# A Coupled k-Nearest Neighbor Algorithm for Multi-label Classification

Chunming Liu and Longbing Cao

AAI, University of Sydney Technology, Australia
Chunming.Liu@student.uts.edu.au; LongBing.Cao@uts.edu.au

**Abstract.** ML-$k$NN is a well-known algorithm for multi-label classification. Although effective in some cases, ML-$k$NN has some defect due to the fact that it is a binary relevance classifier which only considers one label every time. In this paper, we present a new method for multi-label classification, which is based on lazy learning approaches to classify an unseen instance on the basis of its $k$ nearest neighbors. By introducing the coupled similarity between class labels, the proposed method exploits the correlations between class labels, which overcomes the shortcoming of ML-$k$NN. Experiments on benchmark data sets show that our proposed Coupled Multi-Label $k$ Nearest Neighbor algorithm (CML-$k$NN) achieves superior performance than some existing multi-label classification algorithms.

**Keywords:** Multi-label, Coupled, Classification, Nearest neighbor

## 1 Introduction

Although traditional single-label classification approaches have been proved to be successful in handling some real world problems, for the problems which the objects not fit the single-label rule, they may not work well, for example, in image classification, an image may contain several concepts simultaneously, such as beach, sunset and kangaroo. Such tasks are usually denoted as multi-label classification problems. In fact, a conventional single-label classification problem can simply be taken as a special case of the multi-label classification problem where there has only one label in the class label space. Multi-label classification problems exist in many domains, for example, in automatic text categorization, a document can associate with several topics, such as arts, history and Archeology; and in gene functional analysis of bio-informatics, a gene can belong to both metabolism and transcription classes; and in music categorization, a song may labeled as Mozart and sad.

In the last decades, there have been a variety of methods developed for multi-label classifications. These methods are generally grouped into two categories: One is the problem transformation methods and another is the algorithm adaptation methods. Problem transformation methods first transform the multi-label learning tasks into multiple single-label learning tasks, which are then handled by the standard single-label learning algorithms. Another approach is called algorithm adaptation method, which modifies existing single-label learning algorithms in order to extend its ability to handle multi-label data, such as ML-$k$NN [17], IBLR [7], BSVM [2], and BP-MLL [16].

Researchers have tried to extend the $k$NN concept to handle the multi-label classification problem, such as ML-$k$NN. ML-$k$NN applies maximum a posteriori principle for classification and ranking, and the likelihood is estimated by using the $k$ nearest neighbors of an instance. Although simple and powerful, there are some shortcomings in its processing strategy. ML-$k$NN uses the popular binary relevance (BR) strategy [13], which may transfer the problem into many class-imbalance tasks, and then tend to degrade the performance of the classifiers. Another problem of it is the estimation of the posteriori may be affected by the facts that the instances with and without a particular label are typically highly imbalanced. Furthermore, its ignorance of the inter relationship between labels is another issue which limits its usage. Such relationship is described as a Coupled behavior in some previous research [6, 4]. In [14, 8], Can and Liu etc. analysis the coupling relationship on categorical data. These works all proved the effectiveness of considering the dependency between different attributes.

In this paper, we propose a novel $k$NN-based multi-label learning approach (CML-$k$NN for short) based on non-iidness [5]. The major contribution of this paper is summarized as follows:

- We propose a novel multi-label learning algorithm that based on lazy learning and the inner relationship between labels.
- We introduce a new coupled label similarity for multi-label $k$NN algorithm. Rather than only select the neighbors with a specific label, the coupled label similarity will include more similar neighbors in the process to overcome the problem of lacking neighbors with certain label.
- We extended the concept of the nearest neighbor in multi-label classification with coupled label similarity. Based on this extended nearest neighbors, we introduce a new frequency array strategy.

The structure of this paper is organized as follows. Section 2 briefly reviews the ML-$k$NN algorithm. Preliminary definitions are specified in Section 3.1. And section 3 gives a detailed description of the new algorithm we proposed. The experimental results are discussed in Section 4. Finally, the conclusion is discussed in Section 5.

## 2 ML-$k$NN

A number of multi-label learning methods are adapted from $k$NN [3, 11, 15, 17]. ML-$k$NN, the first multi-label lazy learning approach, is based on the traditional $k$NN algorithm and the maximum a posteriori (MAP) principle [17].

