

# Multi-instance Multi-label Learning by Extreme Learning Machine

Chenguang Li, Ying Yin, Yuhai Zhao, Guang Chen and Libo Qin

**Abstract** Multi-instance Multi-label learning is a learning framework, where every object is represented by a bag of instances and associated with multiple labels simultaneously. The existing degeneration strategy based methods often suffer from some common drawbacks: (1) the user-specific parameter for the number of clusters may incur the effective problem; (2) utilizing SVM as the classifiers builder may bring the high computational cost. In this paper, we propose an algorithm, namely MIML-ELM, to address the problems. To our best knowledge, we are the first utilizing ELM in MIML problem and conducting the comparison of ELM and SVM on MIML. Extensive experiments are conducted on the real datasets and the synthetic datasets. The results show that MIML-ELM tends to achieve better generalization performance at a higher learning speed.

**Keywords** Multi-instance multi-label · Extreme learning machine

## 1 Introduction

When utilizing machine learning to solve the practical problems, we often consider an object as a feature vector. Then, we get an instance of the object. Further, associating the instance with a specific class label of the object, we obtain an example. Given a large collection of examples, the task is to get a function mapping from the instance space to the label space. We expect that the learned function can predict the labels of unseen instances correctly. However, in some applications, a real-world object is often of ambiguity, which consists of multiple instances and corresponds to multiple different labels simultaneously.

For example, an image usually contains multiple patches each represented by an instance, while in image classification such an image can belong to several classes

---

C. Li · Y. Yin (✉) · Y. Zhao · G. Chen · L. Qin  
College of Information Science and Engineering, Northeastern University,  
Shenyang 110819, China  
e-mail: yinying@ise.neu.edu.cn

simultaneously, e.g. an image can belong to mountains as well as Africa [1]; Another example is text categorization [1], where a document usually contains multiple sections each of which can be represented as an instance, and the document can be regarded as belonging to different categories if it was viewed from different aspects, e.g. a document can be categorized as scientific novel, Jules Verne's writing or even books on travelling; The MIML problem also arises in the protein function prediction task [2]. A domain is a distinct functional and structural unit of a protein. A multi-functional protein often consists of several domains, each fulfilling its own function independently. Taking a protein as an object, a domain as an instance and each biological function as a label, the protein function prediction problem exactly matches the MIML learning task.

In this context, Multi-instance Multi-label learning was proposed [1]. Similar to two another multi-learning frameworks, i.e. Multi-instance learning (MIL) [3] and Multi-label learning (MLL) [4], the MIML learning framework also results from the ambiguity in representing the real-world objects. Differently, more difficult than two another multi-learning frameworks, MIML studies the ambiguity in terms of both the input space (i.e. instance space) and the output space (i.e. label space) while MIL just studies the ambiguity in the input space and MLL just studies the ambiguity in the output space, respectively. In [1], Zhou et al. proposed a degeneration strategy based framework for MIML, which consists of two phases. First, the MIML problem is degenerated into the single-instance multi-label (SIML) problem through a specific clustering process; Second, the SIML problem is decomposed into multiple independent binary classification (i.e. single-instance single-label) problem using SVM as the classifiers builder. This two-phase framework has been successfully applied to many real-world applications and has been shown effective [5]. However, it could be further improved if the following drawbacks are tackled. On one hand, the clustering process in the first phase requires a user-specific parameter for the number of clusters. Unfortunately, it is often trouble to determine the correct number of clusters in advance. The incorrect number of clusters may affect the accuracy of the learning algorithm; On the other hand, SIML is degenerated into SISL (i.e. single instance single label) in the second phase, as will increase the volume of data to be handled and thus burden the classifier building. Utilizing SVM as the classifiers builder in this phase may suffer from the high computational cost and require much number of parameters to be optimized.

In this paper, we propose to enhance the two-phase framework by tackling the two above issues and make the following contributions: (1) we utilize Extreme Learning Machine (ELM) [6] instead of SVM to improve the efficiency of the two-phase framework. To our best knowledge, we are the first utilizing ELM in MIML problem and conducting the comparison of ELM and SVM on MIML; (2) we design a method of theoretical guarantee to determine the number of clusters automatically while incorporating it into the improved two-phase framework for effectiveness.

The remainder of this paper is organized as follows. In Sect. 2, we give a brief introduction to MIML and ELM; Sect. 3 details the improvements of the two-phase framework; Finally, Sect. 5 concludes this paper.

