ATLANTIS PRESS

Advances in Intelligent Systems Research (AISR), volume 155

2018 International Conference on Advanced Control, Automation and Artificial Intelligence (ACAAI 2018)

A Multi-label Classifier for Human Protein Subcellular Localization Based on LSTM Networks

Zhiying Gao[1], Lijun Sun[2,*] and Zhihua Wei[3]

[1] Department of Computer Science and Technology, Tongji University, Shanghai, China

[2]Research Center of Big Data and Network Security, Tongji University, Shanghai200092, China

[3]Key Laboratory of Embedded System and Service Computing, Tongji University, Shanghai201804, China

[*]Corresponding author

***Abstract*—Nowadays, with the increasing number of protein sequences all over the world, more and more people are paying their attention to predicting protein subcellular location. Since wet experiment is costly and time-consuming, the automatic computational methods are urgent. In this paper, we propose a variant model based on Long Short-Term Memory (LSTM) to predict protein subcellular location. In this model, we employ LSTM to capture long distance dependency features of the sequence data. Moreover, we adjust the loss function of the loss layer to solve multi-label classification problem. Experimental results demonstrate that, compared with the traditional machine learning methods, our method achieves the best performance in various evaluation metrics.**

Keywords-LSTM; multi-label classification; protein subcellular localization

I. INTRODUCTION

Subcellular localization of a new protein is very important for functional realization. However, in post-genome era, protein sequences grow rapidly. It is expensive and time-consuming to use the traditional biochemical methods, such as cell separation method, electronic microscopy, and fluorescence microscopy, to predict protein subcellular location. The use of automatic computational techniques to quickly predict the subcellular locations of proteins is very necessary.

A protein usually has more than one subcellular site, so that it usually can not only be marked by a single label. It is necessary to consider the situation of multi-label annotations. Multi-label classification is an automatic approach for addressing such problems by learning to assign a suitable subset of categories to a given text. In the literature, one can find a number of multi-label classification approaches for a variety of tasks in different domains such as bioinformatics, music, and text.

There are two main approaches to solve multi-label classification problems: (1) Algorithm independency. Through the decomposition of the data sets, a multi-label learning problem is transformed into multiple binary classification or multi-class classification problem, after that we can deal with and integrate each single label classification result as a multi-label learning result. The key of this approach is still the ordinary single label classification, which has no relationship with multi-label learning algorithm. (2) Algorithm dependency, which is based on the single label classification algorithm, can be effectively extended and reconstructed to deal with multi-label classification problems. The common algorithms are Rank-SVM and ML-KNN. This paper uses deep learning model to solve multi-label classification problems.

Recent researches show that neural networks do better in classification than traditional methods. There are several neural networks having been proposed for multi-label classification because they are able to capture label dependencies in the output layer [1]. Inspired by this, we propose a variant model based on Long Short-Term Memory which can deal with multi-label classification. In our model, LSTM can be used to capture long distance dependency features of the sequence data. By changing the loss function, our model achieves good performance on the protein dataset.

The rest of this paper is organized as follows. Section II presents related work. Section III describes the dataset and the proposed model. In Section IV, the experiment results are presented and discussed. Finally, Section V concludes our work.

II. RELATED WORK

Since Nakai K etc. developed an expert system that makes use of various kinds of knowledge organized as “if-then” rules for predicting protein localization sites in Gram-negative bacteria in 1991[2], the related research of protein subcellular localization prediction based on computational methods has begun.

It is found that the focus of protein subcellular localization research is on the selection of sequence feature and the selection of prediction algorithm. Protein feature extraction methods mainly include amino acid composition, dipeptide composition, pseudo-amino acid composition, position specific score matrix, functional domain composition, gene annotation etc.

Chou etc. proposed a covariant discriminant algorithm to predict the subcellular location of a query protein according to its amino acid composition [3]. Z Yuan constructed Markov chain models to the subcellular location, which achieved a higher prediction accuracy than previous methods based on the amino acid composition [4]. In [5-8], they used Support Vector Machine (SVM) in this classification problem. This prediction method is roughly based on two kinds of feature extraction: (1) features based on protein sequence data; (2) features based on ontology data [9]. Furthermore, Liqi Li etc. combined K-

Nnearest Nneighbour (KNN) and Support Vector Machine (SVM) to predict the subcellular localization of eukaryotic proteins from the GO (gene ontology) annotations [10].Jian Guo etc. proposed a novel method for protein subcellular localization based on boosting and probabilistic neural network (PNN). This novel method provides superior prediction performance compared with other existing algorithms based on amino acid composition and can be a complementing method to other existing methods based on sorting signals [11]. All these works regarded the study of protein location as a single label problem. However, a protein can be found in two or more subcellular locations in the experimental data, so it is typically a multi-label classification problem. Recently, X Guo used ensemble multi-label learning techniques to integrate several latest databases to improve prediction performance. Dong Ren adopted feature reduction method to reduce the feature dimension and use MIML algorithm to implement the multi-label classification [12].

