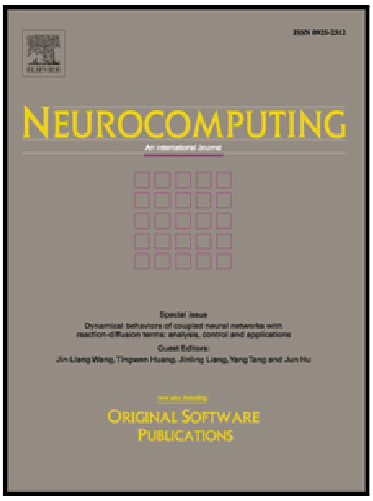

Multi-label Text Categorization Using L_{21}-norm Minimization Extreme Learning Machine

Mingchu Jiang, Zhisong Pan, Na Li

PII: S0925-2312(17)30201-1
DOI: 10.1016/j.neucom.2016.04.069
Reference: NEUCOM 18000

To appear in: *Neurocomputing*

Received date: 30 September 2015
Revised date: 26 January 2016
Accepted date: 20 April 2016

Please cite this article as: Mingchu Jiang, Zhisong Pan, Na Li, Multi-label Text Categorization Using L_{21}-norm Minimization Extreme Learning Machine, *Neurocomputing* (2017), doi: 10.1016/j.neucom.2016.04.069

Multi-label Text Categorization Using L_{21}-norm Minimization Extreme Learning Machine

Mingchu Jiang, Zhisong Pan*, Na Li

College of Command Information System, PLA University of Science and Technology, Nanjing, China, 210007

Abstract

Extreme learning machine (ELM) is extended from the generalized single hidden layer feedforward networks where the input weights of the hidden layer nodes can be assigned randomly. It has been widely used for its much faster learning speed and less manual works. Considering the field of multi-label text classification, in this paper, we propose an ELM based algorithm combined with L_{21}-norm minimization of the output weights matrix called L_{21}-norm Minimization ELM, which not only fully inherits the merits of ELM but also facilitates group sparsity and reduces complexity of the learning model. Extensive experiments on several benchmark data sets show that our proposed algorithm can obtain superior performances compared with other common multi-label classification algorithms.

Keywords: text categorization, multi-label learning, extreme learning machine, L_{21}-norm minimization

1. Introduction

With the development of the Internet and information technology, large numbers of text data have spawned in various forms. It is a big challenge to organize, manage and analyze such a huge data, and find useful information quickly, accurately and comprehensively. Text automatic classification is an important research point in the field of information mining. Compared to the

*Corresponding author
Email address: hotpzs@hotmail.com (Na Li)

traditional single-label classification problem, multi-label text classification has more value in research and application.

In multi-label learning, the text data is always in high dimensionality and sparsity, e.g. in a large number of feature words, only a few are related to the topic of a text and most of the rest are redundant. Therefore, introducing sparsity into machine learning has become a popular technology, which not only meet the need of practical problems but also can simplify the learning model.

In recent years, extreme learning machine (ELM) [17, 18, 19, 20] has attracted increasing attention and been widely used for its distinguishing characteristics: 1) fast learning speed, 2) good generalization performance on classification or regression, 3) less human intervention for hidden layer parameters setted randomly. For these reasons, the theoretical analysis and various improved algorithms of ELM are put forward continuously.

In the ELM network, the function of the random hidden layer nodes can be seen as feature mapping. It maps the data from the input feature space to the hidden layer feature space, which is called the ELM feature space in literature [22]. In this ELM feature space, each instance may still remain the sparsity. So, it is necessary to delete the redundant hidden layer nodes. To achieve this purpose, we could set all weights that connecting these redundant nodes with all output layer nodes to be zero, which can be realized by introducing the L_{21}-norm minimization of the hidden layer output weights matrix. Meantime, considering the advantages of the classifier ELM, in this paper, we propose an embedded model for multi-label text classification, which is derived from a formulation based on ELM with L_{21}-norm minimization of the hidden layer output weights matrix. Through the constraint of the L_{21}-norm regularization, the training model becomes simplified. We can sufficiently preserve the intrinsic relations of different nodes in the ELM feature space and select these nodes by joint multiple related labels, where the labels are not always independent to each other. Experimental results on several benchmark data sets verify the efficiency of our proposed algorithm.

The main contributions of this paper can be summarized below:

- According to the characteristics of the multi-label text data we introduce the sparsity model.
- Applying L_{21}-norm for joint hidden layer nodes selection and avoiding individual training for each label.

- Using ELM for multi-label learning.

The remainder of this paper is organized as follows. After reviewing the related works in Section 2, we present the proposed algorithm L_{21}-ELM in Section 3. Experimental results are presented in Section 4 and we conclude this paper in Section 5.

