2009 International Conference on Machine Learning and Computing
IPCSIT vol.3 (2011) © (2011) IACSIT Press, Singapore

Semi-supervised Multi-label Learning Algorithm Using Dependency Among Labels

Qu Wei [1], Zhang Yang [1+], Zhu Junping [1], Yong Wang [2]

[1] College of Information Engineer, Northwest A&F University, P.R. China

[2] School of Computer, Northwest Polytechnical University, China

Abstract. In this paper, we present a semi-supervised algorithm for multi-label learning by exploring the relationship among labels. Based on the accuracy, we determine the classification order for labels, a list of classifiers is trained by this order, with each classifier being trained by using the outputs of the previous classifiers in the list as additional input features. Experiments on three multi-label data sets show that our algorithm has substantial advantage over the comparing algorithms.

Keywords: semi-supervised, multi-label, feature selection

1. Introduction

Traditional classifiers use only labeled examples for training, however, it is expensive, difficult and time consuming to obtain labeled examples [1], as they require human effort. Meanwhile unlabeled data is relatively easy to collect. Semi-supervised learning addresses this problem by using large amount of unlabeled data, together with labeled data, to build better classifiers [1].

One the other hand, in the real world, an example can be assigned to multiple different categories, which is generally named as multi-label learning problem [2]. For example, in bioinformatics, most genes are associated with more than one functional class [2]. In automatic image annotation, an image can be denoted to more than one class simultaneously. Most previous work on multi-label learning focused on supervised learning [10]. Recently, some semi-supervised approaches have tried to exploit the multi-label relations [2, 3, 4, 5]. In this paper, we proposed a new algorithm by reviewing this problem from a new perspective. Firstly, we explore the relationship among labels by feature selection algorithm, and we adopt cross-validation method to calculate the classification accuracy on each label. Based on the accuracy, the classification order for labels is determined, then we train a list of classifier according to the order, each classifier in list is trained with additional features, which are provided by the outputs of the previous classifiers in the list.

The rest of the paper is organized as follows. In section 2 we review the related works. In section 3 we present our algorithm. Experimental setting and experiment results presented in section 4. Finally, section 5 concludes this paper.

2. Related Works

Liu et al. [2] assume that two examples tent to have large overlap in their assigned class memberships if they share high similarity in their input patterns, they search for the optimal assignment of label memberships to the unlabeled data that minimizes the difference between two similarity set, one based on input patterns and the other based on the labels. The optimization problem is formulated as a constrained non-negative matrix factorization problem. Chen et al. [4] and Zha et al. [3] construct two graphs on feature

[+] Corresponding author. Tel.: +(86-29-87091249); fax: +(86-29-87092353).
E-mail address: (zhangyang@nwsuaf.edu.cn).

and label level respectively, and then explore the multi-label inter-similarity, which leads to the label smoothness over multiple labels for each example, and that is not true in some cases [5].

Wang et al. [5] present an approach based on the discrete hidden Markov random field model, they formulate the multi-label interdependence as a pairwise Markov random field model, which explores all the combinations of relations. This approach does not have the drawbacks of those previously mentioned.

Comparing with the previous works, we review this problem in a new perspective, out approach is simple, effective and also avoid drawbacks mentioned above by taking advantage of dependency information among labels.

3. Semi-supervised Multi-label Learning Algorithm

Here, we introduce some notations that are used throughout the paper. Suppose there are k labeled instances $D=\{(x_1,y_i),...,(x_k,y_k)\}$, and u unlabeled instances $U=\{x_{k+1},...,x_{k+u}\}$, where each $x_i=(x_{i1},...,x_{im})^T$ is a m-dimentional feature vector and each $y_i=(y_{i1},y_{i2},...,y_{i|L|})^T$ is a |L|-dimensional label vector.

3.1. Explore the Relationship among Labels

In supervised multi-label classification scenario, when training, it helps a lot if we select a single label as *currentlabel* , and treat other labels as features. The reason is that there is dependence among labels. For example, if a video can be assigned to “mountain” and it likely can be assigned to “outdoor” [5].

Taking $l\in L$ as *currentlabel* , feature selection algorithm could be used to select a set of related labels to l from $L-\{l\}$. We refer the set of related labels to l as $R(l)$.

