Sparse Representation: Extract Adaptive Neighborhood for Multilabel Classification

Shuo Xiang, Songcan Chen, and Lishan Qiao

Department of Computer Science and Engineering, Nanjing University of Aeronautics and Astronautics, China

Abstract. Unlike traditional classification tasks, multilabel classification allows a sample to associate with more than one label. This generalization naturally arises the difficulty in classification. Similar to the single label classification task, neighborhood-based algorithms relying on the nearest neighbor have attracted lots of attention and some of them show positive results. In this paper, we propose an Adaptive Neighborhood algorithm for multilabel classification. Constructing an adaptive neighborhood is challenging because specified information about the neighborhood, e.g. similarity measurement, should be determined automatically during construction rather than provided by the user beforehand. Few literature has covered this topic and we address this difficulty by solving an optimization problem based on the theory of sparse representation. Taking advantage of the extracted adaptive neighborhood, classification can be readily done using weighted sum of labels of training data. Extensive experiments show our proposed method outperforms the state-of-the-art.

1 Introduction

Multilabel classification has been a popular issue in pattern recognition & machine learning and is encountered in a variety of application domains. For instance, in biology, a gene or protein may posse several functionalities and in natural scene classification, a picture of the beach may also include boats, trees and even a city as its contents. Behind these appearances lies the fact that one object is allowed to associate with more than one labels. Solving classification tasks of multilabel scenario is naturally a generalization of traditional task and posesses much more practical value as well as difficulties.

Several methods taking advantage of traditional classification algorithm, e.g. AdaBoost, SVM, EM, have been proposed to solve this problem. Recent research [1,2] shows that neighborhood-based algorithms relying on the nearest neighbor can achieve good results in multilabel classification task, just like in case of single label. However, the way of choosing neighborhood in these works is based on *K Nearest Neighbor(KNN)*, in which several parameters should be given in advance such as the similarity measurement and the size of the neighborhood K. Constructing an adaptive neighborhood that can get rid of these specifications would be helpful but challenging. In this paper, we address the difficulty by extracting this adaptive neighborhood with an optimization problem based on the theory of sparse representation and further use it for multilabel classification. To our best knowledge, we are not aware of any similar work using this technique to handle the multilabel classficiation problem.

B.-T. Zhang and M.A. Orgun (Eds.): PRICAI 2010, LNAI 6230, pp. 304–314, 2010.

The rest of the paper is organized as follows. In the next section we give a brief review of previous work on the topic of multilabel classification and sparse representation. Then we present our *Adaptive Neighborhood(AN)* algorithm and report the experimental results. Finally we conclude this paper and point out some promising work in the future.

2 Related Work

2.1 Multilabel Classification

Multilabel classification began to be widely concerned due to the work of Schapire and Singer [3]. They presented a boosting-based system BoosTexter for text categorization and also provided several useful measurements that can be extended to other multilabel classification tasks. Besides, they pointed out that controlling the complexity of the overall learning system is an important research issue. To control the this complexity while having a small empirical error, Elisseef and Weston proposed the RankSVM [4] method. As in Support Vector Machine(SVM), a linear model is defined so as to minimize the empirical error measured by the ranking loss and control the complexity of the resulting model simultaneously.

Zhang and Zhou introduced a lazy way of multilabel classification named *ML-KNN* [1]. In their algorithm, *K Nearest Neighbor* in the training set is first computed for an unseen sample, then a *Maximum A Posteriori(MAP)* method is taken to perform the classification, based on the statistical information gained from the label sets of neighbor instances. Motivated by this lazy way method, Cheng and Hullermeier gave *IBLR-ML* algorithm [2] which combines the instance-based learning and logistic regression and allows one to capture the interdependencies between the class labels in a proper way. Experiments on public data sets show that, among several existing multilabel classification algorithm, both *ML-KNN* and *IBLR-ML* show not only positive results but also achieve the state-of-the-art classification performance. However, both of these methods are based on the *KNN* which can easily falls into the predicament of suitable similarity measurements and the size of the neighborhood.