The main idea of the ML-$k$NN approach is that an instance's labels depend on the number of neighbors that possess identical labels. Given an instance $x$ with an unknown label set $L(x) \subseteq L$, ML-$k$NN first identifies the $k$ nearest neighbors in the training data and counts the number of neighbors belonging to each class (i.e. a variable $z$ from 0 to $k$). Then the maximum a posteriori principle is used to determine the label set for the test instance. The posterior probability of $l_i \in L$ is given by

$$P(l_i \in L(x)|z) = \frac{P(z|l_i \in L(x)) \cdot P(l_i \in L(x))}{P(z)} \tag{1}$$

where $z$ is the number of neighbors belonging to each class $(0 \leq z \leq k)$. Then, for each label $l_i \in L$, the algorithm builds a classifier $h_i$ using the rule

$$h_i(x) = \begin{cases} 1 \;\; P(l_i \in L(x)|z) > P(l_i \notin L(x)|z) \\ \\ 0 \qquad\qquad\quad \text{otherwise} \end{cases} \tag{2}$$

where $0 \leq z \leq k$. If $h_i(x) = 1$, it means label $l_i$ is in $x$'s real label set, while 0 means it does not. The prior and likelihood probabilities in Eq. 1 are estimated from the training data set in advance.

ML-$k$NN has two inheriting merits from both lazy learning and MAP principle: One is the decision boundary can be adaptively adjusted due to the varying neighbors identified for each new instance, and another one is the class-imbalance issue can be largely mitigated due to the prior probabilities estimated for each class label. However, ML-$k$NN is actually a binary relevance classifier, because it learns a single classifier $h_i$ for each label independently. In other words, it does not consider the correlations between different labels. The algorithm is often criticized because of this drawback.

## 3 Methodology

### 3.1 Problem Statement

We formally define the multi-label classification problem as this: Let $X$ denotes the space of instances and $Y = \{l_1, \ldots, l_n\}$ denotes the whole label set where $|Y| = n$. $T = \{(x_1, L(x_1)), \ldots, (x_m, L(x_m))\}$ $(|T| = m)$ is the multi-label training data set, whose instances are drawn identically and independently from an unknown distribution $D$. Each instance $x \in X$ is associated with a label set $L(x) \in Y$. The goal of our multi-label classification is to get a classifier $h : X \rightarrow Y$ that maps a feature vector to a set of labels, while optimizing some specific evaluation metrics.

### 3.2 Coupled Label Similarity

It is much easier for numerical data to calculate the distance or similarity, since the existing metrics such as Manhattan distance and Euclidean distance are mainly built for numeric variables, but the labels are categorical data. How to denote the similarity between them is a big issue. As we all know, matching and frequency [1] are the most common ways to measure the similarity of categorical data. Accordingly, two popular similarity measures are defined: For two categorical value $v_i$ and $v_j$, the Overlap Similarity is defined as

$$\text{Sim\_Overlap}(v_i, v_j) = \begin{cases} 1, & \text{if } v_i = v_j \\ 0, & \text{if } v_i \neq v_j, \end{cases}, \tag{3}$$

and the Frequency Based Cosine Similarity between two vectors $V_i$ and $V_j$ is defined as

$$\text{Sim\_Cosine}(V_i, V_j) = \frac{V_i \cdot V_j}{||V_i||\,||V_j||}. \tag{4}$$

The overlap similarity between two categorical values is to assign 1 if they are identical otherwise 0 if different. Further, for two multivariate categorical data points, the similarity between them will be proportional to the number of features in which they match. While for frequency based measures, they assume the different categorical values but with the same occurrence times as the same.

Hence, the Overlap measure and Frequency Based measure are too simplistic by just giving the equal importance to matches and mismatches. The co-occurrence information in categorical data reflects the interaction between features and can be used to define what makes two categorical values more or less similar. However, such co-occurrence information hasn't been incorporated into the existing similarity metrics.

To capture the inner relationship between categorical labels, we introduce an *Intra-Coupling Label Similarity (IaCLS)* and an *Inter-Coupling Label Similarity (IeCLS)* below to capture the interaction of two label values from two different labels.

