A Similarity-based Framework for Classification Task

Zhongchen Ma, and Songcan Chen

Abstract—Similarity-based method gives rise to a new class of methods for multi-label learning and also achieves promising performance. In this paper, we generalize this method, resulting in a new framework for classification task. Specifically, we unite similarity-based learning and generalized linear models to achieve the best of both worlds. This allows us to capture interdependencies between classes and prevent from impairing performance of noisy classes. Each learned parameter of the model can reveal the contribution of one class to another, providing interpretability to some extent. Experiment results show the effectiveness of the proposed approach on multi-class and multi-label datasets.

Index Terms—Similarity-based Learning, multi-class, multi-label, class interdependencies, interpretability.

1 INTRODUCTION

Multi-class classification (MCC) and multi-label classification (MLC) are two typical paradigms in machine learning and can be widely applicable in such fields as computer vision, text classification, medical image, information retrieval, etc. Traditionally, MCC is the problem of classifying instances into one of three or more classes, while MLC is the problem of allowing instances to belong to several classes (labels) simultaneously. Therefore, MCC can be cast into MLC by restricting each instance to have only one label.

Distance-based classifiers, which refer to a class of classifiers basing its decision on the distance calculated from the target instance to the training instances, are simple but effective methods for MCC. Based on what type of distance information is used, such methods can be further categorized as instance-information-based and class-information-based classifiers. The instance-information-based classifiers classify an instance based on the distance information in the feature space, while class-information-based classifiers classify an instance based on the distance information in the space induced by classes. In particular, $k$-nearest neighbor (KNN) classifier [1] and the nearest class mean (NCM) classifier [2] are representative instance-information-based and class-information-based methods, respectively. Specifically, KNN assumes that similar instances in the feature space very likely belong to the same class, thus classifying an instance by a majority vote of its $k$-nearest neighbors, and in another way, NCM represents a class by the mean feature vector of its instances, thus assigning an instance to the class with the closest mean.

As a generalization of MCC, MLC inevitably brings more difficulties to learning. To address these challenges of MLC, many methods have been derived from the above distance-based classifiers [3]. For example, multi-label $k$-nearest neighbor (MLKNN) [4] and instance-based learning by logistic regression (IBLR) [5] are two effective KNN-based multi-label methods. Based on the type of distance information used, both methods can be classified as instance-information-based classifiers.

Compared to class-information-based classifiers, instance-information-based classifiers more likely misclassify an instance in the class boundary, because the assumption of such a type of classifiers that similar instances in the feature space very likely belong to the same class, is more likely violated in that case. However, contrary to extensively used instance-information-based classifiers for MCC and MLC, there are currently few class-information-based classifiers for both tasks. Recently, a similarity-based (analogous to distance-based) multi-label learning (SML) classifier [6] was proposed, which represents a label by all its instances and assigns an instance to the labels with large similarities, thus belonging to class-information-based classifiers. It forms a new class of methods for MLC with promising performance [6]. However, we witness: 1) SML is a lazy learning method and often deals much worse with noise in the training data than inductive learning methods (e.g., generalized linear models) [7]; 2) SML does not take correlations and interdependencies between labels into account, its potentials have not yet been fully exploited.

To remedy the above problems, we try to construct a learning framework, which is able to: 1) achieve the best of SML and generalized linear models, 2) capture interdependencies between classes. Specifically, this framework unifies lazy SML and inductive generalized linear models to achieve the best of both worlds, as opposed to IBLR [5] that unifies lazy KNN and inductive logistic regression model. A key idea of our approach is to consider the similarities between the query $x_0$ and the instance sets of different classes as the input features to the generalized linear models, and then to optimize this model on training data to deduce an inductive
optimal classifier. Intuitively, the so-learned parameters of the model can reflect the contribution of one class to another class, providing interpretability to some extent. Moreover, by using an \(\ell_1\)-norm penalty to sparsify the solution of the model, the proposed approach is further also able to discover irrelevant classes, this way both preventing from impairing performance of them and providing better interpretability.