2.2 Sparse Representation

Theory of Sparse Representation is closely related to our work. It has been quite popular in machine learning area, including face recognition [5], dimensionality reduction [6], image super-resolution [7] and image denoising [8]. Sparse solution of underdetermined systems of linear equations lies at the heart of this theory. As stated in [9], finding such solution can be formulated as the following optimization problem(P_0):

$$\begin{aligned} \min_{w} \quad & ||w||_0 \\ s.t \quad & z = Xw \end{aligned} \tag{1}$$

Unfortunately, although l_0-norm is a straightforward measurement of sparsity, problem P_0 has been proved to be NP-hard [10]. To overcome this prohibitive computation issue, a compromising way is to deal with P_1 instead:

$$\begin{aligned} \min_{w} \quad & ||w||_1 \\ s.t \quad & z = Xw \end{aligned} \tag{2}$$

which is a convex optimization and can be readily solved by linear programming [11]. P_1 is the central focus of sparse representation and has been shown to have exactly the same solution as P_0 when the solution is very sparse. [9]

Sparse representation has been involved in many classification tasks, one of which belongs to Wright's work [5] on robust face recognition. According to their paper, samples from the same class are modeled as lying on a linear subspace. Given sufficient training samples of the *ith* class, $X_i = [x_{i,1}, \cdots, x_{i,n_i}] \in R^{d\times n_i}$, any test sample $z \in R^d$ from the same class would be able to be approximately written as the linear combination of training samples associated with the *ith* class:

$$z = w_1 x_{i,1} + \cdots + w_{n_i} x_{i,n_i} = X_i w$$

Following the idea above, for any unseen sample, finding a sparse representation in all the training samples would typically yield the solution with nonnegative entries associated with training examples of the same class, as shown in the following results, from which we can see that sparse representation is able to capture the discriminant nature behind the samples:

$$z = Xw = [X_1, \cdots, X_c][0, \cdots, 0, w_1, \cdots, w_{n_i}, 0, \cdots, 0]^T$$

The sparse representation can be obtained by solving P_1. In realistic tasks, the exact representation of test sample may not be able to achieved due to noise. Usually a stable version is considered instead:

$$\begin{aligned} \min_{w} \quad & ||w||_1 \\ s.t \quad & ||z - Xw||_2 < \epsilon \end{aligned} \tag{3}$$

where ϵ is an error tolerance. This is an convex programming and can be efficiently solved. With the obtained representation, prediction of a test sample is able to be made by choosing the class with least residual. The algorithm achieve positive results on several public data sets with high accuracy and robustness to occlusion.

3 Adaptive Neighbor(AN) Algorithm

3.1 Problem Setting

Consider the following multilabel classification with:

training set: $Tr = \{(x_i, Y_i)\}_{i=1}^{n}, (x_i \in \mathscr{X}, y_i \in \mathscr{Y})$
test set: $Te = \{z_i\}_{i=1}^{m}, (z_i \in \mathscr{X})$
Our goal is to learn a classifier:

$$f : \mathscr{X} \times \mathscr{Y} \mapsto \mathscr{R}$$

which tends to assign higher value to (z, y_i) if y_i belongs to Y_z. From f we can easily predict the label of an unseen sample, e.g. $\mathit{predict}(z, y_i) = [\![f(x, y_i) \geq \theta]\!]$, θ is a threshold. Another statistic we would like to gain is the rank information between different labels, the function $rank_f(z, y_i)$ ranks different labels according to the corresponding value of $f(z, y_i)$, where higher value of f gets lower(better) rank position.

3.2 Our Method

Extensively applied in different machine learning tasks, ranging from single label classification to dimensionality reduction [12,13] and multilabel classification [1], *KNN* usually serves as an intermediate step to seek the connections between samples. However, neighborhood information gained from *KNN*, largely based on the choice of similarity measurement and the size of the neighborhood, presents a simple but limited portrait of the correlations between samples.

