Vibration indicator-based graph convolutional network for semi-supervised bearing fault diagnosis

S H Wang1, S B Xing1, Y G Lei1, N Lu2 and N P Li1

1 Key Laboratory of Education Ministry for Modern Design and Rotor-Bearing System, Xi’an Jiaotong University, Xi’an, Shanxi, 710049, P.R. China
2 Systems Engineering Institute, Xi’an Jiaotong University, Xi’an, Shanxi, 710049, P.R. China

E-mail: shuhuiwang@stu.xjtu.edu.cn, xingsaibo@stu.xjtu.edu.cn, yaguo lei@mail.xjtu.edu.cn, lvna2009@mail.xjtu.edu.cn, naipengli@mail.xjtu.edu.cn

Abstract. Since fault diagnosis has entered the big data era, deep learning has been more and more widely studied to diagnose faults of rolling element bearings. Generally, existing methods require labeled data for training before they can be used to recognize faults. However, in real scenarios, massive data are usually unlabeled data rather than labeled ones, because labeling data is always a tough issue and consumes much human labor. In order to fully take advantage of the massive unlabeled data, this paper proposes a vibration indicator-based graph convolutional neural network (VI-GCN) for fault diagnosis. The VI-GCN is applied to a benchmark dataset of bearing faults. Experimental results indicate that it is promising for bearing fault diagnosis when there are few labeled data.

Keywords: rolling bearing, fault diagnosis, semi supervised learning, k-nearest neighbor graph (knn graph), graph convolutional neural networks

1. Introduction

With the development of smart manufacturing and cheap sensing technology, the fault diagnosis has entered the big data era. As an important tool to deal with big data, deep learning has been introduced into the field of fault diagnosis [1]. Lei et al. [2] proposed a two-stage learning method to directly mine features from input data, which obtains fairly high diagnosis accuracy. Janssens et al. [3] put forward a new feature learning model based on the convolutional neural network model and achieved at a pretty high accuracy, using only the raw vibration data. As Yann et al. reported in [4], deep learning is able to provide a more accurate representation of raw data, because it is able to learn very complex mapping function by the stacking multiple layers.

Most existing methods require that fault types corresponding to all the training samples should be known. However, in real scenarios, labelling data usually consumes much time and human labor, i.e. labeled data is scarce. To overcome the above contradictions, semi-supervised learning is introduced. One of the most popular algorithms is the graph based semi-supervised learning. In order to infer the class of unlabeled data with graph structure, Zhu et al. [5] introduced label propagation together with an iterative algorithm. Later, Zhou et al. [6] employed the normalized Laplacian and soft clamping solving the classification function, which is more robust to noise than the method in [5]. Generally, graphs are effective tools to describe intrinsic relationship among data points. In the field of fault
diagnosis, Razavi-Far et al. [7] integrated the semi-supervised dimensionality reduction and classification techniques into the diagnostic scheme for decision making under partially labeled sets of observations. Jiang et al. [8] proposed a joint label consistent dictionary learning and adaptive label prediction techniques for machine fault classification.

It can be seen that graphs, in the aforementioned literatures, are either served for feature dimensionality reduction, or used to do label predictions in a semi-supervised manner. Recently, graph convolutional neural network (GCN) [9] has got great success and shows advantages in dealing with graph data. Different from label propagation [5] and label spreading [6], GCN could achieve label predictions using the graph structure and node features at the same time. Motivated by the success of GCN proposed by Kipf [9], we propose a vibration indicator-based graph convolutional neural network (VI-GCN). In VI-GCN, vibration indicator graph is first constructed to relate a few labeled data with a large quantity of unlabeled data. Then, graph labelling is achieved by the stacked graph convolution layers.

2. **VI-GCN**

In this section, we outline the diagnostic framework and detail the basic idea of the VI-GCN. Figure 1 shows a two-layer graph convolution neural network that applied to a vibration indicator graph. The primary procedure of the diagnostic framework is as follows:

1. Collect vibration signal.
2. Calculate vibration indicators of the collected signal.
3. Construct k-nearest neighbour graph using the vibration indicators.
4. Classify faults in a semi-supervised manner using graph convolution neural network.

![Figure 1. Diagnostic framework based on the VI-GCN](image)

2.1. Preliminary on graph

A graph [10] $G$ is defined as $G = (V, E)$, where $V$ is a finite set of vertices and $E$ is a set of edges that depicting the graph’s connectivity. For two vertices $v_1 \in V$, $v_2 \in V$ in a graph, they are adjacent if there is an edge $e \in E$ between these two. A graph can be either directed or undirected. Note that in the rest of this paper, we consider only undirected graph. Adjacency matrix is a useful algebra representation for graphs. For a simple graph $G$, the binary adjacency matrix $A$ can be expressed as

$$A_{ij} = \begin{cases} 1 & \text{if } (v_i, v_j) \in E \text{ and } i \neq j, \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$
when elements of the adjacency matrix are denoted by some numerical weights, i.e. weighted adjacency matrix, we call the corresponding graph an undirected weighted graph, which usually model the similarity between two nodes. Here, the adjacency matrix of an undirected weighted graph is also denoted by $A$.

