An edge prediction proposal for multi-label classification

Ávila-Jiménez, J.L.[1][0000−0001−8006−8256] and Ventura, S.[2][0000−0003−4216−6378]

[1] University of Córdoba. Electronic and Computer Engineering Department.
Leonardo Da Vinci Building. Rabanales University Campus, Córdoba, Spain 14071.
[2] University of Córdoba. Computer Science and Numerical Analisisys Department.
Charles Darwin Building. Rabanales University Campus Córdoba, Spain 14071
{jlavila,sventura}@uco.es

Abstract. Multi-label classification task, where each class can be assigned to several labels simultaneously, has been a growing research area during the last years, due to their ability to deal with many real worlds problems. Besides deep learning techniques have been extensively used in that context, where it ith worth highlighting Graph Neural Network as part of the deep learning specialised to cope with complex data. In the present work, a novedous multi-label classification technique is presented, based on using Graph Neural Network to learn the subjacent structure in multi-label datasets, and it is compared with others well-established multi-label methods.

Keywords: Multi-label Classification · Edge prediction · Graph Neural Network

1 Introduction

Many classification problems like medical diagnosis[12], text categorization[20] or anotation of images[26] can be represented as multi-label classification task (MLC)[13] where each pattern are associated with more than one label. Formally, being $\mathcal{X} = \Re^d$ a d-dimensional input space, $\mathcal{L} = \{\lambda_1, \lambda_2, ..., \lambda_q\}$ an output space of q labels and S a multi-label training set with m examples $\{(x_i, Y_i)|1 \leq i \leq m\}$ where $x_i \in \mathcal{X}$ is a d-dimensional instance which has a set of labels associated $Y_i \subseteq \mathcal{L}$, MLC is the task of learn a classifier that returns $Z_i \subseteq \mathcal{L}$ as close as posible to Y_i.

MLC problems have been dealt with from two points of view[13]. On the one hand, transformation methods, which transform a multi-label problem into single-label problems, on the other hand, adaptation methods imply adapting clasical classification paradigms to MLC.

Transformation methods obtain single label patterns from an MLC dataset to use any single-label classification algorithm, so they are independent of the underlying classification technique. They tend to produce information loss, however are commonly used because they allow working with well-known classical classification approaches[13]. Transformation techniques are Binary Relevance

(BR)[22], that generates one dataset per label, and pairwise methods that creates a new dataset for combinations of labels like Label Powerset (LP) [22]. Also ensembles, which unifies the answers of multiple classifiers to get a more robust one, are considered transformation techniques. Between them, it is worth to foreground RAkEL[18].

Adaptation techniques include many paradigms specifically adapted to deal with multi-label data without preprocessing. Almost every classification model has been adapted to be used in a multi-label context. Without the intention to be exhaustive, we can cite techniques like boosting[20], SVM[11], decision trees[7] , lazy classification(ML-kNN) [25], neural networks (BP-MLL) [8] or bioinspired proposals[1].

Graph Neural networks (GNNs) has been used to generate embedded representations of graphs or subgraphs to obtain less complicated depictions of the data but many works have focused on getting an embedding from a fixed graph [5]. However, most of the real-world tasks need to deal with dynamically generated from unseen nodes or edgesso they need inductive learning over graphs. The GraphSage Algorithm[16] allows generalizing from newly observed subgraph with the previously generated embeddings.

In the present work, we have developed a multi-label classification method, called ML-SAGE, based on GraphSage that generates an embedding from the label-Pattern graph, with is subsequently used to clasify unseen patterns. The paper is structured as follows: the first time we introduce the graph representation and the proposed algorithm, next the experimental framework is exposed, including metrics, datasets and another state of the art proposal used as a baseline to compare our work, to finish with a discussion of the results and the conclusions reached in addition with other works to carry on in the future.

2 ML-SAGE

2.1 Graph representation

Modelling dependencies between labels is one of the handicaps to obtaining accurate multi-label classifiers. The problem has been aborded from several points of view, including graphs and graph neural networks [10]. In this paper, we have used bipartite graphs where two types of nodes and edges can only connect different types of nodes, to model the relationship between patterns and labels. Formally, given a graph $G = \{V, E\}$, it is bipartite if and only if $V = V_1 \cup V_2, V_1 \cap V_2 = \emptyset$ and $v_e^1 \in V_1$ while $v_e^2 \in V_2 \ \forall \ e = (v_e^1, v_e^2) \in E$ [2].

In the present model, a whole multi-label dataset is represented by one bipartite graph, called Paterns-labels graph, composed of pattern nodes and label nodes. Edges in the graph will indicate that one pattern is associated with a label.