3.2. Semi-supervised Multi-label Learning Algorithm

Before we introduce our algorithm, we first introduce Binary Relevance (BR) [6] problem transformation method. For each label $l\in L$, it transforms original multi-label data set into $|L|$ data sets $\{D_1,D_2,...,D_{|L|}\}$. For each example in $\{D_1,D_2,...,D_{|L|}\}$, label it as l if original example contained l , and as $\neg l$ otherwise. In this way, there are $|L|$ binary classifiers trained from $|L|$ data sets $D_l(l\in L)$ respectively. The final labels for each example can be determined by combining the classification results from all the classifiers.

We sort these binary classifiers into $list=<c_{j_1},c_{j_2},...,c_{j_{|L|}}>$, where $j_l(1\le j_l\le|L|)$ is the index of $list$, and use the outputs of classifiers in $\{c_{j_1},c_{j_2},...,c_{j_{i-1}}\}$ as additional input features to classifier c_{j_i} , so as to take advantage of dependency information among labels. The classifiers are sorted according to their classification accuracy, which is calculated with cross-validation method, because if a classifier with low accuracy is listed at the head of $list$, then the wrong output of the classifier will be feed to the following classifiers, so it would have bad effects on their classification performance.

Firstly, the original multi-label data set D is transformed into $|L|$ single-label data set $\{D_1,D_2,...,D_{|L|}\}$. For each $D_l(l\in L)$, we adopt cross-validation method to calculate the label accuracy. Suppose the label with the highest accuracy is i , then a new classifier c_i is learned from D_i and unlabeled data set U by a certain semi-supervised algorithm, c_i is appended to the end of $list$, and it is used to label examples in U . After labeling, we take label i as a feature. Thus, $D_1,D_2,...,D_{i-1},D_{i+1},...,D_{|L|}$ are transformed by taking label i in D as a new feature, and U is transformed by taking the outputs of the c_i on U as a new feature, so they all have $m+1$ features. Meanwhile, the label set $L=L-\{i\}$.

Then, we update the accuracy of $D_l(l\in R(i)\bigcap L)$, after updating, we select the label with highest accuracy in L . This process is repeated until all the labels have been processed.

Algorithm 1 gives an outline of our semi-supervised multi-label learning algorithm. In step 1, D is transformed into $|L|$ single label data sets. In step 2-11, cross-validation method is adopted to calculate the accuracy on each label. In step 13, the label i with the highest accuracy is selected, in step 14, a classifier c_i is trained from D_i and U , in step 15, U is classified by c_i , and is transformed by taking the outputs as a new feature, in step 16-18, each D_l add the original label i as a new feature. In step 19, we update $|L|$ and append c_i in a list. In step 20, we update the accuracy in $R(i)\bigcap L$, with the same method in step 3-10.

Algorithm 1. The Multi-Label Semi-Supervised Learning Algorithm

Input:

D : the labeled multi-label data set

U : the unlabeled data set

L : the label set

Output:

$list$: the list of semi-supervised classifier

transform D into $|L|$ single-label data sets $\{D_1, D_2, ..., D_{|L|}\}$
for each $l \in L$ **do**
  split D_l into k subsets $\{D_{l1}, D_{l2}, ..., D_{lk}\}$ with the same size
  for $i \in \{0...k\}$ **do**
    set D_{li} as test set, and $(D_l - D_{li}) \bigcup U$ as training set
    training a classifier c_{li} by a certain semi-supervised learning algorithm
    calculate accuracy a_{li} by c_{li} on test set
    $sum = sum + a_{li}$
  end for
  $acc_l = sum / k$
end for
while $L \neq \phi$ **do**
  $i = \arg\max(acc_l)$
  train a classifier c_i on $D_i \bigcup U$ by a certain semi-supervised algorithm
  label U by c_i, transform U by taking the outputs as a feature
  for each $l \in L$ **do**
    transform data set D_l by taking label i as a feature
  end for
  $L = L - \{l\}$, $list$.Append(c_i)
  update accuracy with D_k and U, $k \in R(i) \bigcap L$
end while
return $list$

The algorithm for classification is straightforward, and it is omitted here. Basically, given a test example, the final labels are determined by combining the classification results from each classifier in $list$ following the classification order.