2.2. Indicator graph construction

Vibration analysis is widely used in the field of mechanical fault diagnosis [11, 12]. Most works aim at mining sensitive features from bearing vibration signals under time domain, frequency domain and time-frequency domain [1]. We include vibrational data indicators [13] to construct a feature vector that extracted from the three domains, as shown in Table 1. In the graph based semi-supervised learning, $k$-Nearest Neighbors ($k$-NN) and $\varepsilon$-neighborhood are two common methods constructing a graph from a set of points. In fact, nearest neighbor graphs are a subject of computational geometry [14, 15]. Compared to $\varepsilon$-neighborhood based graph, $k$-NN are more adaptive in graph construction. Therefore, we utilize $k$-NN graph. Edges are modeled by distance metric using Gaussian kernel function, which can be expressed as follows

$$A_{ij} = \exp\left(-\frac{d(x_i, x_j)}{2\sigma^2}\right)$$

where $d(x_i, x_j)$ measures the distance between the $i$-th node and $j$-th node.

| Indicator         | Equation |
|-------------------|----------|
| Standard deviation | $\sqrt{\sum_{n=1}^{N} (x(n) - \bar{x})^2 / N}$ (n = 1, 2, ..., N) |
| Peak              | $\max|x(n)|$ |
| Skewness          | $\sum_{n=1}^{N} (x(n) - \bar{x})^3 / (N-1)\sigma^3$ |
| Kurtosis          | $\sum_{n=1}^{N} (x(n) - \bar{x})^4 / (N-1)\sigma^4$ |
| Root mean square  | $\sqrt{\sum_{n=1}^{N} x(n)^2 / N}$ |
| Crest indicator   | $\max|x(n)| / \sqrt{\sum_{n=1}^{N} x(n)^2 / N}$ |
| Clearance indicator | $\max|x(n)| / \left(\sqrt{\sum_{n=1}^{N} x(n)^2 / N}\right)^2$ |
| Shape indicator   | $\sqrt{N \sum_{n=1}^{N} x(n)^2 / \sum_{n=1}^{N} |x(n)|}$ |
| Impulse indicator | $\max|x(n)| / \left(\sum_{n=1}^{N} |x(n)| / N\right)$ |
| Mean frequency    | $\sum_{k=1}^{K} s(k) / K$ |
| Frequency center  | $\sum_{k=1}^{K} f(k)s(k)$ |
| Root mean square frequency | $\sqrt{\sum_{k=1}^{K} f(k)^2 s(k) / \sum_{k=1}^{K} s(k)}$ |
Standard deviation frequency

\[
\sqrt{\sum_{k=1}^{K} \left[ f(k) - \left( \sum_{k=1}^{K} f(k) \right) / \sum_{k=1}^{K} s(k) \right]^2 s(k)}
\]

WPT energy

\[
\sum_{n=1}^{N} |x(n)|^2 / \sum_{i=0}^{2^j-1} \sum_{n=1}^{N} |x(n)|^2, \quad i = 0, 1, \ldots, 2^j - 1
\]

MD energy

\[
\sum_{n=1}^{N} |\text{IMF}_i(n)|^2 / \sum_{i=1}^{NI} \sum_{n=1}^{N} |\text{IMF}_i(n)|^2, \quad i = 1, 2, \ldots, NI
\]

2.3. Graph convolutional neural network

Considering a signal \( x \in \mathbb{R}^N \) (a scalar for every node) on a weighted graph \( G \), its spectral convolutions with a filter \( g_\theta = \text{diag}(\theta) \) parameterized by \( \theta \in \mathbb{R}^N \) in Fourier domain is defined as

\[ g_\theta \ast x = U g_\theta U^T x \]  

Let \( L \in \mathbb{R}^{N \times N} \) denote the normalized graph Laplacian matrix of the graph, then it can be expressed as

\[ L = I_N - D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \]  

where \( I_N \in \mathbb{R}^{N \times N} \) is the identity matrix, \( D \in \mathbb{R}^{N \times N} \) is the degree matrix and the element can be calculated by \( D_{ii} = \sum_j A_{ij} \). The Eigen decomposition of \( L \) is

\[ L = U \Lambda U^T \]  

where \( U = [u_0, \ldots, u_{N-1}] \in \mathbb{R}^{N \times N} \) is the orthonormal eigenvectors and \( \Lambda = \text{diag}(\lambda_0, \ldots, \lambda_{N-1}) \in \mathbb{R}^{N \times N} \) is a diagonal matrix containing its real non-negative eigenvalues.