## 2 The Preliminaries

This research is related to some previous work on multi-instance multi-label (MIML) learning and extreme learning machine (ELM). In what follows, we briefly review some preliminaries of the two related work in Sects. 2.1 and 2.2, respectively.

### 2.1 Multi-instance Multi-label Learning

In the traditional supervised learning, the relationships between an object and its description and its label are always one-to-one correspondence. That is, an object is represented by a single instance and associated with a single class label. In this sense, we refer to it as single-instance single-label learning (SISL for short). Formally, let  $X$  be the instance space (or say feature space) and  $Y$  the set of class labels. The goal of SISL is to learn a function  $f_{SISL}:X \rightarrow Y$  from a given data set  $\{(x_1, Y_1), (x_2, Y_2), \dots, (x_m, Y_m)\}$ , where  $x_i \in X$  is a instance and  $y_i \in Y$  is the label of  $x_i$ . This formalization is prevailing and successful. However, as mentioned in Sect. 1, a lot of real-world objects are complicated and ambiguous in their semantics. Representing these ambiguous objects with SISL may lose some important information and make the learning task problematic [1]. Thus, many real-world complicated objects do not fit in this framework well.

In order to deal with this problem, several multi-learning frameworks have been proposed, e.g. Multi-Instance Learning (MIL), Multi-Label Learning (MLL) and Multi-Instance Multi-Label Learning (MIML). MIL studies the problem where a real-world object described by a number of instances is associated with a single class label. Multi-instance learning techniques have been successfully applied to diverse applications including image categorization [7], image retrieval [8], text categorization [9], web mining [10], computer-aided medical diagnosis [11], etc. Differently, MLL studies the problem where a real-world object is described by one instance but associated with a number of class labels. The existing work of MLL falls into two major categories. The one attempts to divide multi-label learning to a number of two class classification problems [12] or transform it into a label ranking problem [13]; the other tries to exploit the correlation between the labels [14]. MLL has been found useful in many tasks, such as text categorization [15], scene classification [16], image and video annotation [17, 18], bioinformatics [19, 20] and even association rule mining [21, 22]. MIML is a generalization of traditional supervised learning, multi-instance learning and multi-label learning, where a real-world object may be associated with a number of instances and a number of labels simultaneously. MIML is more reasonable than (single-instance) multi-label learning in many cases. In some cases, understanding why a particular object has a certain class label is even more important than simply making an accurate prediction while MIML offers a possibility for this purpose.

## 2.2 A Brief Introduction to ELM

Extreme Learning Machine (ELM for short) is a generalized Single Hidden-layer Feedforward Network. In ELM, the hidden-layer node parameter is mathematically calculated instead of being iteratively tuned, thus it provides good generalization performance at thousands of times faster speed than traditional popular learning algorithms for feedforward neural networks [23].

As a powerful classification model, ELM has been widely applied in many fields, such as protein sequences classification in bioinformatics [24, 25], online social network prediction [26], XML document classification [23], Graph classification [27] and so on. How to classify the data quickly and correctly is an important thing. For example, in [28], ELM was applied for plain text classification by using the one-against-one (OAO) and one-against-all (OAA) decomposition scheme. In [23], an ELM based XML document classification framework was proposed to improve classification accuracy by exploiting two different voting strategies. A protein secondary prediction framework based on ELM was proposed in [29] to provide good performance at extremely high speed. Wang et al. [30] implemented the protein-protein interaction prediction on multi-chain sets and on single-chain sets using ELM and SVM for a comparable study. In both cases, ELM tends to obtain higher Recall values than SVM and shows a remarkable advantage in the computational speed. Zhang et al. [31] evaluated the multicategory classification performance of ELM on three microarray data sets. The results indicate that ELM produces comparable or better classification accuracies with reduced training time and implementation complexity compared to artificial neural networks methods and Support Vector Machine methods. In [32], the use of ELM for multiresolution access of terrain height information was proposed. Optimization method based ELM for classification was studied in [33].

