ADAPTIVE THRESHOLDING FOR MULTI-LABEL SVM CLASSIFICATION WITH APPLICATION TO PROTEIN SUBCELLULAR LOCALIZATION PREDICTION

Shibiao Wan, Man-Wai Mak

Dept. of Electronic and Information Engineering
The Hong Kong Polytechnic University
Hung Hom, Hong Kong SAR, China
email: {shibiao.wan, enmwmak}@polyu.edu.hk

Sun-Yuan Kung

Dept. of Electrical Engineering
Princeton University
New Jersey, USA
email: kung@princeton.edu

ABSTRACT

Multi-label classification has received increasing attention in computational proteomics, especially in protein subcellular localization. Many existing multi-label protein predictors suffer from over-prediction because they use a fixed decision threshold to determine the number of labels to which a query protein should be assigned. To address this problem, this paper proposes an adaptive thresholding scheme for multi-label support vector machine (SVM) classifiers. Specifically, each one-vs-rest SVM has an adaptive threshold that is a fraction of the maximum score of the one-vs-rest SVMs in the classifier. Therefore, the number of class labels of the query protein depends on the confidence of the SVMs in the classification. This scheme is integrated into our recently proposed subcellular localization predictor that uses the frequency of occurrences of gene-ontology terms as feature vectors and one-vs-rest SVMs as classifiers. Experimental results on two recent datasets suggest that the scheme can effectively avoid both over-prediction and under-prediction, resulting in performance significantly better than other gene-ontology based subcellular localization predictors.

Index Terms— Multi-label classification; Protein subcellular localization; Adaptive thresholding; Gene Ontology; Multi-label SVM.

1. INTRODUCTION

In supervised learning, the problem of assigning more than one label to each data instance is known as multi-label classification. In the past decades, multi-label classification has received significant attention in a wide range of problem domains, such as text classification [1], semantic annotation of images [2], and music categorization [3].

The existing methods for multi-label classification can be grouped into two main categories: (1) algorithm adaptation and (2) problem transformation. Algorithm adaptation methods extend single-label algorithms to solve multi-label classification problems. Typical methods include multi-label C4.5 [4], multi-label decision trees [5] and AdaBoost.MH [1]. Problem transformation methods transform a multi-label learning problem into one or more single-label classification problems [2] so that traditional single-label classifiers can be applied without modification. Typical methods include label powerset (LP) [6], binary relevance (BR) [7], ensembles of classifier chains (ECC) [8] and compressive sensing [9]. The LP method reduces a multi-label task to a single-label task by treating each possible multi-label subset as a new class in the single-label classification task. This method is simple, but is likely to generate a large number of classes, many of which are associated with very few examples. BR is a popular problem-transformation method. It transforms a multi-label task into many binary classification tasks, one for each label. Given a query instance, its predicted label(s) are the union of the positive-class labels output by these binary classifiers. BR is effective, but it neglects the correlation between labels, which may carry useful information for multi-label classification. The classifier chain method is a variant of BR but it can take the correlation between labels into account. Similar to BR, a set of one-vs-rest binary classifiers are trained. But unlike BR, the classifiers are linked in a chain and the feature vectors presented to the i-th classifier in the chain are augmented with the binary values representing the label(s) of the feature vectors up to the $(i-1)$-th class. Therefore, label dependence is preserved through the feature space. Classification performance, however, depends on the chain order. This order-dependency can be overcome by ensembles of classifier chains [8]. The compressive sensing approach is motivated by the fact that when the number of classes is large, the actual labels are often sparse. In other words, a typical query instance will belong to a few classes only, even though the total number of classes is large. This approach exploits the sparsity of the output (label) space by means of compressive sensing to obtain a more efficient output coding scheme for large-scale multi-label learning problems.

Compared to algorithm adaptation methods, one advantage of problem transformation methods is that any algorithm which is not capable of dealing with multi-label classification problems can be easily extended to deal with multi-label classification via transformation. It should be pointed out that the multi-label classification methods are different from the multi-class classification methods, such as error-correcting output coding methods [10], pairwise comparison methods [11], and so on. There is probably no multi-class method that outperforms all others in all circumstances [12], so is the same case for multi-label methods.