4. Experiment

In this section, we evaluate the proposed algorithm on three multi-label data sets. The algorithms are implemented in Java with help of WEKA[1], Mulan[2], and SVMlin[3].

4.1. Experimental Setup

Three multi-label data sets[4] are used in the experiments, some statistics on these data sets are tabulated in Table 1. For supervised learning algorithm, we use C4.5 and SVM algorithms. For semi-supervised learning algorithm, we use Tri-train [11], YATSI [12], and TSVM [13] algorithms. For inductive learning algorithm (Tri-train, TSVM), 15% of the examples in the data set are used as testing data set, 20% as training data set, and the rest of the examples as unlabeled data set. For transduction learning algorithm (YATSI), 20% of the examples is used as training data set, and the rest of the examples as unlabeled data set.

[1] http://www.cs.waikato.ac.nz/ml/weka/
[2] Mulan－**Mu**lti**La**bel Classificatio**n,** (http://mlkd.csd.auth.gr/multilabel.html)
[3] http://people.cs.uchicago.edu/~vikass/svmlin.html
[4] http://mlkd.csd.auth.gr/multilabel.html

Five evaluation metrics are used in the experiments, they are hamming loss [7], accuracy [8], precision [8], recall [8], and F1 [9].

Table 1. Experimental Data Sets

name	instances	attributes	labels	cardinality	density	distinct
emotions	593	72	6	1.869	0.311	27
scene	2047	294	6	1.074	0.179	15
yeast	2417	103	14	4.237	0.303	198

4.2. Experimental Results

In order to show semi-supervised learning algorithms usually have better classification performance comparing with supervised learning algorithm. We design another algorithm, which learn from labelled examples only, we use C4.5 and SVM as base classifier which are refered as C4.5-sl, and SVM-sl respectively. In order to show that the relationship among labels helps to train better classifiers. We design an algorithm, which ignore the relationship between labels, that is, applying BR method in semi-supervised learning. In this paper, the experiment results based on the BR method are refered as Tri-BR, YATSI-BR and TSVM-BR, and the experiment results based on the proposed approach are refered as Tri-C, YATSI-C and TSVM-C.

Table 2. Experimental Results For C4.5

Data Set	Metric	C4.5-sl	Tri-BR	Tri-C	YATSI-BR	YATSI-C
emotions	hammingloss	27.26	22.85	21.13	19.54	18.84
	accuracy	40.81	46.59	54.91	52.27	55.00
	precision	53.26	59.69	64.94	66.40	68.19
	recall	52.66	56.69	66.30	62.46	67.03
	F1	52.89	58.05	62.52	64.22	67.55
scene	hammingloss	15.44	12.54	11.51	9.65	8.65
	accuracy	45.89	54.95	65.78	61.19	65.04
	precision	47.28	56.49	68.36	63.28	67.38
	recall	55.95	62.51	67.82	64.49	68.19
	F1	51.23	59.33	68.08	63.86	67.78
yeast	hammingloss	27.31	21.96	21.56	22.89	21.94
	accuracy	40.23	49.49	51.18	47.02	50.23
	precision	56.18	64.57	65.56	68.35	57.65
	recall	55.02	59.56	61.46	54.66	68.07
	F1	55.56	61.95	63.44	60.74	62.43

Table 3. Experimental Results For SVM

Data Set	Metric	SVM-sl	TSVM-BR	TSVM-C	Tri-BR	Tri-C	YATSI-BR	YATSI-C
emotions	hammingloss	30.00	31.11	29.44	28.42	27.54	21.34	27.11
	accuracy	38.61	39.44	41.11	40.62	42.54	47.91	41.81
	precision	48.61	46.39	51.11	51.77	53.18	63.85	59.35
	recall	52.78	56.67	53.89	54.07	53.50	54.64	54.01
	F1	50.61	51.02	52.46	52.80	53.28	58.89	56.55
scene	hammingloss	12.21	12.75	12.84	14.03	13.13	15.95	14.92
	accuracy	52.95	54.92	57.07	50.66	58.54	52.80	56.53
	precision	54.86	56.75	59.09	52.35	60.91	54.43	58.10
	recall	60.91	65.72	65.78	60.20	62.99	65.13	69.13
	F1	57.66	60.85	62.26	55.99	61.92	59.30	63.14
yeast	hammingloss	26.45	22.99	21.67	21.80	22.41	25.19	20.33
	accuracy	41.70	44.86	47.21	45.98	49.16	46.68	50.11
	precision	56.19	62.34	65.16	68.40	64.08	58.16	70.15
	recall	56.32	56.20	57.98	54.60	58.90	65.45	57.94
	F1	56.24	59.08	61.34	60.67	61.36	61.59	63.46

