We are IntechOpen,
the world's leading publisher of
Open Access books
Built by scientists, for scientists

5,500
Open access books available

136,000
International authors and editors

170M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS
BOOK CITATION INDEX INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Multilabel Classification Based on Graph Neural Networks

Wei-Cheng Ye and Jia-Ching Wang

Abstract

Typical Laplacian embedding focuses on building Laplacian matrices prior to minimizing weights of connected graph components. However, for multilabel problems, it is difficult to determine such Laplacian graphs owing to multiple relations between vertices. Unlike typical approaches that require precomputed Laplacian matrices, this chapter presents a new method for automatically constructing Laplacian graphs during Laplacian embedding. By using trace minimization techniques, the topology of the Laplacian graph can be learned from input data, subsequently creating robust Laplacian embedding and influencing graph convolutional networks. Experiments on different open datasets with clean data and Gaussian noise were carried out. The noise level ranged from 6% to 12% of the maximum value of each dataset. Eleven different multilabel classification algorithms were used as the baselines for comparison. To verify the performance, three evaluation metrics specific to multilabel learning are proposed because multilabel learning is much more complicated than traditional single-label settings; each sample can be associated with multiple labels. The experimental results show that the proposed method performed better than the baselines, even when the data were contaminated by noise. The findings indicate that the proposed method is reliably robust against noise.

Keywords: graph neural networks, multilabel classification, deep learning

1. Introduction

Traditional supervised learning deals with the analysis of single-label data, which means that samples are associated with a single label. However, in many real-world data mining applications, such as text classification [1, 2], scene classification [3, 4], crowd sensing/mining [5–11], and gene functional classification [12, 13], the samples are associated with more than one label. From this description, we understand that the challenge of the multilabel classification task is its potential output.

Basically, multilabel learning algorithms can be categorized into two different groups. 1) Problem transformation method. This method takes the multilabel problem and converts it into a single-label problem that can easily be classified using any classifier using the relationship between labels. 2) Adapted algorithm method. This method directly performs multilabel classification rather than transforming the problem into different subsets of problems, and most of these methods use the Euclidean distance between samples.

The main idea of this paper is to aggregate similar samples to obtain better results. To aggregate similar samples, we use the properties of graph neural networks (GNNs) [14]. The main contributions of this study are as follows:

- We propose a method that constructs a multilabel-based Laplacian graph such that each element in it represents the relationship between samples.

- We use similar samples with an aggregation approach that is not used in traditional multilabel learning methods.

The rest of this paper is arranged as follows. Section 2 shows the taxonomy of multilabel learning algorithms and describes their methods. Section 3 presents the details of our proposed method. Section 4 describes the multilabel datasets, evaluation metrics and experimental results, followed by the conclusions in Section 5.

2. Related work

2.1 Multilabel learning algorithms

In this section, we review multilabel learning algorithms. The algorithms that have been applied to multilabel learning over the last decade are not just those mentioned in this paper. **Figure 1** summarizes the algorithms detailed in the next section.

2.1.1 Problem transformation

Binary relevance (BR) is used to address a multilabel problem with a binary classifier, and its advantages are simplicity and efficiency, but correlation between labels is not considered. Classifier chains (CCs) are configured in a chain of binary classifiers where a classifier in the chain is based on the prediction of the previous classifier; their advantage is that they consider the relationship between labels but hence cannot be parallelized. Calibrated label ranking (CAL) performs ranking via the pairwise comparison of labels and has the advantage of considering the
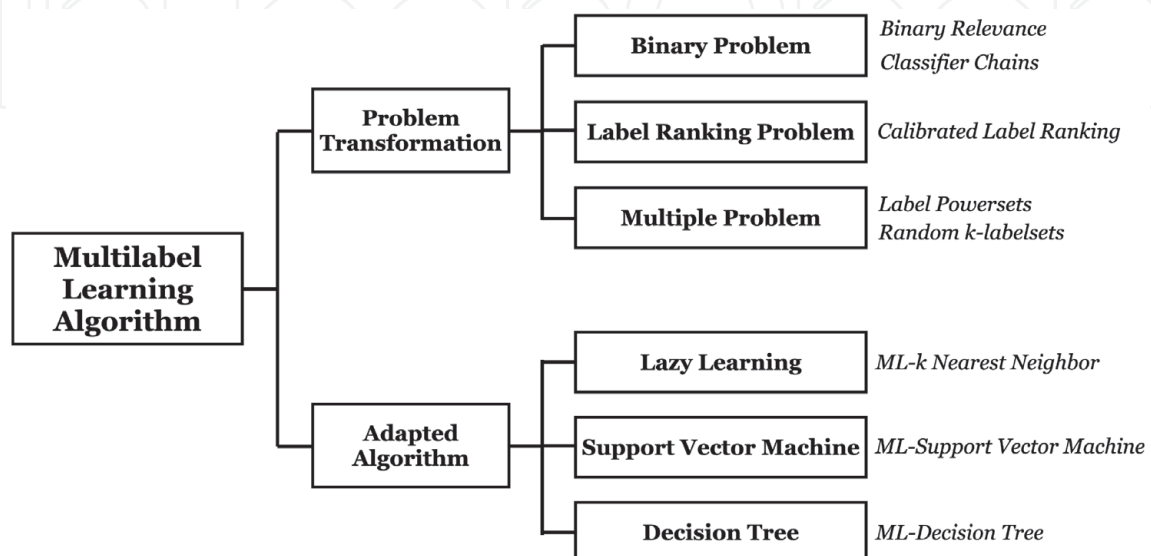

Figure 1.
Taxonomy of multilabel learning algorithms [15].

relationship (but only the pairwise relationship) between labels. Label powersets (LP) treat the situation when multiple labels belong to the same sample as a new label and have the advantage of considering the relationship between labels, but the time complexity grows exponentially with label sets. Random k-labelsets (RKL) are variants of LP models where each classifier is trained with a small random set of labels; their advantage is that they consider the relationship between labels, but they have a low accuracy rate if a worse label set combination is randomly selected.