2. Related Works

In this section, we will give a brief introduction of the multi-label learning as well as its some common evaluation metrics. Then, we will also introduce the regularization techniques used in sparse learning briefly.

2.1. Multi-label learning

Diffident from traditional supervised learning, in multi-label learning each instance may belong to multiple classes and for a new instance we try to predict its associated set of labels. In formal description, let $\mathcal{X} \in \mathbb{R}^n$ denote the n-dimensional feature space of instance, $\mathcal{Y} = \{y_1, ..., y_k\}$ denote the label space with k possible class labels. The task is to learn a multi-label classifier $f : \mathcal{X} \to 2^k$ from the training data set $\{(x_1, Y_1), ..., (x_m, Y_m)\}$, where $x_i \in \mathcal{X}$ and $Y_i \subseteq \mathcal{Y}$. For any unknown instance $x \in \mathcal{X}$, the multi-label classifier $f(\cdot)$ predicts $f(x) \subseteq \mathcal{Y}$ as its set of proper labels.

In most cases, existing multi-label learning algorithms can be divided into two main categories [1, 2].

Problem transformation methods. The main idea of most problem transformation methods is to transform the original multi-label learning problem into multiple single-label learning problems, which usually reconstructs the multi-label data sets and then existing classification algorithms can be applied directly.

The binary relevance (BR) [11] algorithm is a popular kind of this transformation method and has been widely used in many practical applications. This algorithm divides the multi-label classification problem into k independent binary classification problems, however, the assumption of label independence is too implicit and the label correlations are ignored. The label powerset (LP) [3] algorithm is another common transformation method. It considers each unique set of labels in a multi-label training data as one class in the new transformed data. While the computational complexity of LP is too high and it may pose class imbalance problem. The basic idea of the classifier chains (CC) algorithm [12] is to chain the transformed binary classifiers one by one, but the sequence of each classifier is a problem. The ensembles of classifier chains (ECC) [13] improved the CC algorithm and identify the sequence of each classifier effectively.

Algorithm adaptation methods. From another perspective, this method improves conventional algorithms to deal with multi-label data directly. Some representative algorithms include ML-kNN [15] adapting k-nearest neighbor techniques, which has the advantage of both lazy learning and Bayesian but ignores label correlations, ML-DT [24] adapting decision tree techniques, Rank-SVM [16] adapting kernel techniques, etc.

In this paper, the algorithm based on ELM we proposed is designed to deal with multi-label data directly, therefore, it can be considered as a kind of algorithm adaptation method.

2.2. Evaluation metrics

To measure the performance of each algorithm, we adopt five evaluation metrics in multi-label learning, including: hamming loss, one-error, coverage, ranking loss and average precision [1, 14, 23]. The following is a look at these measures based on a test data set $S = \{(x_i, Y_i) \mid 1 \leqslant i \leqslant n\}$ and a trained model $f(\cdot,\cdot)$ or $g(\cdot)$.

Hamming loss. It evaluates the error rate of all instances on all labels, i.e. a relevant label of an instance is not predicted or irrelevant one is predicted. The smaller the value of hamming loss, the better the performance.

$$hloss_S(g) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{m} \mid g(x_i) \triangle Y_i \mid \tag{1}$$

where m stands for the total number of labels, $\triangle$ for the symmetric difference between two sets. It is worth noting that when most of these instances have little correlative labels, it can also get a small hamming loss even if all labels are predicted in negative. Therefore, we should integrate it with other metrics.

One-error. This metric evaluates the error rate that the top-ranked label of an instance is not in its relevant label set.

$$one - error_S(f) = \frac{1}{n}\sum_{i=1}^{n}\left[arg\max_{y\in\mathcal{Y}} f(x_i,y) \notin Y_i\right] \tag{2}$$

One-error mainly focuses on the most relevant label being correct or not, and it don't pay attention to other labels. Note that, it is equal to ordinary error identically in single-label classification problems.

Coverage*.* This metric evaluates the average steps we need to go down the ranked-label list for the sake of covering all the relevant labels.

$$coverage_S(f) = \frac{1}{n}\sum_{i=1}^{n}\max_{\ell\in Y_i} rank_f(x_i,\ell) - 1 \tag{3}$$

Where the $rank_f(\cdot,\cdot)$ is derived from the real-valued function $f(\cdot,\cdot)$, and the bigger the value of f , the smaller the $rank_f$ is. The performance is perfect when $coverage_S(f) = 0$.