The rest of this paper is organized as follows: The problems of MLC and MCC are introduced in a more formal way in Sect. 2, and our novel method is then described in Sect. 3. Section 4 is devoted to experiments on MLC and MCC datasets. The paper ends with a summary and some concluding remarks in Sect. 5.

2 MCC AND MLC

Let \(X \subset \mathbb{R}^d\) denote an instance space and let \(L = \{\lambda_1, \lambda_2, \cdots, \lambda_m\}\) \((m \geq 3)\) be a finite set of class labels. Moreover, we are given a training set \(D = \{(x_i, L_i)\}_{i=1}^N\), where \(N\) is the number of instances and \(x_i \in X\). Based on the label association to the instances, the classification problem can be categorized into MCC and MLC. Formally, for \(\forall i \in \{1, 2, \cdots, N\}\),

1) if \(L_i\) is an element of \(L\), the classification problem is MCC;

2) if \(L_i\) is a subset of \(L\), the classification problem is MLC.

This subset is often called the set of relevant labels, while the \(\forall\) on the label association to the instances, the classification problem is MLC, as stated in the introduction section, it has some promising performance. However, when applied to MLC and MCC, as stated in the introduction section, it has some key shortcomings needing to be addressed. Thus, in this paper, we are especially interested in SML to improve its performance in MLC and MCC tasks.

3 SIMILARITY-BASED LEARNING

3.1 SML classifier

Let us assume that all the given feature vectors are normalized from its original range to the range \([-1, 1]\) and define the subset \(D_k \subseteq D\) of training instances with class \(\lambda_k \in L\) as

\[
D_k = \{(x_i, L_i) \in D | \lambda_k \in L_i\},
\]

then, the weight \(f_k(x_i)\) of class \(\lambda_k\) for an unseen instance \(x_i \in X\) is estimated as:

\[
f_k(x_i) = \sum_{x_j \in D_k} \Phi(x_i, x_j)
\]

where \(\Phi\) is an arbitrary similarity function. Then, all class weights denoted by \(f(x_i)\) are estimated as:

\[
f(x_i) = \begin{bmatrix}
f_1(x_i) \\
\vdots \\
f_m(x_i)
\end{bmatrix} = \begin{bmatrix}
\sum_{x_j \in D_1} \Phi(x_i, x_j) \\
\vdots \\
\sum_{x_j \in D_m} \Phi(x_i, x_j)
\end{bmatrix}
\]

After estimating \(f(x_i) = [f_1(x_i), \cdots, f_m(x_i)]^T \in \mathbb{R}^m\), the class set \(L_i\) of \(x_i\) can be predicted as the classes with \(\xi\) largest values of \(f(x_i)\), where \(\xi\) is a threshold that need to be estimated differently for MCC and MLC. In MCC, for \(\forall i \in \{1, \cdots, N\}\), \(\xi_i\) should be set to 1 without estimating. In MLC, a straightforward way to estimate \(\xi = [\xi_1, \cdots, \xi_N]\) is to transform the original MLC problem into a general MCC problem to predict the label set size. The details are shown below.

Let \(D' = \{x_i, \xi_i\}_{i=1}^N\) denote the newly-formed training dataset of the transformed MCC problem, and \(L' = \{\xi_i\}_{i=1}^N\) forms the transformed label space. Then, the similarity of instances in \(D'\) of the same set size with respect to \(x_i\) is

\[
f'_k(x_i) = \sum_{x_j \in D'_k} \Phi(x_i, x_j)
\]

where \(D'_k \subseteq D'\) is the subset of training instances with class \(k \in L'\). Therefore, the set size of \(x_i\) is predicted by the following decision function:

\[
\xi(x_i) = \arg\max_{k \in Y} \sum_{x_j \in D'_k} \Phi(x_i, x_j)
\]

where \(\xi(x_i)\) is the predicted label set size for \(x_i\). Obviously, \(\xi(x_i)\) is the label set size with maximum similarity.