Several algorithms based on support vector machines (SVM) [13] have been proposed to tackle multi-label classification problems. In [14], a ranking-SVM approach that minimizes the margin and the ranking loss [1] at the same time is proposed. In [15], three improvements to enhance the BR method for SVM classifiers are presented. The first improvement is to extend the original data set with some additional features indicating the relationship between classes. The second improvement is to remove negative training instances if they are similar to the positive training instances. And the third improvement is to remove very similar negative training instances that are within a pre-defined distance from the decision boundary.

This work was in part supported by The Hong Kong Research Grant Council, Grant No. PolyU5264/09E and HKPolyU Grant No. G-U877.

978-1-4799-0356-6/13/$31.00 ©2013 IEEE

In the past decades, multi-label classification methods have been increasingly applied in bioinformatics, especially in protein subcellular localization. Protein subcellular localization is a problem of predicting which part(s) in a cell a protein resides. This information is vitally important for understanding the functions of proteins and for identifying drug targets [16]. This problem has been extensively studied in previous decades and many computational methods [17–22] have been developed. More information about protein subcellular localization can be found in a comprehensive review [23]. Recently, several multi-label predictors have been proposed to deal with the prediction of multi-label proteins, such as Virus-mPLoc [24], iLoc-Virus [25], KNN-SVM ensemble classifier [26], and mGOASVM [27]. They all use the Gene Ontology (GO)[1] information as the features and apply different multi-label classifiers to tackle the multi-label classification problem. Among them, Virus-mPLoc and iLoc-Virus use algorithm adaptation methods, while KNN-SVM and mGOASVM use problem transformation methods.

This paper proposes a multi-label SVM classifier for predicting subcellular localization of multi-label proteins. The method extends the BR methods with an adaptive thresholding decision scheme that essentially converts the linear SVMs in the classifier into piecewise linear SVMs, which effectively reduces the over-prediction instances while having little influence on the correctly predicted ones. Results on two recent benchmark datasets demonstrate that the proposed predictor can substantially outperform the state-of-the-art predictors.

2. FEATURE EXTRACTION

2.1. Retrieval of GO Terms

Given a query protein, the predictor described in this paper can use either its protein accession number (AC) or its protein sequence as input. For proteins with known ACs, their respective GO terms are retrieved from the Gene Ontology Annotation (GOA) database[2] using the ACs as the searching keys. For a protein without an AC, its amino acid sequence is presented to BLAST [28] to find its homologs, whose ACs are then used as keys to search against the GOA database.

While the GOA database allows us to associate the AC of a protein with a set of GO terms, for some novel proteins, neither their ACs nor the ACs of their top homologs have any entries in the GOA database; in other words, no GO terms can be retrieved by their ACs or the ACs of their top homologs. In such case, the ACs of the homologous proteins, as returned from BLAST search, will be successively used to search against the GOA database until a match is found.

2.2. Construction of GO Vectors

Given a dataset, we used the procedure described in Section 2.1 to retrieve the GO terms of all of its proteins. Then, we determined the number of distinct GO terms corresponding to the dataset. Suppose T distinct GO terms were found; these GO terms form a GO Euclidean space with T dimensions. For each sequence in the dataset, we constructed a GO vector by matching its GO terms to all of the T GO terms. Unlike the conventional 1-0 value [24, 29] to determine the elements of the GO vectors, we used Term-Frequency [30] to construct the GO vectors.

The approach of Term-Frequency (TF) is similar to the 1-0 value approach in that a protein is represented by a point in a Euclidean space. However, unlike the 1-0 approach, it uses the number of occurrences of individual GO terms as the coordinates. Specifically, the GO vector $\mathbf{P}_i$ of the i-th protein is defined as:

$$\mathbf{P}_i = [b_{i,1}, \cdots, b_{i,j}, \cdots, b_{i,T}]^{\mathsf{T}}, b_{i,j} = \begin{cases} f_{i,j} & \text{, GO hit} \\ 0 & \text{, otherwise} \end{cases} \tag{1}$$

where $f_{i,j}$ is the number of occurrences of the j-th GO term (term-frequency) in the i-th protein sequence. The rationale is that the term-frequencies may also contain important information for classification and therefore should not be quantized to either 0 or 1. Note that $b_{i,j}$'s are analogous to the term-frequencies commonly used in document retrieval.