Ranking loss*.* This metric evaluates the average fraction of the reversely ordered label pairs.

$$rloss_S(f) = \frac{1}{n}\sum_{i=1}^{n}\frac{\mid \left\{(y,\overline{y})|f(x_i,y) \le f(x_i,\overline{y}),(y,\overline{y}) \in Y_i \times \overline{Y_i}\right\} \mid}{\mid Y_i \mid\mid \overline{Y_i} \mid} \tag{4}$$

Where Y_i and $\overline{Y_i}$ denote the possible and impossible label sets of the instance x_i. When the value is zero, it means that all impossible labels follow possible ones.

Average precision*.* This metric evaluates the average fraction of relevant labels ranked above a particular one $\ell \in Y_i$. It is typically used in information retrieval (IR) system to evaluate the document ranking performance query retrieval [25]. The bigger the value of $avgpec_S(f)$, the better the performance.

$$avgpec_S(f) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{\mid Y_i \mid}\sum_{y\in Y_i}\frac{\mid L_i \mid}{rank_f(x_i,y)} \tag{5}$$

Where $L_i = \{y' \mid rank_f(x_i,y') \le rank_f(x_i,y), y' \in Y_i\}$. $avgpec_S(f)$ =1 ranks perfectly, it means that there is no instance whose label not in Y_i being ranked above on the one in Y_i.

2.3. L_{21}-norm regularization

In recent years, sparsity-promoting regularization has be widely used in machine learning and statistics. Perhaps L_1-norm regularization is the most common and successful method to promote sparsity for the parameter vector (the lasso approach). Along with the development of multi-task learning, in 2006, Obozinski et al. [6, 7] proposed to constraint the sum of L_2-norms of the blocks of weights connected with each feature, and then leading to the L_{21}-norm regularized optimization problem (the group lasso).

And then, we will review the basics of this technique briefly. Usually, the optimization problem can be described as following:

$$\min_{X} : loss(X) + \lambda \parallel X \parallel_{2,1} \tag{6}$$

where $\lambda > 0$ is the regularization parameter, $X \in \mathbb{R}^{n \times k}$ is the weights matrix, $\parallel X \parallel_{2,1} = \sum_{i=1}^{n} \parallel X \parallel_2$ and $loss(X)$ is a smooth and convex loss function (e.g. the logistic loss, the least square loss and the hinge loss). Take the least squares problem as an example, the Eq.6 is expressed as:

$$\min_{X} : \frac{1}{2} \parallel AX - Y \parallel_2^2 + \lambda \parallel X \parallel_{2,1} \tag{7}$$

where $A \in \mathbb{R}^{m \times n}$, $Y \in \mathbb{R}^{m \times k}$ are the data matrices, each row of X forms a feature group. Fig.1 visualizes this optimization problem.

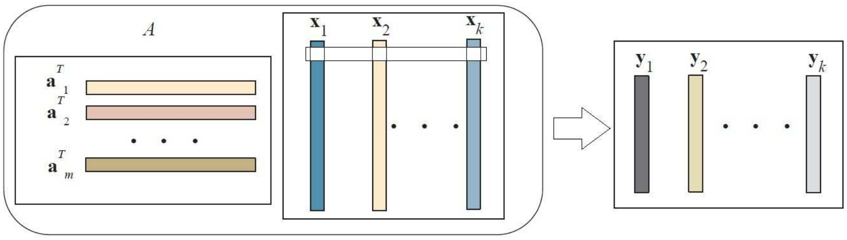

Figure 1: Illustration of the data matrix A, Y, and the weights matrix X.

This optimization problem will be more challenging to solve due to the non-smoothness and non-differential of the L_{21}-norm regularization. In this paper, we apply the strategy proposed in literature [5] to solve this problem, which reformulates the non-smooth L_{21}-norm regularized problem to an equivalent smooth convex optimization problem and can be solved in linear time.

3. L_{21}-minimization ELM (L_{21}-ELM)

In this section, we propose L_{21}-ELM algorithm for multi-label learning problem, which takes the significant advantages of ELM like affording good generalization performance at extremely fast learning speed and offering us some additional characteristics. Firstly, we will review the theories of ELM, then, introduce the algorithm we proposed.

3.1. Extreme learning machine

Extreme learning machine [18, 19] was originally proposed for single hidden layer feedforward neural networks and then extended to the generalized single hidden layer feedforward networks where the hidden layer need not be neuron alike [17]. Fig.2 shows the structure of ELM network. It contains an input layer, a hidden layer and an output layer.

