The 2010 ACM International Conference on Bioinformatics and Computational Biology

ACM-BCB 2010

Niagara Falls, New York, U.S.A.
August 2 - 4, 2010

Editors:
Li Liao, Guozheng Li, Aidong Zhang, Mark Borodovsky
Gultekin Ozsoyoglu, Armin R. Mikler

Association for Computing Machinery
Advancing Computing as a Science & Profession

Department of Computer Science and Engineering
University at Buffalo The State University of New York

National Science Foundation
WHERE DISCOVERIES BEGIN

ACM ISBN: 978-1-4503-0192-3

Feature Selection for Semi-Supervised Multi-Label Learning with Application to Gene Function Analysis

Guo-Zheng Li
The MOE Key Laboratory of Embedded System and Service Computing
Tongji University
Shanghai 201804, China
gzli@tongji.edu.cn

Mingyu You
Department of Control Science & Engineering
Tongji University
Shanghai 201804, China
myyou@tongji.edu.cn

Lei Ge
School of Computer Science & Engineering
Shanghai University
Shanghai 200072, China
grame@shu.edu.cn

Jack Y Yang
Purdue Univ ECE, West Lafayette, IN 47907 and Indiana Univ Bioinformatics Center, Indianapolis, IN 46202 USA
Dr.Yang@JHU.edu

Mary Qu Yang
Purdue Univ ECE, West Lafayette, IN 47907 and Indiana Univ Bioinformatics Center, Indianapolis, IN 46202 USA
yangma@mail.nih.gov

ABSTRACT
This paper investigates gene function annotation of Yeast by using semi-supervised multi-label learning. Multi-label learning has been a hot topic in the bioinformatics field, but there are many samples unlabeled. Semi-supervised learning may be employed to utilize the unlabeled data. This paper proposes a novel semi-supervised multi-label learning algorithm COMN by combining Co-Training with ML-kNN to utilize the unlabeled yeast gene data to improve modeling accuracy of function annotation. Furthermore, an embedded feature selection algorithm PRECOMN is proposed to perform feature selection for COMN to remove the irrelevant and redundant features. Experimental results on one benchmark data set of Yeast show COMN and PRECOMN perform better than the original multi-label learning algorithm ML-kNN. Furthermore PRECOMN improves generalization performance of COMN.

Categories and Subject Descriptors
I.5.2 [**Design Methodology**]: Feature Evaluation and Selection; I.2.6 [**Artificial Intelligence**]: Learning; J.3 [**Life and Medical Science**]: Biology and Genetics

General Terms
Bioinformatics, Machine Learning, Algorithms

Keywords
Semi-Supervised Learning, Multi-Label Learning, Feature Selection, Gene Function Analysis

1. INTRODUCTION
Multi-label learning (MLL) studies how to analyze data sets with multi-labels in one sample, which is still a challenge problem in the bioinformatics field [1]. Multi-label problems have been existing widely, e.g. in the Yeast data set the gene YAL062w belongs to several different function classes like metabolism, transcription, and protein synthesis [4]. In all multi-label problems, the instances in the training data set have relation with many labels, but which are unknown for the test cases.

There are two types of tasks in supervised MLL, i.e. multi-label classification and label ranking. In multi-label classification, we need to learn from training samples to produce a model and output a collection of labels with respect to the samples in the test set. Many scholars contribute to this topic and develop a lot of algorithms [11], where ML-kNN as an adaptation algorithm of MLL based on k nearest neighbor (kNN) has achieved satisfied results. ML-kNN uses the maximum a posteriori principle in order to determine the label set of the test instance, based on prior and posterior probabilities for the frequency of each label within the k nearest neighbors [12, 11].

As the size of data set increases, there are a lot of samples without labels due to the cost, which are useless in supervised learning. To utilize the unlabeled samples, semi-supervised learning (SSL) is becoming a hot topic. More and more SSL algorithms are proposed [14], of which Co-Training series algorithms are good choices [2, 13, 8]. Here we propose a novel semi-supervised multi-label learning algorithm by combining Co-training with ML-kNN to solve the novel application of gene function analysis.

In this paper, we improve MLL in two levels, one is combining SSL to utilize the unlabeled data, the other is removing irrelevant and redundant features. The rest is arranged as follows, Section 2 presents two novel algorithms, the semi-supervised multi-label learning algorithm COMN, and PRECOMN with feature selection; Section 3 introduces the used data set Yeast and multi-label measure; Results are presented in Section 4 and conclusion in Section 5.