2.1.2 Adapted algorithm

The multilabel k-nearest neighbor (MLkNN) method is derived from the traditional k-nearest neighbor algorithm. Each sample is identified with k nearest neighbors in the training set, and information is obtained from these identified neighbors. Multilabel support vector machine (ML-SVM) classification determines an optimal hyperplane that separates observations according to their labels. A multilabel decision tree (ML-DT) is constructed by building a decision tree, where each node corresponds to a set of samples in the data set.

2.2 Graph neural networks

GNNs were mentioned for the first time and further elaborated by [16]. The goal of a GNN is to learn a node's representation of the acquisition of its information by propagation. Currently, there are many deep learning tasks that need to process data with graph structures. Convolutional neural networks (CNNs) [17] have been successfully developed in the field of computer vision [18, 19] but are unable to process graph structured data [20]. The method used in this paper is called a graph convolutional network (GCN). A GCN can aggregate similar samples by propagating neighbor information, giving it the ability to infer, and there is no need to consider the sequence. GCNs have appeared in many top machine learning conferences and many applications across different tasks and domains, such as manifold learning [21, 22], computer vision [23–25], text classification [26, 27], hashing [28, 29], and hyperspectral image classification [30, 31].

3. The proposed method

This section presents the overall flow of our proposed method, as shown in **Figure 2**. The multilabel data matrix is first converted into a similarity matrix generated from a Laplacian graph. We call this a multilabel-based Laplacian graph and use this graph as inputs to the GCN model. Each node in the output layer predicts the probability of class membership for the label.

3.1 Multilabel-based Laplacian graph

This section presents the proposed method. Before this, let us describe some notational conventions. Matrices are written in boldface capital letters (e.g., $\mathbf{X}$). The transpose of a matrix is denoted as $\mathbf{X}^\top$. Vectors are written in boldface lowercase letters (e.g., $\mathbf{x}$). For a matrix $\mathbf{X} \in \mathbb{R}^{n \times m}$, the j-th column and the ij-th entry are denoted by $\mathbf{x}_j$ and x_{ij}, respectively. $\mathbf{I}$ denotes the identity matrix, $\|\cdot\|_2$ is the l_2-norm, and $\mathbf{1}$ denotes a column vector with all elements equal to ones.

Based on [32], we formally present our multilabel-based Laplacian graph. For a multilabel dataset, let $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_n] \in \mathbb{R}^{n \times m}$ be the data matrix with n and m
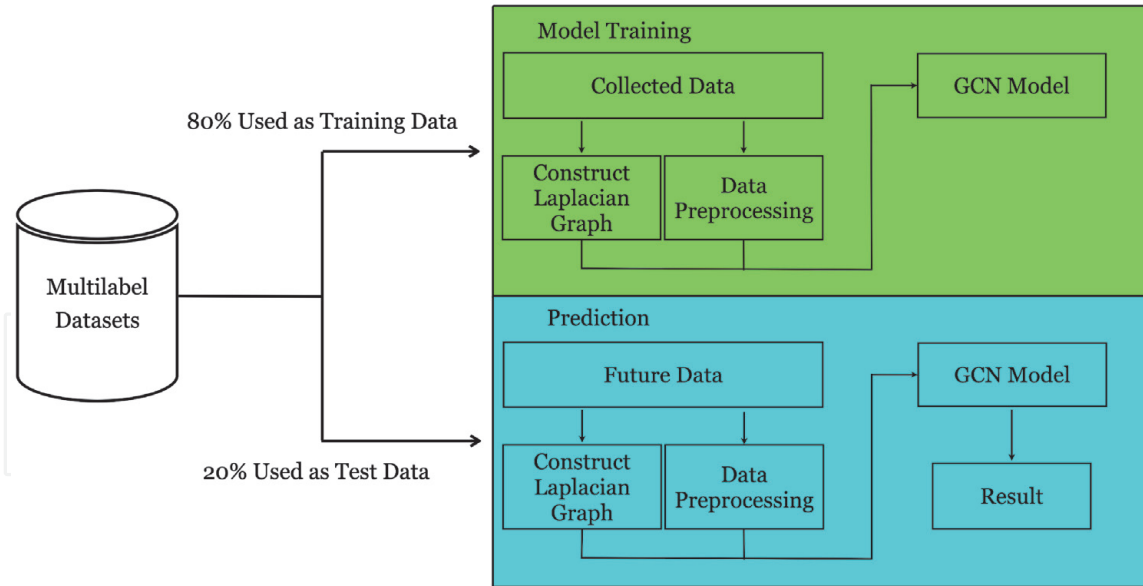
Figure 2.
An illustration of the work flow of the proposed method. Fully green color represents the training model; fully blue color represents the test model.

representing the number of samples and the dimensions, respectively. $\mathbf{S} \in \mathbb{R}^{n \times n}$ is the multilabel-based Laplacian graph, and we use a sparse representation method to construct this graph as follows:

$$\min_{\mathbf{S}} \quad \sum_{i,j=1}^{n} \left\| \mathbf{x}_i - \mathbf{x}_j \right\|_2^2 S_{ij} + \beta \sum_{i=1}^{n} \left\| \mathbf{s}_i \right\|_2^2 \tag{1}$$
$$\text{s.t.} \quad \forall S_{ii} = 0, \quad S_{ij} \geq 0, \quad \mathbf{1}^T \mathbf{s}_i = 1.$$

We normalize $\mathbf{1}^T \mathbf{s}_i = 1$ which represents a sparse constraint on $\mathbf{S}$ because sparse representation is robust to noise [33], and β is an adjustable parameter. The second term is added to regularize the loss function.