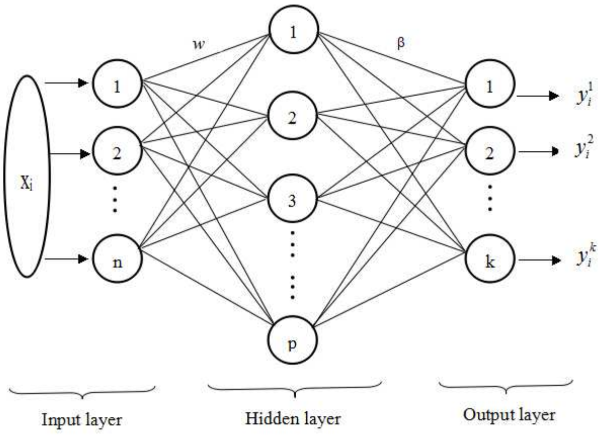

Figure 2: Structure of ELM network

In ELM, the hidden layer parameters are chosen randomly, and the output function of one node can be represented as following:

$$f_{output}(x) = \sum_{i=1}^{p} \beta_i h_i(x) = \mathbf{h}(x)\beta \tag{8}$$

where $x \in \mathbb{R}^n$ is the input variable, $\beta = (\beta_1, \cdots, \beta_p)^T$ is the weights vector between the hidden and the output layer nodes. $\mathbf{h}(x) = [h_1(x), \cdots, h_p(x)]$ is the output vector of the hidden layer with respect to the input vector x . $h_i(x)$ is the i^{th} activation function, its input weights vector and bias are w_i and b_i.

Fig.2 shows that $\mathbf{h}(x)$ actually maps the input variables from the n-dimension to the p-dimensional hidden layer space (ELM feature space), thus, it can be seen as a feature mapping function.

The ELM reliably approximates m samples, $X = [x_1, ..., x_m]$, with minimum error:

$$\min_{\beta} :\parallel H\beta - Y \parallel_2^2 \tag{9}$$

where H is hidden layer output matrix,

$$H = \begin{bmatrix} h(x_1) \\ \vdots \\ h(x_m) \end{bmatrix} = \begin{bmatrix} g(w_1 \cdot x_1 + b_1) & \cdots & g(w_p \cdot x_1 + b_p) \\ \vdots & \ddots & \vdots \\ g(w_1 \cdot x_m + b_1) & \cdots & g(w_p \cdot x_m + b_p) \end{bmatrix}_{m \times p} \tag{10}$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_p^T \end{bmatrix}_{p \times k} \quad and \quad Y = \begin{bmatrix} y_1^T \\ \vdots \\ y_m^T \end{bmatrix}_{m \times k} \tag{11}$$

The analytical result of this least squares equation is:

$$\hat{\beta} = H^{\dagger} Y \tag{12}$$

Where $H^{\dagger}$ is called Moore-Penrose generalized inverse of matrix H.

3.2. L_{21}-norm minimization ELM for multi-label learning

In this section, we consider adapting the ELM network to solve the multi-label learning problem. Given the multi-label training data with m samples (x_i, y_i), where $x_i = (x_{i1}, ..., x_{in})^T \in \mathbb{R}^n$ and $y_i = (y_{i1}, ..., y_{ik}) \in \mathbb{R}^k$. As shown in the Fig.2, we set the number of output layer nodes k , which equals the number of labels, and set the number of hidden layer nodes p randomly.

Inspired by ELM, we consider combining the smallest training error of ELM with the L_{21}-norm minimization of output weights matrix. It is reformulated as following:

$$\min_{\beta} :\| H\beta - Y \|_2^2 + \lambda \| \beta \|_{2,1} \tag{13}$$

This optimization problem is nonsmooth as described in section 3.2. To solve this problem, the literature [5] proposed to employ the Nesterov's optimal method by optimizing its equivalent smooth convex reformulation. When using a constraint to replace the nonsmooth L_{21}-norm, the original problem can be equivalent to the L_{21}-ball constrained smooth convex optimization problem as following:

$$\min_{\beta} :\| H\beta - Y \|_2^2 \ s.t. \ \| \beta \|_{2,1} \leq z \tag{14}$$

When applying the Nesterov's optimal method to solve Eq.(14), one key building block of this method is Euclidean projection onto the L_{21}-ball. The Euclidean projection of a given $U \in \mathbb{R}^{p \times k}$ onto the domain Z can be defined as:

$$\pi_Z(U) = arg \min_{\beta \in Z} \frac{1}{2} \| \beta - U \|_2^2 \tag{15}$$

where $Z = \left\{ \beta \in \mathbb{R}^{p \times k} \, | \| \beta \|_{2,1} \leq z \right\}$ is the L_{21}-ball and $z \geq 0$[1] is the radius of L_{21}-ball. The Lagrangian variable α can be introduced to the inequality constrain $\| \beta \|_{2,1} \leq z$ to solve Eq.(15), then we can lead to its Lagrangian function as:

$$\text{Ł}(\beta, \alpha) = \frac{1}{2} \| \beta - U \|_2^2 + \alpha(\| \beta \|_{2,1} - z) \tag{16}$$