The essence of SML is to perform transformation of the feature space. From the perspective of [8], SML is actually a data preprocessing method. Unlike the traditional methods such as PCA [9] or LPP [10], which maintain the data structural of itself, SML is a data preprocessing method that contains class information. Although simple, it has proven to be effective [6]. This provides a promising next direction for multi-label data preprocessing.

3.2 Similarity-based framework

As can be seen from the last subsection, SML predicts the class \(\lambda_k \in L\) for unseen instance \(x\) only by using the similarity from instance subset \(D_k \subseteq D\), ignoring the similarities from other subsets \(\{D_i \subseteq D\}_{i \in [m] \setminus k}\), where \([m] \setminus k\) denotes the positive integer set \{1, 2, \cdots, k-1, k+1, \cdots, m\}.

To compensate such a limitation, our idea is to train one classifier \(h_k\) for each class by using all the normalized similarities \(f(x_1), \cdots, f(x_N)\). Specifically, we use all these similarities as the input to the generalized linear models to predict the class \(\lambda_k \in L\) for unseen instance \(x\). The differences of predicting unseen instance \(x_0\) between our approach and SML are graphically shown in Figure 1.

More formally, for the \(k\)-th class \(\lambda_k\), the corresponding classifier is derived from the following generalized linear model

\[
\pi^k_{0} = g^{-1}(\beta^k_0 + \sum_{t=1}^m \beta^k_t f_t(x_0))
\]

where \(\pi^k_{0}\) denotes the expected response for \(x_0\), and \(f_t(x_0)\) denotes the normalized \(f_t(x_0) = \sum_{x \in D_t} \Phi(x_0, x)\), which is the similarity evidence from instance subset \(D_t \subseteq D\) in favor of \(\pi^k(x_0) = 1\). \(g\) is called the link function in the generalized linear model. Ideally, \(\pi^k_{0} = 1\) if class \(\lambda_k\) is relevant for \(x_0\), 0 otherwise.
Our approach classes for preventing from impairing performance of them. \(\lambda\) impose an \(\ell\) practice, some similarity evidences from other classes do \(\lambda\) the contribution degree of other classes to class \(\beta\) (Sparse Similarity-Based multi-label learning by Logistic Regression). It can be solved by means of sparse logistic regression and a useful toolbox is provided by \cite{12}. We refer to the model derived from (9) as SparseSBLR \(\beta\) are then computed by \(\beta\) is relevant for \(\theta\) and our approach. SML predicts the class \(\lambda\) between SML and our approach. SML predicts the class \(\lambda\) only by using the similarity evidence \(f_k\), while our approach predicts the class \(\lambda\) by using all similarity evidences \(\{f_k\}_{k \in [m]}\)

Note that various link functions can be chosen depending on the assumed distribution of \(\pi_0^k\) \cite{11}, in this paper, we use the logit link \(g(a) = \log(a/(1-a))\) for simplicity. This reduces to logistic regression model. Specifically,

\[
\log \left( \frac{\pi_0^k}{1 - \pi_0^k} \right) = \beta_0^k + \sum_{t=1}^{m} \beta_t^k f_t(x_0) \tag{7}
\]

In this case, \(\pi_0^k\) denotes the posterior probability that \(\lambda_k\) is relevant for \(x_0\). An optimal specification of \(\{\beta_0^k\}_{t=0}^m\) can be accomplished by adapting this parameter to the data \(D\) by using the method of maximum likelihood estimation. The corresponding negative log-likelihood function is then given by

\[
\ell(\beta^k) = \sum_{i=1}^{n} \left( -y_i^k \beta^k z_i + \ln(1 + e^{\beta^k z_i}) \right) \tag{8}
\]

where \(y_i^k = 1\) if class \(\lambda_k\) is relevant for \(x_i\), 0 otherwise. \(\beta^k = \begin{bmatrix} \beta_0^k, \beta_1^k, \cdots, \beta_m^k \end{bmatrix}^T\) and \(z_i = [1, f_1(x_i), \cdots, f_m(x_i)]^T\).