3. ADAPTIVE THRESHOLDING FOR SVM

3.1. Multi-label SVM Scoring

GO vectors are used for training the multi-label one-vs-rest SVMs. Specifically, for an M-class problem (here M is the number of subcellular locations), M independent binary SVMs are trained, one for each class. Denote the GO vector created by using the true AC of the i-th query protein as $\mathbf{q}_{i,0}$ and the GO vector created by using the accession number of the k-th homolog as $\mathbf{q}_{i,k}$, $k = 1, \ldots, k_{\max}$, where $k_{\max}$ is the number of homologs retrieved by BLAST with the default parameter setting. Then, given the i-th query protein $\mathbf{Q}_i$, the score of the m-th SVM is:

$$s_m(\mathbf{Q}_i) = \sum_{r \in \mathcal{S}_m} \alpha_{m,r} y_{m,r} K(\mathbf{P}_r, \mathbf{q}_{i,k}) + b_m \tag{2}$$

where $\mathcal{S}_m$ is the set of support vector indexes corresponding to the m-th SVM, $y_{m,r} \in \{-1, +1\}$ are the class labels, $\alpha_{m,r}$ are the Lagrange multipliers, $K(\cdot, \cdot)$ is a kernel function; here, the linear kernel is used. Note that $\mathbf{P}_r$'s in Eq. 2 represents the GO training vectors, which may include the GO vectors created by using the true AC of the training sequences or their homologous ACs.

3.2. Adaptive Thresholding

To predict the subcellular locations of datasets containing both single-label and multi-label proteins, an adaptive thresholding decision scheme for multi-label SVM classifiers is proposed in this paper. Unlike the single-label problem where each protein has one predicted label only, a multi-label protein could have more than one predicted labels. Thus, the predicted subcellular location(s) of the i-th query protein are given by:
If $\exists\, s_m(\mathbf{Q}_i) > 0$,

$$\mathcal{M}(\mathbf{Q}_i) = \bigcup_{m=1}^{M} \{\{m : s_m(\mathbf{Q}_i) > 1.0\} \cup \{m : s_m(\mathbf{Q}_i) \geq f(s_{\max}(\mathbf{Q}_i))\}\} \tag{3}$$

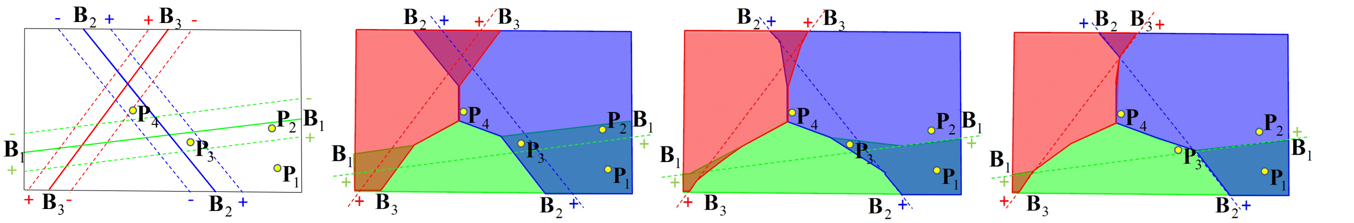
otherwise,

$$\mathcal{M}(\mathbf{Q}_i) = \arg\max_{m=1}^{M} s_m(\mathbf{Q}_i). \tag{4}$$

In Eq. 3, $f(s_{\max}(\mathbf{Q}_i))$ is a function of $s_{\max}(\mathbf{Q}_i)$, where $s_{\max}(\mathbf{Q}_i) = \max_{m=1}^{M} s_m(\mathbf{Q}_i)$. In this paper, we use a linear function, i.e., $f(s_{\max}(\mathbf{Q}_i)) = \theta s_{\max}(\mathbf{Q}_i)$, where $\theta \in [0.0, 1.0]$ is a parameter. Because $f(s_{\max}(\mathbf{Q}_i))$ is linear, Eq. 3 and Eq. 4 turn the linear SVMs into piecewise linear SVMs. Eq. 3 also suggests that the predicted labels depend on $s_{\max}(\mathbf{Q}_i)$, a function of the test instance (or protein). This means that the threshold is adaptive to the test protein. For ease of reference, we refer to the proposed predictor as AT-SVM.