Let β^* be the primal optimal point, and α^* be the dual optimal point. This two points must satisfy the condition: $\| \beta^* \|_{2,1} \leq z$ and $\alpha^* \geq 0$. Since considering the strong duality holds of the Slater's condition, and values of the primal and dual optimal points are equal: $\alpha^*(\| \beta \|_{2,1} - z) = 0$. Therefore, the primal optimal point β^* can be given by Eq.(17) if the dual optimal point α^* is known.

$$\beta_i^* = \begin{cases} (1 - \frac{\alpha^*}{\|u^i\|})u^i, & \alpha^* > 0, \| u^i \| > \alpha^* \\ 0, & \alpha^* > 0, \| u^i \| \leq \alpha^* \\ u^i, & \alpha^* = 0 \end{cases} \tag{17}$$

where $u^i \in \mathbb{R}^{1\times k}$ is the i^{th} row of U.

According to Eq.(17), β^* can be computed as long as α^* is solved. Now, the key step is how to compute the unknown dual optimal point α^*. Liu et al. [5] gives the theorem : if $\parallel U \parallel_{2,1} \leq z$ the value of α^* is zero, otherwise, it can be solved as the unique root of the following auxiliary function.

$$\omega(\alpha) = \sum_{i=1}^{p} max(\parallel u^i \parallel - \alpha, 0) - z \tag{18}$$

The Eq.(18) can be solved in $O(p)$ flops by the bisection [9], and it costs $O(pk)$ flops to compute β^* by Eq.(17). Above all, for solving Eq.(13) the time complexity is $O(pk)$. When testing an unseen instance, we will use a threshold function $t(x)$ to determine its associated label set. For an actual outputs c_j , $Y = \{j \mid c_j > t(x)\}$. An usual solution is setting the $t(x)$ to be zero. In this paper, we adopt the threshold category used in literature [21].

4. Experiments and Results Analysis

To evaluate the performance of our proposed algorithm, we apply it on four benchmark data sets, i.e. Enron for email analysis, Reuters for text categorization, BibTeX for tags of paper and Yahoo for web page categorization. All of them belong to the field of text. For Enron and Reuters without pre-separated training and testing sets, therefore, we decide to select 1,500 instances for training randomly, and the rest for testing. We repeat the data partition for thirty times randomly, and give the average results. Table 1 describes some statistics information of the data sets in detail.

Experiments are all carried out in Matlab 2011a environment running in Core i5 CPU with 8 GB memory. In these experiments, we compare our proposed algorithm L_{21}-ELM with some existing multi-label classification algorithms including ECC, ML-kNN, Rank-SVM as well as the original ELM approach. The parameters for these algorithms are setted as the same as in these references [13, 15, 16]. In L_{21}-ELM, the regularization parameter $\lambda = 0.02$, the optimized starting point is setted to (0,0), and the maximum number of iterations is 100. The number of ELM hidden layer nodes is setted randomly but not more than the number of the training samples and the best results are selected.

Table 2 - Table 4 shows the comparison results on each data set. Among them, the symbol ”↓” means the smaller the better, ”↑” on the contrary.

Table 1: Information of the Data sets

items	size	train	test	features	classes	average labels
Enron	1702	–	–	1001	53	3.38
Reuters	2000	–	–	243	7	1.15
BibTeX	7395	4880	2515	1836	159	2.40
Arts	5000	2000	3000	462	26	1.64
Business	5000	2000	3000	438	30	1.59
Computers	5000	2000	3000	681	33	1.51
Education	5000	2000	3000	550	33	1.46
Entertain-	5000	2000	3000	640	21	1.42
Health	5000	2000	3000	612	32	1.66
Recreation	5000	2000	3000	606	22	1.42
Reference	5000	2000	3000	793	33	1.17
Science	5000	2000	3000	743	40	1.45
Social	5000	2000	3000	1047	39	1.28
Society	5000	2000	3000	636	27	1.69

HL, OE, Co, RL and AP are the abbreviations of hamming loss, one-error, coverage, ranking loss and average precision respectively.