The problem (8) can be computed by means of standard methods from logistic regression \cite{12}. Here, we refer to this method as SBLR (Similarity-Based multi-label learning by Logistic Regression).

Obviously, these learned parameters \(\{\beta_t^k\}_{t \in [m]}\) reveal the contribution degree of other classes to class \(\lambda_k\). In practice, some similarity evidences from other classes do not provide useful information to class \(\lambda_k\), even conversely bring noisy information. To overcome this problem, we impose an \(\ell_1\)-norm penalty on SBLR to detect the irrelevant classes for preventing from impairing performance of them. This leads to an objective:

\[
\ell(\beta^k) = \frac{1}{n} \sum_{i=1}^{n} \left( -y_i^k \beta^k z_i + \ln(1 + e^{\beta^k z_i}) \right) + \lambda \|\beta^k\|_1 \tag{9}
\]

We refer to the model derived from (9) as SparseSBLR (Sparse Similarity-Based multi-label learning by Logistic Regression). It can be solved by means of sparse logistic regression and a useful toolbox is provided by \cite{12}.

After learning all the parameters \(\{\beta_t^k\}_{t \in [m]}\) for each class \(\lambda_k\), The posterior probabilities \(\{\pi_0^k\}_{k \in [m]}\) for the query \(x_0\) are then computed by

\[
\pi_0^k = \frac{1}{1 + e^{\beta_0^k x_0}}, \tag{10}
\]

then classify \(x_0\) with applying the decision rule

\[
h(x_0) = \begin{cases} \lambda_k \in L | \pi_0^k \geq 1/2, k \in [m] \end{cases} \tag{11}
\]

for MLC, and the decision rule

\[
h(x_0) = \arg \max_{\lambda_k} \{ \pi_0^k \}_{k \in [m]} \tag{12}
\]

for MCC, respectively.

3.3 Complexity analysis

In this subsection, we detail the computational complexities of the proposed methods. Given \(N\) training instances and a test instance \(x_0\), the time complexity of constructing the similarity evidences for all training instances is \(O(dN^2)\). Then, the time complexity for training SBLR is \(O(\eta Nm^2)\) if we use the gradient descent method, where \(\eta\) is the number of iterations. And training SparseSBLR costs also \(O(\eta Nm^2)\) if using the accelerated proximal gradient method. The time complexity for computing posterior probabilities \(\{\pi_0^k\}_{k \in [m]}\) for \(x_0\) is \(O(m^2)\). In a nutshell, the total time complexities for training SBLR and SparseSBLR are both \(O(\eta Nm^2 + dN^2)\). The time complexities of SBLR and SparseSBLR for predicting \(x_0\) are both \(O(dN + m^2)\).

4 Experiments

Due to the effectiveness of using the sum of similarities from the instances of each class as features has been validated in \cite{6}, in this section, we focus on demonstrating the effectiveness of the proposed methods, i.e., SparseSBLR and SBLR, on both MLC and MCC datasets. The codes of our approaches are publicly available on https://github.com/John986/paperCodes.

4.1 Experiments on MLC datasets

We use five commonly MLC criteria, including Hamming loss, one error, Coverage, Rank loss and Average precision \cite{3}. For each criterion, we use \(\uparrow(\downarrow)\) to denote the larger (smaller) the value, the better the performance. Before presenting the results of our experiments, we detail the datasets, compared methods used for evaluation.

4.1.1 Datasets

We use seven real-world datasets \cite{1} and give an overview of these datasets in Table 1.