**Definition 1** *Given a training multi-label data set $D$ and two different labels $l_i$ and $l_j$ $(i \neq j)$, the label value is $v_i^x, v_j^y$ respectively. The* ***Intra-Coupling Label Similarity*** *(IaCLS) between label values $v_i^x$ and $v_j^y$ of label $l_i$ and $l_j$ is formalized as:*

$$\delta^{Intra}(v_i^x, v_j^y) = \frac{RF(v_i^x) \cdot RF(v_j^y)}{RF(v_i^x) + RF(v_j^y) + RF(v_i^x) \cdot RF(v_j^y)}, \tag{5}$$

*where $RF(v_i^x)$ and $RF(v_j^y)$ are the occurrence frequency of label value $v_i^x$ and $v_j^y$ in label $l_i$ and $l_j$, respectively.*

The Intra-coupling Label Similarity reflects the interaction of two different label values in the label space. The higher these similarities are, the closer such two values are. Thus, Equation (5) is designed to capture the label value similarity in terms of occurrence times by taking into account the frequencies of categories. Besides, since $1 \leq RF(v_i^x), RF(v_j^y) \leq m$, then $\delta^{Intra} \in [1/3, m/(m+2)]$.

In contrast to the Intra-Coupling, we also define an *Inter-Coupling Label Similarity* below to capture the interaction of two different label values according to the co-occurrence of some value (or discretized value group) from feature spaces.

**Definition 2** *Given a training multi-label data set $D$ and two different labels $l_i$ and $l_j$ $(i \neq j)$, the label value is $v_i^x, v_j^y$ respectively. $v_i^x$ and $v_j^y$ are defined to be Inter-Coupling related if there exists at least one pair value $(v_p^{zx})$ or $(v_p^{zy})$ that occurs in feature $a_z$ and labels of instance $U_p$. The* ***Inter-Coupling Label Similarity*** *(IeCLS) between label values $v_i^x$ and $v_i^y$ according to feature value $v_p^z$ of feature $a_z$ is formalized as:*

$$\delta^{Inter}(v_i^x, v_j^y | v_p^z) = \frac{\min\left(F(v_p^{zx}), F(v_p^{zy})\right)}{\max(RF(v_i^x), RF(v_j^y))}, \tag{6}$$

*where $F(v_p^{zx})$ and $F(v_p^{zy})$ are the co-occurrence frequency count function for value pair $v_p^{zx}$ or $v_p^{zy}$, and $RF(v_i^x)$ and $RF(v_j^y)$ is the occurrence frequency of related class label. $v_p^z$ is the value in categorical feature $a_z$ or the discretized value group in numerical feature $a_z$.*

Accordingly, we have $\delta^{Ie} \in [0, 1]$. The Inter-Coupling Label Similarity reflects the interaction or relationship of two label values from label space but based on the connection to some other features.

**Definition 3** *By taking into account both the Intra-Coupling and the Inter-Coupling, the* ***Coupled Label Similarity*** *(CLS) between two label values $v_i^x$ and $v_j^y$ is formalized as:*

$$CLS(v_i^x, v_j^y) = \delta^{Intra}(v_i^x, v_j^y) \cdot \sum_{k=1}^{n} \delta^{Inter}(v_i^x, v_j^y | v_k), \tag{7}$$

*where $v_i^x$ and $v_j^y$ are the label values of label $l_i$ and $l_j$, respectively. $\delta^{Intra}$ and $\delta^{Inter}$ are the intra-coupling label similarity (Eq. 5) and inter-coupling label similarity (Eq. 6), respectively. The $n$ is the number of attributes and $v_k$ denotes the values in the* k*th feature $a_k$.*

**Table 1.** An Example of Multi-label Data

| Instances | Label1 | Label2 | Label3 | Label4 |
|---|---|---|---|---|
| $u_1$ | $l_1$ | | | $l_4$ |
| $u_2$ | | | $l_3$ | $l_4$ |
| $u_3$ | $l_1$ | | $l_3$ | |
| $u_4$ | | $l_2$ | $l_3$ | |
| $u_5$ | | $l_2$ | $l_3$ | $l_4$ |

The *Coupled Label Similarity* defined in Eq. 7 reflects the interaction or similarity of two different labels. The higher the *CLS*, the more similar two labels be. In Table 1, for example, $CLS(l_1, l_4) = 0.33$, $CLS(l_1, l_3) = 0.25$, so in the data set, an instance with label $l_4$ is more similar or close to instances with label $l_1$ than those instances with label $l_3$ do. That is to say, label pair ($l_1$,$l_4$) is closer to each other than the label pair ($l_1$, $l_3$). For Table 1, we got the coupled label similarity array which showed in Table 2.