Table 2: Results on data set Enron

	Rank-SVM	ML-kNN	ECC	ELM	L_{21}-ELM
HL ↓	0.071±0.004	0.051±0.002	0.055±0.002	0.053±0.002	**0.048±0.002**
OE ↓	0.714±0.087	0.299±0.031	**0.224±0.036**	0.281±0.036	0.236±0.0276
Co ↓	31.27±2.23	12.96±0.83	21.08±1.27	17.12±1.18	**12.81±0.91**
RL ↓	0.338±0.037	0.091±0.008	0.249±0.023	0.121±0.012	**0.084±0.008**
AP ↑	0.312±0.045	0.639±0.018	0.636±0.023	0.649±0.019	**0.683±0.015**

Table 3: Results on data set Reuters

	Rank-SVM	ML-kNN	ECC	ELM	L_{21}-ELM
HL↓	0.093±0.007	0.049±0.003	0.036±0.003	0.044±0.004	**0.033±0.003**
OE↓	0.205±0.056	0.126±0.013	0.068±0.009	0.091±0.011	**0.062±0.011**
Co↓	0.639±0.163	0.439±0.035	0.350±0.036	0.380±0.034	**0.276±0.029**
RL↓	0.078±0.027	0.045±0.004	0.040±0.006	0.034±0.004	**0.019±0.003**
AP↑	0.867±0.037	0.920±0.007	0.953±0.006	0.940±0.006	**0.962±0.006**

Table 4: Results on data set BibTeX

	ML-kNN	ECC	ELM	L_{21}-ELM
HL↓	**0.014**	0.017±0.0001	0.014±0.0001	0.015±0.0002
OE↓	0.585	**0.371±0.007**	0.409±0.005	0.461±0.018
Co↓	56.218	60.113±0.369	37.266±0.329	**23.041±0.436**
RL↓	0.217	0.463±0.002	0.128±0.001	**0.081±0.001**
AP↑	0.345	0.486±0.003	0.516±0.003	**0.528±0.015**

From above three experiments, in most cases L_{21}-ELM could achieve the best performance compared with other algorithms. Especially, on the Reuters, our algorithm is the best. Moreover, it shows the absolute advantage by coverage, ranking loss and average precision on these three datasets. Compared with the hamming loss, it is worse than ML-kNN only on the BibTeX. ECC achieves better performance comparatively by the metric of one-error, but L_{21}-ELM can also outperform other approaches.

Compared with the original ELM approach, L_{21}-ELM achieves obviously better performance on almost all datasets over all the 5 criteria. This validates the effectiveness of the sparse strategy by the L_{21}-norm regularization, which can improve the original method with eliminating redundant information and get better accuracy.

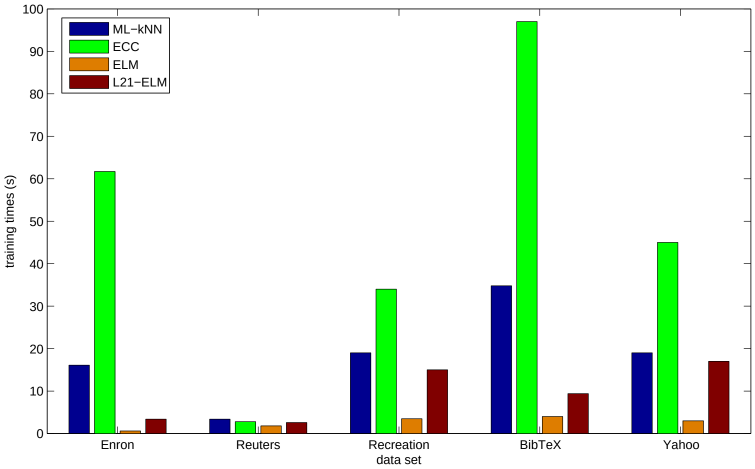

Figure 3: Training time on each data set.

Fig.3 shows the training time on each data set. Note that we do not give the performance of Rank-SVM for its much high complexity. We also reduce an order of magnitude on the BibTeX in order to be displayed in the same figure. As the figure shows, the L_{21}-ELM has faster training speed than ML-kNN and ECC approach, let alone the Rank-SVM. This validates that it could fully inherit the merits of ELM with extreme learning speed. Compared with original ELM, L_{21}-ELM consumes more training time, but it is worth for its better performance.

Note that Yahoo is comprised of 11 independent data sets, including: Arts, Business, Computers, Education, Entertainment, Health, Recreation, Reference, Science, Social and Society. Average results over the 11 data sets can be found in Table 5. As it shows, our approach could also achieve a better performance relatively.

