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

Mingchu Jiang, Na Li and Zhisong Pan

Abstract Extreme learning machine (ELM) was 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 a more desirable performance compared with other common multi-label classification algorithms.

Keywords Text categorization · Multi-label learning · L_{21}-norm minimization · Extreme learning machine

1 Introduction

Continued development of the Internet and information technology has spawned a large number of text data in various forms. How to organize, manage and analyze such a huge data, and find the user information quickly, accurately and comprehensively is a big challenge. Text automatic classification is an important research point in the field of information mining. Compared to the traditional single classification problem, multi-label text classification has more value of research and application.

In multi-label learning, the text data are 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

M. Jiang (✉) · N. Li · Z. Pan
College of Command Information System, PLA University of Science and Technology, Nanjing 210007, China
e-mail: 735906675@qq.com

Z. Pan
e-mail: hotpzs@hotmail.com

J. Cao et al. (eds.), *Proceedings of ELM-2015 Volume 1*,
Proceedings in Adaptation, Learning and Optimization 6,
DOI 10.1007/978-3-319-28397-5_10

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 resent years, extreme learning machine (ELM) [1–4] 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 with randomly setted hidden layer parameters. For these reasons, the theoretical analysis and various improvement algorithms of ELM are put forward continuously.

In 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 ELM feature space in literature [5]. In this ELM feature space, each instance may still remains the sparsity. Meantime, considering the characteristics of multi-label learning and 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 output weights matrix. Through the constraint of the L_{21}-norm regularization, the training model becomes simplified, also we can sufficiently preserve the intrinsic relation of different nodes in the ELM feature space and select them 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 text classification.

The remainder of this paper is organized as follows. After reviewing the related works in Sect. 2, we present the algorithm L_{21}-ELM in Sect. 3 and describe the evaluation measures of multi-label learning in Sect. 4. Experimental results are presented in Sect. 5 and we conclude this paper in Sect. 6.

2 Related Work

2.1 Multi-label Learning

Unlike 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. This is a generalized case of the prevalent multi-class problems where in multiple classes each instance has only one class restrictedly.

Let $\mathcal{X} \in \mathbb{R}^d$ denote the d-dimensional space of instances, $\mathcal{Y} = \{y_1, \ldots, y_k\}$ denote the label space with k possible class labels. Given the training data set $\{(x_1, Y_1), \ldots, (x_n, Y_n)\}$ where $x_i \in \mathcal{X}$ and $Y_i \subseteq \mathcal{Y}$. the task of multi-label learning is to learn a multi-label classifier $f : \mathcal{X} \to 2^k$ from the training data set. For any unknown instance $x \in \mathcal{X}$, the multi-label classifier $f(\cdot)$ predicts $f(x) \subseteq \mathcal{Y}$ as the set of proper labels. Existing multi-label learning algorithms can be divided into two main categories [6, 7].

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) [8] 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) [9] 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 big and it may pose class imbalance problem. The basic idea of the classifier chains (CC) [10] 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) [11] 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 [12] adapting k-nearest neighbor techniques, which has the advantage of both lazy learning and Bayesian but ignores label correlations. ML-DT [13] adapting decision tree techniques, Rank-SVM [14] 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 L_{21}-norm Regularization for Parameter Estimation

In recent years, parameter estimation via sparsity-promoting regularization has be widely used in machine learning and statistics. Perhaps L_1-norm regularization is the most successful and common method to promote sparsity for the parameter vector (the lasso approach). Along with the development of multi-task learning, in 2006, Obozinski et al. [15, 16] 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).

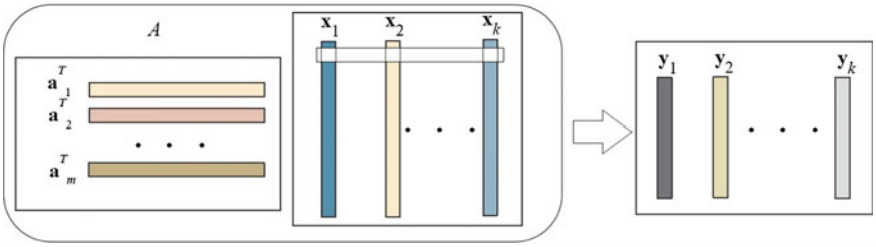

Fig. 1 Illustration of the data matrix A, Y, and the weights matrix X

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

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

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 (such as the logistic loss, the least square loss and the hinge loss). Take the least squares problem as an example, the Eq.1 is expressed as:

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

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. Figure 1 visualizes this optimization problem.