**Table 2.** CLS Array

| | **Label1** | **Label2** | **Label3** | **Label4** |
|---|---|---|---|---|
| **Label1** | 1.0 | 0 | 0.25 | 0.33 |
| **Label2** | 0 | 1.0 | 0.50 | 0.33 |
| **Label3** | 0.25 | 0.50 | 1.0 | 0.50 |
| **Label4** | 0.33 | 0.33 | 0.50 | 1.0 |

### 3.3 Extended Nearest Neighbors

Based on the Coupled Label Similarity, we introduce our extended nearest neighbors. Based on the similarity between labels, we can transfer a label set into a set with only a certain label, it also means a multi-label instance can be extended to a set of single-label. If we specify a basic label $l_b$, then any instance can be transformed into a set with only one label $l_b$. For example, in Table 1, instance $u_5$ has a label set of $\{l_2, l_3, l_4\}$, then according to the label similarity array Table 2, it can be transformed into $\{1 \cdot l_2, 0.5 \cdot l_2, 0.33 \cdot l_2\}$ if we choose label $l_2$ as the basic label. We can then call the original multi-label instance $u_5$ equals a single-label instance with a label of $\{1.83 \cdot l_2 | l_2\}$. If $u_5$ is the neighbor of some instance, when we consider the label $l_2$, the instance $u_5$ can be presented as an instance which contains $1 + 0.5 + 0.33 = 1.83$ label $l_2$, and vice versa, instance $u_5$ also presents there are $(1-1)+(1-0.5)+(1-0.33) = 1.17$ instances which not contain the label $l_2$, and there will have $(1.83 + 1.17 = 3 = |L(u_5)|)$. This is the basic idea when we finding our extended nearest neighbors.

**Table 3.** Extended Nearest Neighbors

| **instance** | **Extended Neighbors** | **To Label** |
|---|---|---|
| $u_5$ | $0 \cdot l_1 + 0.25 \cdot l_1 + 0.33 \cdot l_1$ | $l_1$ |
| $u_5$ | $1 \cdot l_2 + 0.5 \cdot l_2 + 0.33 \cdot l_2$ | $l_2$ |
| $u_5$ | $0.5 \cdot l_3 + 1 \cdot l_3 + 0.5 \cdot l_3$ | $l_3$ |
| $u_5$ | $0.33 \cdot l_4 + 0.5 \cdot l_4 + 1 \cdot l_4$ | $l_4$ |

### 3.4 Coupled ML-*k*NN

For the unseen instance $x$, lets $N(x)$ represents the set of its $k$ nearest neighbors identified in data set $D$. For the $j$-th class label, CML-$k$NN chooses to calculate the following statistics:

$$C_j = Round(\sum_{i=1}^{k} \delta_{L_i^*|j}) \tag{8}$$

Where $L_i$ is the label set of the $i$-th neighbor and $L_i \in N(x)$, and $\delta_{L_i^*|j}$ denotes the sum of the CLS values of the $i$-th neighbor's label set to the $j$-th label $l_j$, and $Round()$ is the rounding function.

Namely, $C_j$ is a rounding number which records all the $CLS$ value of all $x$'s neighbors to label $l_j$.

Let $H_j$ be the event that $x$ has label $l_j$ , and $P(H_j|C_j)$ represents the posterior probability that $H_j$ holds under the condition that $x$ has exactly $C_j$ neighbors with label $l_j$ . Correspondingly, $P(\neg H_j|C_j)$ represents the posterior probability that $H_j$ doesn't hold under the same condition. According to the MAP rule, the predicted label set is determined by deciding whether $P(H_j|C_j)$ is greater than $P(\neg H_j|C_j)$ or not:

$$Y = \{l_j|\frac{P(H_j|C_j)}{P(\neg H_j|C_j)} > 1, 1 \leq j \leq q\} \tag{9}$$

According to the Bayes Theory, we have:

$$\frac{P(H_j|C_j)}{P(\neg H_j|C_j)} = \frac{P(H_j) \cdot P(C_j|H_j)}{P(\neg H_j) \cdot P(C_j|\neg H_j)} \tag{10}$$

Here, $P(H_j)$ and $P(\neg H_j)$ represents the prior probability that $H_j$ holds and doesn't hold. Furthermore, $P(C_j|H_j)$ represents the likelihood that $x$ has exactly $C_j$ neighbors with label $l_j$ when $H_j$ holds, and $(P(Cj|\neg Hj))$ represents the likelihood that $x$ has exactly $C_j$ neighbors with label $l_j$ when $H_j$ doesn't hold.

