Multi-Label Classification: Inconsistency and Class Balanced K-Nearest Neighbor

Hua Wang, Chris Ding and **Heng Huang**
Department of Computer Science and Engineering, University of Texas at Arlington, Arlington, TX 76019
huawang2007@mavs.uta.edu, {heng,chqding}@uta.edu

Abstract

Many existing approaches employ one-vs-rest method to decompose a multi-label classification problem into a set of 2-class classification problems, one for each class. This method is valid in traditional single-label classification, it, however, incurs training inconsistency in multi-label classification, because in the latter a data point could belong to more than one class. In order to deal with this problem, in this work, we further develop classical K-Nearest Neighbor classifier and propose a novel Class Balanced K-Nearest Neighbor approach for multi-label classification by emphasizing balanced usage of data from all the classes. In addition, we also propose a Class Balanced Linear Discriminant Analysis approach to address high-dimensional multi-label input data. Promising experimental results on three broadly used multi-label data sets demonstrate the effectiveness of our approach.

Introduction

Multi-label classification frequently arises in many real life applications, such as document categorization, image annotation, *etc*. Different from traditional *single-label classification* where each object belongs to only one class, multi-label classification deals with problems where an object may belong to more than one class. Formally, for a classification task, given K predefined classes and n data points, each data point $\mathbf{x}_i \in \mathbb{R}^p$ is associated with a subset of class labels represented by a binary label indicator vector $\mathbf{y}_i \in \{1,0\}^K$, such that $\mathbf{y}_i(k) = 1$ if $\mathbf{x}_i$ belongs to the k-th class, and 0 otherwise. In single label classification, it holds that $\sum_{k=1}^{K} \mathbf{y}_i(k) = 1$; while for multi-label classification, $\sum_{k=1}^{K} \mathbf{y}_i(k) \geq 1$. Given a training data set $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{l}$ with l $(l < n)$ data points, our goal is to predict labels $\{\mathbf{y}_i\}_{i=l+1}^{n}$ for the rest $n-l$ testing data points $\{\mathbf{x}_i\}_{i=l+1}^{n}$.

Inconsistency of One-vs-Others Method in Multi-label Classification

Some of the most powerful classification algorithms, such as Support Vector Machine (SVM), are built upon 2-class classifiers. One of the most popular approach to construct a multi-class classifier from these algorithms is to employ "one-vs-others" method, in which a multi-class problem is decomposed into a set of 2-class classification problems, one for each class. In each 2-class subproblem, data points belonging the given class are considered as positive samples, while those belonging to other classes are considered as negative samples. This treatment is reasonable in single-label classification, because the classes here are assumed mutually exclusive and a data point belongs to only one class.

As one of our contribution, in this paper, we point out that the above one-vs-others method has a fundamental deficiency when it is applied to multi-label classification, although it has been broadly used in many existing multi-label classification algorithms.

The deficiency occurs at the training (or parameter estimation) stage. The standard training process of the one-vs-others method for multi-label classification is as following. In a 2-class classifier for a given class, the training data points belonging to the class are used as positive samples, while those not belonging to the class are used as negative samples. This, however, is inconsistent in multi-label classification. For example, let us consider a 4-class multi-label classification problem, in which we have four target classes, which are denoted as c_1, c_2, c_3 and c_4, respectively. For the 2-class subproblem c_1-vs-others, *i.e*., $\{c_1\}$-vs-$\{c_2 \cup c_3 \cup c_4\}$, suppose data point $\mathbf{x}_1$ has two labels c_1 and c_2, it should simultaneously belong to both class $\{c_1\}$ and class $\{c_2 \cup c_3 \cup c_4\}$. More precisely, $\mathbf{x}_1$ belongs to c_1 with probability 1, and belongs to class $(c_2 \cup c_3 \cup c_4)$ with probability 1/3. Therefore, it can not be completely considered as a positive sample of class $\{c1\}$. Suppose another data point $\mathbf{x}_2$ has three labels $c1$, $c2$ and $c3$. When training the same c_1-vs-$\{c_1 \cup c_2 \cup c_3\}$ 2-class $c1$-vs-others classifier, $\mathbf{x}_2$ should belong to class c_1 with probability 1, and belong to class $\{c_2 \cup c_3 \cup c_4\}$ with probability 2/3. Therefore, the training process in the standard one-vs-others method for multiple-label classification is **inconsistent**.