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 [17] 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, meantime, offers us some additional characteristics. Firstly, we will review the theories of ELM, then, introduce the algorithm we proposed.

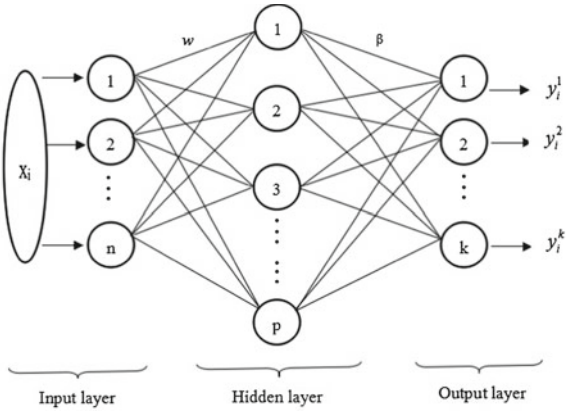

Fig. 2 Structure of ELM network

3.1 Extreme Learning Machine

Extreme learning machine [2, 3] 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 [1]. Figure 2 shows the structure of ELM network. It contains an input layer, a hidden layer and an output layer.

In ELM, the hidden layer parameters are chosen randomly, and the output function can be represented as following (take the case of p hidden layer nodes and one output layer node as an example):

$$f_{output}(x) = \sum_{i=1}^{p} \beta_i h_i(x) = \mathbf{h}(x)\beta \tag{3}$$

where $x \in \mathbb{R}^n$ is the input variable, $\beta = (\beta_1, \beta_2, \ldots, \beta_p)^T$ is the weights vector between the hidden layer nodes and the output layer nodes. $\mathbf{h}(x) = \left[h_1(x), h_2(x), \ldots, h_p(x)\right]$ is the output vector of the hidden layer with respect to the input vector x. $h_i(x)$ is the ith activation function, its input weights vector and bias are w_i and b_i.

Figure 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 appears to be a feature mapping function.

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

$$\min_{\beta} : \| H\beta - Y \|_2^2 \tag{4}$$

where H is hidden layer output matrix,

$$H = \begin{bmatrix} h(x_1) \\ \vdots \\ h(x_m) \end{bmatrix} = \begin{bmatrix} g(w_1 \cdot x_1 + b_1) & \cdots & g(w_p \cdot x_1 + b_p) \\ \vdots & \ddots & \vdots \\ g(w_1 \cdot x_m + b_1) & \cdots & g(w_p \cdot x_m + b_p) \end{bmatrix}_{m \times p} \tag{5}$$

and $Y = [y_1, \ldots, y_m]^T$ is the target vector.

The analytical result of this least squares equation is:

$$\hat{\beta} = H^{\dagger} Y \tag{6}$$

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 $x_i = (x_{i1}, x_{i2}, \ldots, x_{in})^T \in \mathbb{R}^n$ and $y_i = (y_{i1}, y_{i2}, \ldots, y_{ik}) \in \mathbb{R}^k$. 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:

$$\min_{\beta} : \| H\beta - Y \|_2^2 + \lambda \| \beta \|_{2,1} \tag{7}$$

where $\| \beta \|_{2,1} = \sum_{i=1}^{p} \| \beta_i \|_2$ is the L_{21}-norm of the matrix β, and $\beta_i = (\beta_{i1}, \beta_{i2}, \ldots, \beta_{ik})$, λ is the regularization parameter.