When we count the prior probabilities, we integrated our coupled label similarity into the process:

$$\begin{aligned} P(H_j) &= \frac{s + \sum_{i=1}^{m} \delta_{L_i^*|j}}{s \times 2 + m \times n}; \\ P(\neg H_j) &= 1 - P(H_j); \end{aligned} \tag{11}$$

where $(1 \leq j \leq n)$ and $m$ is the records number in training set, and $s$ is a smoothing parameter controlling the effect of uniform prior on the estimation which generally takes the value of 1 (resulting in Laplace smoothing).

Same as ML-$k$NN, for the $j$-th class label $l_j$, our CML-$k$NN maintains two frequency arrays $\alpha_j$ and $\beta_j$. As our method considers the other labels which have a similarity to a specific label, the frequency arrays will contain $k \times n + 1$ elements:

$$\begin{aligned} \alpha_j[r] &= \sum_{i=1}^{m} \delta_{L_i^*|j} | C_j(x_i) = r \qquad (\delta_{L_i^*|j} \geq 0.5) \\ \beta_j[r] &= \sum_{i=1}^{m} (n - \delta_{L_i^*|j}) | C_j(x_i) = r \quad (\delta_{L_i^*|j} < 0.5) \end{aligned} \tag{12}$$

Where $(0 \leq r \leq k \times n)$. We take an instance with $\delta_{L_i^*|j} \geq 0.5$ as an instance which does have label $j$ and we take an instance with $\delta_{L_i^*|j} < 0.5$ as an instance which doesn't have label $j$. Therefore, $\alpha_j[r]$ counts the sum of CLS values to label $j$ of training examples which have label $l_j$ and have exactly $r$ neighbors with label $l_j$, while $\beta_j[r]$ counts the CLS to label $j$ of training examples which don't have label $l_j$ and have exactly $r$ neighbors with label $l_j$. Afterwards, the likelihood can be estimated based on elements in $\alpha_j$ and $\beta_j$:

$$\begin{gathered} P(C_j|H_j) = \frac{s + \alpha_j[C_j]}{s \times (k \times n + 1) + \sum_{r=0}^{k \times n} \alpha_j[r]} \\ P(C_j|\neg H_j) = \frac{s + \beta_j[C_j]}{s \times (k \times n + 1) + \sum_{r=0}^{k \times n} \beta_j[r]} \\ (1 \leq j \leq n, 0 \leq C_j \leq k \times n) \end{gathered} \tag{13}$$

Thereafter, by combing the prior probabilities (Eq.11) and the likelihoods (Eq.13) into Eq.(10), we will get the predicted label set in Eq.(9).

### 3.5 Algorithm

Given an unknown test instance $x_t$, the algorithm determines the final label set of the instance. Algorithm 1 illustrates the main idea of our process. Our proposed CML-$k$NN contains of six main parts. a)Maintain the label similarity array; b)Finding the nearest neighbors for every instance in training set; c)Getting the prior probabilities and frequency arrays; d)Finding the nearest neighbors for the target instance; e)Calculate the statistics value; f)Calculate the result.

Firstly, we calculate the label similarity according to their inter-relationships and maintain the Coupled Label Similarity Array $A(L)$ from the training data set. Secondly, for every training instance, we identify its traditional $k$ nearest neighbors. After that, for every different label, we calculate its prior probability which combined with $CLS$. Simultaneously, we expand the neighbors set for every instance to a new label-coupled neighbors set using the $CLS$, and calculate the frequency array for every label. After these works done, we identify the $k$ neighbors of the test instance $x_t$. After applying $CLS$ on this neighbor set and calculate the label statistics, we can finally get the predicted label set.

It is worth noting that our key idea is the label similarity, which tries to learn the label distance and then transfer any label into a specific label.