On the other hand, one-vs-others method is particularly appropriate to classify new data points, because it naturally assigns multiple labels to a new/testing data point.

A straightforward remedy to the training inconsistency problem when using one-vs-others method in multi-label classification is to weight the training error. For example, we can assign a weight of 1 to the training error to misclassify $\mathbf{x}_1$ to class $\{c_2 \cup c_3 \cup c4\}$, and assign $1/3$ to the training error to misclassify $\mathbf{x}_1$ to class $\{c_1\}$.

Another multi-class/label classifier construction approach from 2-class classifiers is "one-vs-one" method, which also decomposes a multi-class/label problem into a set of 2-class subproblems, one for each class pair. This approach solves the above training inconsistency problem. When training c_k-vs-c_l 2-class classifier, if (1) data points with both class labels c_k and c_l are excluded, and (2) data point not belonging to either class are excluded, the rest data points are uniquely used by either c_k or c_l. However, the classification of new data points for multi-label problem requires a threshold to determine how many labels the new data point should acquire. In addition, as well known, one-vs-one method requires 2^K sub-classifiers, which could easily lead to explosion of computation.

An entirely new direction to avoid the training inconsistency problem is to use non-parametric classification methods, such as K-Nearest Neighbor (KNN) classifier, either directly on the original input space or on a reduced feature space, because there is no training phase involved. The KNN approach, however, suffers from unbalanced data distribution problem, which become more severe for multi-label problem. Fundamentally, because we need to assign multiple labels to a test data point, a simple 1NN or 3NN do not have enough information for this multi-label assignment (in contrast to single label), which will be discussed in detail later in algorithm description section.

In this work, we deal with the training inconsistency problem incurred by one-vs-others method in multi-label classification. We take the perspective to use non-parametric classification method and propose our Class Balanced K-Nearest Neighbor (BKNN) approach for multi-label classification by emphasizing balanced usage of data from all the classes. The aforementioned problems in non-parametric classification methods are solved in our BKNN approach. We also propose a Class Balanced Linear Discriminant Analysis (BLDA) method to embed high-dimensional multi-label input data into a lower dimensional space, which can be used for optional dimensionality reduction prior to classification.

Class Balanced Linear Discriminant Analysis

Same as single-label classification, high-dimensional input data could make multi-label classification computationally infeasible due to "curse-of-dimensionality" (Fukunaga 1990). Therefore, dimensionality reduction to prune irrelevant features and reduce dimensionality is necessary prior to classification. In this work, we further develop the classical LDA to reduce the dimensionality of multi-label data. We first point out the difficulties in computing the scatter matrices when using traditional single-label definitions in multi-label classification, and then propose our class-wise multi-label scatter matrices to deal with the problem. Meanwhile the powerful classification capability inherited from classical LDA is preserved.

In traditional single-label multi-class classification, the scatter matrices S_b and S_w are well defined in standard LDA algorithm as per the geometrical dispersion of data points. These definitions, however, become obscure when applied to multi-label classification. Because a data point with multiple labels belong to different classes at the same time, how much it should contribute to the between-class and within-class scatters remains unclear.

Therefore, instead of computing the scatter matrices from data points perspective as in standard LDA, we propose to formulate them by class-wise, *i.e.*, $S_b = \sum_{k=1}^{K} S_b(k)$, $S_w = \sum_{k=1}^{K} S_w(k)$, and $S_t = \sum_{k=1}^{K} S_t(k)$. In this way, the structural variances of the training data are represented more lucid and the construction of the scatter matrices turns out easier. Especially, the ambiguity, how much a data point with multiple labels should contribute to the scatter matrices, is avoided. The *multi-label between-class scatter matrix* is defined as:

$$S_b = \sum_{k=1}^{K} S_b^{(k)},\ S_b^{(k)} = \left(\sum_{i=1}^{l} Y_{ik}\right)(\mathbf{m}_k - \mathbf{m})(\mathbf{m}_k - \mathbf{m})^T, \quad (1)$$

the *multi-label within-class scatter matrix* S_w is defined as:

$$S_w = \sum_{k=1}^{K} S_w^{(k)},\ S_w^{(k)} = \sum_{i=1}^{l} Y_{ik}\,(\mathbf{x}_i - \mathbf{m}_k)\,(\mathbf{x}_i - \mathbf{m}_k)^T, \quad (2)$$

where $\mathbf{m}_k$ is the mean of class k and $\mathbf{m}$ is the *multi-label global mean*, which are defined as follows:

$$\mathbf{m}_k = \frac{\sum_{i=1}^{l} Y_{ik}\mathbf{x}_i}{\sum_{i=1}^{l} Y_{ik}}, \quad \mathbf{m} = \frac{\sum_{k=1}^{K}\sum_{i=1}^{l} Y_{ik}\mathbf{x}_i}{\sum_{k=1}^{K}\sum_{i=1}^{l} Y_{ik}}. \quad (3)$$

Note that, the multi-label global mean $\mathbf{m}$ defined in Eq. (3) is different from the global mean in single-label sense as in standard LDA. The latter is defined as $\frac{1}{l}\sum_{i=1}^{l}\mathbf{x}_i$.

Equipped with the class-wise scatter matrices defined in Eqs. (1–2), we can compute the transformation matrix $U \in \mathbb{R}^{p\times r}$ following the standard LDA algorithm (Fukunaga 1990), where r is the reduced dimensionality. The projected data points are hence computed by $\mathbf{q}_i = U^T\mathbf{x}_i$.

Class Balanced K-Nearest Neighbor Classifier

Given input data, either original features $\mathbf{x}_i$ or reduced features $\mathbf{q}_i$, we may use statistical learning method to conduct classification. Considering the training inconsistency problem of one-vs-others method in multi-label classification, we consider to use KNN classifier due to its non-parametric property. However, the following difficulties prevent us from directly using the classical KNN classifier.

The first one is the unbalanced data distribution. The second one is the thresholding problem, which is also caused by the nature of multi-label data. For example, for a 4-class multi-label classification problem same as before, we use 3NN and assume a testing data point $\mathbf{x}_i$ has the following nearest neighbors, $\mathbf{x}_1$ with labels c_3 and c_4, $\mathbf{x}_2$ with labels c_2, c_3 and c_4, $\mathbf{x}_3$ with labels c_1 and c_4. Therefore, the most frequently appearing labels in the neighbors of the testing data point $\mathbf{x}_i$ are sorted as following: c_4 for 3 times, c_3 for 2 times, c_1 for 1 times and c_2 for 1 times. Apparently, c_4 will be assigned to $\mathbf{x}_i$ if this is a single-label problem. However, in multi-label case, a threshold is required to make classification and it is hard to select an optimal one.

In order to exploit its non-parametric property, in this section, we further develop classical KNN method, and propose a Class Balanced K-Nearest Neighbor (BKNN) approach for multi-label classification. The aforementioned

two problems in classical KNN classifier are solved by our BKNN approach.

Algorithm. Step 1. Given a test data point $\mathbf{x}_i (l+1 < i \leq n)$, we pick up b nearest neighboring data points in each class. This leads to at most $K \times b$ data points, denoted as Γ_i. Let $\{\mathbf{x}_{ij}^{(k)}\}_{j=1}^{b}$ be the data points in Γ_i for the kth class, given a similarity matrix $W \in \mathbb{R}^{n \times n}$ with W_{ij} indicating the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$, we define:

$$s_i^{(k)} = \sum_{j=1}^{b} W(\mathbf{x}_{ij}^{(k)}, \mathbf{x}_i), \quad \text{and} \quad \bar{s_i} = \frac{\sum_{k=1}^{K} s_i^{(k)}}{K}, \quad (4)$$

where $W(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma\right)$ in this work.