3.2 Graph convolutional network

Based on [34], we fit the GCN used for single-label classification to multilabel classification. The GCN has been modified from a first-order Chebyshev approximation [35]. In order to create a multidimensional input, ChevNet convolution with an input vector $\mathbf{x}$ and a filter g_θ is formulated as follows:

$$\mathbf{x} \star g_\theta = \theta_0 \mathbf{x} - \theta_1 \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}} \mathbf{x}, \tag{2}$$

where $\star$ means the convolution operator, $\mathbf{A}$ is the adjacency matrix and $\mathbf{D}$ is the degree matrix. By using the single parameter $\theta = \theta_0 = -\theta_1$ to avoid overfitting, Eq. (2) can be rewritten as:

$$\mathbf{x} \star g_\theta = \theta \left(\mathbf{I}_n + \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}} \right) \mathbf{x}. \tag{3}$$

Repeated use of this graph convolution operation may cause serious problems such as vanishing gradients. Therefore, $\mathbf{I}_n + \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}}$ in Eq. (3) is modified to $\tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}}$ with $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_n$ and $\tilde{\mathbf{D}}_{ii} = \sum_j \tilde{\mathbf{A}}_{ij}$, finally giving a layerwise propagation rule to support multidimensional inputs as follows:

$$\mathbf{H}^{(l+1)} = \sigma\left(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^{(l)}\mathbf{W}^{(l)}\right). \tag{4}$$

Here, $\mathbf{H}^{(l)}$ is the output of an activation function in the l-th layer of the GCN. $\mathbf{W}^{(l)}$ is a trainable weight matrix corresponding to the l-th layer of GCN. $\mathbf{H}^{(0)}$ is the data matrix. $\sigma(\cdot)$ denotes a specific activation function such as a sigmoid activation function.

This paper considers only a two-layer GCN model as the proposed method, and we modify Eq. (4) by placing the adjacent matrix into a multilabel-based Laplacian graph to obtain the formula of the two-layer GCN method proposed in this paper as follows:

$$\mathbf{H}^{(1)} = \sigma\left(\hat{\mathbf{D}}^{-\frac{1}{2}}\hat{\mathbf{S}}\hat{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^{(0)}\mathbf{W}^{(0)}\right)$$
$$\mathbf{H}^{(2)} = \sigma\left(\hat{\mathbf{D}}^{-\frac{1}{2}}\hat{\mathbf{S}}\hat{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^{(1)}\mathbf{W}^{(1)}\right), \tag{5}$$

where $\hat{\mathbf{S}} = \mathbf{S} + \mathbf{I}_n$ and $\hat{\mathbf{D}}_{ii} = \sum_j \hat{\mathbf{S}}_{ij}$. For semi-supervised multilabel classification, we evaluate the mean square error over all labeled samples:

$$\text{Mean Square Error} = \frac{1}{t}\sum_{i=1}^{t}\left(\mathbf{H}_i^{(2)} - \mathbf{Y}_i\right)^2, \tag{6}$$

where $\mathbf{Y} \in [0,1]^{n \times c}$ is the ground truth label matrix with c labelsets, and t is the number of labeled samples.

4. Experiments

4.1 Datasets

The multilabel datasets used in this paper and their associated statistics are shown in **Table 1**.

4.2 Experimental setup

In this study, we have added probabilistic classifier chains [36], CSMLC [37] and RethinkNet [38] as baselines for comparison. The experimental settings are as follows: First, multilabel datasets are preprocessed to [0,1] as inputs, 80% of the

Datasets	Domain	# of features	# of samples	# of training data	# of test data	# of classes
Emotions[*]	Audio	72	593	474	119	6
Water Quality[*]	Chemistry	16	1060	848	212	14
CIE Image[*]	Image	294	2000	1600	400	5
Natural Scenes[*]	Image	294	2407	1925	482	6
Yeast[*]	Biology	103	2417	1933	484	14
AR Face[**]	Image	1024	30303	24242	6061	6

[*]*Multilabel datasets are available at http://mulan.sourceforge.net/datasets-mlc.html*
[**]*AR Face dataset is available at http://www2.ece.ohio-state.edu/_aleix/ARdatabase.html*

Table 1.
Statistics of the multilabel datasets.

samples are used for model (both multilabel learning and proposed method) training, and the last 20% of the samples are used as test sets. We also add Gaussian noise ranging from 6% to 12% of each test sample to test the robustness of the model. The overall framework is shown in **Figure 2**.

For deep learning, we train all models for 200 epochs using Adam [39] with a learning rate of 0.01 and the mean square error as the loss function.

4.3 Evaluation metrics

In multilabel learning, the evaluation metrics must be more rigorous than traditional single-label learning because one sample may be associated with multiple labels. These evaluation metrics [15] are divided into three groups, as shown in **Figure 3**. The higher the values of the F1 score, precision, mean average precision and recall, the better the performance is. The lower the values of the Hamming loss, one-error, coverage and ranking loss, the better the performance is. We consider the Hamming loss, one-error and mean average precision as three major metrics.