1. http://mulan.sourceforge.net/datasets-mlc.html

| Data set  | Domain        | #Instances | #Attributes | #Labels | Cardinality |
|-----------|---------------|------------|-------------|---------|-------------|
| emotions  | music         | 593        | 72          | 6       | 1.869       |
| scene     | image         | 2407       | 294         | 6       | 1.074       |
| yeast     | biology       | 2417       | 103         | 14      | 4.237       |
| birds     | audio         | 645        | 260         | 19      | 1.014       |
| genbase   | biology       | 662        | 1186        | 27      | 1.252       |
| medical   | text          | 978        | 1449        | 45      | 1.245       |
| CAL500    | music         | 502        | 68          | 174     | 26.044      |

TABLE 1

Statistics for the MLC datasets used in the experiments. The symbol #A indicates the numbers of A; cardinality is the average number of labels per instance.
4.1.2 Compared methods

We compare the proposed methods SBLR and SparseSBLR to three MLC methods:

1) BR-SVM [13] works by training a number of independent SVM classifiers, one per class space. Therefore, BR-SVM does not consider dependencies among class spaces in model induction.

2) SML [6], as reviewed in the introduction, gives rise to a new class of methods for MLC and also achieves promising performance. The code of SML is not publicly available. Thus, we implement this algorithm by ourselves and the code is publicly available on https://github.com/John986/paperCodes2.

3) MLKNN [3] learns a single classifier for each label by means of a combination of KNN and Bayesian inference, and still can be considered as the state-of-the-art MLC classifier. It is parameterized by the size k of the neighborhood, for which we adopted k = 10 as recommended in [4], the value yields the best performance. This method is implemented by using the codes directly provided by the authors.

4) IBLR3 [5] learns a single classifier for each label by combining KNN and logistic regression. It is also parameterized by the size of the neighborhood, for which we adopted the same value as stated in [5]. Likewise, we use its implementation in the MULAN package [14].

BR-SVM, SML and our proposed methods SBLR, SparseSBLR all use the RBF similarity function. The RBF hyperparameter and the hyperparameter λ of SparseSBLR are both selected from the set \{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^0\} via cross-validation on 10% of the training data.

4.1.3 Results

In the experiments, we adopt a 10-fold cross-validation strategy to compute the mean of classification performance. The results are presented in Tables 2-6, where the number in brackets behind the performance value is the rank of the method on the corresponding data set and the average rank is the average of the ranks across all datasets.

As we can see from the above tables, our proposed model SparseSBLR, achieves the best average ranking in terms of Hamming loss, Rank loss, One error, Coverage and Average precision. More specifically, 1) both SparseSBLR and SBLR achieve better average rank performance than SML in terms of all evaluation measures, which verifies that incorporating class dependencies into SML is useful to further promote its classification performance for MLC; 2) compared to the compared methods, SparseSBLR nearly achieves best performance on the datasets with relatively more classes, i.e., birds, genbase, medical and CAL500. This phenomenon indicates that SparseSBLR can indeed detect the irrelevant classes and prevent them from impairing classification performance.

To verify whether the differences between different methods are significant in terms of all evaluation measures, the Nemenyi tests are conducted. The results are shown in Figures 2(a)-2(e) from which we can see that, in Hamming loss, Rank loss, One error, Coverage and Average precision, SparseSBLR is significantly different from SML, while SBLR, BR-SVM, MLKNN and IBLR achieve comparable performance with SML. Thus, we can conclude that SparseSBLR achieves excellent performance on all evaluation measures across all the datasets here.

4.2 Experiments on MCC datasets

Similar to the above subsection, we also give the experimental datasets and compared methods to evaluate the efficacy of the proposed methods on MCC task.