The \( g_\theta \) could be understood as a function of the eigenvalues of \( L \), thus the spectral convolutions can be expressed as

\[ g_\theta \ast x = U g_\theta (\Lambda) U^T x \]

In order to learn \( k \)-localized filter and to reduce computational complexity, eq. (5) can be simplified as

\[ g_\theta \ast x = \theta(I_N + D^{-\frac{1}{2}} A D^{-\frac{1}{2}}) x \]  

For an input graph signal \( X \in \mathbb{R}^{N \times F} \) (\( F \) denotes the input dimensions for every node), with renormalization trick, we can get a layer-wise linear model

\[ H = \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} X W \]  

where \( \tilde{A} = A + I_N \) and \( \tilde{D}_{ii} = \sum_j \tilde{A}_{ij} \). \( H \in \mathbb{R}^{N \times M} \) is the filtered signal after a graph convolution and \( M \) denotes the number of neurons in the hidden layer. \( W \in \mathbb{R}^{F \times M} \) denotes the filter weights.

The layer-wise propagation rule is formulated as

\[ H^{(l+1)} = \sigma \left( \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l)} W^{(l)} \right) \]  

where \( \sigma(\cdot) \) denotes an activation function. \( H^{(l)} \) is the filtered signal in the \( l \)-th layer and \( W^{(l)} \) is the trainable weights.

Finally, with a series of convolutional layers, VI-GCN is able to aggregate features from one node to its neighbors. The output layer utilizes a softmax activation function with cross-entropy loss minimized over all labeled data. Hence, graph labeling of nodes through layer-wise graph convolution can be achieved.
3. Experimental validation

In this section, we considered a case study of rolling bearing fault in electromechanical drive systems from Paderborn University [16] to show the performance of the method.

3.1. Dataset

The test rig is shown in Figure 2. It consists an electric motor, a torque-measurement shaft, a rolling bearing test module, a flywheel, and a load motor. The tested bearings are ball bearings (type: 6203) with eight rolling elements. The faults were introduced to different locations of the bearings, using electric discharge machining, drilling or manual electric engraving. The shaft rotating speeds are set to 1500 and 900 rpm, respectively. The sampling frequency is set to 64 kHz.

The dataset mainly contains 4 bearing health states, i.e. normal state (N), inner race fault (IR), outer race fault (OR), and compound fault (CF) of IR and OR. Hence the 4 health states are used for analysis.

The shaft rotating speed of test data is under 1500 rpm, the load torque is 0.7 Nm and the radial force is 1000 N.

![Test rig](image)

**Figure 2. Test rig**

3.2. Experimental analysis

We consider the VI-GCN with two graph convolutional layers and select ReLU as the activation function. Then, the forward propagation can be expressed as

\[
Z = f(X, A) = \text{softmax}(\hat{A}\text{ReLU}(\hat{A}XW^{(0)})W^{(1)})
\]

(10)

where \(\hat{A} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}\), \(W^{(0)} \in \mathbb{R}^{F \times M}\) and \(W^{(1)} \in \mathbb{R}^{M \times C}\) denote the trainable weights of the two graph convolutional layers. To train the model in a semi-supervised manner, the cross-entropy error over all labeled data can be calculated by

\[
\text{Loss} = -\sum_{i \in y_L} \sum_{c=1}^{C} y_{lc} \ln Z_{lc}
\]

(11)

where \(y_L\) is the set of sample indices that have labels and \(C\) denotes the number of the fault classes.

We randomly select 10 data files for each health state and then obtain 4000 samples. Each sample contains 2560 data points. A vibration indicator graph is constructed by the 4000 samples. The nearest 20 neighbors for each node are considered in the graph. Only 20 samples are labelled per class for the training process. The experimental parameter settings are summarized in Table 2.

| Node | \(k\) | \(F\) | \(M\) | \(C\) | Label rate | Learning rate | Metric          |
|------|------|------|------|------|------------|--------------|----------------|
| 4000 | 20   | 27   | 200  | 4    | 0.02       | 0.003        | Euclidean distance |

To show some details about the diagnosis results, the confusion matrix is presented in Figure 3(a). As we can see, the proposed VI-GCN performs well on the normal state and the single fault states in
the case study, but it misclassifies 0.5% of the inner race fault as the compound fault, 9.88% of the outer race fault as the compound fault, 14.34% of the compound fault as the outer race fault, and 0.3% of the compound fault as the inner race fault. The misclassification is due to the fact that the compound fault contains information of both the inner race fault and the outer race fault. The results indicate that the pairwise similarity metric is good enough for the single fault identification, even though it does not perform very well with the compound fault. Different from raw signals, vibration indicators represent statistical features of the monitoring signals, therefore it is not hard to understand why Euclidean distance works in cooperating with the VI-GCN in the case study.