2.2 Proposed method

The proposed method works by creating low dimensional embeddings for pattern nodes in the Patterns-labels graph by sampling and aggregating features from their local neighbourhood. It generalizes by learning an embedding function that will be applied to unseen pattern nodes to determine which are the probable label nodes that would be connected with them. In other words, a multi-label classification task is modelled as an edge prediction problem over the patterns-labels graph.

During the training phase nodes calculates embeddings for labels and patterns nodes based on the information obtained by a sample of n of their neighbour located at a K distance, using the mean agregation (eq.1). Two loops are made with $K = 1$ and $K = 2$, to generates embedded which learns the common characteristics of patterns with the same label and also allows to learn relations between labels.

$$h_v^k = \frac{1}{|N_r(v)|} \sum_{u \in N_r(v)} D_p h_u^{k-1} \tag{1}$$

Forward pass through each layer are calculated using eq. 2, being the output of node v at layer k ${W^k}_v$ and ${W^k}_n$ trainable parameters , h_v the embedded calculated using 1 and D_p a random dropout. σ represents the activation function of the layer.

$$Output_v^k = \sigma((h_v^{k-1} + {W^k}_n {h^k}_v)) \tag{2}$$

The neural network architecture is shown in figure 1 where the connections between layers appear. Two layers calculate node embeddings, to distance 1 and 2 respectively. The node embeddings are used to generate an edge embedding that feeds a fully connected neural network which generates the classifier results. The whole net structure is training end to end. Vectors W are actualized using a back-propagation algorithm and stochastic gradient descent.

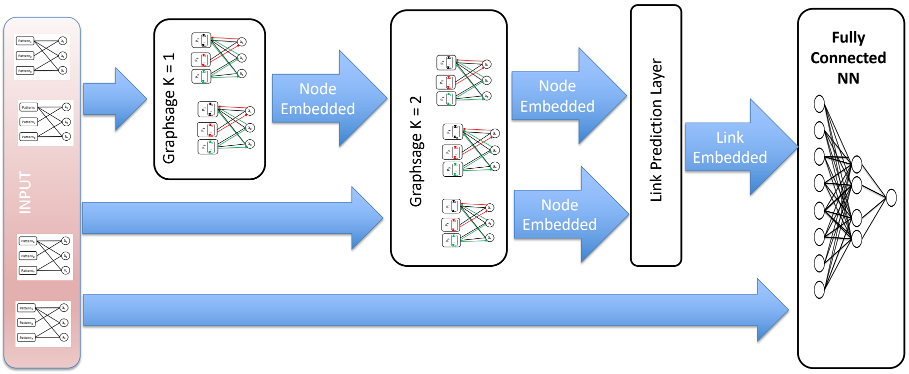

Fig. 1. Neural network architecture

In the inference process, when a new pattern is shown to the classifier, embedded is used to determine which edges fit better with the pattern.

3 Experimental setup

In this section, the main characteristics of the performed experiments are presented, including metrics, implementation details and used datasets.

The performance of our proposal has been compared to a set of representative state-of-the-art methods for multi-label classification. All algorithms have been tested using 10-fold cross-validation on all datasets.

3.1 Parameters and implementation

The proposed algorithm has been developed under the StellarGraph library [9] that implements several methods of GNN over Keras[6] and Tensorflow [14]. MULAN implementation of the comparation methods later mentioned has been used [23]. Before the main experimentation, several texts have been carried out to tune up the algorithm parameter: The optimal parameters are 300 epoch, dropout of 0.1 batch size 20 and Adam is used as optimizer with learnig ratio 0.01.

3.2 Performance metrics

Performance of trained classifiers has been compared amongs the othes using Fscore, Accuracy y Hamming Loss Metrics

The Fscore (eq. 3), harmonic mean between precision and recall, gives a good idea of the overall performace of the classifier. In MLC experiments, there is a contingency table for each label, so it is necessary to average values in the metric. In the experiments carried out, the micro approach[21] has been used to calculate FScore because it is widely used and it gives equal weight to each label.

$$FScore = \sum_{i=1}^{m} \frac{\sum_{\lambda=1}^{q} tp_{i\lambda}}{\sum_{\lambda=1}^{q} tp_{i\lambda} + \frac{1}{2}(\sum_{\lambda=1}^{q} fp_{i\lambda} + \sum_{\lambda=1}^{q} fn_{i\lambda})} \tag{3}$$

Accuracy (eq. 4) is the fraction of correctly classified label values. It is worth noting is irrelevant how it is averaged across labels[22].

$$Accuracy = \frac{1}{q}\sum_{\lambda=1}^{q}\frac{1}{m}\sum_{i=1}^{m}\frac{tp_{i\lambda}+tn_{i\lambda}}{tp_{i\lambda}+tn_{i\lambda}+fp_{i\lambda}+fn_{i\lambda}} \tag{4}$$

Hamming loss (eq. 5), considers both classification errors (predict a wrong label) and omission errors (label not predicted) definning Δ as the symmetric difference between Y_i, the set of true labels of the instance i, and P_i, labels assigned by the classifier.

$$HammingLoss = \frac{1}{m}\sum_{i=1}^{m}\frac{|Y_i \Delta P_i|}{q} \tag{5}$$

3.3 Datasets

6 datasets have been used to carry out the experiments: *birds*[4], *enron*[17], *flags*[15], *mediamill*[24],*medical*[19] and*scene*[3]. The datasets include a great variety of domains like images and audio classification, medical diagnosis or text classification.