In the experiments, we use Information Gain feature selection algorithm to learn the dependency among labels. Table 2 and Table 3 gives the experiment result when C4.5 and SVM is used as supervised classifier (base classifier) respectively. Comparing C4.5-sl with Tri-C and YATSI-C in table 2, and SVM-sl with Tri-C and YATSI-C in table 3, it is shown that unlabeled data helps to improve the performance of classifiers. Comparing Tri-BR with Tri-C, YATSI-BR with YATSI-C in both table 2 and table3, and TSVM-BR with

TSVM-C in table 3, it is shown that our algorithm has better performance, thus the dependence information could helps to improve the classification performance of classifiers.

5. Conclusion and Future Work

In this paper, we proposed a semi-supervised multi-label learning algorithm from a new perspective. In our algorithm, we determine the classification order for labels, and then a list of classifiers is trained according to the classification order, with each classifier in the list is trained by using the outputs of previous classifiers in the list as additional features. Compared with the algorithm which does not use unlabeled data, and the algorithm which ignore the dependency. The experiment result shows that our algorithm has substantial advantage over the other two algorithms. In our future work, we plan to adopt hierarchy method to learn from the domains with large number of labels.

6. Acknowledgements

This research is supported by the National Natural Science Foundation of China (60873196) and Young Cadreman Supporting Program of Northwest A&F University (01140301).

7. References

[1] J. Zhu. Semi-supervised Learning Literature Survey. *Computer Sciences Technical Report* TR 1530, University of Wisconsin-Madison, 2005.

[2] Y. Liu, R. Jin, L. Yang. Semi-supervised Multi-label Learning by Constrained Non-Negative Matrix Factorization. In: *AAAI*, 2006.

[3] Z. Zha, T. Mie, Z. Wang, X. Hua. Graph-Based Semi-Supervised Learning with Multi-label. In *ICME*. page 1321-1324, 2008.

[4] G. Chen, Y. Song, C. Zhang. Semi-supervised Multi-label Learning by Solving a Sylvester Equation. In *SDM*, 2008.

[5] J. Wang, Y. Zhao, X. Wu, X, Hua. Transductive Multi-label Learning for Video Concept Detection. In: *Proceeding of the 1st ACM international conference on Multimedia information retrieval*, 2008.

[6] K. Brinker, J. Furnkranz, E. Hullermeier. A unified model for multilabel classification and ranking. In: *Proceedings of the 17th European Conference on Artificial Intelligence*, Riva del Garda, Italy, 489-493, 2006.

[7] E. Schapire, Y. Singer. Boostexter. A boosting-based system for text categorization. *Machine Learning*. 135-168, 2000.

[8] S. Godbole, S. Sarawagi. Discriminative methods for multi-labeled classification. In: *Proceedings of the 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 22-30, 2004.

[9] S. Gao, W. Wu, C. Lee, T. Chua. A maximal figure-of-merit approach to text categorization. In: *SIGIR'03*, 174-181, 2003.

[10] G. Tsoumakas, I. Katakis. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining*, 3(3):1-13, 2007.

[11] Z. Zhou, M. Li. Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Transactions on Knowledge and Data Engineering*, 17(11): 1529-1541, 2005.

[12] K. Driessens, P. Reutemann, B. Pfahringer, C. Leschi. Using weighted nearest neighbor to benefit from unlabeled data. In: *Proceedings of the 10th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp.60-69, 2006.

[13] V. Sindhwani, S. Keerthi. Large Scale Semi-supervised Linear SVMs. In: *Proceedings of the 29th annual international ACM SIGIR conference on research and development in information retrieval*, 2006.