2. COMPUTATIONAL METHODS

2.1 COMN – a semi-supervised multi-lable learning algorithm

Multi-label learning (MLL) studies how to model the instances with multi-labels, whose challenge exists in the cross relationship among the different labels. There are multi-label problems in the bioinformatics field, e.g. gene function annotation [4]. A lot of MLL algorithms are developed in recent years, of which one type is algorithm adaptation, e.g. multi-label text categorization, multi-label decision tree, multi-label kernel, multi-label neural networks, multi-label k nearest neighbor (ML-kNN) and multi-label ensemble. ML-kNN proposed by Zhang and Zhou [12, 11] is based on the prior and posterior probabilities for the frequency of each label within the k nearest neighbors, and determines the labels of test instances by posterior principle, which has obtained satisfied performance. Without loss of generality, ML-kNN is employed as the baseline MLL algorithm in this paper.

Semi-supervised learning (SSL) techniques utilize the unlabeled samples to help improve generalization performance of base learners [14], of which Co-Training series algorithms are state-of-arts [2, 13, 8]. Co-Training was proposed by Blum and Mitchell [2], which supposes there are two independent and redundant views or feature sets in the data set, this is very strict. CoReg is proposed by using a pair of heterogynous learners with different parameters, i.e. a pair of kNNs with different distance metrics [13]. Based on Co-Reg, FESCOT is proposed to solve classification with feature selection for modeling of quantitive structure activity relationship [8]. FESCOT uses a pair of heterogynous learners with different parameters, i.e. a pair of kNNs with different distance metrics. kNN is used as base learners. There are two reasons, one is CoReg is a loop, it needs to repeat training the learners, its computational complexity is high. If kNN is used, training is ignored. The other is that CoReg needs to estimate the confidence interval, where kNN is efficient to do. Here ML-kNN is used as base learners of MLL for SSL which inherits the advantages of kNN.

In this paper, we fuse both state-of-arts algorithms of ML-kNN [12] and FESCOT [8] to constitute a novel algorithm named COMN (Co-Training ML-kNN). As shown in Algorithm 1, COMN inherits the idea of FESCOT and CoReg [13], which is trained on the same data set by using a pair of ML-kNN [12] classifiers with two different sets of parameters. Both classifiers label the unlabeled instances and fertilize the training data set for each other. The final prediction results are determined by fusing both classifiers. COMN adapts the previous algorithms as follows:

- In COMN, Δu is defined as

  $$\Delta u = \text{hloss}_{N(x_u)}(h) - \text{hloss}_{N(x_u)}(h')$$

  where $N(x_u)$ represents that the set of k instances near the unlabeled instance x_u, h means the original classifier, h' means the classifier trained on the new training data set with the newly labeled instances which are originally unlabeled. $\hat{Y}_u$ are prediction results produced from the original classifier. hloss Mean the measure function of hamming loss as in Section 3.

- Labels of an new instance x are determined by fusing both final ML-kNN learners, output of COMN is changed to be:

  $$\vec{r}_u(l) = \vec{r}_u^{\,1}(l) + \vec{r}_u^{\,2}(l), l \in Y$$

  $$\vec{y}_u(l) = 1, \text{when } \ \vec{r}_u(l) > 1; \vec{y}_u(l) = 0, \text{ otherwise}$$

  where $\vec{r}_u^{\,1}(l)$ and $\vec{r}_u^{\,2}(l)$ represent the evaluation values produced from the pair of ML-kNN classifiers respectively. $\vec{r}_u(l)$ means the final evaluation value produced by the whole semi-supervised learner, COMN here. $\vec{y}_u(l)$ means the relation between the instance l with the set of instances y_u, it is 1, when $l \in y_u$; otherwise 0.

In Algorithm 1, please refer to ML-kNN [12] for the calculation method of $P(H)$ and $P(E|H)$.

2.2 PRECOMN – feature selection for COMN

In data sets, there are irrelevant and/or redundant features, which hurt the prediction performance. Feature selection is needed in the learning process. Supervised feature selection utilizes the labels to improve prediction performance, while in semi-supervised learning, there are some instances without labels. Feature selection for SSL is still a challenge, only in FESCOT, embedded feature selection is proposed for SSL [8].

Feature selection meets the challenge from multi-labels, few works have been done. Researchers transform the multi-label problems to single label ones, then perform feature selection [3, 10]. MEFS is proposed by using the embedded feature selection model and has obtained better performance that other feature extraction methods like PCA, LSI and MDDM on Yahoo web pages data sets [5].