4.4 Experimental results

All experiments use different combinations of training and test data to verify the trained model and average the results after repeating the training ten times. According to the observations in **Figures 4–6**, the following conclusions are reached:

- Regardless of whether the Gaussian noise is added to the data set, the classification results of the problem transformation methods (BR, CCs, CAL, LP and RKL) are almost worse than the adaptive algorithms (MLkNN, ML-SVM and ML-DT)

- Deep learning may not obtain the best performance.

- We found that our method was raised on average by 1.8% and 8% higher in Hamming loss and mean average precision, respectively. And also has excellent performance even if the dataset were contaminated by noise.

- Regardless of whether noise is added to the data, our method in one-error evaluation is not as good as other baselines.
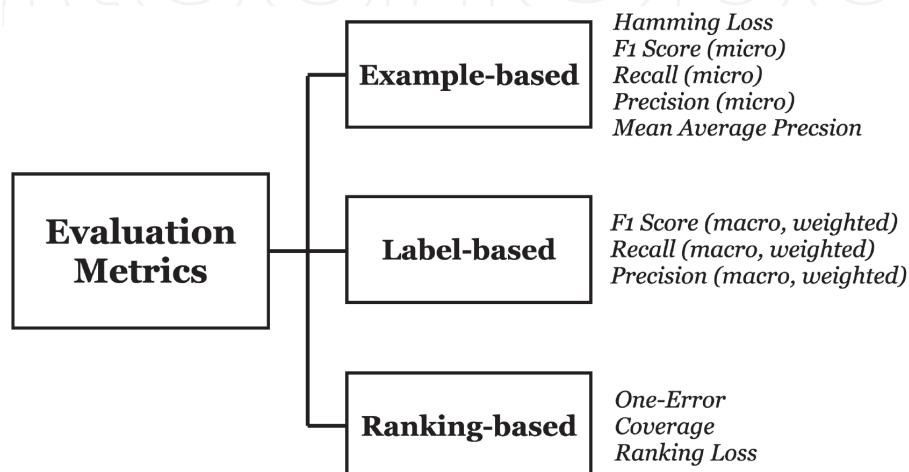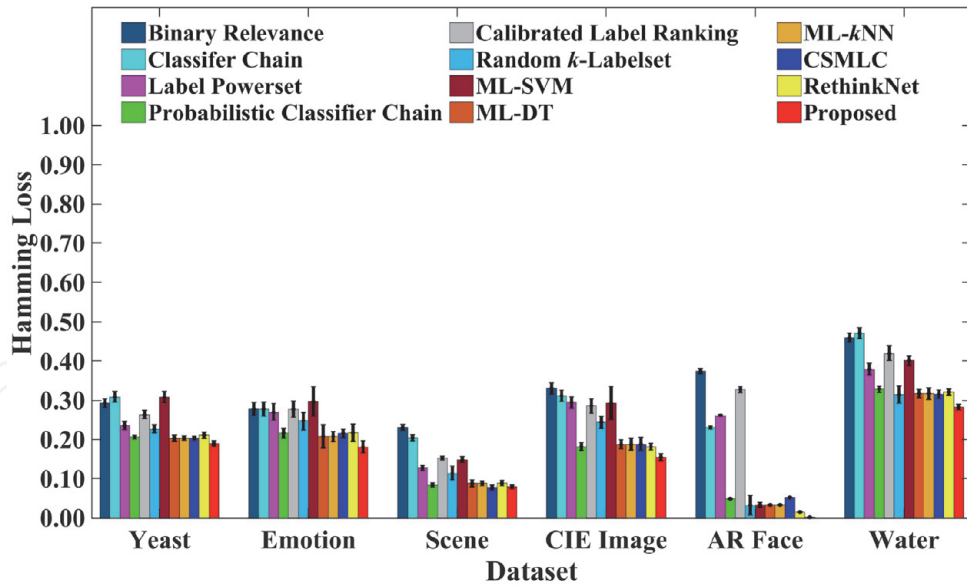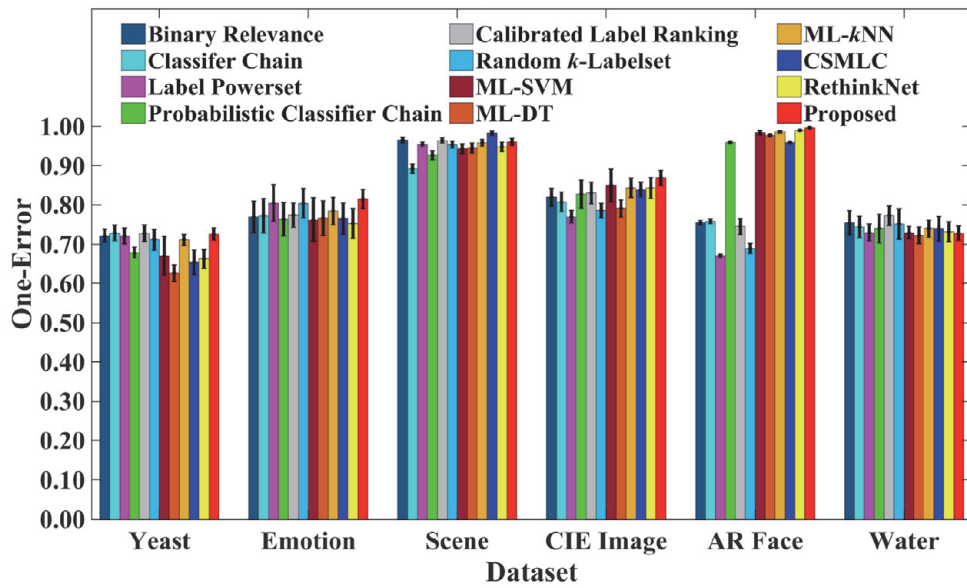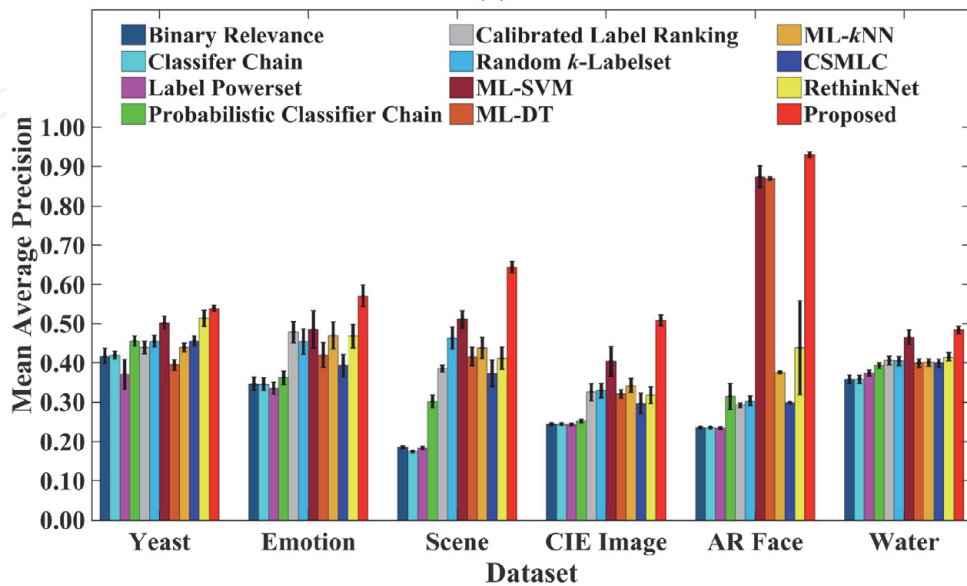