4 Results, conclusions and future work

4.1 Experimental results

Table 1 summarize the results of the proposed algorithm compared with the others MLC proposals in terms of Accuracy, FScore and Hamming Loss.

The ML-SAGE proposal obtains, broadly speaking, better results for all metrics in most of the tested datasets. It gets better results in terms of Accuracy, FScore and Hamming Loss.

It has better values of accuracy in 3, better FScore in 5 and better Hamming Loss in 4 of the 6 tested datasets. On top of that, none of the other MLC proposals are able to defeat it consistently across metrics and datasets.

On the other hand, it is worth noting that less performance is associated with datasets like medical and enron that have a relatively high number of labels per patter(high cardinality). It can be explained by the way the embedded are built, which may make it more difficult to learn representations of relations that includes more than two labels.

4.2 Conclusions and future work

The current paper presents the ML-SAGE algorithm, a Graph Neural Network model that deals with MLC classification problems modelling it as an edge prediction task in a bipartite graph. ML-SAGE creates embeddings representations of the patterns that share the same label and combine it with embedded that represent relations between labels.

Several experiments have been made to compare the performance of the proposal with other well established MLC techniques, showing that ML-SAGE obtains better results in generating MLC classifiers among several application domains.

Further studies could be carried out to check its functioning with extreme multi-label learning, the performance of other Neural network topologies and the parameter sensitivity, especially to label cardinality. In addition, ML-SAGE can be modified to be used in a map-reduced architecture to apply it in Big Data contexts.

Acknowledgements This work has been supported by the Spanish Ministry of Science and Innovation, project PID2020-115832GB-I00. The project was also supported by the European Fund of Regional Development.

Table 1. Experimental results.

	BP-MLL	ML-kNN	BR	LP	RAkEL	ML-SAGE
	Accuracy ↑					
birds	0.124	0.590	0.125	0.107	0.563	0.649
enron	0.179	0.660	0.406	0.469	0.526	0.491
flags	0.585	0.591	0.593	0.570	0.593	0.750
mediamill	0.512	0.698	0.416	0.485	0.457	0.521
medical	0.068	0.565	0.693	0.735	0.733	0.708
scene	0.499	0.673	0.536	0.588	0.622	0.783
	FScore ↑					
birds	0.457	0.284	0.379	0.345	0.400	0.315
enron	0.250	0.562	0.483	0.418	0.569	0.661
flags	0.250	0.562	0.483	0.418	0.569	0.807
mediamill	0.487	0.585	0.393	0.545	0.496	0.971
medical	0.120	0.680	0.802	0.755	0.791	0.971
scene	0,631	0.726	0.623	0.629	0.673	0.808
	Hamming Loss ↓					
birds	0.131	0.051	0.337	0.104	0.057	0.024
enron	0.250	0.047	0.042	0.054	0.041	0.028
flags	0.275	0.289	0.275	0.297	0.275	0.097
mediamill	0.043	0.123	0.047	0.083	0.47	0.020
medical	0.312	0.015	0.010	0.012	0.009	0.021
scene	0.141	0.085	0.156	0.127	0.102	0.169

References

1. Ávila, J.L., Gibaja, E.L., Zafra, A., Ventura, S.: Journal of Multiple-Valued Logic and Soft Computing. Journal of Multiple-Valued Logic and Soft Computing **17**(2-3), 183–206 (2011)
2. Bhatia, K., Jain, H., Kar, P., Varma, M., Jain, P.: Sparse local embeddings for extreme multi-label classification. In: Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 28. Curran Associates, Inc. (2015),
3. Boutell, M.R., Luo, J., Shen, X., Brown, C.M.: Learning multi-label scene classification. Pattern Recognition **37**(9), 1757–1771 (2004)
4. Briggs, F., Huang, Y., Raich, R., Eftaxias, K., Lei, Z., Cukierski, W., Hadley, S., Hadley, A., Betts, M., Fern, X., Irvine, J., Neal, L., Thomas, A., Fodor, G., Tsoumakas, G., Huttunen, H., Ruusuvuori, P., Manninen, T., Diment, A., Virtanen, T.: The 9th annual mlsp competition: New methods for acoustic classification of multiple simultaneous bird species in a noisy environment. In: 2013 IEEE International Workshop on Machine Learning for Signal Processing, 22-25 September 2013, Southampton, UK. IEEE International Workshop on Machine Learning for Signal Processing (2013). Publisher name: Institute of Electrical and Electronics Engineers IEEE
5. Cao, S., Lu, W., Xu, Q.: Grarep: Learning graph representations with global structural information. In: Proceedings of the 24th ACM International on Conference

