International Journal of Approximate Reasoning 54 (2013) 1373–1387

Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

www.elsevier.com/locate/ijar

Neighborhood rough sets based multi-label classification for automatic image annotation

Ying Yu [a,b,c,e,*], Witold Pedrycz [b,d], Duoqian Miao [a,c]

[a] Department of Computer Science and Technology, Tongji University, Shanghai 201804, PR China
[b] Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, T6G 2G7, Canada
[c] Key Laboratory of Embedded System and Service Computing, Ministry of Education, Tongji University, Shanghai 201804, PR China
[d] System Research Institute, Polish Academy of Sciences, Warsaw, Poland
[e] School of Software Design, Jiangxi Agricultural University, Nanchang 330013, PR China

ARTICLE INFO

Article history:
Received 4 December 2012
Received in revised form 3 June 2013
Accepted 6 June 2013
Available online 13 June 2013

Keywords:
Multi-label classification
Automatic image annotation
Neighborhood rough sets

ABSTRACT

Automatic image annotation is concerned with the task of assigning one or more semantic concepts to a given image. It is a typical multi-label classification problem. This paper presents a novel multi-label classification framework MLNRS based on neighborhood rough sets for automatic image annotation which considers the uncertainty of the mapping from visual feature space to semantic concepts space. Given a new instances, its neighbors in the training set are firstly identified. After that, based on the concept of upper and lower approximations of neighborhood rough sets, all possible labels of the given instance are found. Then, based on the statistical information gained from the label sets of the neighbors, maximum a posteriori (MAP) principle is utilized to determine the label set for the given instance. Experiments completed for three different image datasets show that MLNRS achieves more promising performance in comparison with to some well-known multi-label learning algorithms.

1. Introduction

Nowadays, we are faced with a plethora of images. As the images databases grow in size and number, a large body of research has been carried out to explore effective and efficient image retrieval (IR) approaches. In general, IR research pursuits can be categorized into three main areas [1]. In the first area, we are concerned with traditional text based image retrieval (TBIR) where the images are annotated manually by human. However, manual image annotation is time-consuming and expensive. Then the research focuses on the content-based image retrieval (CBIR) [2], where images are retrieved based on content features such as color, shape and texture. However, recent research has shown that there is a significant semantic gap between low-level content features and high-level semantic concepts. To bridge the semantic gap, the IR research has been shifted to semantic-level approaches [3,4], where the images are annotated with semantic labels based on the automatic image annotation (AIA) technology and then can be retrieved by keywords similar to TBIR.

With much effort devoted to semantic-level image retrieval, automatic image annotation has started drawing more attention [5–10]. Previous main approaches towards automatic image annotation modeled the learning problem as machine translation [5,6], correlations learning tasks [7,8]. In addition, some other researches regarded the automatic image annotation as multi-label classification problem [9,10] for the reason that an image could be related to more than one semantic concept simultaneously. However, the previously mentioned methods suffer from some limitations. First, most of these approaches

* Corresponding author at: Department of Computer Science and Technology, Tongji University, Shanghai 201804, PR China.
E-mail address: yuyingtongzhi@hotmail.com (Y. Yu).

0888-613X/$ – see front matter
http://dx.doi.org/10.1016/j.ijar.2013.06.003

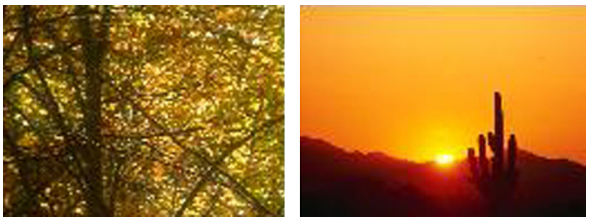

Fig. 1. Examples of multi-label images.

are based on a strong assumption that the visual similarity guarantees semantic similarity which is in conflict with the semantic gap. In fact, two images with the similar visual contents may correspond to quite different semantic concepts. Taking Fig. 1 for example, an autumn scene and a sunset scene may share some warm, bright colors. Therefore there is confusion between the two scenes in the feature space when the color features are used. Second, they ignore the impact from the limited quantity of the training instances which leads to inexact distribution of each class during the process of classification. For these reasons, there exists some uncertainty in the mapping of visual feature space to semantic concepts space.