Table 5: Results on data set Yahoo

	Rank-SVM	ML-kNN	ECC	ELM	L_{21}-ELM
HL↓	0.046±0.014	0.043±0.014	0.051±0.021	0.050±0.019	**0.042±0.014**
OE↓	0.441±0.118	0.471±0.157	0.383±0.123	0.437±0.134	**0.379±0.125**
Co↓	**3.564±1.043**	4.098±1.237	8.563±1.867	6.362±1.207	4.836±1.080
RL↓	**0.083±0.031**	0.102±0.045	0.329±0.080	0.154±0.051	0.111±0.034
AP↑	0.661±0.089	0.625±0.117	0.621±0.085	0.631±0.104	**0.685±0.095**

Fig.4 shows the plots for performance vs. the original ELM approach on 11 independent data sets of Yahoo. Here we give the result by six metrics. In each plot, the horizontal axis contains the 11 independent data sets of Yahoo and the evaluation metrics is represented in the vertical axis. As seen in this figure, compared to the original ELM, the L_{21}-ELM approach has the best performance on all data sets of Yahoo by all metrics. So the performance of the original ELM could be improved by the constraint of the L_{21}-norm regularization greatly.

5. Conclusion

Multi-label text classification presents much challenges for the label dependencies and the high dimensional features of the text data. In this paper, we propose a L_{21}-norm minimization ELM algorithm for multi-label learning problem, which not only inherits the advantage of ELM but also offers us

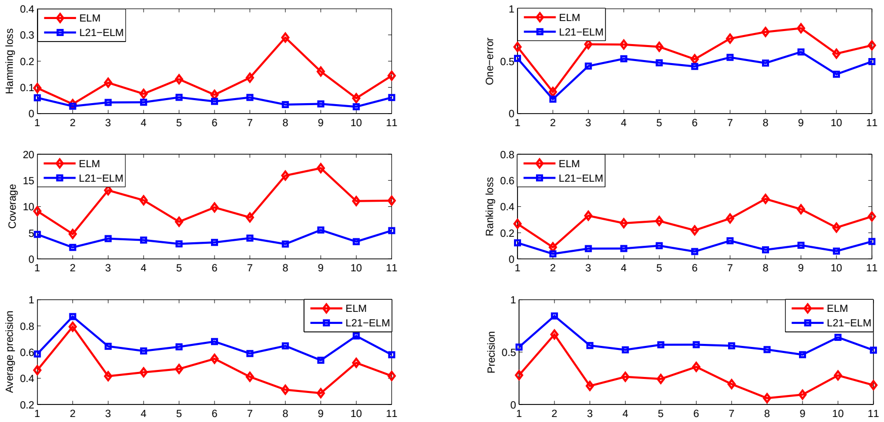

Figure 4: Performance on 11 independent data sets of Yahoo.

additional characteristics. Through the constraint of the L_{21}-norm regularization on the original ELM, the output weights matrix of the hidden layer nodes become sparse. Then we can extract sufficient features of the data and lead to the simplification of the training model. What's more, it can avoid individual training of each label. Experimental results on benchmark data sets validate that L_{21}-ELM is highly superior to state-of-the-art multi-label algorithms, especially in training time. Our approach also improves the performance of the original ELM greatly, although it sacrifices more time.

Acknowledgements

Our work is supported by the National Technology Research and Development Program of China (863 Program) with grant number 2012AA01A510, and the Natural Science Foundation of Jiangsu Province, China (Grant NO.BK20140073). We would like to express the sincere appreciation to the anonymous reviewers for their helpful comments and valuable suggestions.

References

[1] Zhang, M. L., & Zhou, Z. H. (2014). A review on multi-label learning algorithms. Knowledge & Data Engineering IEEE Transactions on, 26(8), 1819-1837.

[2] Tsoumakas, G., & Katakis, I. (2007). Multi-label classification: an overview. Int J Data Warehousing & Mining, 2007(3), 1-13.

[3] Grigorios Tsoumakas, & Ioannis Vlahavas. (2007). Random k-labelsets: an ensemble method for multilabel classification. Lecture Notes in Computer Science, 406-417.

[4] Liu, J., Ji, S., & Ye, J. (2010). Slep: sparse learning with efficient projections. Arizona State University.

[5] Liu, J., Ji, S., & Ye, J. (2009). Multi-task feature learning via efficient L_{21}-norm minimization. In Proceedings of the twenty-fifth conference on Uncertainty in Artificial Intelligence, 339-348.

[6] Obozinski G, Taskar B, & Jordan M I. (2006). Multi-task feature selection. Statistics Department, UC Berkeley, Tech. Rep, 1693-1696.

[7] Obozinski G, Taskar B, & Jordan M I. (2007). Joint covariate selection for grouped classification. Statistics Department, UC Berkeley, Tech. Rep, 2007.

[8] Obozinski G, Taskar B, & Jordan M I. (2010). Joint covariate selection and joint subspace selection for multiple classification problems[J]. Statistics and Computing, 20(2):231-252.