Given  $N$  arbitrary distinct samples  $(x_i, t_i)$ , where  $x_i = [x_{i1}, x_{i2}, \dots, x_{im}]^T \in \mathbf{R}^n$  and  $t_i = [t_{i1}, t_{i2}, \dots, t_{im}]^T \in \mathbf{R}^m$ , standard SLFNs with  $L$  hidden nodes and activation function  $g(x)$  are mathematically modeled as

$$f(x) = \sum_{i=1}^L \beta_i g(\mathbf{a}_i, b_i, \mathbf{x}) \quad (1)$$

where  $a_i$  and  $b_i$  are the learning parameters of hidden nodes and  $\beta_i$  is the weight connecting the  $i$ th hidden node to the output node.  $g(\mathbf{a}_i, b_i, \mathbf{x})$  is the output of the  $i$ th hidden node with respect to the input  $x$ . In our case, sigmoid type of additive hidden nodes are used. Thus, Eq. (1) is given by

$$f(x) = \sum_{i=1}^L \beta_i g(\mathbf{a}_i, b_i, \mathbf{x}) = \sum_{i=1}^L \beta_i g(\mathbf{w}_i \cdot \mathbf{x}_j + b_i) = \mathbf{o}_j, (j = 1, \dots, N) \quad (2)$$

where  $\mathbf{w}_i = [w_{i1}, w_{i2}, \dots, w_{in}]^T$  is the weight vector connecting the  $i$ -th hidden node and the input nodes,  $\beta_i = [\beta_{i1}, \beta_{i2}, \dots, \beta_{im}]^T$  is the weight vector connecting the  $i$ th hidden node and the output nodes,  $b_i$  is the bias of the  $i$ th hidden node, and  $o_j$  is the output of the  $j$ th node [34].

If an SLFN with activation function  $g(x)$  can approximate the  $N$  given samples with zero error that  $\sum_{j=1}^L \|o_j - t_j\| = 0$ , there exist  $\beta_i$ ,  $a_i$  and  $b_i$  such that

$$\sum_{i=1}^L \beta_i g(\mathbf{w}_i \cdot \mathbf{x}_j + b_i) = \mathbf{t}_j, j = 1, \dots, N \quad (3)$$

Equation (3) can be expressed compactly as follows

$$\mathbf{H}\beta = \mathbf{T} \quad (4)$$

where

$$\mathbf{H}(\mathbf{w}_1, \dots, \mathbf{w}_L, b_1, \dots, b_L, \mathbf{x}_1, \dots, \mathbf{x}_N) = \begin{bmatrix} g(\mathbf{w}_1 \cdot \mathbf{x}_1 + b_1) & \dots & g(\mathbf{w}_L \cdot \mathbf{x}_1 + b_L) \\ \vdots & \dots & \vdots \\ g(\mathbf{w}_1 \cdot \mathbf{x}_N + b_1) & \dots & g(\mathbf{w}_L \cdot \mathbf{x}_N + b_L) \end{bmatrix}_{N \times L},$$

$$\beta = [\beta_1^T, \dots, \beta_L^T]_{m \times L}^T, \text{ and } \mathbf{T} = [\mathbf{t}_1^T, \dots, \mathbf{t}_L^T]_{m \times N}^T$$

$\mathbf{H}$  is called the hidden layer output matrix of the network. The  $i$ th column of  $\mathbf{H}$  is the  $i$ th hidden nodes output vector with respect to inputs  $x_1, x_2, \dots, x_N$  and the  $j$ th row of  $\mathbf{H}$  is the output vector of the hidden layer with respect to input  $x_j$ .

For the binary classification applications, the decision function of ELM [33] is

$$f(x) = \text{sign} \left( \sum_{i=1}^L \beta_i g(\mathbf{a}_i, b_i, \mathbf{x}) \right) = \text{sign}(\beta \cdot h(x)) \quad (5)$$

$h(\mathbf{x}) = [g(\mathbf{a}_1, b_1, \mathbf{x}), \dots, g(\mathbf{a}_L, b_L, \mathbf{x})]^T$  is the output vector of the hidden layer with respect to the input  $\mathbf{x}$ .  $h(\mathbf{x})$  actually maps the data from the  $d$ -dimensional input space to the  $L$ -dimensional hidden layer feature space  $\mathbf{H}$ .