Rough sets form a vehicle to deal with the ambiguous, vague, and uncertain knowledge. In order to reduce the bias between visual similarity and semantic similarity, we propose a multi-label classification framework based on the neighborhood rough sets, named MLNRS. By introducing the concept of upper and lower approximations of neighborhood rough set model, the framework can find all the possibly related labels of the given instance and then confirm the final labels according to the information of the neighborhood of the given instance. Empirical results using three image annotation data sets suggest that the proposed approach can improve the performance and reduce the training time compared with other standard multi-label algorithms.

The rest of this paper is organized as follow. Sections 2 and 3 provide background material on multi-label classification and neighborhood rough sets respectively. Section 4 introduces the proposed approach. Section 5 contains experimental results obtained by applying our algorithms and other multi-label learning algorithms to multi-labeled scene classification. Finally Section 6 concludes this work and points to future research direction.

2. Multi-label classification

Multi-label classification [11] is a supervised learning problem that an instance may be associated with multiple labels. This is different from the traditional single-label classification tasks [12]. In single-label classification tasks, an instance is only associated with one label and the classes are mutually exclusive by definition. Let Γ be the domain of the instances to be classified, Y be a finite set of labels and H be the set of the classifiers for $X \to Y$. The goal of the single-label classification task is to find the classifier $h \in H$ for each given instance $x \in X$ by maximizing the probability of $h(x) = y$, i.e. $y = \arg\max_i P(y_i|x)$. While in multi-label classification tasks, the base classes are non-mutually exclusive and may overlap in the selected feature space. As before, let X be the domain of the examples to be classified and L be the finite set of labels. Now let Y be a set of binary vectors, each of length $|L|$. Each vector $y \in Y$ indicates membership in the base classes in $L(1 = \text{member}, 0 = \text{non-member})$. H is the set of classifiers for $X \to Y$. The goal of the multi-label classification task is to output a classifier $h \in H$ which optimizes the specific evaluation metric (e.g., Hamming loss) for $h(x) = y$.

2.1. Learning algorithms

Multi-label classification algorithms can be categorized into 2 different groups [11]: (i) problem transformation methods and (ii) algorithms adaptation methods. The first group includes methods that are algorithm independent. They transform the multi-label problem into one or more single-label problems. The representative transformation method include binary relevance method (BR) [9,13,14], binary pairwise classification approach (PW) [15] and label combination or label power-set method (LC) [16,17]. BR transforms a multi-label problem into multiple binary problems. Each binary model is trained to predict one label; Classifier Chain (CC) [18] is a BR-based methods which can overcome the label-independent defect while maintaining the acceptable computational complexity of BR. PW can also be used to address multi-label problem, where a binary model is trained for each pair of labels; LC transforms a multi-label problem into a single-label problem by treating all label sets as atomic labels. The second group includes methods that extend specific learning algorithms in order to handle multi-label data directly. Well-known approaches include Adaboost [19], decision trees [20], and lazy methods [21–24]. Adaboost.MH and Adaboost.MR [19] are two extensions of Adaboost for multi-label classification. Comité et al. [20] extended the alternating decision tree learning algorithm for multi-label classification. ML-*k*NN [21] is an adaptation of the *k*NN lazy learning algorithm for multi-label data. In ML-*k*NN, in order to assign a set of label to an instance, a decision is made separately for each label by taking into account the number of neighbors containing the label to be assigned. ML-*k*NN fails to take into account the dependency between labels, while DML*k*NN [22] considers the dependencies between classes. In order to decide whether a particular label should be included among the unseen instance's labels, DML*k*NN takes into

account the numbers of different labels in the neighborhood, instead of considering only the number of neighbors having the label in question. Younes et al. [23] and Denoeux et al. [24] extended the D-S theory to manipulate multi-label data and proposed a new method EML–kNN for multi-label classification based on multi-label evidence-theoretic k-NN rule where the uncertain and imprecise is represented by D-S theory. MLNRS proposed in this paper is also a lazy method based algorithm, where the relation among labels is considered and the uncertain is represented by Rough sets.