Figure 3.
Taxonomy of evaluation metrics.

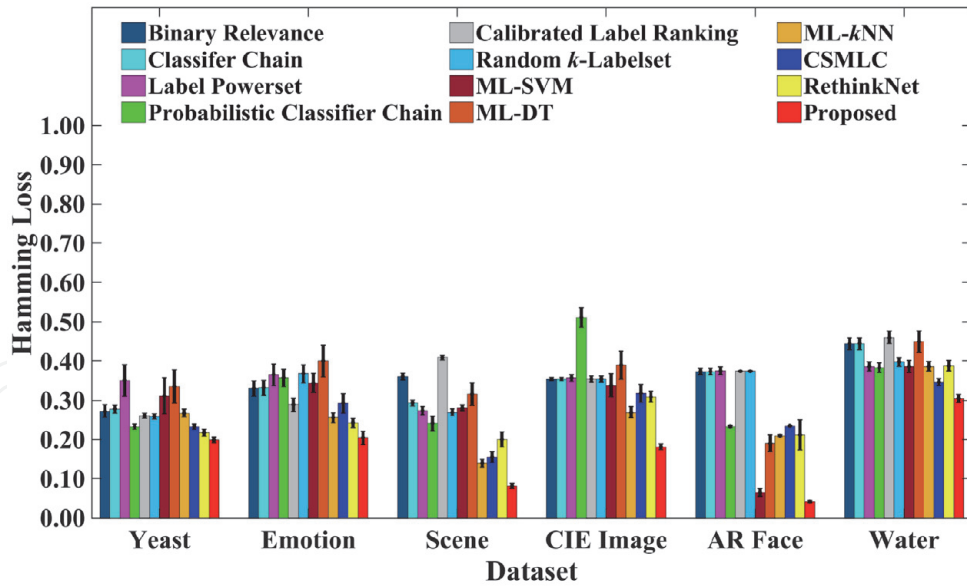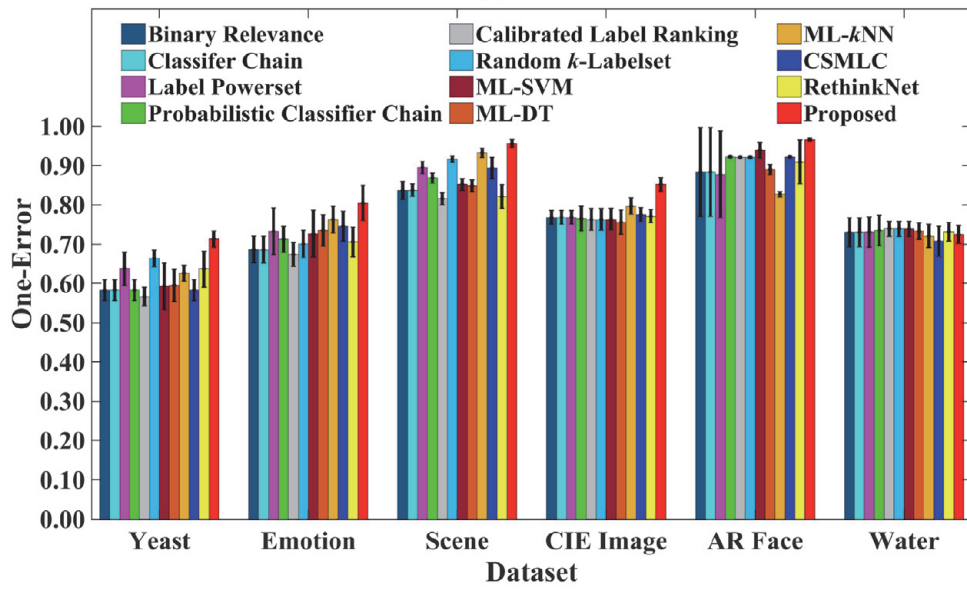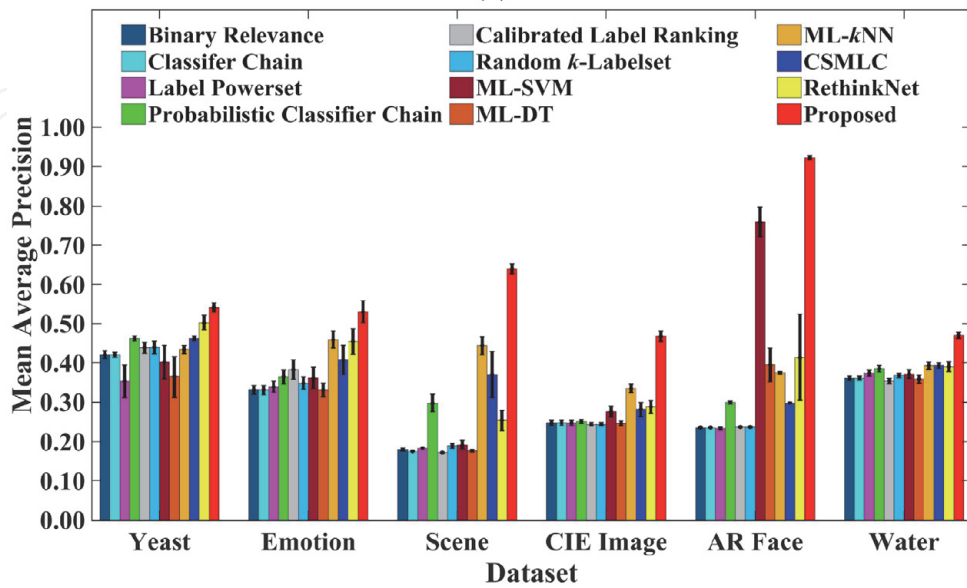**Figure 4.**
Results of the proposed method compared with multilabel learning algorithms on the used multilabel datasets. (a)–(c) show the results without adding Gaussian noise.
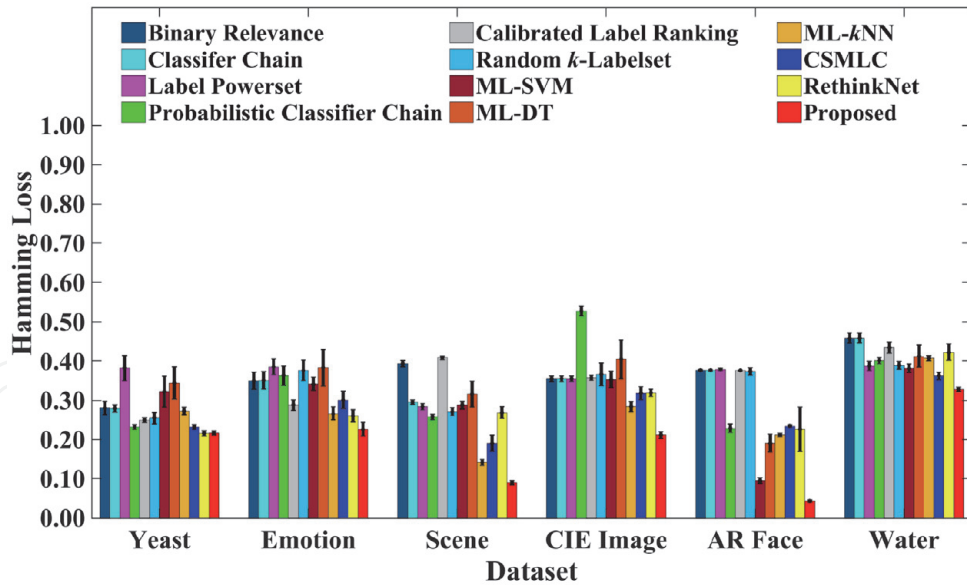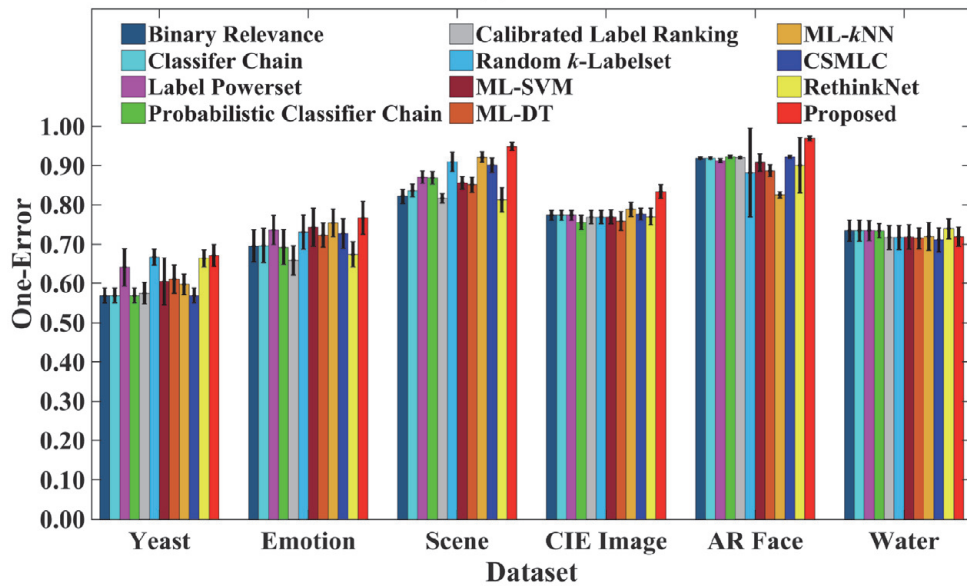