[1] http://www.geneontology.org

[2] http://www.ebi.ac.uk/GOA



(a) The original SVM (b) $\theta = 0.0$ (c) $\theta = 0.5$ (d) $\theta = 1.0$

Fig. 1. A 3-class example illustrating how the adaptive thresholding scheme changes the decision boundaries from linear to piecewise linear and how the resulting SVMs assign label(s) to test points when θ changes from 0 to 1. In (a), the solid and dashed lines respectively represent the decision boundaries and margins of individual SVMs. In (b)–(d), the input space is divided into three 1-label regions (green, blue and red) and three 2-label regions (green ∩ blue, blue ∩ red, and red ∩ green).

To facilitate discussion, let's define two terms: *over-prediction* and *under-prediction*. Specifically, over (under) prediction means that the number of predicted labels of a query protein is larger (smaller) than the ground-truth. In this paper, both over- and under-predictions are considered as incorrect predictions, which will be reflected in the "actual accuracy" to be defined in Section 4.

Conventional methods use a fixed threshold to determine the predicted classes. When the threshold is too small, the prediction results are liable to over-prediction; on the other hand, when the threshold is too large, the prediction results are susceptible to under-prediction. To overcome this problem, the adaptive thresholding scheme in the classifier uses the maximum score ($s_{\max}(\mathbf{Q}_i)$) among the one-vs-rest SVMs in the classifier as a reference. In particular, $s_{\max}(\mathbf{Q}_i)$ in Eq. 3 adaptively normalizes the scores of all one-vs-rest SVMs so that for SVMs to be considered as runner-ups, they need to have a sufficiently large score relative to the winner. This strategy effectively reduces the chance of over-prediction. The first condition in Eq. 3 ($s_m(\mathbf{Q}_i) > 1$) aims to avoid under-prediction when the winning SVM has very high confidence (i.e., $s_{\max}(\mathbf{Q}_i) \gg 1$) but the runners-up still have enough confidence ($s_m(\mathbf{Q}_i) > 1$) in making a right decision.[3] On the other hand, when the maximum score is small (say $0 < s_{\max}(\mathbf{Q}_i) \leq 1$), θ in the second term of Eq. 3 can strike a balance between over-prediction and under-prediction. When all of the SVMs have very low confidence (say $s_{\max}(\mathbf{Q}_i) < 0$), the classifier switches to single-label mode via Eq. 4.

To further illustrate how this decision scheme works, an example is shown in Fig. 1. Suppose there are 4 test data points ($\mathbf{P}_1, \ldots, \mathbf{P}_4$) which are possibly distributed into 3 classes: {green, blue, red}. The decision boundaries of individual SVMs and the 4 points are shown in Fig. 1(a). Suppose $s_m(\mathbf{P}_i)$ is the SVM score of $\mathbf{P}_i$ with respect to class m, where $i = \{1, \ldots, 4\}$ and $m \in$ {green, blue, red}. Fig. 1(a) suggests the following conditions:

$$\begin{array}{rrr} s_{\text{green}}(\mathbf{P}_1) > 1, & s_{\text{blue}}(\mathbf{P}_1) > 1, & s_{\text{red}}(\mathbf{P}_1) < 0; \\ 0 < s_{\text{green}}(\mathbf{P}_2) < 1, & s_{\text{blue}}(\mathbf{P}_2) > 1, & s_{\text{red}}(\mathbf{P}_2) < 0; \\ 0 < s_{\text{green}}(\mathbf{P}_3) < 1, & 0 < s_{\text{blue}}(\mathbf{P}_3) < 1, & s_{\text{red}}(\mathbf{P}_3) < 0; \\ s_{\text{green}}(\mathbf{P}_4) < 0, & s_{\text{blue}}(\mathbf{P}_4) < 0, & s_{\text{red}}(\mathbf{P}_4) < 0. \end{array}$$