2.2. Related studies

Nowadays, the research of the automatic image annotation (AIA) as well as text or music categories [13,15,25] that uses multi-label classification methods is receiving increased attention. Boutell et al. [9] present a framework which uses a new training strategy Cross-training to build classifiers and three classification criteria in testing. In addition, Boutell et al. [9] propose a generic evaluation metric that can be tailored to applications needing different error forgiveness. Nasiereling et al. [10] propose a framework that comprises an initial clustering phase that breaks the original training set into several disjoint clusters of data. It then trains a multi-label classifier from the data of each cluster. Given a new test instance, the framework first finds the nearest cluster and then applies the corresponding model. The empirical results show that the proposed approach can improve the performance and reduce the training time in the case of large number of label. More recently, a hidden-concept driven image annotation and label ranking algorithm (HDIALR) [26] is explored which conducts label propagation based on the similarity over a visually semantically consistent hidden-concepts space. Particularly, to our interests, M. Wang et al. [27], Tang et al. [28] and C. Wang et al. [29] handled the semantic gap problem in multi-label image annotation from different perspective. M. Wang et al. explore an image annotation approach based on weighted kNN multi-label classification [27], which focus on establishing the correlations between semantic concepts and low-level features. They develop an iterative solution to achieve 'optimal margins' in both of the semantic feature space and the visual feature space in order to reduce the semantic gap problem. An active learning method is explored to tackle the semantic gap [28]. The user interaction is regarded as an effective way to handle the semantic gap problem in image annotation and the size of the semantic gap of each concept is looked as an important factor that affects the performance of user feedback. It combines the semantic gap measures of concepts with the information minimization criterion to account for the effect of semantic gap in the sample selection strategy. C. Wang et al. [29] present a multi-label sparse coding framework for feature extraction and classification within the context of automatic image annotation. It answers the question that how to effectively measure the semantic similarity between two image ages with multiple objects/semantics. It claims that the semantic similarity of two images with overlapped labels should be measured in a reconstruction-based way rather than in a one-to-one way.

2.3. Evaluation measures

Performance evaluation of multi-label classification algorithm is different from that of classic single-label classification algorithm. We calculate a variety of multi-label evaluation metrics that are available in the Mulan library [30] including Hamming loss, coverage, one-error, ranking loss, average precision.

Before introducing the evaluation metrics, we first present the formal notation that we use throughout. Let $X = R^d$ denote the domain of input instances. The set $L = \{l_1, l_2, \ldots, l_m\}$ is a finite domain of possible labels. An instance is represented as a d-dimensions vector $x = [x^1, x^2, \ldots, x^d]$ $(x \in X)$. Each instance $x \in X$ is associated with a subset of L. The subset is denoted by an m-dimension vector $y = [y^1, y^2, \ldots, y^m]$, where $y^j = 1$ only if instance x has label l_j and 0 otherwise.

Let $T = \{(x_i, y_i) \mid i = 1, 2, \ldots, n\}$ be a training data set of n labeled instances and $D = \{(x_i, y_i) \mid i = 1, 2, \ldots, q\}$ be a testing set composed of q labeled instances. We use subscripts here to avoid ambiguity with the label dimension. Therefore y_i^j corresponds to the binary relevance of the jth label belonging to the ith instance.

(1) Hamming loss: evaluates how many times an instance–label pair is misclassified between the predicted set of labels y' and the ground-truth set of labels y, i.e. a label not belonging to the instance is predicted or a label belonging to the instance is not predicted. The smaller the value of Hamming loss, the better the performance.

$$hloss = 1 - \frac{1}{mq}\sum_{i=1}^{q}\sum_{j=1}^{m} 1_{y_i'^j = y_i^j}$$

(2) Coverage: evaluates how far we need, on average, to go down the ranked list of labels in order to cover all the relevant labels of the example. The smaller the value of coverage, the better the performance. $r_i(l)$ is a ranking function, which maps the predicted outputs for any $l \in y_i$.

$$cov = \frac{1}{q}\sum_{i=1}^{q}\max_{l \in y_i} r_i(l) - 1$$

(3) One-error: evaluates how many times the top-ranked label is not in the set of relevant labels of the instance. The smaller the value of one-error, the better the performance.

$$one\text{-}error = \frac{1}{q}\sum_{i=1}^{q}\delta\left(\arg\max_{l\in L} r_i(l)\right)$$

where

$$\delta(\lambda) = \begin{cases} 1 & \text{if } l \notin L \\ 0 & \text{otherwise} \end{cases}$$

(4) Ranking loss: expresses the number of times that irrelevant labels are ranked higher than relevant labels:

$$rloss = \frac{1}{q}\sum_{i=1}^{q}\frac{1}{|y_i||\overline{y_i}|}\left|\left\{l_a, l_b\colon r_i(l_a) > r_i(l_b), (l_a, l_b) \in y_i \times \overline{y}_i\right\}\right|$$

where $\overline{y}_i$ denotes the complementary set of y_i with respect to L.