Figure 5.
Results of the proposed method compared with multilabel learning algorithms on the used multilabel datasets.
(a)–(c) show the results of adding 6% Gaussian noise.
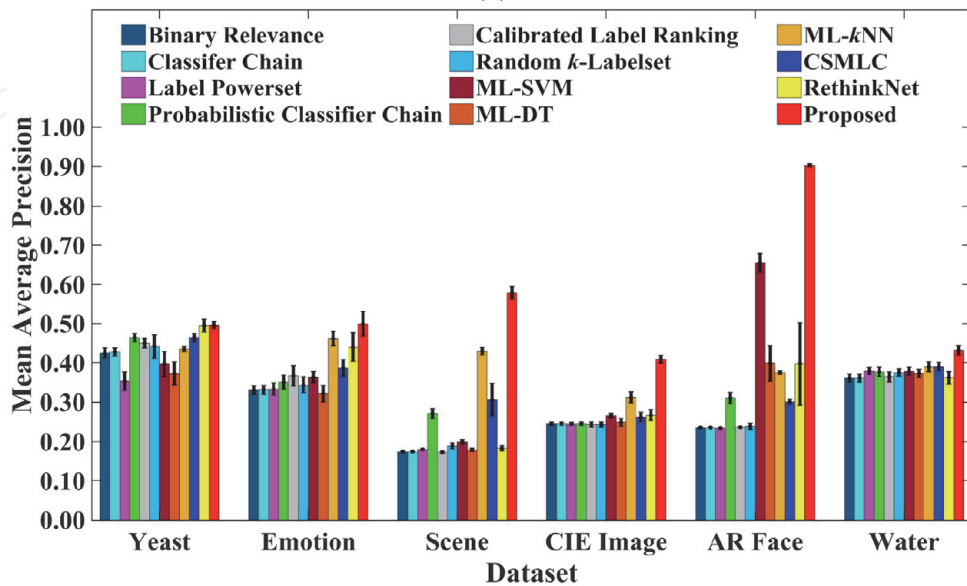
Figure 6.
Results of the proposed method compared with multilabel learning algorithms on the used multilabel datasets.
(a)–(c) show the results of adding 12% Gaussian noise.