In order to capture the discriminant nature behind the data, our work focuses on designing an effective construction of an adaptive neighborhood on which multilabel classification task can be efficiently carried out. By adaptive, we mean, this neighborhood is determined by the natural structure behind the data and we don't have to prescribe the parameter like the number of neighbors K or a specific way of similarity measurement. Motivated by sparse representation in face recognition [5], we summarize this procedure in a similar optimization problem(P_{AN}):

$$\begin{aligned} \min_{w} \quad & ||z - Xw||_2^2 + \lambda||w||_1 \\ s.t \quad & w \geq 0 \end{aligned} \tag{4}$$

X is a d by n matrix whose columns contain the training data of dimension d. z is a single test sample and our goal is to seek the sparsest coefficient w while keeping the residual as small as possible. This formulation is able to capture exactly the same kind discriminant nature as sparse representation stated in the previous section. However, our method still differs from sparse representation in the objective function and the constraint as follows:

- Different from sparse representation which aims at finding a sparse solution with best reconstruction results, our method concerns more to find out the information of neighborhood in which the nonnegativity is necessary.
- The nonnegativity constraint can provide us a straightforward interpretation of the relation between the test sample and the training sample, where larger value of w_i means that the ith training sample is "more similar" to the test sample z and vice versa.

Based on the facts above, we claim that an adaptive neighborhood for each test sample is obtained by noticing that we don't need to prescribe any concrete way of similarity measurement between samples or the size of the neighborhood. Unlike sparse representation's choosing class with least residual in classification [5], we design the classifier in a simpler weighted sum way: for a label $l \in L$ and a given test sample z, $f(z, y_l) = \sum_j w_j * Y_{lj}$, Y contains the true label of training data, each in a column. Algorithm 1 shows the the complete description.

3.3 Comparison with Previous Work

Compared to previous the state-of-the-art works like *ML-KNN* and *IBLR-ML*, several remarkable differences should be emphasized for our method which makes multilabel classification done effectively and efficiently.

First, the neighbors chosen by our algorithm is generally different from that of *KNN*. Inherited from sparse representation, *AN* tends to select those neighbors that share the same underlying subspace, as can be seen from Figure 1.

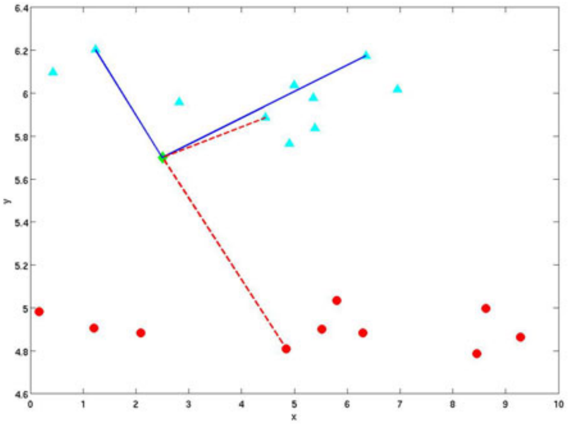

Fig. 1. Data from two affine subspace($y = 5.0$,$y = 6.0$) with gaussian noise added. The solid line shows the neighbors selected by *AN* and the dashed line gives that of *2-Nearest Neighbors(2NN)*. *2NN* selects neighbors with least distances while the neighbors chosen by *AN* automatically(with the size of two coincidentally) tend to lie on the same subspace which are much more discriminative [5].

Second, we don't need to prescribe the size of the neighborhood K as in *ML-KNN* and *IBLR-ML*. K is set to the number of nonnegative elements in w which is naturally obtained from the above optimization. Although we can still fix the value of K by choosing the K largest elements of w, it is advisable that different samples would belong to the different neighborhoods which have different sizes.

In addition, due to the natural discriminant property of sparsity, no further complicated classifier is required, a simple weighted sum would suffice. This makes the classification procedure more efficient.

4 Experiments

In this section, experiments are conducted on public multilabel classification data sets, which serve both to demonstrate the efficacy of the proposed method and to validate the claim we have made in the previous sections. We compare our results with the state-of-the-art, including *ML-KNN* and *IBLR-ML*, of which the implementations are provided

Algorithm 1. Adaptive Neighborhood

```
Input:
  $X$: training data
  $Y$: training label set
  $z$: test sample
  $\theta$: threshold, $\lambda$: regularizer
Output:
  $f$: classifier
  $predict$: predicting function
Procedure:
for all test sample $z$ do
  Solve the optimization Problem:
      $\min_{w} \quad ||z - Xw||_2^2 + \lambda||w||_1 \quad s.t. \quad w \geq 0$
  Normalize $w$
  $f(z, \cdot) = Yw$
  for $j = 1$ to $|L|$ do
    if $f(z, y_j) \geq \theta$ then
      $predict(z, y_j) = 1$
    else
      $predict(z, y_j) = -1$
    end if
  end for
end for
```

by their original authors. Our algorithm can be efficiently implemented using the sparse learning package l1_ls [1] or SLEP [14].