---

### Algorithm 1: ELM

---

**Input:** DB:dataset, HN: Number of Hidden Layer nodes, AF: ActivationFunction

**Output:** Results

```

1 for  $i=1$  to  $L$  do
2   | randomly assign input weight  $w_i$ ;
3   | randomly assign bias  $b_i$ ;
4 end
5 calculate  $\mathbf{H}$ ;
6 calculate  $\beta = \mathbf{H}^\dagger \mathbf{T}$ 

```

---

In ELM, the parameters of hidden layer nodes, i.e.  $w_i$  and  $b_i$ , can be chosen randomly without knowing the training data sets. The output weight  $\mathbf{L}$  is then calculated with matrix computation formula  $\mathbf{L} = \mathbf{H}^\dagger \mathbf{T}$ , where  $\mathbf{H}^\dagger$  is the Moore-Penrose Inverse of  $\mathbf{H}$ . ELM not only tends to reach the smallest training error but also the smallest norm of weights [6]. Given a training set  $\mathfrak{N} = \{(\mathbf{x}_i, \mathbf{t}_i) | \mathbf{x}_i \in \mathbf{R}^n, \mathbf{t}_i \in \mathbf{R}^m, i = 1, \dots, N\}$ , activation function  $g(x)$  and hidden node number  $L$ , the pseudo code of ELM [34] is given in Algorithm 1.

### 3 The Proposed Two-Phase MIMLELM Framework

MIMLSVM is a representative two-phase MIML algorithm successfully applied in many real-world tasks [2]. It was first proposed by Zhou et al. in [1], and recently improved by Li et al. in [5]. MIMLSVM solves the MIML problem by first degenerating it into single-instance multi-label problems through a specific clustering process and then decomposing the learning of multiple labels into a series of binary classification tasks using SVM. However, as ever mentioned, MIMLSVM may suffer from some drawbacks in either of the two phases. For example, in the first phase, the user-specific parameter for the number of clusters may incur the effective problem; in the second phase, utilizing SVM as the classifiers builder may bring the high computational cost and require much number of parameters to be optimized.

---

#### Algorithm 2: The MIMLELM Algorithm

---

**Input:** DB:dataset, HN: Number of Hidden Layer nodes, AF: ActivationFunction

**Output:** Results

- 1  $DB = \{(X_1, Y_1), (X_2, Y_2), \dots, (X_m, Y_m)\}, \Gamma = X_1, X_2, \dots, X_m;$
  - 2 Determine the number of clusters,  $k$ , using AIC;
  - 3 randomly select  $k$  elements from  $\Gamma$  to initialize the  $k$  medoids  $\{M_1, M_2, \dots, M_k\};$
  - 4 **repeat**
  - 5  $\Gamma_t = \{M_t\} (t = 1, 2, \dots, k);$
  - 6 **foreach**  $X_u \in (\Gamma - \{M_t\})$  **do**
  - 7  $index = \arg \min_{t \in \{1, 2, \dots, k\}} d_H(X_u, M_t);$
  - 8  $\Gamma_{index} = \Gamma_{index} \cup \{X_u\}$
  - 9 **end**
  - 10  $M_t = \arg \min_{A \in \Gamma_t, B \in \Gamma_t} \sum d_H(A, B) (t = 1, 2, \dots, k);$
  - 11 Transform  $(X_u, Y_u)$  into into an SIML example  $(z_u, Y_u)$ , where  
 $z_u = (d_H(X_u, M_1), d_H(X_u, M_2), \dots, d_H(X_u, M_k));$
  - 12 **until**  $M_t (t = 1, 2, \dots, k)$  don't change;
  - 13 **foreach**  $z_u (u \in \{1, 2, \dots, m\})$  **do**
  - 14 **foreach**  $y \in Y_u$  **do**
  - 15 decompose  $(z_u, Y_u)$  into  $|Y_u|$  SISL examples
  - 16 **end**
  - 17 **end**
  - 18 Train ELM<sub>y</sub> for every class  $y$
-



Four real datasets are utilized in our experiments. The data set is *Image* [1], which comprises 2000 natural scene images and 5 classes. The percent of images of more than one class is over 22%. On average, each image is of  $1.24 \pm 0.46$  class labels and  $1.36 \pm 0.54$  instances.

In the next experiments, we study the efficiency of MIMLELM by testing its scalability. That is, the data set is replicated different number of times, and then we observe how the training time and the testing time vary with the data size increasing. Again, MIMLSVM+ is utilized as the competitor. Similarly, the MIMLSVM+ algorithm is implemented with a Gaussian kernel while the penalty factor Cost is set to be 1, 2, 3, 4 and 5, respectively. The experimental results are given in Figs. 1. The Image data set is replicated 0.5–2 times with step size set to be 0.5. When the number of copies is 2, the efficiency improvement could be up to one 92.5% (from about 41.2 s down to about 21.4 s). As we observed, as the data size increasing, the superiority of MIMLELM becomes more and more significant.