## 4 Experiments and Evaluation

### 4.1 Experiment Data

A total of eight commonly used multi-label data sets are tested for experiments in this study, and the statistics of the data sets are shown in Table 4. Given a multi-label data set $M = \{(x_i, L_i)|1 \leq i \leq q\}$, we use $|M|$, $f(M)$, $La(M)$, $F(M)$ to represent the number of instances, number of features, number of total labels, and feature type respectively. In addition, several multi-label statistics [9] are also shown in the Table. The Label cardinality ($LC(M)$) measures the average number of labels per example; the Label density ($LD(M)$) normalizes $LC(M)$ by the number of possible labels; the Distinct label sets ($DL(M)$) counts the number of distinct label combinations appeared in the data set; the Proportion

**Algorithm 1:** : Coupled ML-$k$NN Algorithm

```
Input: An unlabeled instance $x_t$ and a labeled dataset
    $T\{(x_1, L(x_1)), \ldots, (x_m, L(x_m))\}$, where $|T| = m$ and $|L| = n$
Output: The label set $L(x_t)$ of instance $x_t$
Calculate the $CLS$ array $A(L)$ according to Eq.(7);
for $i = 1$ to $m$ do;
    Identify the $k$ nearest neighbors $N(x_i)$ for $x_i$
end for
for $j = 1$ to $n$ do
    Calculate $P(H_j)$ and $P(\neg H_j)$ according to Eq.(11)
    Maintain the label-coupled frequency arrays $\alpha_j, \beta_j$ using Eq.(12)
end for
Identify the $k$ nearest neighbors $N(x_t)$ for $x_t$
for $j = 1$ to $n$ do
    Calculate the statistic $C_j$ according to Eq.(8)
end for
Return the label set $L(x_t)$ of instance $x_t$ according to Eq.(9)
```

**Table 4.** Experiment Data Sets

| Data Set | \|M\| | f(M) | La(M) | LC(M) | LD(M) | DL(M) | PDL(M) | F(M) |
|---|---|---|---|---|---|---|---|---|
| emotions | 593 | 72 | 6 | 1.869 | 0.311 | 27 | 0.046 | n |
| yeast | 2417 | 103 | 14 | 4.237 | 0.303 | 198 | 0.082 | n |
| image | 2000 | 294 | 5 | 1.236 | 0.247 | 20 | 0.010 | n |
| scene | 2407 | 294 | 6 | 1.074 | 0.179 | 15 | 0.006 | n |
| enron | 1702 | 1001 | 53 | 3.378 | 0.064 | 753 | 0.442 | c |
| genbase | 662 | 1185 | 27 | 1.252 | 0.046 | 32 | 0.048 | c |
| medical | 978 | 1449 | 45 | 1.245 | 0.028 | 94 | 0.096 | c |
| bibtex | 7395 | 1836 | 159 | 2.402 | 0.015 | 2856 | 0.386 | c |

of distinct label sets ($PDL(M)$) which normalizes $DL(M)$ by the number of instances. As shown in Table 4, eight data sets are included and are ordered by Label density $LD(M)$.

### 4.2 Experiment Setup

In our experiments, we compare the performance of our proposed CML-$k$NN with that some state-of-the-art multi-label classification algorithms: ML-$k$NN, IBLR and BSVM. All nearest neighbor based algorithms are parameterized by the size of the neighborhood $k$. We repeat the experiments with $k = 5, 7, 9$ respectively (odd number for voting), and use the Euclidean metric as the distance function when computing the nearest neighbors. For BSVM, models are learned via the cross-training strategy[2]. We also choose the BR-$k$NN as the basic algorithm to compare with. We perform 10-fold cross-validation three times on all the above data sets.

### 4.3 Evaluation Criteria

Multi-label classification requires different metrics than those used in traditional single-label classification. A lot of criteria have been proposed for evaluating the performance of multi-label classification algorithms [12]. In this paper, we use three popular evaluation criteria for multi-label classification: the **Hamming Loss**, the **One Error** and the **Average Precision**. The definitions of them can be found in [10].

### 4.4 Experiment Results

The experiment results are shown in Table 5 - Table 7. For each evaluation criterion, "↓" indicates "the smaller the better", while "↑" indicates "the bigger the better". And the numbers in parentheses denote the rank of the algorithms among the five compared algorithms. The result tables indicate that CML-*k*NN and BSVM outperforms other algorithms significantly, which implies that exploiting the frequency of neighbors' label is effective, and especially for our