With the promotion of deep neural networks, some people use deep neural networks to predict protein subcellular location. For example, Søren Kaae Sønderby used Convolutional LSTM Networks to predict protein subcellular location. However, they use it to predict the single-site protein problem. Different from that, we use LSTM Networks to predict multi-label protein, which will be described in detail in the next section

III. MATERRIALS AND METHOD

A. Dataset

The dataset used in this work are obtained from [13].There are 4802 different protein sequences, among which 3448 are single-location proteins, 1354 are multiple-location proteins. The protein sequences used in this work are provided as a set of flat files in fasta format, each sequence is represented as an identifier and lines of data.

These protein sequences are distributed in 37 subcellular locations: (1)Lipid Particles, (2)Extracellular, (3)Early Endosomes, (4)Endoplasmic Reticulum, (5)Nuclear Envelope, (6)Mitochondria, (7)Cytoplasmic Vesicles, (8)Centrosome, (9)Endosomes, (10)Cellular Component Unknown, (11)Golgi Cis Cisterna, (12)Cytoskeleton, (13)Transport Vesicle, (14)Microtubule, (15)Peroxisome, (16)Cytoplasm, (17)Outer Mitochondria Membrane, (18)Nucleolus, (19)Apical Plasma Membrane, (20)Melanosome, (21)Late Endosomes, (22)Golgi Trans Face, (23)Secretory Vesicles, (24)Golgi Trans Cisterna, (25)Plasma Membrane, (26)Tight Junction, (27)Medial Golgi, (28)ERGIC, (29)Microtubule Organizing Center, (30)Inner Mitochondrial Membrane, (31)Secretory Granule, (32)Sarcolemma, (33)Golgi Apparatus, (34)Basolateral Plasma Membrane, (35)Synaptic Vesicles, (36)Lysosomes, (37)Nucleus.

B. Features

In order to develop a powerful predictor for identifying subcellular localization based on the sequence information, the most important thing is to extract enough feature information on subcellular locations of proteins. By extracting features from the sequences, we can represent them with a new set of vectors to train the model.

A protein P with a sequence of L amino acid residues. The P can be expressed as:

$$P = R_1 R_2 R_3 \mathrm{L}\ R_L \tag{1}$$

Where R_1 represents the residue at chain position 1, it is the same with other residues.

To completely avoid losing its sequence order information, we use the pseudo amino acid composition (PseAAC) to represent the sequence of a protein.

After feature extraction, the transformed vector can be formulated as:

$$P = [p_1, p_2, \mathrm{L}\ p_{20}, p_{20+1}, \mathrm{L}\ p_{20+\lambda}]^T \tag{2}$$

Where the first 20 elements characterize the contents of the 20 amino acids in the sequence, while the additional λ dimensional feature vector reveals the effect of residue in protein sequences on the physical and chemical property of protein.

The PseAAC for a protein P can be generally formulated as:

$$p_u = \begin{cases} \dfrac{f_u}{\sum_{i=1}^{20} f_i + w\sum_{j=1}^{\lambda} \delta_j}, (1 \le u \le 20) \\ \dfrac{w\delta_u}{\sum_{i=1}^{20} f_i + w\sum_{j=1}^{\lambda} \delta j}, (20+1 \le u \le 20+\lambda; \lambda < L) \end{cases} \tag{3}$$

In the above formula, f_i is the frequency of the 20 amino acids in the protein sequence [14]. w is the weight factor. δ_λ is the correlation factor of the protein sequence with the most adjacent λ amino acid residues.

C. Methodology

1) Long Short Term Memory Networks (LSTM)

The purpose of Recurrent Neural Networks (RNN) is to process sequence data. In a traditional neural network, we suppose that all inputs (and outputs) are independent of each other. Namely, from the input layer to the hidden layer and then to the output layer, the nodes between the adjacent layers are fully connected, but the nodes within one layer has no connection. This kind of ordinary neural network is not competitive for many problems. RNN are recurrent because they perform the same task for each element of the sequence, and the output depends on the previous computations. Another way to consider RNN is that they have a "memory" that captures the information that has been calculated so far. In theory, RNN can use information in arbitrarily long sequence, but in practice they are limited to a few steps back.