| Average Precision | SparseSBLR | SML | MLKNN | IBLR | BR-SVM |
|------------------|-----------|-----|-------|------|--------|
| emotions        | 0.652(2)  | 0.645(4) | 0.656(5) | 0.648(3) | 0.655(4) |
| scene           | 0.760(2)  | 0.754(4) | 0.760(5) | 0.758(3) | 0.761(4) |
| yeast           | 0.630(2)  | 0.624(4) | 0.632(5) | 0.629(3) | 0.631(4) |
| birds           | 0.530(2)  | 0.525(4) | 0.535(5) | 0.532(3) | 0.534(4) |
| genbase         | 0.590(1)  | 0.576(2) | 0.591(3) | 0.585(4) | 0.595(3) |
| medical         | 0.690(1)  | 0.676(2) | 0.691(3) | 0.685(4) | 0.695(3) |
| CAL500          | 0.560(1)  | 0.546(2) | 0.561(3) | 0.555(4) | 0.565(3) |
| Average rank    | 1.43       | 2.71   | 5.00   | 3.97   | 4.00   | 4.29 |

TABLE 6 Experimental results in terms of Average Precision.
4.2 Datasets

We use eleven datasets from UCI machine learning repository [15] and give an overview of them in Table 7.

4.2.2 Compared methods

We compare the proposed methods SBLR and SparseSBLR to four MCC methods:

1) BR-SVM [13] works by training a number of independent SVM classifiers, one per class. The class with the largest score is the final predicted output.

2) Logistic Regression model (LR) [16] is an inductive classification method and used as a baseline for MCC task.

3) KNN [1] is a simple but effective method for classification. An instance is classified by a majority vote of its neighbors, with the instance being assigned to the class most common among its k nearest neighbors.

4) NCM [2] is a classification model that assigns to an instance the class of training instances whose mean is closest to the instance.

5) SML [6] has been used as a compared method for MLC task. Because it can be naturally applied to MCC, we also used here as a compared method.

BR-SVM, SML, SBLR and SparseSBLR still use the RBF similarity function. And, their hyperparameters are also chosen in the same way as that in the MLC experiments.

4.2.3 Results

In the experiments, we adopt a 10-fold cross-validation strategy to compute the mean of classification accuracy. The classification results are presented in Table 8. From the Table 8, we can see that SparseSBLR and SBLR achieve the highest and the third-highest average ranking, respectively. To verify whether the differences between different methods are significant, the Nemenyi tests are conducted. The results are shown in Figure 2(f), from which we can see that SparseSBLR is significantly different from SML, while SBLR, KNN, NCM and LR achieve comparable performance with SML. Therefore, we can conclude that SparseSBLR achieves excellent classification performance in MCC.

4.3 Discussion

The learned parameters of generalized linear models indicate the contribution of one class to another class, providing interpretability to some extent. We show the absolute values of the SparseSBLR parameter and the SBLR parameter on the birds dataset in Figure 3 and Figure 4. In order to see
what kind of class dependency is portrayed, we show the learned parameters \(\{\beta_1, \ldots, \beta_n\}\) more detailedly on the birds dataset in Table 9, where the parameters reflect the contributions from other classes to class \(k\) (class ‘Pacific Wren’). As can be seen from Table 9, the contributions from the low co-occurrence classes often have smaller parameter values. For example, the contributions, learned by SparseSBLR, from three classes ‘Dark-eyed Junco’, ‘Hammond’s Flycatcher’ and ‘Swainson’s Thrush’ to class ‘Pacific Wren’ are \(|0| < |1.07| < |3.23|\), while their corresponding counts of co-occurrence are \(0 < 13 < 18\).

5 Conclusion

In this paper, we propose a novel framework for MLC and MCC by uniting SML and generalized linear models. In particular, by uniting SML and logistic regression model, we present two novel approaches, called SBLR and SparseSBLR respectively. Both methods can capture interdependencies between classes and moreover, SparseSBLR can also discovery noisy classes to prevent them from impairing classification performance, thus providing the interpretability to some extent. What’s more, extensive experiments on MLC and MCC datasets not only show excellent classification performance, but also reveal what kind of class dependency is portrayed by our model.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Nos. 62076124 and 62006098) and the China Postdoctoral Science Foundation (No. 2020M681515). It is completed in the Nanjing University of Aeronautics and Astronautics.