To solve the nonsmooth optimization problem in Eq. (7), the literature [17] 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:

$$\min_{\beta} : \| H\beta - Y \|_2^2 \; s.t. \; \| \beta \|_{2,1} \leq z \tag{8}$$

When applying the Nesterov's optimal method to solve Eq. (8), one key building block of this method is Euclidean projection onto the L_{21}-ball. The Euclidean projection problem is defined as:

$$\pi_Z(U) = arg \min_{\beta \in Z} \frac{1}{2} \| \beta - U \|_2^2 \tag{9}$$

where $Z = \left\{\beta \in \mathbb{R}^{p\times k} \mid\mid \beta \mid\mid_{2,1} \leq z\right\}$ is the L_{21}-ball and $z \geq 0$[1] is the radius of L_{21}-ball. To solve the problem in Eq. (9), the Lagrangian variable α is introduced for the inequality constrain $\| \beta \|_{2,1} \leq z$, then we can lead to the Lagrangian function of Eq. (9) as:

$$Ł(\beta, \alpha) = \frac{1}{2} \| \beta - U \|_2^2 + \alpha(\| \beta \|_{2,1} - z) \tag{10}$$

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. (11) if the dual optimal point α^* is known.

$$\beta_i^* = \begin{cases} (1 - \frac{\alpha^*}{\|u^i\|})u^i, & \alpha^* > 0, \| u^i \| > \alpha^* \\ 0, & \alpha^* > 0, \| u^i \| \leq \alpha^* \\ u^i, & \alpha^* = 0 \end{cases} \tag{11}$$

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

According to Eq. (11), 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. [17] gives the theorem : if $\| U \|_{2,1} \leq z$ the value of α^* is zero, otherwise, it can be solved as the unique root of the following auxiliary function.

$$\omega(\alpha) = \sum_{i=1}^{p} max(\| u^i \| - \alpha, 0) - z \tag{12}$$

The Eq. (12) can be solved in $O(p)$ flops by the bisection [18], and it costs $O(pk)$ flops to compute β^* by Eq. (11). Above all, for solving Eq. (7) 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, $Y = \left\{j \mid c_j > t(x)\right\}$. An usual solution is to set $t(x)$ to be zero. In this paper, we adopt the threshold category used in literature [19].

4 Evaluation Measures

Being different from the traditional single-label learning system, in multi-label learning an instance usually have one or more labels simultaneously, therefore those classical evaluation methods would be no longer applied in multi-label learning system. For this reason, a series of evaluation metrics for multi-label learning are proposed.

In order to compare our algorithm with other commonly used methods, we adopt five evaluation measures in multi-label learning in this section, including: hamming

loss, one-error, coverage, ranking loss and average precision [6, 20, 21]. 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. This measure evaluates the error rate of all instances on all labels, e.g. a relevant label of an instance is not predicted or an irrelevant one is predicted. the smaller the value of hamming loss, the better the performance.

$$hloss_S(g) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{m} \mid g(x_i) \triangle Y_i \mid \tag{13}$$

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

One-error. This measure evaluates the times that the top-ranked label of an instance is not in its relevant label set. The smaller the value of $one - error_S(f)$, the better the performance.

$$one - error_S(f) = \frac{1}{n}\sum_{i=1}^{n}\left[arg\max_{y\in y} f(x_i, y) \notin Y_i\right] \tag{14}$$

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 measure evaluates the average steps we need to go down the ranked-label list for the sake of covering all the relevant labels. The smaller the value of coverage, the better the performance.

$$coverage_S(f) = \frac{1}{n}\sum_{i=1}^{n}\max_{\ell\in Y_i} rank_f(x_i, \ell) - 1 \tag{15}$$

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 measure evaluates the average fraction of the reversely ordered label pairs. The smaller the value of $rloss_S(f)$, the better the performance.

$$rloss_S(f) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{\mid Y_i \mid\mid \overline{Y_i} \mid} \mid \left\{(y, \overline{y}) \lfloor f(x_i, y) \leq f(x_i, \overline{y}), (y, \overline{y}) \in Y_i \times \overline{Y_i}\right\} \mid \tag{16}$$

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 measure 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 [22]. The bigger the value of $avgpec_S(f)$, the better the performance.

$$avgpec_S(f) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{\mid Y_i \mid}\sum_{y\in Y_i}\frac{\mid L_i \mid}{rank_f(x_i, y)} \tag{17}$$

where $L_i = \left\{y' \mid rank_f(x_i, y') \leq rank_f(x_i, y), y' \in Y_i\right\}$. Note that $avgpec_S(f) = 1$ ranks perfectly, that means there is no instance x_i for which a label not in Y_i is ranked above on a label in Y_i.

