategy is that the same type of bearing defect or failure may have different length (e.g., longer or shorter) degradation processes.

- **Strategy III: Using the bootstrap aggregating (i.e., bagging) method to improve the accuracy of RUL prediction.**

  The bagging method is an ensemble learning method that reduces the variance of estimated result by averaging multiple estimates. Using this method, we train multiple RNN-LSTM models on different subsets of learning sequence data (selected randomly with replacement) and then produce the multiple RUL estimates for the test instance correspondingly. As illustrated in Fig. 6, the average and standard deviation of the multiple estimates can be considered the final RUL estimate and the prediction confidence interval, respectively.

  The procedure for the step of degradation prediction can be summarized as follow:
1) **Data Preparation**
   a) Specify the defect-propagation segment in the feature sequence of the learning instance.
   b) Populate the learning sequences based on the segment specified in 1.a).
   c) Normalize the learning sequence and the corresponding feature sequence of the test instance using the mean and standard deviation obtained from the learning sequences.
   d) Specify the defect-propagation segment in the feature sequence of the test instance.

2) **Model Training and RUL Prediction (one trail for each selected subset of learning data)**
   a) Select 80% subset randomly from learning data for training and the rest 20% for validation.
   b) Train the RNN-LSTM model using the training and validation subsets selected in 2.a).
   c) Generate the RUL estimates by applying the RNN-LSTM model trained in 2.b) on the defect-propagation segment of the test instance specified in 1.d).
   d) Smooth the RUL estimates along the time direction using the moving average method.

3) **Estimate Ensemble (after multiple trails of model training and RUL prediction)**
   a) Compute the average and standard deviation of the multiple RUL estimates obtained in 2.d).

In the example where Bearing1_1 is considered the similar learning instance of Bearing1_3, a RNN-LSTM (20, 10, 10, 10, 1) model is constructed for RUL prediction. The 1st PC feature sequences are identified as the pair of similar sequences between the two instances. The defect-propagation segment [1300, 2750] is specified in the learning feature sequence. Additional twenty learning sequences are generated using the length shrinking or stretching ratios in the range [-0.8, 0.2] as shown in Fig. 5. Using the bagging method, final prediction result is obtained by averaging ten trails of training-and-prediction. The prediction confidence interval is measured by the standard deviation (i.e., +/- 3σ). At the last time cycle [1800], the final predicted RUL is equal to 561 (time cycles), while the actual RUL is known as 573 (time cycles). The prediction result achieves a high accuracy with the relative error = 2% and the confidence interval 3σ = 3*26 (time cycles).

### 4. Experimental Evaluation

The experimental evaluation results for the five test instances are summarized in Table 2. In the experimental evaluation, we applied the proposed IBL approach for Bearing1_3 and Bearing1_7, with Bearing1_1 and Bearing1_2 as their similar learning instances, and achieved the relative prediction errors 2% and 11%, respectively.

The RUL estimates for the test instances Bearing1_4, Bearing1_5, and Bearing1_6 are directly made using their similar learning instance Bearing1_2 as reference, without applying the RNN-LSTM prediction models. Referring to the Bearing1_2, Bearing1_4 passes over the maximum failure feature value at its last time cycle [1137], thus the RUL of Bearing1_2 can be estimated as 0. Since both Bearing1_5 and Bearing 1_6 at their last time cycles approach closely to the turning point from the defect-propagation stage to the near-failure stage, their RULs are conservatively estimated as 45 (time cycles) that is the duration of the near-failure stage of Bearing1_2.
The bearing RUL prediction datasets were first published for the IEEE PHM 2012 Prognostic Challenge, two tentative solutions (as shown in Table 3) were ranked in the top two in the competition [6, 7]. However, both solutions encountered difficulties in extracting representative features that can effectively reflect the degradation process trends. Since then, no more effective approaches have been reported in the research literatures, due to the multiple challenges including limited learning instances, no information about failure modes, and a wide range of failure times.

Table 2: RUL prediction results for test instances at their last time cycles.

<table>
<thead>
<tr>
<th>Test Instance (Bearing)</th>
<th>1_3</th>
<th>1_4</th>
<th>1_5</th>
<th>1_6</th>
<th>1_7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning Instance (Bearing)</td>
<td>1_3</td>
<td>1_4</td>
<td>1_5</td>
<td>1_6</td>
<td>1_7</td>
</tr>
<tr>
<td>Last Time Cycle</td>
<td>1800</td>
<td>1137</td>
<td>2303</td>
<td>2300</td>
<td>1500</td>
</tr>
<tr>
<td>Predicted RUL (time cycles)</td>
<td>561</td>
<td>0</td>
<td>45</td>
<td>45</td>
<td>677</td>
</tr>
<tr>
<td>Standard Deviation (the cycles)</td>
<td>26</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>32</td>
</tr>
<tr>
<td>Relative Error (time cycles)</td>
<td>2%</td>
<td>100%</td>
<td>72%</td>
<td>69%</td>
<td>11%</td>
</tr>
<tr>
<td>Actual RUL (time cycles)</td>
<td>573</td>
<td>34</td>
<td>161</td>
<td>146</td>
<td>757</td>
</tr>
</tbody>
</table>

Table 3: RUL prediction results of the first and second place solutions in IEEE HPM 2012 [6, 7].

<table>
<thead>
<tr>
<th>Test Instance (Bearing)</th>
<th>1_3</th>
<th>1_4</th>
<th>1_5</th>
<th>1_6</th>
<th>1_7</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st Place Solution - Predicted RUL (time cycles)</td>
<td>360</td>
<td>27</td>
<td>147</td>
<td>156</td>
<td>772</td>
</tr>
<tr>
<td>2nd Place Solution - Predicted RUL (time cycles)</td>
<td>49</td>
<td>10</td>
<td>49</td>
<td>49</td>
<td>49</td>
</tr>
</tbody>
</table>

5. Conclusion

In this paper, we proposed an IBL approach to deal with bearing RUL prediction problem, particularly where limited learning instances are available in the historic datasets. Based on the spectrograms of bearing vibration signals, the PCA technique is employed to generate representative features reflecting the bearing degradation trends. Comparing the test and learning instances, their similar feature sequences are identified using the DTW based evaluation method. In the degradation prediction, multiple learning feature sequences are synthetically populated for training the RNN-LSTM model. With the trained prediction model, the bagging method is used to generate and ensemble RUL estimates. The proposed IBL approach provides the promising prediction results on the experimental bearing datasets provided for IEEE PHM 2012 Prognostic Challenge. Further development with real-world RUL prediction problems is expected in our future work.

REFERENCES