## 5 Conclusion

MIML is a framework for learning with complicated objects, and has been proved to be effective in many applications. However, the existing two-phase MIML approaches may suffer from the effectiveness problem arising from the user-specific clusters number and the efficiency problem arising from the high computational cost. In this paper, we propose the MIMLELM approach to learn with MIML examples fast. On one hand, the efficiency is highly improved by integrating Extreme Learning Machine into the MIML learning framework. To our best knowledge, we are the first utilizing ELM in MIML problem and conducting the comparison of ELM and SVM on MIML. On the other hand, we develop a method of theoretical guarantee to determine the number of clusters automatically and exploit a genetic algorithm based ELM ensemble to further improve the effectiveness.

**Acknowledgments** National Natural Science Foundation of China (61272182, 61100028, 61572117, 61173030, 61173029), State Key Program of National Natural Science of China (61332014, U1401256), New Century Excellent Talents (NCET-11-0085) and the Fundamental Research Funds for the Central Universities under grants (No.130504001).

## References

1. Zhou, Z.H., Zhang, M.L.: Multi-instance multi-label learning with application to scene classification. In: Schölkopf, B., Platt, J., Hoffman, T. (eds.) *Advances in Neural Information Processing Systems 19*, pp. 1609–1616. MIT Press, Cambridge (2007)
2. Wu, J., Huang, S., Zhou, Z.: Genome-wide protein function prediction through multi-instance multi-label learning. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **11**, 891–902 (2014)
3. Dietterich, T.G., Lathrop, R.H., Lozano-Pérez, T.: Solving the multiple instance problem with axis-parallel rectangles. *Artif. Intell.* **89**, 31–71 (1997)



4. Schapire, R.E., Singer, Y.: Boostexter: a boosting-based system for text categorization. *Mach. Learn.* **39**, 135–168 (2000)
5. Li, Y., Ji, S., Kumar, S., Ye, J., Zhou, Z.: Drosophila gene expression pattern annotation through multi-instance multi-label learning. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **9**, 98–112 (2012)
6. Huang, G.B., Zhu, Q.Y., Siew, C.K.: Extreme learning machine: a new learning scheme of feedforward neural networks. In: *Proceedings of International Joint Conference on Neural Networks (IJCNN2004)*, vol. 2, pp. 985–990, Budapest, Hungary, 25–29 July 2004
7. Chen, Y., Bi, J., Wang, J.Z.: MILES: multiple-instance learning via embedded instance selection. *IEEE Trans. Pattern Anal. Mach. Intell.* **28**, 1931–1947 (2006)
8. Yang, C., Lozano-Pérez, T.: Image database retrieval with multiple-instance learning techniques. In: *ICDE*, pp. 233–243 (2000)
9. Andrews, S., Tsochantaridis, I., Hofmann, T.: Support vector machines for multiple-instance learning. In: *Advances in Neural Information Processing Systems 15 [Neural Information Processing Systems, NIPS 2002, December 9–14, 2002, Vancouver, British Columbia, Canada]*, pp. 561–568 (2002)
10. Zhou, Z., Jiang, K., Li, M.: Multi-instance learning based web mining. *Appl. Intell.* **22**, 135–147 (2005)
11. Fung, G., Dundar, M., Krishnapuram, B., Rao, R.B.: Multiple instance learning for computer aided diagnosis. In: *Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems*, pp. 425–432, Vancouver, British Columbia, Canada, 4–7 Dec 2006
12. Joachims, T.: Text categorization with support vector machines: learning with many relevant features. In: *Proceedings Machine Learning: ECML-98, 10th European Conference on Machine Learning*, pp. 137–142, Chemnitz, Germany, 21–23 April 1998
13. Elisseeff, A., Weston, J.: A kernel method for multi-labelled classification. In: *Advances in Neural Information Processing Systems 14 [Neural Information Processing Systems: Natural and Synthetic, NIPS 2001, December 3–8, 2001, Vancouver, British Columbia, Canada]*, pp. 681–687 (2001)
14. Liu, Y., Jin, R., Yang, L.: Semi-supervised multi-label learning by constrained non-negative matrix factorization. In: *Proceedings, The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference*, pp. 421–426, Boston, Massachusetts, USA, 16–20 July 2006
15. Godbole, S., Sarawagi, S.: Discriminative methods for multi-labeled classification. In: *Proceedings Advances in Knowledge Discovery and Data Mining, 8th Pacific-Asia Conference, PAKDD 2004, 22–30, Sydney, Australia, 26–28 May 2004*
16. Boutell, M.R., Luo, J., Shen, X., Brown, C.M.: Learning multi-label scene classification. *Pattern Recogn.* **37**, 1757–1771 (2004)
17. Kang, F., Jin, R., Sukthankar, R.: Correlated label propagation with application to multi-label learning. In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006)*, pp. 1719–1726, New York, NY, USA, 17–22 June 2006
18. Qi, G., Hua, X., Rui, Y., Tang, J., Mei, T., Zhang, H.: Correlative multi-label video annotation. In: *Proceedings of the 15th International Conference on Multimedia 2007, 17–26, Augsburg, Germany, 24–29 Sept 2007*
19. Barutcuoglu, Z., Schapire, R.E., Troyanskaya, O.G.: Hierarchical multi-label prediction of gene function. *Bioinformatics* **22**, 830–836 (2006)
20. Brinker, K., Fürnkranz, J., Hüllermeier, E.: A unified model for multilabel classification and ranking. In: *Proceedings ECAI 2006, 17th European Conference on Artificial Intelligence, August 29–September 1, 2006, Riva del Garda, Italy, Including Prestigious Applications of Intelligent Systems (PAIS 2006)*, pp. 489–493 (2006)
21. Rak, R., Kurgan, L.A., Reformat, M.: Multi-label associative classification of medical documents from MEDLINE. In: *Fourth International Conference on Machine Learning and Applications, ICMLA 2005, Los Angeles, California, USA, 15–17 Dec 2005*