on Information and Knowledge Management. p. 891–900. CIKM '15, Association for Computing Machinery, New York, NY, USA (2015).
6. Chollet, F., et al.: Keras (2015), https://github.com/fchollet/keras
7. Clare, A., King, R.D.: Knowledge Discovery in Multi-label Phenotype Data. Lecture Notes in Computer Science **2168**, 42–53 (2001),
8. Crammer, K., Singer, Y.: A family of additive online algorithms for category ranking. The Journal of Machine Learning Research **3**, 1025–1058 (2003)
9. Data61, C.: Stellargraph machine learning library. https://github.com/stellargraph/stellargraph (2018)
10. Do, K., 0001, T.T., Nguyen, T., Venkatesh, S.: Attentional multilabel learning over graphs: a message passing approach. Mach. Learn **108**(10), 1757–1781 (2019)
11. Elisseeff, A., Weston, J.: A Kernel Method for Multi-Labelled Classification. Advances in Neural Information Processing Systems **14**, 681–687 (2001)
12. Gérardin, C., Vaillant, P., Wajsbürt, P., Gilavert, C., Bellamine, A., Kempf, E., Tannier, X.: Classification multilabel de concepts médicaux pour l'identification du profil clinique du patient (multilabel classification of medical concepts for patient's clinical profile identification). In: Grouin, C., Grabar, N., Illouz, G. (eds.) Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles. Atelier DÉfi Fouille de Textes, DEFT@TALN 2021, Lille, France, June 28 - July 2, 2021. pp. 21–30. ATALA (2021),
13. Gibaja, E., Ventura, S.: A tutorial on multilabel learning. ACM Comput. Surv. **47**(3) (apr 2015).
14. Girija, S.S.: Tensorflow: Large-scale machine learning on heterogeneous distributed systems. Software available from tensorflow. org **39**(9) (2016)
15. Goncalves, E.C., Plastino, A., Freitas, A.A.: A genetic algorithm for optimizing the label ordering in multi-label classifier chains. In: 2013 IEEE 25th International Conference on Tools with Artificial Intelligence. pp. 469–476 (2013).
16. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017),
17. Keila, P., Skillicorn, D.: Structure in the enron email dataset. Computational & Mathematical Organization Theory **11**(3), 183–199 (October 2005)
18. Partalas, I., Tsoumakas, G., Katakis, I., Vlahavas, I.: Ensemble pruning using reinforcement learning. Advances in Artificial Intelligence pp. 301–310 (2006)
19. Sasaki, Y., Rea, B., Ananiadou, S.: Multi-topic aspects in clinical text classification. In: BIBM '07: Proceedings of the 2007 IEEE International Conference on Bioinformatics and Biomedicine. pp. 62–70. IEEE Computer Society, Washington, DC, USA (2007)
20. Schapire: Boostexter: a boosting-based system for text categorization. Machine Learning **39**(2/3), 135–168 (2000)
21. Tsoumakas, G., Katakis, I.: Multi Label Classification: An Overview. International Journal of Data Warehousing and Mining **3**(3), 1–13 (2007)
22. Tsoumakas, G., Katakis, I., Vlahavas, I.: Data Mining and Knowledge Discovery Handbook, chap. Mining Multi-label Data. Springer (2009)
23. Tsoumakas, G., Spyromitros-Xioufis, E., Vilcek, J., Vlahavas, I.: Mulan: A java library for multi-label learning. J. Mach. Learn. Res. **12**(null), 2411–2414 (jul 2011)
24. Worring, M., Snoek, C.G.M., de Rooij, O., Nguyen, G.P., Smeulders, A.W.M.: The mediamill semantic video search engine. In: Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on. vol. 4, pp. IV–1213–IV–1216 (2007)

25. Zhang, M.L., Zhou, Z.H.: ML-KNN: A lazy learning approach to multi-label learning. Pattern Recognition **40**(7), 2038–2048 (Jul 2007).
26. Zhu, P., Tan, Y., Zhang, L., Wang, Y., Mei, J., Liu, H., Wu, M.: Deep learning for multilabel remote sensing image annotation with dual-level semantic concepts. IEEE Trans. Geosci. Remote. Sens. **58**(6), 4047–4060 (2020).