**Table 5.** Experiment Result1 - Hamming Loss↓

| | **CML-*k*NN** | **BR-*k*NN** | **ML-*k*NN** | **IBLR** | **BSVM** |
|---|---|---|---|---|---|
| emotions | 0.189(1) | 0.219(5) | 0.194(2) | 0.201(4) | 0.199(3) |
| yeast | 0.194(1) | 0.205(5) | 0.195(2) | 0.198(3) | 0.199(4) |
| image | 0.157(1) | 0.189(5) | 0.172(2) | 0.182(4) | 0.176(3) |
| scene | 0.078(1) | 0.152(5) | 0.084(2) | 0.089(3) | 0.104(4) |
| enron | 0.061(4) | 0.052(2) | 0.052(2) | 0.064(5) | 0.047(1) |
| genbase | 0.003(2) | 0.004(3) | 0.005(4) | 0.005(4) | 0.001(1) |
| medical | 0.013(1) | 0.019(4) | 0.016(3) | 0.026(5) | 0.013(1) |
| bibtex | 0.013(1) | 0.016(4) | 0.014(2) | 0.016(4) | 0.015(3) |
| AvgRank | **(1.50)** | 4.13 | 2.38 | 4.00 | 2.50 |

**Table 6.** Experiment Result2 - One Error↓

| | **CML-*k*NN** | **BR-*k*NN** | **ML-*k*NN** | **IBLR** | **BSVM** |
|---|---|---|---|---|---|
| emotions | 0.244(1) | 0.318(5) | 0.263(3) | 0.279(4) | 0.253(2) |
| yeast | 0.222(1) | 0.235(4) | 0.228(2) | 0.237(5) | 0.232(3) |
| image | 0.267(1) | 0.601(5) | 0.319(3) | 0.432(4) | 0.314(2) |
| scene | 0.197(1) | 0.821(5) | 0.219(2) | 0.235(3) | 0.251(4) |
| enron | 0.308(3) | 0.237(1) | 0.313(4) | 0.469(5) | 0.245(2) |
| genbase | 0.008(2) | 0.012(5) | 0.009(3) | 0.011(4) | 0.002(1) |
| medical | 0.158(2) | 0.327(4) | 0.252(3) | 0.414(5) | 0.151(1) |
| bibtex | 0.376(1) | 0.631(5) | 0.589(3) | 0.576(2) | 0.599(4) |
| AvgRank | **(1.50)** | 4.25 | 2.88 | 4.00 | 2.38 |

**Table 7.** Experiment Result3 - Average Precision↑

| | **CML-$k$NN** | **BR-$k$NN** | **ML-$k$NN** | **IBLR** | **BSVM** |
|---|---|---|---|---|---|
| emotions | 0.819(1) | 0.595(5) | 0.799(3) | 0.798(4) | 0.807(2) |
| yeast | 0.769(1) | 0.596(5) | 0.765(2) | 0.759(3) | 0.749(4) |
| image | 0.824(1) | 0.601(5) | 0.792(3) | 0.761(4) | 0.796(2) |
| scene | 0.885(1) | 0.651(5) | 0.869(2) | 0.862(3) | 0.849(4) |
| enron | 0.591(3) | 0.435(5) | 0.626(2) | 0.564(4) | 0.702(1) |
| genbase | 0.994(3) | 0.992(4) | 0.989(5) | 0.994(2) | 0.998(1) |
| medical | 0.876(1) | 0.782(4) | 0.806(3) | 0.686(5) | 0.871(2) |
| bibtex | 0.567(1) | 0.329(5) | 0.351(4) | 0.476(3) | 0.531(2) |
| AvgRank | **(1.50)** | 4.75 | 3.00 | 3.50 | 2.25 |

CML-$k$NN, the improvement is significant compared to BR-$k$NN, that means incorporating the label relationship will greatly improve the BR strategy. Meanwhile, ML-$k$NN, IBLR and BR-$k$NN do not perform as well compared to the other algorithms. This implies that only exploiting the exact neighbor information is not sufficient, and the similar neighbor (correlations between labels) should also be considered.

Overall, our proposed CML-$k$NN outperforms all the compared methods on all three measures. The average ranking of our method on these data sets using three different metrics is the first one, with (1.50, 1.50, 1.50) respectively, while the second best algorithm, BSVM, only achieves (2.50, 2.38, 2.25). The BR-$k$NN performs the worst, which only achieves (4.13,4.25,4.75).