5. Conclusions

In this paper, we proposed a method of constructing a relation matrix by considering the correlation and sparsity of paired samples. We then added the characteristics of a GCN, which aggregates similar samples, to finally obtain the probability of occurrence of each label. Experimental results on six datasets showed that our proposed method can deliver superior performance in comparison with eleven baselines. Our future work will include designing a general framework that can reduce the use of memory and increase the efficiency of a GCN and extending this framework to unsupervised learning.

Acknowledgements

Author details

Wei-Cheng Ye[1] and Jia-Ching Wang[2*]

1 Department of Electrical Engineering, National Sun Yat-sen University, Kaohsiung, Taiwan

2 Department of Computer Science and Information Engineering, National Central University, Taoyuan, Taiwan

*Address all correspondence to: jcw@csie.ncu.edu.tw

IntechOpen

References

[1] R. E. Schapire and Y. Singer, "BoosTexter: A boosting-based system for text categorization," *Machine Learning*, vol. 23, no. 3, pp. 135–168, May 2020.

[2] H. Peng, J. Li, S. Wang, L. Wang, Q. Gong, R. Yang, B. Li, P. S. Yu, and L. He,"Hierarchical taxonomy-aware and attentional graph capsule RCNNs for large-scale multi-label text classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 6, pp. 2505–2519, Jun. 2021.

[3] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, "Learning multi-label scene classification," *Pattern Recognition*, vol. 37, no. 9, pp. 1757–1771, May 2004.

[4] L. Chen, W. Zhan, W. Tian, Y. He, and Q. Zou, "Deep integration: A multi-label architecture for road scene recognition," *IEEE Transactions on Image Processing*, vol. 28, no. 10, pp. 4883–4898, Oct. 2019.

[5] B.-W. Chen, M. Imran, N. Nasser, and M. Shoaib, "Self-aware autonomous city: From sensing to planning," *IEEE Communications Magazine*, vol. 57, no. 4, pp. 33–39, Apr. 2019.

[6] W. Ji, J. Xu, H. Qiao, M. Zhou, and B. Liang, "Visual IoT: Enabling internet of things visualization in smart cities," *IEEE Network*, vol. 33, no. 2, pp. 102–110, Mar.–Apr. 2019.

[7] W. Ji, B. Liang, Y. Wang, R. Qiu, and Z. Yang, "Crowd V-IoE: Visual internet of everything architecture in AI-driven fog computing," *IEEE Wireless Communications*, vol. 27, no. 2, pp. 51–57, Apr. 2020.

[8] B.-W. Chen, "Novel kernel orthogonal partial least squares for dominant sensor data extraction," *IEEE Access*, vol. 8, pp. 36131–36139, Feb. 2020.

[9] H. Chuang, K.-L. Hou, S. Rho, and B.-W. Chen, "Cooperative comodule discovery for swarm-intelligent drone arrays," *Computer Communications*, vol. 154, pp. 528–533, Mar. 2020.

[10] B.-W. Chen, "Symmetric nonnegative matrix factorization based on box-constrained half-quadratic optimization," *IEEE Access*, vol. 8, pp. 170976–170990, Sep. 2020.

[11] W. Ji, L.-Y. Duan, X. Huang, and Y. Chai, "Astute video transmission for geographically dispersed devices in visual IoT systems," *IEEE Transactions on Mobile Computing*, Jul. 2020.