5 Experimental Results

In this section, L_{21}-ELM is compared with the performance of the original ELM as well as other common multi-label classification algorithms. The benchmark data sets we used are all in text areas, including: Enron for email analysis, Reuters for text categorization, BibTeX for tags of paper and Yahoo for web page categorization. Table 1 describes the datasets in detail. For Enron and Reuters without pre-separated training and testing sets, therefore, we decide to select 1,500 instances of them for

Table 1 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
Entertainment	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

Table 2 Results on data set Enron

Measure	Rank-SVM	ML-kNN	ECC	ELM	L_{21}-ELM
HL ↓	0.071 ± 0.0044	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 ↓	31.269 ± 2.233	12.959 ± 0.832	21.079 ± 1.265	17.118 ± 1.176	**12.809 ± 0.906**
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**
Time	>100	16.1	61.7	0.6	3.4

Table 3 Results on data set Reuters

Measure	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**
Time	>100	3.4	2.8	1.8	2.6

Table 4 Results on data set Recreation

Measure	Rank-SVM	ML-kNN	ECC	ELM	L_{21}-ELM
HL↓	0.061	0.064	0.070 ± 0.001	0.084 ± 0.001	**0.058 ± 0.001**
OE↓	0.499	0.746	**0.485 ± 0.005**	0.577 ± 0.002	0.501 ± 0.023
Co↓	4.066	5.432	6.365 ± 0.128	6.169 ± 0.060	**3.955 ± 0.012**
RL↓	0.140	0.208	0.364 ± 0.008	0.228 ± 0.003	**0.136 ± 0.001**
AP↑	0.608	0.422	0.569 ± 0.006	0.528 ± 0.002	**0.611 ± 0.014**
Time	95	19	34	3.5	15

Table 5 Results on data set BibTeX

Measure	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**
Time	348	1007	40	94

training randomly and the rest data for testing. We repeat the data partition for thirty times randomly, and give the "average results" ± "standard deviations".

Table 2, 3, 4 and 5 shows the comparison results on a single data set. Among them, the symbol "↓" means the smaller the better, "↑" on the contrary. HL, OE, Co,

Table 6 Results on data set Yahoo

Measure	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	$\mathbf{0.042 \pm 0.014}$
OE↓	0.441 ± 0.118	0.471 ± 0.157	0.383 ± 0.123	0.437 ± 0.134	$\mathbf{0.379 \pm 0.125}$
Co↓	$\mathbf{3.564 \pm 1.043}$	4.098 ± 1.237	8.563 ± 1.867	6.362 ± 1.207	4.836 ± 1.080
RL↓	$\mathbf{0.083 \pm 0.031}$	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	$\mathbf{0.685 \pm 0.095}$
Time	213	19	45	3	17

RL and AP are the abbreviations of hamming loss, one-error, coverage, ranking loss and average precision respectively, unit of Time (training) is seconds. The number of ELM hidden layer nodes is randomly setted but not more than the training samples and the best results are selected.

Overall, compared with other algorithms, L_{21}-ELM achieves the best performance in most case. Especially, it shows the absolute advantage on coverage, ranking loss and average precision in all datasets. On hamming loss it is worse than Rank-SVM only on BibTeX data set, and performs better on other cases. On one-error, ECC achieves comparable performance with other approaches. Without consideration of ECC, L_{21}-ELM outperforms other approaches by the metric of one-error.

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 L_{21}-norm regularization on the original ELM and eliminating redundant information.

On the training time, the ELM group has faster training time than other approaches. This validates that L_{21}-ELM could fully inherit the merits of ELM with extreme learning speed. Compared with original ELM, L_{21}-ELM consumes more training time, but considering its better performance it is worth.

Note that Yahoo is comprised of 11 independent data sets, including: Arts, Business, Computers, Education, Entertainment, Health, Recreation, Reference, Science, Social and Society. We just give the average results over the 11 data sets. From the results as Table 6 shows, our approach could also achieve a better performance relatively.