We continue the embedded model employed in FESCOT [8] and MEFS [5] and propose a novel algorithm PRECOMN to perform feature selection for COMN. PRECOMN is the abbreviation of Prediction Risk based Embedded feature selection for COMN where the sequential backward search algorithm is employed to search feature subsets and the prediction risk criterion [9] is to evaluate feature subsets. Prediction risk has been used in other learners like neural networks [9], support vector machines [7], ensemble learning [6], semi-supervised learning [8] and multi-label learning [5] and obtained satisfied results. Here it is defined as:

$$S_i = \text{avgprec}(x) - \text{avgprec}(x_i) \qquad (1)$$

where avgprec is the computational function of Average precision as in Section 3, and avgprec(x_i) means the Average precision value on the training data set with the ith feature being replaced by its average value.

Suppose n is the original number of features, d is the target dimension, $D = (X, Y) = \{L \cup U\}$ including the feature set X and the label set Y represents the data set fusing the labeled data set L and unlabeled U, length(u) is the feature

Algorithm 1 The COMN Algorithm

```
Input:   Labeled data $L$
         Unlabeled Data $U$,
         Test instance $t$,
         Number of nearest neighbor $k_1$ and $k_2$,
         Maximum number of iterations $T$,
         Parameter of distance metric $d_1$ and $d_2$,
         Smooth parameters $s_1$ and $s_2$
Output:  Label vector $\vec{y_t}$,
         Ranking labels $\vec{r_t}$

1:  Begin
    % Train the learner on $L$ and $U$
2:  $L_1 \leftarrow L$; $L_2 \leftarrow L$
3:  Get $U'$ by randomly choosing from $U$
4:  $h_1 \leftarrow$ ML-kNN$(L_1, k_1, d_1, s_1)$,
    $h_2 \leftarrow$ ML-kNN$(L_2, k_2, d_2, s_2)$
5:  Calculate the prior probability $P_j(H_b^l)$ and posterior
    probability $P_j(E_j^l|H_b^l)(j \in \{1,2\})$ by using ML-kNN
6:  for $i = 1$ to $T$ do
7:    for $j = 1$ to 2 do
8:      for any $x_u \in U'$ do
9:        $\hat{Y}_u \leftarrow h_j(x_u)$
10:       $N(x_u) \leftarrow$ Neighbors$(x_u, L_j, k_j, d_j)$
11:       $h'_j \leftarrow$ ML-kNN$(L_j \cup (x_u, \hat{Y}_u), k_j, d_j, s_j)$ Recalcu-
          late the prior probability $P'_j(H_b^l)$ and posterior
          probability $P'_j(E_j^l|H_b^l)$
12:       $\Delta u = \text{hloss}_{N(x_u)}(h_j) - \text{hloss}_{N(x_u)}(h'_j)$
13:     end for
14:     if $\Delta u > 0$ then
15:       $\tilde{x}_j \leftarrow \arg\max_{x_u \in U'} \Delta u$; $\tilde{Y}_j \leftarrow h_j(\tilde{x}_j)$
16:       $\prod_j \leftarrow \{(\tilde{x}_j, \tilde{Y}_j)\}$; $U' \leftarrow U' - \prod_j$;
17:     else
18:       $\prod_j \leftarrow \Phi$
19:     end if
20:   end for
21:   $L_1 \leftarrow L_1 \cup \prod_2$; $L_2 \leftarrow L_2 \cup \prod_1$
22:   if $L_1$ and $L_2$ are changed then
23:     $h_1 \leftarrow$ ML-kNN$(L_1, k_1, d_1, s_1)$
        $h_2 \leftarrow$ ML-kNN$(L_2, k_2, d_2, s_2)$
24:     Recalculate the prior probability $P_j(H_b^l)$ and pos-
        terior probability $P_j(E_j^l|H_b^l)(j \in \{1,2\})$
25:     Reconstitute $U'$ by randomly choosing from $U$
26:   end if
27: end for
    % Test on the instance $t$
28: for $l \in Y$ do
29:   for $j = 1$ to 2 do
30:     $\vec{C}_t^j(l) = \sum_{a \in N_j(t)} \vec{Y}_a(l)$
31:     $\vec{r}_t^j(l) = \frac{P_j(H_1^l)P(E^l_{\hat{C}_t^j(l)}|H_1^l)}{\sum_{b \in \{0,1\}} P_j(H_b^l)P_j(E^l_{\vec{C}_t^j(l)}|H_b^l)}$;
32:   end for
33:   $\vec{r}_t(l) = \vec{r}_t^1(l) + \vec{r}_t^2(l)$
34:   if $\vec{r}_t(l) > 1$ then
35:     $\vec{y}_t(l) = 1$
36:   else
37:     $\vec{y}_t(l) = 0$
38:   end if
39: end for
40: End
```