[12] A. Elisseeff and J. Weston, "A kernel method for multi-labelled classification," in *Proc. International Conference on Neural Information Processing Systems: Natural and Synthetic*, Vancouver, British Columbia, Canada, 2001, Dec. 03–08, pp. 681–687.

[13] Y.-X. Li, S. Ji, S. Kumar, J. Ye, and Z.-H. Zhou, "Drosophila gene expression pattern annotation through multi-instance multi-label learning," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 9, no. 1, pp. 98–112, Feb. 2012.

[14] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–21, Mar. 2020.

[15] M.-L. Zhang and Z.-H. Zhou, "A review on multi-label learning algorithms," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 8, pp. 1819–1837, Aug. 2014.

[16] F. Scarselli, M. Gori, A.-C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model,"

IEEE Transactions on Neural Networks, vol. 20, no. 1, pp. 61–80, Jan. 2009.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. International Conference on Neural Information Processing Systems*, Lake Tahoe, Nevada, United States, 2012, Dec. 03–06, pp. 1097–1105.

[18] X. Mai, H. Zhang, X. Jia, and M. Q.-H. Meng, "Faster R-CNN with classifier fusion for automatic detection of small fruits," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 3, pp. 1555–1569, Jul. 2020.

[19] R. Li, Q. Liu, J. Gui, D. Gu, and H. Hu, "Indoor relocalization in challenging environments with dual-stream convolutional neural networks," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 2, pp. 651–662, Apr. 2018.

[20] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad, "Collective classification in network data," *AI Magazine*, vol. 29, no. 3, p. 93, Sep. 2008.

[21] F. Monti, D. Boscaini, J. Masci, E. Rodolà, J. Svoboda, and M. M. B., "Geometric deep learning on graphs and manifolds using mixture model CNNs," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, Hawaii, United States, 2017, Jul. 21–26, pp. 5425–5434.

[22] W. Liu, S. Fu, Y. Zhou, Z.-J. Zha, and L. Nie, "Human activity recognition by manifold regularization based dynamic graph convolutional networks," *Neurocomputing*, vol. 444, pp. 217–225, Jul. 2021.

[23] Z.-M. Chen, X.-S. Wei, P. Wang, and Y. Guo, "Multi-label image recognition with graph convolutional networks," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, California, United States, 2019, Jun. 15–20, pp. 5172–5181.

[24] X. Zhang, C. Xu, and D. Tao, "Context aware graph convolution for skeleton-based action recognition," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, United States, 2020, Jun. 14–19, pp. 14333–14342.

[25] R. Zeng, W. Huang, M. Tan, Y. Rong, P. Zhao, J. Huang, and C. Gan, "Graph convolutional networks for temporal action localization," in *Proc. IEEE/CVF International Conference on Computer Vision*, Seoul, Korea, 2019, Oct. 27–Nov. 02, pp. 7094–7103.

[26] L. Yao, C. Mao, and Y. Luo, "Graph convolutional networks for text classification," in *Proc. AAAI Conference on Artificial Intelligence*, Honolulu, Hawaii, United States, 2019, Jan. 27–Feb. 01, pp. 7370–7377.

[27] R. Ragesh, S. Sellamanickam, A. Iyer, R. Bairi, and V. Lingam, "Hetegcn: Heterogeneous graph convolutional networks for text classification," in *Proc. ACM International Conference on Web Search and Data Mining*, New York, United States, 2021, Mar. 08–12, pp. 860–868.

[28] X. Zhou, F. Shen, L. Liu, W. Liu, L. Nie, Y. Yang, and H. T. Shen, "Graph convolutional network hashing," *IEEE Transactions on Cybernetics*, vol. 50, no. 4, pp. 1460–1472, Apr. 2020.

[29] R. Xu, C. Li, J. Yan, C. Deng, and X. Liu, "Graph convolutional network hashing for cross-modal retrieval," in *Proc. International Joint Conference on Artificial Intelligence*, Macao, China, 2019, Aug. 10–16, pp. 982–988.

[30] D. Hong, L. Gao, J. Yao, B. Zhang, A. Plaza, and J. Chanussot, "Graph convolutional networks for hyperspectral image classification,"

IEEE Transactions on Geoscience and Remote Sensing, pp. 1–13, Aug. 2020.

[31] A. Qin, Z. Shang, J. Tian, Y. Wang, T. Zhang, and Y. Y. Tang, "Spectral–spatial graph convolutional networks for semisupervised hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 16, no. 2, pp. 241–245, Feb. 2019.

[32] H. Wang, Y. Yang, and B. Liu, "GMC: Graph-based multi-view clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 6, pp. 1116–1129, Jun. 2019.

[33] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, Feb. 2009.

[34] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks.," in *Proc. International Conference on Learning Representations*, Toulon, France, 2017, Apr. 24–26.

[35] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, May 2013.

[36] K. Dembczyński, W. Cheng, and E. Hüllermeier, "Bayes optimal multilabel classification via probabilistic classifier chains," in *Proc. International Conference on Machine Learning*, Haifa, Israel, 2010, Jun. 21–24, pp. 279–286.

[37] K.-H. Huang and H.-T. Lin, "Cost-sensitive label embedding for multi-label classification," *Machine Learning*, vol. 106, no. 9–10, pp. 1725–1746, Oct. 2017.

[38] Y.-Y. Yang, Y.-A. Lin, H.-M. Chu, and H.-T. Lin, "Deep learning with a rethinking structure for multi-label classification," in *Proc. Asian Conference on Machine Learning*, Nagoya, Japan, 2019, Nov. 17–19, pp. 125–140.

[39] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. International Conference for Learning Representations*, San Diego, California, United States, 2015, May 07–09.