[9] Liu, J. & Ye, J. (2009). Efficient euclidean projections in linear time. In Proceedings of the twenty-sixth Annual International Conference on Machine Learning, pages 657-664, ACM, 2009.

[10] Yang, Y., Shen, H. T., Ma, Z., Huang, Z., & Zhou, X. (2011). L_{21}-norm regularized discriminative feature selection for unsupervised learning[J]. Proceedings of the Twenty-Second international joint conference on Artificial Intelligence - Volume Volume Two (pp.1589-1594). AAAI Press.

[11] Boutell, M. R., Luo, J., Shen, X., & Brown, C. M. (2004). Learning multi-label scene classification. Pattern Recognition, 37(9): 1757-1771.

[12] Read, J., Pfahringer, B., Holmes, G., & Frank, E. (2009). Classier chains for multi-label classification. Proc. of ECML-KDD, 22(4), 829-840.

[13] Read, J., Pfahringer, B., Holmes, G., & Frank, E. (2011). Classifier chains for multi-label classification. Machine Learning, 85(3):333-359.

[14] Gjorgji M A, & Dejan G A. (2012). Two Stage Architecture for Multi-label Learning[J]. Pattern Recognition, 45(3):1019-1034.

[15] Zhang, M. L., & Zhou, Z. H. (2007). ML-kNN: a lazy learning approach to multi-label learning. Pattern Recognition, 40(7): 2038-2048.

[16] A.Elisseeff., & J.Weston. (2002). A kernel method for multi-labelled classification. USA:MIT Press, 681-687.

[17] Huang, G. B., & Chen, L. (2007). Convex incremental extreme learning machine. Neurocomputing, 70, 3056-3062.

[18] Huang, G. B., Zhu, Q. Y., & Siew, C. K. (2004, July). Extreme learning machine: a new learning scheme of feedforward neural networks. In Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on(Vol. 2, pp. 985-990). IEEE.

[19] Huang, G. B., Chen, L., & Siew, C. K. (2006, Jul). Universal approximation using incremental constructive feedforward networks with random hidden nodes. IEEE Trans. Neural Netw. (Vol. 17, pp. 879-892). IEEE.

[20] Huang, G. B., & Siew, C. K. (2005). Extreme learning machine with randomly assigned RBF kernels. International Journal of Information Technology,11(1), 16-24.

[21] Zhang, M. L., & Zhou, Z. H. (2006). Multi-label neural networks with applications to functional genomics and text categorization. IEEE Transactions on Knowledge and Data Engineering, 18(10): 1338-1351.

[22] Huang, G. B., Ding, X., & Zhou, H. (2010). Optimization method based extreme learning machine for classification. Neurocomputing, 74(1-3), 155-163.

[23] Schapire, R. E., & Singer, Y. (2000). Boostexter: a boosting-based system for text categorization. Machine Learning, 39(2-3), 135-168.

[24] Amanda Clare, & Ross D. King. (2001). Knowledge discovery in multi-label phenotype data. Lecture Notes in Computer Science, 42-53.

[25] Salton, G. (1991). Developments in automatic text retrieval. Science, 253(5023), 974-980.

[26] Feng, L., Wang, J., Liu, S., & Xiao, Y. (2014). Multi-label dimensionality reduction and classification with extreme learning machines. Journal of Systems Engineering & Electronics, 25(3), 502-513.

[27] Zuo B, Huang G B, & Wang D, et al. (2014). Sparse Extreme Learning Machine for Classification[J]. Cybernetics IEEE Transactions on, 44(10):1858-1870.

[28] Zheng W, Tang H, & Qian Y. (2015). Collaborative work with linear classifier and extreme learning machine for fast text categorization[J]. World Wide Web-internet and Web Information Systems, 18(2):235-252.

[29] Li, D., Hu, G., Wang, Y., & Pan, Z. (2015). Network traffic classification via non-convex multi-task feature learning. Neurocomputing, 152, 322-332.

[30] Wang, Y., Li, D., Du, Y., & Pan, Z. (2015). Anomaly detection in traffic using L1-norm minimization extreme learning machine. Neurocomputing, 149, 415-425.

Biography of Authors

Jiang Mingchu (1989-) Male, Master Degree Candidate, Research direction: pattern recognition, machine learning and text classification; E-mail：jiangmingchu@foxmail.com.

Li Na (1990-) Female, Master Degree Candidate, Research direction: pattern recognition, machine learning and multi-label learning.

Pan Zhisong(1973-) Male, Professor, doctoral tutor,Research direction: pattern recognition, machine learning and network security, E-mail：hotpzs@hotmail.com.

Jiang Mingchu

Li Na

Pan Zhisong