Decision tree [17] (DT) and support vector machine [18] are benchmark methods in machine learning area. Label spreading [6] (LS) is also the well-known semi-supervised learning method. Area under the Receiver operating characteristic (AUC-ROC) curves of LS, DT, linear support vector classification (linear SVC) and the VI-GCN, are shown in Figure 3(b). The comparison methods are implemented using scikit-learn [19]. It can be seen that in the case study, DT performs better than linear SVC in a complete supervised manner, while the VI-GCN performs better than LS by semi-supervised learning. Unlike LS propagating labels though the graph structure, graph convolutional neural network propagates node features (explained in section 2.3) instead, which could explain why VI-GCN remains competitive in the case study.

![Confusion matrix of the VI-GCN](image1)

![ROC curves of different methods](image2)

**Figure 3.** Test results

4. Conclusions

This paper proposes VI-GCN for semi-supervised bearing fault diagnosis. Firstly, vibration signal indicators are extracted from different domains. Secondly, a graph based on the vibration indicators is constructed using pairwise similarity. Convolutions on the vibration indicator graph finally map the unlabeled data to its corresponding fault state. Experiment results indicate that the proposed VI-GCN is effective in diagnosing faults when there are only a few labeled data. In this work, the graph is built by a user defined parameter $k$. In the future, the adaptive $k$-nearest neighbors algorithm [20] will be investigated to build graph rely upon the data itself.

Acknowledgments

This research is supported by NSFC-Zhejiang Joint Fund for the Integration of Industrialization and Informatization (U1709208), National Natural Science Foundation of China (61673311), and Fundamental Research Funds for the Central Universities.
References

[1] Lei Y G 2016 Intelligent Fault Diagnosis and Remaining Useful Life Prediction of Rotating Machinery. (Oxford: Butterworth-Heinemann Elsevier Ltd)

[2] Lei Y G, Jia F, Lin J, Xing S B and Ding S X 2016 IEEE Transactions on Industrial Electronics. 63(5) 3137-47

[3] Janssens O, Slavkovikj V, Vervisch B, Stockman K, Loccuif M, Verstockt S, Van de Walle R and Van Hoecke S 2016 Journal of Sound and Vibration. 377 331-45

[4] Yann L, Yoshua B and Geoffrey H 2015 Nature. 521(7553) 436-44

[5] Zhu X J, Gahramani Z B, and Lafferty J D 2002 Technical Report CMU-CALD-02-107

[6] Zhou D, Bousquet O, Lal T N, Weston J, and Schölkopf B 2004 In Advances in neural information processing systems. 321-28

[7] Razavi-Far R, Hallaji E, Saif M and Rueda L 2017 IEEE International Conference on Machine Learning and Applications. 61-7

[8] Jiang W M, Zhang Z, Li F Z, Zhang L, Zhao M B and Jin X H 2016 IEEE Transactions on Industrial Informatics. 12(1) 248-56

[9] Kipf T N and Welling M 2016 Semi-supervised classification with graph convolutional networks arXiv:1609.02907 (arXiv preprint)

[10] Chung F R K 1997 Spectral Graph Theory (American Mathematical Society)

[11] Wang D, Peng Z K and Xi L F 2020 Mechanical Systems and Signal Processing 140 106725

[12] Wang D, Peng Z K and Xi L F 2020 IEEE Transactions on Automation Science and Engineering 1-13

[13] Wang S H, Xiang J W, Zhong Y T and Tang H S 2018 Mechanical Systems and Signal Processing 112 154-70

[14] Preparata F P and Shamos M I 1985 Computational Geometry-An Introduction. (Berlin: Springer-Verlag)

[15] Eppstein D, Paterson M S and Yao F 1997 Discrete and Computational Geometry 17 (3) 263–82

[16] Lessmeier C, Kimotho J K, Zimmer D and Sextro W 2016 Proc. of the European conference of the prognostics and health management society vol. 7

[17] Breiman L, Friedman J, Olshen R and Stone C 1984 Classification and Regression Trees (Wadsworth, Belmont, CA)

[18] Cortes C and Vapnik V 1995 Machine Learning 20 273–97

[19] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M and Duchesnay E 2011 Journal of Machine Learning Research 12(10) 2825-30

[20] Ziemann A K and Messinger D W 2014 Proc. of SPIE 9088 90880B