Note that points whose scores lie between 0 and 1 are susceptible to over-prediction because they are very close to the decision boundaries of the corresponding SVM. The decision scheme used in [27] (i.e., $\theta = 0.0$) leads to the decision boundaries shown in Fig. 1(b). Based on these boundaries, $\mathbf{P}_1$, $\mathbf{P}_2$ and $\mathbf{P}_3$ will be assigned to class green ∩ blue , and $\mathbf{P}_4$ will be assigned to the class with the highest SVM score (using Eq. 4). If θ increases to 0.5, the results shown in Fig. 1(c) will be obtained. The assignments of $\mathbf{P}_1$, $\mathbf{P}_3$ and $\mathbf{P}_4$ remain unchanged but $\mathbf{P}_2$ will be changed from class green ∩ blue to class blue. Similarly, when θ increases to 1.0 (Fig. 1(d)), then the class of $\mathbf{P}_3$ will also be determined by the SVM with the highest score. This analysis suggests that when θ increases from 0 to 1, the decision criterion becomes more stringent, which has the effect of shrinking the 2-label regions in Fig. 1, thus reducing the over-prediction. Provided that θ is not close to 1, this reduction in over-prediction will not compromise the decisions made by the high scoring SVMs.

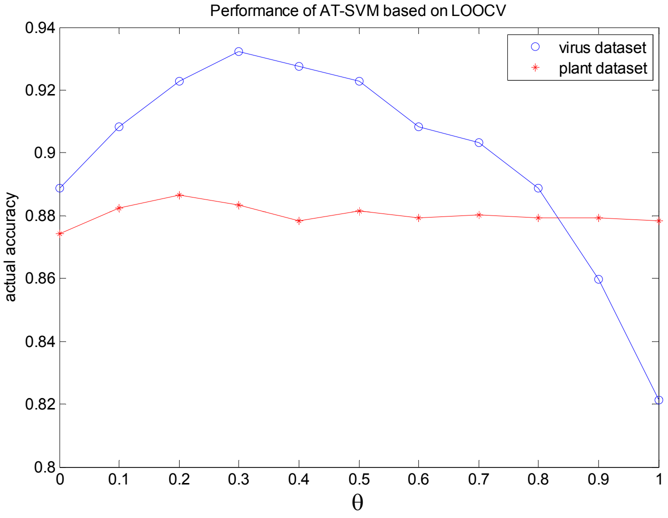


Fig. 2. Performance of AT-SVM based on leave-one-out cross-validation (LOOCV) varying with θ. $\theta = 0$ represents the performance of mGOASVM.

4. EXPERIMENTS AND RESULTS

4.1. Datasets and Performance Measures

In this paper, a virus dataset [24, 25] and a plant dataset [31] were used to evaluate the performance of the proposed predictor. The virus and the plant datasets were created from Swiss-Prot 57.9 and 55.3, respectively. The virus dataset contains 207 viral proteins distributed in 6 locations. Of the 207 viral proteins, 165 belong to one subcellular locations, 39 to two locations, 3 to three locations and none to four or more locations. This means that about 20% of proteins are located in more than one subcellular location. The plant dataset contains 978 plant proteins distributed in 12 locations. Of the 978 plant proteins, 904 belong to one subcellular locations, 71 to

[3] SVM scores larger than one means that the test proteins fall beyond the margin of separation; therefore, the confidence is fairly high.

Table 1. Comparing AT-SVM with state-of-the-art multi-label predictors based on leave-one-out cross validation (LOOCV) using the virus dataset. "–" means the corresponding references do not provide the overall actual accuracy. *Host ER*: Host endoplasmic reticulum.

Label	Subcellular Location	LOOCV Locative Accuracy				
		Virus-mPLoc [24]	KNN-SVM [26]	iLoc-Virus [25]	mGOASVM [27]	AT-SVM
1	Viral capsid	8/8 = 100.0%	8/8 = 100.0%	8/8 = 100.0%	8/8 = 100.0%	8/8 = 100.0%
2	Host cell membrane	19/33 = 57.6%	27/33 = 81.8%	25/33 = 75.8%	32/33 = 97.0%	32/33 = 97.0%
3	Host ER	13/20 = 65.0%	15/20 = 75.0%	15/20 = 75.0%	17/20 = 85.0%	17/20 = 85.0%
4	Host cytoplasm	52/87 = 59.8%	86/87 = 98.8%	64/87 = 73.6%	85/87 = 97.7%	83/87 = 95.4%
5	Host nucleus	51/84 = 60.7%	54/84 = 65.1%	70/84 = 83.3%	82/84 = 97.6%	82/84 = 97.6%
6	Secreted	9/20 = 45.0%	13/20 = 65.0%	15/20 = 75.0%	20/20 = 100.0%	20/20 = 100.0%
Overall Locative Accuracy		152/252 = 60.3%	203/252 = 80.7%	197/252 = 78.2%	244/252 = **96.8%**	242/252 = 96.0%
Overall Actual Accuracy		–	–	155/207 =74.8%	184/207 = 88.9%	193/207 = **93.2%**

two locations, 3 to three locations and none to four or more locations. The sequence identity of both datasets was cut off at 25%.