It is worth noting that although our proposed method runs the best on average, it does not mean that it is suitable for all kinds of data. For example, when used on data set "enron" and "genbase", the result is not as good as on other data sets. Sometimes it even got a worse result than BR-$k$NN. For example, when used on "enron" and evaluated by the Hamming Loss, our supposed CML-$k$NN only achieved a *4th* rank(0.061), while BR-$k$NN can get a second well result(0.052). The reason is because of the weak or loose connection between different labels in those data sets, and our extended neighbors may introduce more noisy information than useful information. But in terms of average performance, our method performs the best (the first rank).

## 5 Conclusions and Future Work

ML-$k$NN learns a single classifier $h_i$ for each label $l_i$ independently, so it is actually a binary relevance classifier. In other words, it does not consider the correlations between different labels. The algorithm is often criticized for this drawback. In this paper, we introduced a coupled label similarity, which explores the inner-relationship between different labels in multi-label classification according to their natural co-occupance. This similarity reflects the distance of the different labels. Furthermore, by integrating this similarity into the multi-label $k$NN algorithm, we overcome the ML-$k$NN's shortcoming and improved the performance. Evaluated over three commonly-used multi-label data sets and in terms of Hamming Loss, One Error and Average Precision, the proposed method outperforms ML-$K$NN, BR-$k$NN, IBLR and even BSVM. This result shows that our supposed coupled label similarity is appropriate for multi-label learning problems and can work more effectively than other methods.

Our future work will focus on expanding our coupled similarity to categorical multi-label data, and even mixed type multi-label data for which current numerical distance metrics is not suitable.

## References

1. Boriah, S., Chandola, V., Kumar, V.: Similarity measures for categorical data: A comparative evaluation. red 30(2), 3 (2008)
2. Boutell, M.R., Luo, J., Shen, X., Brown, C.M.: Learning multi-label scene classification. Pattern recognition 37(9), 1757–1771 (2004)
3. Brinker, K., Hüllermeier, E.: Case-based multilabel ranking. In: IJCAI. pp. 702–707 (2007)
4. Cao, L.: Coupling learning of complex interactions. Information Processing & Management (2014)
5. Cao, L.: Non-iidness learning in behavioral and social data. The Computer Journal 57(9), 1358–1370 (2014)
6. Cao, L., Ou, Y., Yu, P.S.: Coupled behavior analysis with applications. Knowledge and Data Engineering, IEEE Transactions on 24(8), 1378–1392 (2012)
7. Cheng, W., Hüllermeier, E.: Combining instance-based learning and logistic regression for multilabel classification. Machine Learning 76(2-3), 211–225 (2009)
8. Liu, C., Cao, L., Yu, P.S.: Coupled fuzzy k-nearest neighbors classification of imbalanced non-iid categorical data. In: Neural Networks (IJCNN), 2014 International Joint Conference on. pp. 1122–1129. IEEE (2014)
9. Read, J., Pfahringer, B., Holmes, G., Frank, E.: Classifier chains for multi-label classification. Machine learning 85(3), 333–359 (2011)
10. Schapire, R.E., Singer, Y.: Boostexter: A boosting-based system for text categorization. Machine learning 39(2-3), 135–168 (2000)
11. Spyromitros, E., Tsoumakas, G., Vlahavas, I.: An empirical study of lazy multilabel classification algorithms. In: Artificial Intelligence: Theories, Models and Applications, pp. 401–406. Springer (2008)
12. Tsoumakas, G., Katakis, I.: Multi-label classification: An overview. International Journal of Data Warehousing and Mining (IJDWM) 3(3), 1–13 (2007)
13. Vembu, S., Gärtner, T.: Label ranking algorithms: A survey. In: Preference learning, pp. 45–64. Springer (2011)
14. Wang, C., Cao, L., Wang, M., Li, J., Wei, W., Ou, Y.: Coupled nominal similarity in unsupervised learning. In: Proceedings of the 20th ACM international conference on Information and knowledge management. pp. 973–978. ACM (2011)
15. Wieczorkowska, A., Synak, P., Raś, Z.W.: Multi-label classification of emotions in music. In: Intelligent Information Processing and Web Mining, pp. 307–315. Springer (2006)
16. Zhang, M.L., Zhou, Z.H.: Multilabel neural networks with applications to functional genomics and text categorization. Knowledge and Data Engineering, IEEE Transactions on 18(10), 1338–1351 (2006)
17. Zhang, M.L., Zhou, Z.H.: Ml-knn: A lazy learning approach to multi-label learning. Pattern recognition 40(7), 2038–2048 (2007)