Step 2. We compute the *Balanced K-Nearest Neighbor Decision Score* of a test data point $\mathbf{x}_i$ for the k-th class as:

$$f_i^{(k)} = \frac{s_i^{(k)} - \bar{s_i}}{\bar{s_i}}. \quad (5)$$

Step 3. In order to decide the class membership of $\mathbf{x}_i$, we learn a threshold from training data. We first compute $f_i^{(k)}$ for all the training data points for the kth class. Given a threshold h_k, let $p_k(h_k)$ and $r_k(h_k)$ denote the corresponding "precision" and "recall" on the training data using $\{f_i^{(k)}\}_{i=1}^{l}$, we select our *Adaptive Decision Boundary* as:

$$h_k^{\text{opt}} = \arg\max_{h_k} \left(\frac{2}{\frac{\alpha}{p_k(h_k)} + \frac{1-\alpha}{r_k(h_k)}} \right), \quad (6)$$

where α is an application dependent parameter to determine how much the "weighted F1 score" should be biased to precision, and empirically selected as 0.5 in this work. Finally, the labels $\mathbf{y}_i$ for a test point $\mathbf{x}_i$ is determined by

$$\mathbf{y}_i(k) = \text{sign}\left(f_i^{(k)} - h_k^{\text{opt}}\right). \quad (7)$$

Obviously, although different class could have very different number of labeled data points, for a given test data point, we pick up the most representative training data points from every class with equal number, *i.e.*, b for each class, such that the label of a test data point is determined via the information from all the classes in a balanced manner.

Note that, b in our approach is a free parameter like K in KNN, which is normally fine tuned by cross validation. Empirically, small b gives good classification results.

Empirical Studies

We use standard 5-fold cross validation to evaluate the proposed BKNN approach in multi-label classification, and compare the experimental results with the following most recent multi-label classification methods: (1) Semi-supervised learning by Sylvester Equation (SMSE) (Chen et al. 2008) method, (2) Multi-Label Least Square (MLLS) (Ji et al. 2008) method and (3) Multi-label Correlated Green's Function (MCGF) (Wang, Huang, and Ding 2009) method. We follow the implementation details as in the original works. The input data are first projected to a lower r dimensional subspace before being fed into the respective classification approaches. In our evaluations, we set $r = K - 1$.

Table 1: Multi-label classification performance measured by "average precision" for the four compared approaches.

Approaches	data sets		
	TRECVID	Yahoo	Music
SMSE	10.7%	13.5%	21.3%
MLLS	23.7%	27.6%	31.2%
MCGF	24.9%	24.3%	30.3%
BKNN ($b = 5$)	**25.6%**	**26.8%**	**36.2%**

We apply the compared approaches on the following three broadly used multi-label datasets from different applications: **TRECVID 2005** dataset (Smeaton, Over, and Kraaij 2006) (image), **Music emotion** dataset (Trohidis et al. 2008) (music), and **Yahoo** dataset ("Science" topic) (Ueda and Saito 2002) (document). For performance evaluation, we adopt "Average Precision" as recommended by TRECVID (Smeaton, Over, and Kraaij 2006), which computes the precision for each class and average them over all the classes.

We adopt "Average Precision" as our performance metric as recommended by TRECVID (Smeaton, Over, and Kraaij 2006), which computes the precision for each class and average them over all the classes.

Table 1 presents the overall classification performance comparisons measured by average precision on the three data sets. The results show that the proposed BKNN approach outperforms all the other compared approaches, which quantitatively demonstrate the advantage of our approach.

Acknowledgments

This research is supported by NSF CCF-0830780, NSF CCF-0939187, NSF CCF-0917274, NSF DMS-0915228.

References

Chen, G.; Song, Y.; Wang, F.; and Zhang, C. 2008. Semi-supervised Multi-label Learning by Solving a Sylvester Equation. In *SDM'08*.

Fukunaga, K. 1990. *Introduction to statistical pattern recognition*. Academic Press.

Ji, S.; Tang, L.; Yu, S.; and Ye, J. 2008. Extracting shared subspace for multi-label classification. In *SIGKDD'08*.

Smeaton, A. F.; Over, P.; and Kraaij, W. 2006. Evaluation campaigns and trecvid. In *MIR'06*.

Trohidis, K.; Tsoumakas, G.; Kalliris, G.; and Vlahavas, I. 2008. Multilabel classification of music into emotions. In *ISMIR'08*.

Ueda, N., and Saito, K. 2002. Single-shot detection of multiple categories of text using parametric mixture models. In *SIGKDD'02*.

Wang, H.; Huang, H.; and Ding, C. 2009. Image Annotation Using Multi-label Correlated Greens Function. In *ICCV'09*.