To facilitate comparison, the locative accuracy and actual accuracy [27] were used to assess the prediction performance. Specifically, denote $\mathcal{L}(\mathbf{P}_i)$ and $\mathcal{M}(\mathbf{P}_i)$ as the true label set and the predicted label set for the i-th protein $\mathbf{P}_i$ ($i = 1, \ldots, N$), respectively.[4] Then, the overall locative accuracy is:

$$\Lambda_{\text{loc}} = \frac{1}{\sum_{i=1}^{N} |\mathcal{L}(\mathbf{P}_i)|} \sum_{i=1}^{N} |\mathcal{M}(\mathbf{P}_i) \cap \mathcal{L}(\mathbf{P}_i)| \tag{5}$$

where $|\cdot|$ means counting the number of elements in the set therein and $\cap$ represents the intersection of sets. And the overall actual accuracy is:

$$\Lambda_{\text{act}} = \frac{1}{N} \sum_{i=1}^{N} \Delta[\mathcal{M}(\mathbf{P}_i), \mathcal{L}(\mathbf{P}_i)] \tag{6}$$

where

$$\Delta[\mathcal{M}(\mathbf{P}_i), \mathcal{L}(\mathbf{P}_i)] = \begin{cases} 1 & \text{, if } \mathcal{M}(\mathbf{P}_i) = \mathcal{L}(\mathbf{P}_i) \\ 0 & \text{, otherwise.} \end{cases} \tag{7}$$

Note that the actual accuracy is more objective and stricter than the locative accuracy [27].

4.2. Performance of AT-SVM

Fig. 2 shows the performance of AT-SVM on the virus dataset and the plant dataset with respect to the parameter θ based on leave-one-out cross-validation. As can be seen, for the virus dataset, as θ increases from 0.0 to 1.0, the overall actual accuracy increases first, reaches the peak at $\theta = 0.3$ (with an actual accuracy of 93.2%), and then decreases. An analysis of the predicted labels $\{\mathcal{L}(\mathbf{P}_i); i = 1, \ldots, N\}$ suggests that the increases in actual accuracy is due to the reduction in the number of over-prediction, i.e., the number of cases where $|\mathcal{M}(\mathbf{P}_i)| > |\mathcal{L}(\mathbf{P}_i)|$ has been reduced. When $\theta > 0.3$, the benefit of reducing the over-prediction diminishes because the criterion in Eq. 3 becomes so stringent that some of the proteins were under-predicted, i.e., the number of cases where $|\mathcal{M}(\mathbf{P}_i)| < |\mathcal{L}(\mathbf{P}_i)|$ increases. Note that the performance at $\theta = 0.0$ is equivalent to the performance of mGOASVM [27], and that the best actual accuracy (93.2% when $\theta = 0.3$) obtained by the proposed decision scheme is more than 4% (absolute) higher than mGOASVM (88.9%).

For the plant dataset, when θ increases from 0.0 to 1.0, the overall actual accuracy increases from 87.4%, and then fluctuates around 88%. If we take the same θ as that for the virus dataset, i.e., $\theta = 0.3$, the performance of AT-SVM is 88.3%, which is still better than that of mGOASVM at $\theta = 0.0$.

4.3. Comparing with State-of-the-Art Predictors

Table 1 compares the performance of AT-SVM against several state-of-the-art multi-label predictors on the virus dataset. All the predictors use the information of GO terms as features. From the perspective of classifiers, Virus-mPLoc [24] uses an ensemble OET-KNN (optimized evidence-theoretic K nearest neighbor) classifier; iLoc-Virus [25] uses a multi-label KNN classifier; KNN-SVM ensemble classifier [26] uses an ensemble classifier combining KNN and SVM; mGOASVM [27] uses a multi-label SVM classifier; and the proposed AT-SVM uses a multi-label SVM classifier incorporated with the proposed adaptive thresholding scheme.