6 Conclusion

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 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 and then leading to the simplification of the learning model. Experimental

results validate that L_{21}-ELM has highly competition to state-of-the-art multi-label algorithms (e.g. Rank-SVM, ML-kNN and ECC) especially in training time. Our approach greatly improves the performance of the original ELM, although it sacrifices more time.

References

1. Huang, G.B., Chen, L.: Convex incremental extreme learning machine. Neurocomputing **70**, 3056–3062 (2007)
2. Huang, G.B., Zhu, Q.Y., Siew, C.K.: Extreme learning machine: a new learning scheme of feedforward neural networks. In: 2004 IEEE International Joint Conference on Neural Networks, 2004. Proceedings, vol. 2, pp. 985–990. IEEE (2004)
3. Huang, G.B., Chen, L., Siew, C.K.: Universal approximation using incremental constructive feedforward networks with random hidden nodes. IEEE Trans. Neural Netw. **17**, 879–892 (2006)
4. Huang, G.B., Siew, C.K.: Extreme learning machine with randomly assigned RBF kernels. Int. J. Inf. Technol. **11**(1), 16–24 (2005)
5. Huang, G.B., Ding, X., Zhou, H.: Optimization method based extreme learning machine for classification. Neurocomputing **74**(1–3), 155–163 (2010)
6. Zhang, M.L., Zhou, Z.H.: A review on multi-label learning algorithms. IEEE Trans. Knowl. Data Eng. **26**(8), 1819–1837 (2014)
7. Tsoumakas, G., Katakis, I.: Multi-label classification: an overview. Int. J. Data Warehousing Min. **2007**(3), 1–13 (2007)
8. Boutell, M.R., Luo, J., Shen, X., Brown, C.M.: Learning multi-label scene classification. Pattern Recogn. **37**(9), 1757–1771 (2004)
9. Tsoumakas, G., Vlahavas, I.: Random k-labelsets: an ensemble method for multilabel classification. Lecture Notes in Computer Science, pp. 406-417 (2007)
10. Read, J., Pfahringer, B., Holmes, G., Frank, E.: Classier chains for multi-label classification. In: Proceedings of ECML-KDD, vol. 22, no. 4, pp. 829–840 (2009)
11. Read, J., Pfahringer, B., Holmes, G., Frank, E.: Classifier chains for multi-label classification. Mach. Learn. **85**(3), 333–359 (2011)
12. Zhang, M.L., Zhou, Z.H.: ML-kNN: a lazy learning approach to multi-label learning. Pattern Recogn. **40**(7), 2038–2048 (2007)
13. Clare, A., King, R.D.: Knowledge Discovery in Multi-label Phenotype Data. Lecture Notes in Computer Science, pp. 42–53 (2001)
14. Elisseeff, A., Weston, J.: A Kernel Method for Multi-labelled Classification, pp. 681–687. MIT Press, USA (2002)
15. Obozinski, G., Taskar, B., Jordan, M.I.: Multi-task feature selection. Statistics Department, UC Berkeley, Technical Report, 1693–1696 (2006)
16. Obozinski, G., Taskar, B., Jordan, M.I.: Joint covariate selection for grouped classification. Statistics Department, UC Berkeley, Technical Report (2007)
17. Liu, J., Ji, S., Ye, J.: Multi-task feature learning via efficient L_{21}-norm minimization. In: Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, pp. 339–348 (2009)
18. Liu, J., Ye, J.: Efficient Euclidean projections in linear time. In: Proceedings of the Twenty-Sixth Annual International Conference on Machine Learning, pp. 657–664. ACM, (2009)
19. Zhang, M.L., Zhou, Z.H.: Multi-label neural networks with applications to functional genomics and text categorization. IEEE Trans. Knowl. Data Eng. **18**(10), 1338–1351 (2006)
20. Gjorgji, M.A., Dejan, G.A.: Two stage architecture for multi-label learning. Pattern Recogn. **45**(3), 1019–1034 (2012)

21. Schapire, R.E., Singer, Y.: Boostexter: a boosting-based system for text categorization. Mach. Learn. **39**(2–3), 135–168 (2000)
22. Salton, G.: Developments in automatic text retrieval. Science **253**(5023), 974–980 (1991)