(5) Average precision: evaluates the average fraction of labels ranked above a particular label $\lambda \in y_i$ which actually are in y_i. The bigger the value of *avgprec*, the better the performance.

$$avgprec = \frac{1}{q}\sum_{i=1}^{q}\frac{1}{|y_i|}\sum_{\lambda\in y_i}\frac{|\{\lambda' \in y_i\colon r_i(\lambda') \leqslant r_i(\lambda)\}|}{r_i(\lambda)} \tag{1}$$

3. Neighborhood rough sets

Rough set theory [31,32], introduced by Pawlak, has been developed as a tool to conceptualize, organize and analyze various types of data, in particular, to deal with inexact, uncertain or vague knowledge in applications [33,34] related to Artificial Intelligence.

Pawlak's rough set model is built on equivalence relations and equivalence classes. Equivalence relations are directly induced from nominal attributes based on the attribute values. However, some attributes in data are numeric in real-world applications. In order to extend the rough set model to support numeric attributes, Yao [35] and Hu et al. [36–38] proposed the neighborhood rough set model based on the neighborhood relations. Here we use the framework proposed by Hu et al., where nominal features generate equivalence information granules and numeric features produce neighborhood information granules, and then they are both used to approximate the decision class in the rough sets. In the following, we present several important concepts of neighborhood rough sets which are the basis of this paper.

Formally, let universe $IS = \langle U, A\rangle$ be the samples for classification, where U is the nonempty set of samples $\{x_1, x_2, \ldots, x_n\}$ and A denotes the nonempty set of attributes. To be more specific, $A = C \cup D$, where C is a set of condition attributes and D is a decision attribute.

Definition 1. (See [36].) Given arbitrary $x_i \in U$, $B \subseteq C$ and metric function, the neighborhood $\delta_B(x_i)$ of x_i in the subspace B is defined as

$$\delta_B(x_i) = \left\{x_j \mid \Gamma(x_i, x_j) \leqslant \tau, x_j \in U\right\}, \quad \text{where } \tau \geqslant 0$$

$\delta_B(x_i)$ is called a neighborhood information granule induced by attribute B and object x_i. There are some ways to define the metric function. One can define it with the fixed radius or define it with fixed k instances in the neighborhood, like k-nearest-neighbor. Here we select the second approach. Obviously, $\delta_B(x_i)$ is a subset of instances close to x_i and the family of neighborhood information granules $\{\delta_B(x) \mid x \in U\}$ forms a set of elemental concepts in the universe.

Definition 2. (See [36].) Given a neighborhood decision table $NDT = \langle U, C \cup D\rangle$, X is the subsets of instances with decisions attribute value ω_j $(j = 1, 2, \ldots, c)$, $\delta_B(x_i)$ $(x_i \in U,\ i = 1, 2, \ldots, n)$ is the neighborhood information granules including x_i and generated by attributes $B \subseteq C$. Then the lower and upper approximations of the decision class X with respect to attributes B are respectively defined as

$$\underline{R_B}X = \left\{x_i \mid \delta_B(x_i) \subseteq X,\ x_i \in U\right\}$$
$$\overline{R_B}X = \left\{x_i \mid \delta_B(x_i) \cap X \neq \emptyset,\ x_i \in U\right\}$$

For each decision class X, the neighborhood rough set model divides the universe into three subsets: (1) the positive region of decision class X, (2) the boundary region of decision class X and (3) the negative region of decision class X. The positive region, namely the lower approximation of the decision class X, is denoted by $POS_B(X) = \underline{R_B}X$. The positive region is a subset of instances whose neighborhoods consistently belong to decision class X. The boundary region of decision class X with respect to attributes B is defined as $BN_B(X) = \overline{R_B}X - \underline{R_B}X$. The boundary region is a subset of instances whose neighborhoods come from more than one decision class. Therefore, they are inconsistent and the instance in the boundary region is easy to be misclassified. The size of the boundary region reflects the degree of roughness of the decision. Usually we hope that the boundary region of the decision is as little as possible for decreasing uncertainty in decision. In addition,

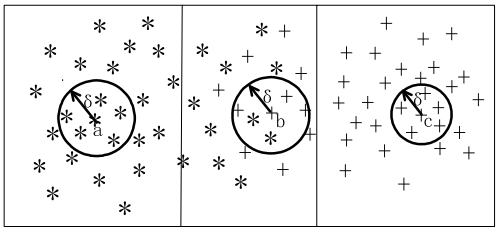

Fig. 2. An example of binary classification using neighborhood decision function.