As shown in Table 1, AT-SVM performs significantly better than Virus-mPLoc and iLoc-Virus. Both the overall locative accuracy and overall actual accuracy of AT-SVM are more than 17% (absolute) higher than iLoc-Virus (96.0% vs 78.2% and 93.2% vs 74.8%, respectively). AT-SVM also performs significantly better than the KNN-SVM ensemble classifier in terms of overall locative accuracy (96.0% vs 80.7%). When comparing with mGOASVM, although the locative accuracy of the proposed predictor is a bit lower than that of the mGOASVM (96.0% vs 96.8%), the overall actual accuracy of the proposed predictor performs more than 4% higher than mGOASVM, The results suggest that the improved multi-label SVM classifier using the proposed adaptive thresholding decision scheme performs better than the state-of-the-art classifiers. As for the individual locative accuracy, except for the "viral capsid" for which all the predictors reach 100%, the locative accuracies of the proposed predictor are remarkably higher than those of Virus-mPLoc, iLoc-Virus and KNN-SVM, and are comparable to mGOASVM.

5. CONCLUSIONS

This paper proposes an efficient multi-label SVM classifier, namely AT-SVM, incorporated with an adaptive thresholding decision scheme to predict subcellular localization of multi-label proteins. Given a query protein, the GO information is extracted by using either its accession number or its homologous accession number as keys to search against GO annotation database, which is subsequently used to construct GO vectors. After scoring the GO vectors by the multi-label SVM classifier, the predicted results are determined by an adaptive thresholding decision scheme. Results on two benchmark datasets demonstrate that the adaptive threshold scheme can be readily integrated into multi-label SVM classifiers.

[4] Here, $N = 207$ for the virus dataset and $N = 978$ for the plant dataset.

6. REFERENCES

[1] R. E. Schapire and Y. Singer, “Boostexter: A boosting-based system for text categorization,” *Machine learning*, vol. 39, no. 2/3, pp. 135–168, 2000.

[2] M. Boutell, J. Luo, X. Shen, and C. Brown, “Learning multi-label scene classification,” *Pattern Recognition*, vol. 37, no. 9, pp. 1757–1771, 2004.

[3] T. Li and M. Ogihara, “Toward intelligent music information retrieval,” *IEEE Transactions on Multimedia*, vol. 8, no. 3, pp. 564–574, 2006.

[4] A. Clare and R. D. King, “Knowledge discovery in multi-label phenotype data,” in *Proceedings of the 5th European Conference on Principles of Data Mining and Knowledge Discovery*, 2001, pp. 42–53.

[5] C. Vens, J. Struyf, L. Schietgat, S. Dzeroski, and H. Blockeel, “Decision trees for hierarchical multi-label classification,” *Machine Learning*, vol. 2, no. 73, pp. 185–214, 2008.

[6] G. Tsoumakas, I. Katakis, and I. Vlahavas, “Mining multi-label data,” in *Data Mining and Knowledge Discovery Handbook, O. Maimon, l. Rokach (Ed.). Springer, 2nd edition*, 2010.

[7] G. Tsoumakas and I. Katakis, “Multi-label classification: An overview,” *International Journal of Data Warehousing and Mining*, vol. 3, pp. 1–13, 2007.

[8] J. Read, B. Pfahringer, G. Holmes, and E. Frank, “Classifier chains for multi-label classification,” in *Proceedings of European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, 2009, pp. 254–269.

[9] D. Hsu, S. M. Kakade, J. Langford, and T. Zhang, “ Multi-label prediction via compressed sensing,” in *Advances in Neural Information Processing Systems 22*, 2009, pp. 772–780.

[10] T. G. Dietterich and g. Bakari, “ Solving multiclass learning problem via error-correcting output codes,” *Journal of Artificial Intelligence Research*, pp. 263–286, 1995.

[11] U. Kressel, “Pairwise classification and support vector machines,” in *Advances in Kernel Methods: Support Vector Learning, Chap. 15. MIT Press*, 1999.

[12] B. Scholkopf and A. J. Smola, “Learning with kernels,” in *MIT Press*, 2002.

[13] V. N. Vapnik, “Statistical learning theory,” in *John Wiley & Sons*, 1998.