The most commonly used RNN type is LSTM, which is much better at capturing long term temporal dependencies than

RNN. An LSTM has three gates to protect and control the cell state: a forget gate f to control whether to forget the current state; an input gate i to control whether to read the input; an output gate o to control whether to output the state [15].The memory cell of the LSTM network is shown in Figure I. A single LSTM layer is formulated as:

$$\begin{aligned} i_t &= \sigma(W_x \cdot [h_{t-1}, x_t] + b_i) \\ f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\ g_t &= \tanh(W_g \cdot [h_{t-1}, x_t] + b_g) \\ c_t &= f_t \text{ e } c_{t-1} + i_t \text{ e } g_t \\ o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\ h_t &= o_t \text{ e } \tanh(c_t) \end{aligned} \tag{4}$$

Where $\sigma(\cdot)$ is an activation function, e is elementwise multiplication, and W matrices are learned parameters. Note that for the first hidden layer h_t^1 the input x_t is the amino acid feature [16].

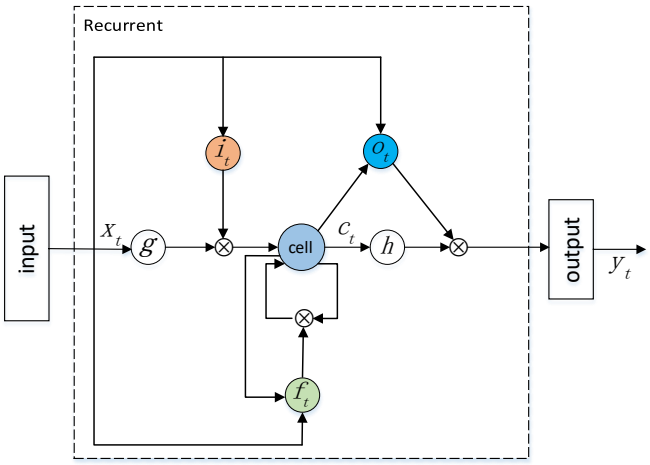

FIGURE I. LSTM ARCHITECTURE

2) *Model*

We use LSTM framework for multi-label classification task. For original feature sequence, the LSTM can employ context information. Then we process the word vector with a different size window for one-dimensional convolution. This method is similar to implicit n-gram.

It is necessary to design a loss function that can compare multiple labels to the actual labels. Combining the loss function of the single label classification and the various situations of multi-label loss function, we design a loss function based on multi-label ranking.

Softmax regression can be seen as a generalization of logistic regression on classification issues. Logistic regression is often used on binary classification. However, softmax regression can take k different values. When k is 2, softmax regression is equal to logistic regression.

Assume that there are C categories, N protein sample $\{x_1, x_2, \ldots, x_i, \ldots, x_N\}$, where each x_i denotes the $i-th$ protein sequence. Let $f_j(x_i)$ be the activation value of the protein sequence x_i of label j, then the posteriori probability p_{ij} of the protein sequence x_i been annotated as label j is:

$$p_{ij} = \frac{\exp(f_j(x_i))}{\sum_{k=1}^{C} \exp(f_k(x_i))} \tag{5}$$

Then minimize Kullback-Leibler (KL) divergence between the predicted label and the actual label. KL divergence, also known as relative entropy, is an asymmetric measure of the difference between the two probability distributions, which can be used to measure the difference between the two probability distributions.

For multiple labels of each protein sequence, we suppose that each label is independent of others. Let $y = \{y_1, y_2, \ldots, y_C\}$ be the label vector, where y_i means the value of label *i*, $y_i = 1$ represents that the protein sequence is own to the $j-th$ category and $y_i = 0$ represents that the protein sequence is not own to the $j-th$ category. According to normalization, the probability $\hat{p}_{ij}$ of the protein sequence x_i been annotated as label j is:

$$\hat{p}_{ij} = \frac{y}{\|y\|_1} \tag{6}$$

The cost function is:

$$J = -\frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{c} \hat{p}_{ij} \log(p_{ij}) \tag{7}$$

According to the above, we use J as the loss function to train model. When the network is trained, the result of the loss function will be transmitted in reverse. In the test stage, the optimal number of labels will be selected according to the loss function and be compared with the actual labels. In order to avoid the gradient vanishing or exploding, we apply the rmsprop optimization algorithm [17].

IV. EXPERIMENTS

A. *Evaluation*

The evaluation of methods learned from multi-label data requires different measures from those used in the case of single-label data [18]. There are two main evaluation metrics.