4.1 Measurement

Unlike traditional loss function of single label classification, special criterion should be considered while evaluating the performance of multilabel task. Here we utilize the measurements that provided in [3].

- Hamming Loss:

$$hloss(predict, x, Y) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{|L|}|predict(x_i) \oplus Y|$$

- OneError:

$$OneError(f, x, Y) = \frac{1}{n}\sum_{i=1}^{n}[\![arg \max_{y} f(x_i, y) \notin Y_i]\!]$$

[1] http://www.stanford.edu/ boyd/l1_ls/

- Coverage:

$$Coverage(f, x, Y) = \frac{1}{n}\sum_{i=1}^{n} \max_{y \in Y_i} rank_f(x_i, y) - 1$$

- Ranking Loss:

$$rloss(f, x, Y) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{|Y_i||\overline{Y}_i|} \times$$
$$|\{(y_1, y_2)|f(x_i, y_1) \le f(x_i, y_2), (y_1, y_2) \in Y_i \times \overline{Y}_i\}|$$

- Average Precision:

$$AvePrec(f, x, Y) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{Y_i} \times$$
$$\sum_{y \in Y_i} \frac{|\{y'|rank_f(x_i, y') \le rank_f(x_i, y), y' \in Y_i\}|}{rank_f(x_i, y)}$$

The operator $\oplus$ in *Hamming Loss* means symmetric difference which measures the number of labels that we have misclassified during the test phase. In *One Error*, function $[\![\cdot]\!]$ takes 1 if the parameter it takes holds true and the whole statistic calculates the times the label we classified with most confidence is actually incorrect. *Coverage* measures how far we need to go down the label list to cover all the positive label and *Ranking Loss* provides the average fraction of pairs that are not correctly ordered. Similar to the concept of precision in Information Retrieval, *Average Precision* gives the mean precision on every label.

Table 1. Statistics of the data sets used in the experiments

DATA SET	INSTANCE	ATTRIBUTE	LABEL
genbase	662	1186	27
medical	978	1449	45
enron	1702	1001	53
bibtex	7358	1836	159

4.2 Data Sets

Four data sets[1]: *genbase*, *medical*, *enron* and *bibtex* are chosen for our experiments. Data set *genbase* is derived from the task of protein classification [15], where each protein can associate with at most 27 labels. The data set contain 662 instances of 1185 dimensions. 978 instances of dimension 1449 each with 45 labels are contained in the data set of *medical*. It comes from the international challenge of classifying clinical free text using natural language processing, which aims to create and train computational intelligence algorithms that automate the assignment of *ICD-9-CM* codes to clinical

[1] http://mlkd.csd.auth.gr/multilabel.html

free text. Data set *enron* is derived from the *UC Berkeley Enron Email Analysis Project* and contains Email data from about 150 users, mostly senior management of Enron. After processing, the current data set is comprised of 1702 instances with the dimension of 1001 and 53 labels are involved. The last data set we use is relative large. *bibtex* was used to solve the automated tag suggestion problem [16], containing 7395 instances of 1836 dimension with 159 labels. An overview of all the data is provided in Table 1.

4.3 Parameter Setting

As pointed in the previous section, *K Nearest Neighbors* are involved in both algorithms of *ML-KNN* and *IBLR-ML*. In their experiments, the size of the neighborhood is fixed at 10 by which positive results have been achieved. We also use this value in our experiments for fairness. The regularizer λ in algorithm *AN* should also be carefully chosen. Although various methods have been proposed to deal with this issue, there is currently no reliable way to get the optimal value. Cross validation can be adopted for better performance, however, that would be time-consuming. Therefore we simply fix λ at 1.0 in all our experiments. Actually it will be shown in our experiments that a small change in λ does not affect the performance much.