22. Thabtah, F.A., Cowling, P.I., Peng, Y.: MMAC: a new multi-class, multi-label associative classification approach. In: Proceedings of the 4th IEEE International Conference on Data Mining (ICDM 2004), pp. 217–224, Brighton, UK, 1–4 Nov 2004
23. Zhao, X., Wang, G., Bi, X., Gong, P., Zhao, Y.: Xml document classification based on elm. *Neurocomputing* **74**, 2444–2451 (2011)
24. Zhao, Y., Wang, G., Yin, Y., Li, Y., Wang, Z.: Improving elm-based microarray data classification by diversified sequence features selection. *Neural Comput. Appl.* **2014**, 1–12 (2014)
25. Zhao, Y., Wang, G., Zhang, X., Yu, J.X., Wang, Z.: Learning phenotype structure using sequence model. *IEEE Trans. Knowl. Data Eng.* **26**, 667–681 (2014)
26. Sun, Y., Yuan, Y., Wang, G.: An on-line sequential learning method in social networks for node classification. *Neurocomputing* **149**, 207–214 (2015)
27. Wang, Z., Zhao, Y., Wang, G., Li, Y., Wang, X.: On extending extreme learning machine to non-redundant synergy pattern based graph classification. *Neurocomputing* **149**, 330–339 (2015)
28. Zhang, R., Huang, G.B., Sundararajan, N., Saratchandran, P.: Multi-category classification using an extreme learning machine for microarray gene expression cancer diagnosis. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **4**, 485–495 (2007)
29. Wang, G., Zhao, Y., Wang, D.: A protein secondary structure prediction framework based on the extreme learning machine. *Neurocomputing* **72**, 262–268 (2008)
30. Wang, D.D., Wang, R., Yan, H.: Fast prediction of protein-protein interaction sites based on extreme learning machines. *Neurocomputing* **128**, 258–266 (2014)
31. Zhang, R., Huang, G.B., Sundararajan, N., Saratchandran, P.: Multicategory classification using an extreme learning machine for microarray gene expression cancer diagnosis. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **4**, 485–495 (2007)
32. Yeu, C.W.T., Lim, M.L., Huang, G.B., Agarwal, A., Ong, Y.S.: A new machine learning paradigm for terrain reconstruction. *IEEE Geosci. Remote Sens. Lett.* **3**, 382–386 (2006)
33. Huang, G.B., Ding, X., Zhou, H.: Optimization method based extreme learning machine for classification. *Neurocomputing* **74**, 155–163 (2010)
34. Huang, G.B., Zhu, Q.Y., Siew, C.K.: Extreme learning machine: theory and applications. *Neurocomputing* **70**, 489–501 (2006)