One is based on Binary Prediction measures, another is based on Rank-based evaluation metrics.

1) *Binary prediction measures*

Based on binary prediction measures, we use the standard recall, precision and F1 measures to evaluate the effectiveness of classification. Formula (8) is precision measure, Formula (9) is recall measure and Formula (10) is F1 measure.

$$\text{Precision}=\frac{1}{|\mathrm{D}|}\sum_{i=1}^{|\mathrm{D}|}\frac{|Y_i \cap Z_i|}{|Z_i|} \tag{8}$$

$$\text{Recall}=\frac{1}{|\mathrm{D}|}\sum_{i=1}^{|\mathrm{D}|}\frac{|Y_i \cap Z_i|}{|Y_i|} \tag{9}$$

$$\text{Fbeta}=\frac{(1+\beta^2)*(\text{Precision}*\text{Recall})}{(\beta^2*\text{Precision}+\text{Recall})} \tag{10}$$

Where D denotes the multi-label evaluation data set. There are $|D|$ multiple label samples. $Y_i \subseteq L$ is the set of true labels and L is the label set. Given instance x_i, the set of labels that are predicted by a multi-label method is denoted as z_i.

2) *Rank-based evaluation metrics*

Based on rank-based evaluation metrics, LRAP（label ranking average precision）was employed. This index is based on the notion of label ranking instead of precision and recall. This index will yield better scores if you are able to give better rank to the labels associated with each sample. The value is between 0 and 1. The higher value, the better performance:

$$AP(f)=\frac{1}{|T|}\sum_{i=1}^{|T|}\left(\frac{1}{|Y(x_i)|}\sum_{\lambda\in Y(x_i)}\frac{|\lambda' \mid rank_f(x_i,\lambda')\le rank_f(x_i,\lambda),\lambda'\in Y(x_i)|}{rank_f(x_i,\lambda)}\right) \tag{11}$$

Here T is the number of all samples; $|Y(x_i)|$ is the number of the sample with label $|Y(x_i)|$; $rank(x_i,\lambda)$ means the probability of the sample x_i with label λ.

B. *Result and Discussion*

Table 1 shows the comparison result of our model with several traditional methods, such as MLKNN and BRKNN. The experimental result denotes that our proposed method works better than all the traditional methods used in [13]. Our proposed method achieves 69.1% average precision, almost 15% higher than MLKNN, which is the best result in [13]. Our method achieves higher Micro-averaged F-Measure score than traditional methods, about 16% improvement. In addition, our method achieves higher Macro-averaged F-Measure score (about 25% improvement), higher Macro-averaged Precision score (about 12% improvement) and higher Micro-averaged Precision score (about 2% improvement).

In Figure II, we compare the result of our model with several ensemble approaches, such as MeanEnsemble (ME), MajorityVoteEnsemble (MVE) and TopKEnsemble (TopK-E). We can see that the proposed method performs better than these ensemble approaches.

TABLE I: AP VALUE COMPARISON ON 9 DIFFERENT BASIC MULTI-LABEL CLASSIFIERS IN 420-D FEATURES.

Methods	Average Precision	Micro-averaged F-Measure	Macro-averaged F-Measure	Macro-averaged Precision	Micro-averaged Precision
RF	0.418	0.155	0.214	0.256	0.574
J48	0.481	0.189	0.290	0.222	0.400
IBK	0.378	0.161	0.276	0.207	0.467
MLKNN	0.545	0.148	0.151	0.175	0.570
BRKNN	0.525	0.147	0.184	0.151	0.588
HOMER	0.309	0.159	0.256	0.159	0.253
BRKNN	0.525	0.147	0.184	0.151	0.588
HOMER	0.309	0.159	0.256	0.159	0.253
LSTM	**0.691**	**0.301**	**0.402**	**0.391**	**0.592**