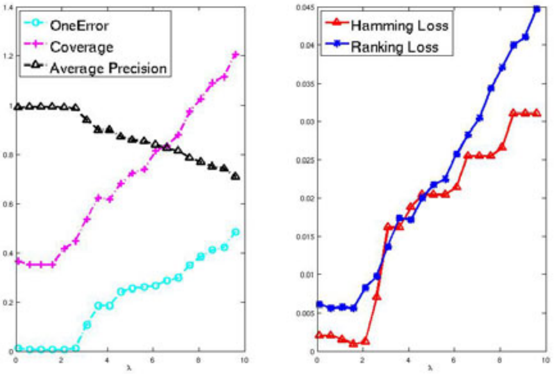

Fig. 2. Indexes' values of *AN* vs. the λ on *genbase*: small λ tends to give better performance which decreases as λ increases, however, a small change at its manually-chosen value(e.g. 1.0 here) does not affect the efficacy much

4.4 Experimental Results and Analysis

First, we test the stability of our *AN* algorithm to the parameter setting of λ by assigning different values of λ in a relative large range on the data sets of *genbase*, as shown in Figure 2. We can see that the a small value of λ tents to give better performance. This can be explained that, as λ increases, the optimization will exert more penalty on the sparsity of the w. A very large λ would typically result in very few number(e.g. only 1) of neighbors which are chosen for further classification, which yields a bad classification results. However, it can also be recognized that, for small value of λ, its

Table 2. Comparative Results on *genbase*: *AN* achieves the best performance on all statistic except for Ranking Loss. *IBLR-ML* gets better performance in all statistic than *ML-KNN*.

Algorithm	AN	ML-KNN	IBLR-ML
hloss ↓	**0.0020**	0.0050	**0.0020**
OneError ↓	**0.0056**	0.0090	0.0070
Coverage ↓	**0.3518**	0.5610	0.4220
rloss ↓	0.0058	0.0060	**0.0040**
AvePrec ↑	**0.9920**	0.9890	0.9900

Table 3. Comparative Results on *medical*: *AN* has the best result but for coverage on which *ML-kNN* gets the best performance. On this data set, *IBLR-ML* was surpassed by *ML-KNN* in all statistics.

Algorithm	AN	ML-KNN	IBLR-ML
hloss ↓	**0.0165**	0.0171	0.0223
OneError ↓	**0.1381**	0.2643	0.3844
Coverage ↓	1.7177	**0.7237**	4.7960
rloss ↓	**0.0253**	0.0425	0.0833
AvePrec ↑	**0.8876**	0.7957	0.7045

Table 4. Comparative Results on *enron*: Except for Hamming Loss, *AN* achieves the best performance. *ML-KNN* outperforms *IBLR-ML* consistently in all statistics.

Algorithm	AN	ML-KNN	IBLR-ML
hloss ↓	0.0540	**0.0520**	0.0572
OneError ↓	**0.3005**	0.3040	0.3834
Coverage ↓	**12.8532**	13.2055	14.9551
rloss ↓	**0.0891**	0.0938	0.1124
AvePrec ↑	**0.6598**	0.6232	0.6020

small change does not affect the performance much. Secondly, we compare our *AN* algorithm with the *ML-KNN* and *IBLR-ML* on the aforementioned measurements. The ↓ beside each measurement means that smaller value yields better performance while ↑ represents the opposite. Table 2 shows the testing results on *genbase*, from which we can see that *AN* algorithm dramatically outperforms the other methods in all statistic except for *Ranking Loss*, on which *IBLR-ML* achieves the best result.

Similarly, Table 3 to Table 5 give the effectiveness of the three algorithms on data sets *medical*, *enron*, *bibtex* respectively. From the experimental results we can see that *IBLR-ML* outperforms *ML-KNN* in data set *genbase* while the opposite results are achieved in data sets *medical* and *enron* and none is guaranteed better than the other. However, although *AN* does not posses the best results in all statistics, it still can be recognized that *AN* dominates the experimental results and outperforms the other two.