number of a feature vector u. PRECOMN is shown in Algorithm 2, whose main idea is to rank the features by using prediction risk, and then to evaluate the feature subsets with different number of top features by using COMN, at last to choose the number of features with the best performance of COMN as the output.

Algorithm 2 The PRECOMN algorithm

```
Input:   Data set $D = \{L \cup U\}$
Output:  Number of selected features $d$,
         Feature subset $D'$

1:  Begin
2:  Initialize the remaining feature list vector $u = [1, ..., n]$,
    the removed feature list vector $r = []$, the evaluation
    result on the validation set $e = []$
3:  Randomly choose 10% to form validation set $D_v$ from
    training set $D$
4:  for length$(u) > 0$ do
5:    $D_t = D(:, u)$, $D_v = D_v(:, u)$
6:    Train COMN on $D_t$ and Validate COMN on $D_v$ ,
      Obtain $e_v = \text{avgprec}_{D_v}$ and update $e = [e_v, e]$
7:    Calculate the prediction risk value $S$ by using equa-
      tion (1) for all features
8:    Find the worst feature $h = \arg\max(S)$
9:    Update the removed feature list $r = [u(h), r]$ and up-
      date $u = u - \{u(h)\}$
10: end for
11: Find the best feature subset $h = \arg\max(e)$ and pro-
    duce the subset by $ub = [r(1:h)]$
12: Obtain the number of the best feature subset $d =$
    length$(ub)$ and produce the best training subset $D' =$
    $D(:, ub)$
13: End
```

3. DATA SETS AND MEASURE

Two proposed novel algorithms COMN and PRECOMN are tested on one benchmark data set of Yeast.

The Yeast data set is in microarray, there are 2147 instances, each one has 103 features. There are 14 labels, the average is 4.24 [4].

The measure of multi-label learning is more complex than single label, five popular measures are used in this paper, i.e. hamming loss, one-error, coverage, ranking loss and average precision [12].

4. RESULTS AND DISCUSSIONS

Two novel algorithms, COMN and PRECOMN are compared with ML-kNN [12] on the real world application of gene function annotation of Yeast. Settings of COMN, PRECOMN and ML-kNN are the same. $k_1 = 10$, $k_2 = 12$, $d_1 = 2$, $d_2 = 5$, $s_1 = s_2 = 1$ and the euclidean distance is used in all the three learners.

On Yeast, 75% of samples are used as the training set, and the rest 25% are test, so there are 1610 samples for training and 537 for test. Then 50% of training set are set as labeled, i.e. 805 samples, while the other 50% are unlabeled. 10% of 805 labeled training samples are randomly chosen as the validation set for PRECOMN, i.e. 81 samples. The split and experiment are repeated 10 times. Experimental results are averaged on the obtained 10 times results. Results on Yeast are listed in Table 1.

Table 1: Statistical results on Yeast by using ML-kNN, COMN, and PRECOMN

Criterion	ML-kNN	COMN	PRECOMN
Hamming loss$\downarrow$	0.212	0.203	0.195
One-error$\downarrow$	0.25	0.242	0.232
Coverage$\downarrow$	6.848	6.563	6.297
Ranking loss$\downarrow$	0.183	0.176	0.169
Average precision$\uparrow$	0.835	0.869	0.894

From Table 1, we can see that: 1) On the four measures of hamming loss, one-error, coverage and ranking loss, the smaller results, the better classifiers. Results of COMN and PRECOMN are all better than those of ML-kNN, while results of PRECOMN are better than those of COMN. So PRECOMN is the best classifier of all. 2) On the average precision measure, the higher, the better of the classifiers. The same phenomena take place on three classifiers, PRECOMN performs absolutely better than COMN and ML-kNN does.