[14] A. Elisseeff and J. Weston, “Kernel methods for multi-labelled classification and categorical regression problems.,” in *In Advances in Neural Information Processing Systems 14*. 2001, pp. 681–687, MIT Press.

[15] S. Godbole and S. Sarawagi, “Discriminative methods for multi-labeled classification,” in *Proceedings of the 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining*. 2004, pp. 22–30, Springer.

[16] G. Lubec, L Afjehi-Sadat, J. W. Yang, and J. P. John, “Searching for hypothetical proteins: theory and practice based upon original data and literature,” *Prog. Neurobiol*, vol. 77, pp. 90–127, 2005.

[17] H. Nakashima and K. Nishikawa, “Discrimination of intracellular and extracellular proteins using amino acid composition and residue-pair frequencies,” *J. Mol. Biol.*, vol. 238, pp. 54–61, 1994.

[18] S. Wan, M. W. Mak, and S. Y. Kung, “Protein subcellular localization prediction based on profile alignment and Gene Ontology,” in *2011 IEEE International Workshop on Machine Learning for Signal Processing (MLSP’11)*, Sept 2011, pp. 1–6.

[19] K. C. Chou and H. B. Shen, “Predicting eukaryotic protein subcellular location by fusing optimized evidence-theoretic K-nearest neighbor classifiers,” *J. of Proteome Research*, vol. 5, pp. 1888–1897, 2006.

[20] M.S. Scott, D.Y. Thomas, and M.T. Hallett, “Predicting subcellular localization via protein motif co-occurrence,” *Genome research*, vol. 14, no. 10a, pp. 1957–1966, 2004.

[21] S. Wan, M. W. Mak, and S. Y. Kung, “GOASVM: Protein subcellular localization prediction based on gene ontology annotation and SVM,” in *2012 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP’12)*, 2012, pp. 2229–2232.

[22] S. Y. Mei, W. Fei, and S. G. Zhou, “Gene ontology based transfer learning for protein subcellular localization,” *BMC Bioinformatics*, vol. 12, pp. 44, 2011.

[23] K. C. Chou and H. B. Shen, “Recent progress in protein subcellular location prediction,” *Analytical Biochemistry*, vol. 1, no. 370, pp. 1–16, 2007.

[24] H. B. Shen and K. C. Chou, “Virus-mPLoc: a fusion classifier for viral protein subcellular location prediction by incorporating multiple sites,” *J. Biomol. Struct. Dyn.*, vol. 26, pp. 175–186, 2010.

[25] X. Xiao, Z. C. Wu, and K. C. Chou, “iLoc-Virus: A multi-label learning classifier for identifying the subcellular localization of virus proteins with both single and multiple sites,” *Journal of Theoretical Biology*, vol. 284, pp. 42–51, 2011.

[26] L. Q. Li, Y. Zhang, L. Y. Zou, Y. Zhou, and X. Q. Zheng, “Prediction of protein subcellular multi-localization based on the general form of Chou’s pseudo amino acid composition,” *Protein & Peptide Letters*, vol. 19, pp. 375–387, 2012.

[27] S. Wan, M. W. Mak, and S. Y. Kung, “mGOASVM: Multi-label protein subcellular localization based on gene ontology and support vector machines,” *BMC Bioinformatics*, vol. 13, pp. 290, 2012.

[28] S. F. Altschul, T. L. Madden, A. A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman, “Gapped BLAST and PSI-BLAST: A new generation of protein database search programs,” *Nucleic Acids Res.*, vol. 25, pp. 3389–3402, 1997.

[29] K. C. Chou and H. B. Shen, “Plant-mPLoc: A top-down strategy to augment the power for predicting plant protein subcellular localization,” *PLoS ONE*, vol. 5, pp. e11335, 2010.

[30] S. Wan, M. W. Mak, and S. Y. Kung, “GOASVM: A subcellular location predictor by incorporating term-frequency gene ontology into the general form of Chou’s pseudo-amino acid composition,” *Journal of Theoretical Biology*, vol. 323, pp. 40–48, 2013.

[31] Z. C. Wu, X. Xiao, and K. C. Chou, “iLoc-Plant: A multi-label classifier for predicting the subcellular localization of plant proteins with both single and multiple sites,” *Molecular BioSystems*, vol. 7, pp. 3287–3297, 2011.