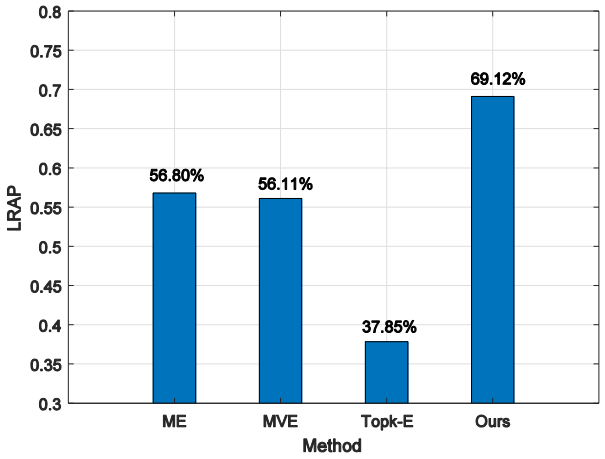

FIGURE II. LRAP VALUE COMPARISON WITH ENSEMBLE METHOD

V. CONCLUSION

In this paper, we introduced an automatic computational method based on deep learning to predict protein subcellular location. First and foremost, we utilized LSTM to obtain long distance dependency features of the sequence data to deal with protein subcellular location problems. Furthermore, considering the various situations of multi-label loss function, we adjusted the loss function of single label network to solve multi-label classification problems. Through comparing with several existing traditional machine learning methods, the results demonstrated the superiority of our method. In future work, we will focus on processing unbalanced data.

ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China under Grant 61573259, 61673301, 61573255 and 61673299.

REFERENCES

[1] Nam J, Kim J, Mencía E L, et al. Large-scale multi-label text classification—revisiting neural networks[C]//Joint european conference on machine learning and knowledge discovery in databases. Springer, Berlin, Heidelberg, 2014: 437-452.

[2] Nakai K, Kanehisa M. Expert system for predicting protein localization sites in gram - negative bacteria[J]. Proteins: Structure, Function, and Bioinformatics, 1991, 11(2): 95-110.

[3] Chou, Kuo-Chen, and David W. Elrod. "Protein subcellular location prediction." Protein engineering 12.2 (1999): 107-118.

[4] Yuan Z. Prediction of protein subcellular locations using Markov chain models.[J]. Febs Letters, 1999, 451(1):23-26.

[5] Hua S, Sun Z. Support vector machine approach for protein subcellular localization prediction[J]. Bioinformatics, 2001, 17(8):721-8.

[6] Li, Yan Fu, and J. Liu. "Predicting Subcellular Localization of Proteins Using Support Vector Machine with N-Terminal Amino Composition."International Conference on Advanced Data Mining and ApplicationsSpringer Berlin Heidelberg, 2005:618-625.

[7] Wan S, Mak M W, Kung S Y. GOASVM: Protein subcellular localization prediction based on Gene ontology annotation and SVM[C]// IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2012:2229-2232.

[8] Rahman J, Mondal M N, Islam M K, et al. Feature Fusion Based SVM Classifier for Protein Subcellular Localization Prediction.[J]. Journal of Integrative Bioinformatics, 2016, 13(1):23-33.

[9] Kim J K, Raghava G P S, Bang S Y, et al. Prediction of subcellular localization of proteins using pairwise sequence alignment and support vector machine[J]. Pattern Recognition Letters, 2006, 27(9):996-1001.

[10] Liqi Li, Hong Kuang, Yuan Zhang, Yue Zhou, et al. Prediction of eukaryotic protein subcellular multi-localisation with a combined KNN-SVM ensemble classifier.[J].Journal of Computational Biology and Bioinformatics Research , 2011, Vol. 3(2):15-24.

[11] Guo, J., Lin, Y., & Sun, Z. (2004). A Novel Method for Protein Subcellular Localization Based on Boosting and Probabilistic Neural Network. APBC.

[12] Ren D, Ma L, Zhang Y, et al. Online biomedical publication classification using multi-instance multi-label algorithms with feature reduction[C]//Cognitive Informatics & Cognitive Computing (ICCI* CC), 2015 IEEE 14th International Conference on. IEEE, 2015: 234-241

[13] Guo X, Liu F, Ju Y, et al. Human Protein Subcellular Localization with Integrated Source and Multi-label Ensemble Classifier[J]. Scientific Reports, 2016, 6.

[14] Xiao X, Wu Z C, Chou K C. A Multi-Label Classifier for Predicting the Subcellular Localization of Gram-Negative Bacterial Proteins with Both Single and Multiple Sites[J]. Plos One, 2011, 6(6):e20592.

[15] Wang, Jiang, et al. "Cnn-rnn: A unified framework for multi-label image classification." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.

[16] Sønderby S K, Sønderby C K, Nielsen H, et al. Convolutional LSTM networks for subcellular localization of proteins[C]//International Conference on Algorithms for Computational Biology. Springer, Cham, 2015: 68-80.

[17] T. Tieleman and G. Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning, 4, 2012. 5

[18] Tsoumakas G, Katakis I, Vlahavas I. Mining multi-label data[M]//Data mining and knowledge discovery handbook. Springer, Boston, MA, 2009: 667-685.