Results are out of our expectation, PRECOMN performs the best of all three classifiers on all five measures and all three data sets. COMN performs better than ML-kNN does on all five measures and all three data sets. The improvements of COMN from ML-kNN and PRECOMN from COMN are apparent. Experimental results indicates that semi-supervised learning may improve the generalization performance of ML-kNN, multi-label learners. Furthermore, irrelevant and redundant features really and greatly hurt performance of semi-supervised multi-label learning algorithms, a great need to remove the irrelevant and redundant features before learning is raised for semi-supervised multi-label learning algorithms.

5. CONCLUSIONS

This paper studies semi-supervised multi-label learning, proposing two novel algorithms COMN and PRECOMN to solve the semi-supervised multi-label data sets with irrelevant and redundant features. Experimental results on one benchmark data sets show COMN works well with semi-supervised multi-label data sets, and PRECOMN further improves its generalization performance when there are irrelevant and redundant features. This proves that semi-supervised learning improves the performance of multi-label learning when there are unlabeled samples. Both semi-supervised learning and multi-label learning algorithms suffer from irrelevant features, feature selection is needed in semi-supervised multi-label learning.

This paper just proves that feature selection for semi-supervised multi-label learning does work, future works are needed to improve this paper. Firstly, introducing more multi-label learning algorithms into semi-supervised learning may boost its performance. Secondly, an efficient heuristic feature subset criterion for semi-supervised multi-label learning is valuable.

6. ACKNOWLEDGMENTS

This work was supported by NSFC (60873129, 30901897), the STCSM ”Innovation Action Plan” Project (07DZ19726), the Shanghai Rising-Star Program (08QA1403200) and the Open Projects Program of National Laboratory of Pattern Recognition.

7. REFERENCES

[1] Z. Barutcuoglu, R. E. Schapire, and O. G. Troyanskaya. Hierarchical multi-label prediction of gene function. *Bioinformatics*, 22(7):830–836, 2006.

[2] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the 1998 Conference on Computational Learning Theory*, pages 92–100. Morgan Kaufmann Publishers, 1998.

[3] W. Chen, J. Yan, B. Zhang, Z. Chen, and Q. Yang. Document transformation for multi-label feature selection in text categorization. In *Proc. 7th IEEE International Conference on Data Mining*, pages 451–456, 2007.

[4] A. Elisseeff and J. Weston. A kernel method for multi-labelled classification. In *Advances in Neural Information Processing Systems 14*, 2002.

[5] L. Ge, G.-Z. Li, and M. You. Embedded feature selection for multi-label learning. *Journal of Nanjing University (in Chinese)*, 45(5):671–676, 2009.

[6] G.-Z. Li and T.-Y. Liu. Feature selection for bagging of support vector machines. In *PRICAI2006, Lecuture Notes in Computer Science 4099*, pages 271–277. Springer, 2006.

[7] G.-Z. Li, J. Yang, G.-P. Liu, and L. Xue. Feature selection for multi-class problems using support vector machines. In *PRICAI2004, Lecture Notes in Artificial Intelligence 3157*, pages 292–300. Springer, 8 2004.

[8] G.-Z. Li, J. Y. Yang, W.-C. Lu, D. Li, and M. Q. Yang. Improving prediction accuracy of drug activities by utilizing unlabeled instances with feature selection. *International Journal of Computational Biology and Drug Design*, 1(1):1–13, 2008.

[9] J. Moody and J. Utans. Principled architecture selection for neural networks: Application to corporate bond rating prediction. In J. E. Moody, S. J. Hanson, and R. P. Lippmann, editors, *Advances in Neural Information Processing Systems*, pages 683–690. Morgan Kaufmann Publishers, Inc., 1992.

[10] K. Trohidis, G. Tsoumakas, G. Kalliris, and I. Vlahavas. Multilabel classification of music into emotions. In *Proc. 9th International Conference on Music Information Retrieval (ISMIR2008)*, 2008.

[11] G. Tsoumakas, I. Katakis, and I. Vlahavas. *Data Mining and Knowledge Discovery Handbook (2nd edition)*, chapter Mining Multi-label Data. Springer, 2009.

[12] M.-L. Zhang and Z.-H. Zhou. Ml-knn: A lazy learning approach to multi-label learning. *Pattern Recognition*, 40(7):2038–2048, 2007.

[13] Z.-H. Zhou and M. Li. Semi-supervised regression with co-training. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI'05)*, pages 908–913, Edinburgh, Scotland, 2005.

[14] X. Zhu. *Semi-Supervised Learning with Graphs*. PhD thesis, Carnegie Mellon University, 2005. CMU-LTI-05-192.