Table 5. Comparative Results on *bibtex*: *AN* leads in all statistics and significantly improvement is achieved in Ranking Loss and Coverage

ALGORITHM	AN	ML-KNN	IBLR-ML
HLOSS ↓	**0.0137**	0.0140	0.0189
ONEERROR ↓	**0.4064**	0.5853	0.6294
COVERAGE ↓	**26.6282**	56.2179	48.7797
RLOSS ↓	**0.0896**	0.2173	0.1961
AVEPREC ↑	**0.5378**	0.3449	0.3349

5 Conclusion and Future Work

In this paper, we propose an Adaptive Neighborhood algorithm for multilabel classification. We construct an adaptive neighborhood by an optimization procedure similar to sparse representation but with more interpretability of relation between neighborhood. Based on this automatically-formed neighborhood, classification can be easily carried out. Experiments show our algorithm outperforms the state-of-the-art.

Some issues of this framework should still be ameliorated in the following points which will be our future work:

- The quadratic programming behind the algorithm is time consuming. Solving the optimization more efficiently can be helpful.
- How to take the labels' correlations into account explicitly under the *AN* framework is another issue.
- Exploring other ways to classification under *AN* other than our current weighted sum method is desirable.

Acknowledgments

This work was supported by the National Science Foundation of China under the Grant No.60973097 and Higher Education Doctoral Foundation under the grant No.200802870003 respectively. We thank Zhang and Cheng very much for providing the codes of *ML-KNN* and *IBLR-ML*.

References

1. Zhang, M., Zhou, Z.: Ml-knn: A lazy learning approach to multilabel learning. Pattern Recognition 40(7), 2038–2048 (2007)
2. Cheng, W., Hullermeier, E.: Combining instance-based learning and logistic regression for multilabel classification. Machine Learning 76(2/3), 211–225 (2009)
3. Schapire, R., Singer, Y.: Boostexter: A boosting-based system for text categorization. Machine Learning 39(2/3), 135–168 (2000)
4. Elisseeff, A., Weston, J.: A kernel method for multi-labelled classification. Advances in Neural Information Processing Systems 14, 681–687 (2002)
5. Wright, J., Yang, A.Y., Ganesh, A., Sastry, S.S., Ma, Y.: Robust face recognition via sparse representation. IEEE Transactions on Pattern Analysis and Machine Intelligence 31(2-3), 210–227 (2009)

6. Qiao, L., Chen, S., Tan, X.: Sparsity preserving projections with applications to face recognition. Pattern Recognition 43(1), 331–341 (2010)
7. Yang, J., Wright, J., Huang, T., Ma, Y.: Image super-resolution as sparse representation of raw image patches. In: Computer Vision and Pattern Recognition, CVPR 2008, pp. 1–8 (June 2008)
8. Elad, M., Aharon, M.: Image denoising via sparse and redundant representations over learned dictionaries. IEEE Transactions on Image Process. 15(12), 3736–3745 (2006)
9. Bruckstein, A.M., Donoho, D.L., Elad, M.: From sparse solutions of systems of equations to sparse modeling of signals and images. SIAM Review 51(1), 34–81 (2009)
10. Natarajan, B.K.: Sparse approximation solutions to linear systems. SIAM J. Comput. 24(2), 227–234 (1995)
11. Chen, S.S., Donoho, D.L., Saunders, M.A.: Atomic decomposition by basis pursuit. SIAM Review 43(1), 129–159 (2001)
12. Roweis, S., Saul, L.: Nonlinear dimensionality reduction by locally linear embedding. Science 290(22), 2323–2326 (2000)
13. Tenenbaum, J., Silva, V., Langford, J.: A global geometric framework for nonlinear dimensionality reduction. Science 290(22), 2319–2322 (2000)
14. Liu, J., Ji, S., Ye, J.: SLEP: Sparse Learning with Efficient Projections. Arizona State University (2009)
15. Diplaris, S., Tsoumakas, G., Mitkas, P., Vlahavas, I.: Protein classification with multiple algorithms. In: Bozanis, P., Houstis, E.N. (eds.) PCI 2005. LNCS, vol. 3746, pp. 448–456. Springer, Heidelberg (2005)
16. Katakis, I., Tsoumakas, G., Vlahavas, I.: Multilabel text classification for automated tag suggestion. In: Proceedings of the ECML/PKDD 2008 Discovery Challenge (2008)