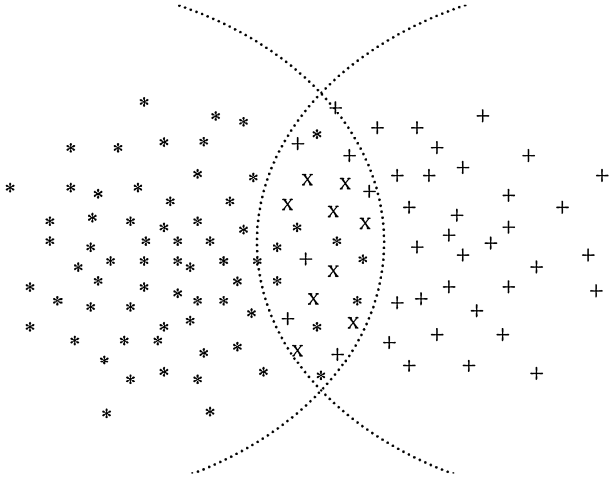

Fig. 3. Illustration of multi-label classification with two classes.

the negative region is denoted as $NEG_B(X) = U - \overline{R_B}X$. The negative region of decision class X is a subset of instances whose neighborhoods have no relation with decision class X.

Definition 3. (See [37].) Given $NDT = \langle U, C \cup D \rangle$, $x_i \in U$, $\delta(x_i)$ is the neighborhood of x_i and $P(\omega_j|\delta(x_i))$ $(j = 1, 2, \ldots, c)$ is the probability of instance x_i belonging to class ω_j. The neighborhood decision function of x_i is defined as $ND(x_i) = \omega_l$, if $P(\omega_l|\delta(x_i)) = \max_j P(\omega_j|\delta(x_i))$, where $P(\omega_j|\delta(x_i)) = n_j/k$, k is the number of instances in the neighborhood and n_j is the number of instances with decision ω_j in $\delta(x_i)$.

$ND(x_i)$ is the class assigned to x_i according to the classification probability in the neighborhood of x_i. Obviously, $ND(x_i) = \omega$ if x_i is located in the positive region of class ω, or $ND(x_i) \neq \omega$ if x_i is located in the negative region of class ω, or if x_i is located in the boundary region of class ω, we should compute the probability for giving a class label to x_i according to $\delta(x_i)$ and $ND(x_i) = \omega$ if majority of the instances in $\delta(x_i)$ belong to class ω.

Fig. 2 shows an example of binary classification using neighborhood decision function in a 2-D numerical space, where class $X1$ is marked with '$*$' and class $X2$ is marked with '$+$'. We associate a neighborhood to every object in the sample space such as a, b, c and the number of instances in the neighborhood is fixed at 5. It is easy to find that a and c are respectively located in the positive regions of class $X1(\delta(a) \subseteq X1)$ and class $X2(\delta(c) \subseteq X2)$, while b is located in the boundary region $(\delta(b) \cap X_1 \neq \emptyset$ and $\delta(b) \cap X_2 \neq \emptyset)$ and the majority of instances in $\delta(b)$ belong to X_2. According to the above definitions, $ND(a) = X_1$, $ND(c) = X_2$ and $ND(b) = X_2$.

4. Multi-label classification based on neighborhood rough sets

Contrary to single-label classification, the base classes in multi-label classification are non-mutually exclusive and may overlap by definition and the objects could be associated with a set of labels simultaneously. Generally speaking, the object with multiple labels is located in the overlapped region. Fig. 3 illustrates a binary multi-label classification task. As before, two classes X_1 and X_2 are denoted by '$*$' and '$+$' respectively. Examples simultaneously belonging to '$*$' and '$+$' classes are denoted by 'x'. It can be seen from Fig. 3 that the instances in the overlapped region have one or two labels while the instances in non-overlapped region are only related with one label. Because some instances belong to multiple classes, the multi-label classification problem can be regarded as an inconsistent decision problem that two objects which have the same condition values may not have the same decision value. The neighborhood rough set theory introduced above aims at the single-label consistent decision problem. In order to predict the labels of instances in inconsistent decision problem, we extend the neighborhood decision function for multi-label classification as follows.

Definition 4. Given a multi-label neighborhood decision table $NDT = \langle U, C \cup D \rangle$, $x_i \in U$, $\delta(x_i)$ is the neighborhood of x_i. For each label l_j $(j = 1, 2, \ldots, m)$, $P(l_j^1|E_{l_j}^\lambda)$ is the probability of instance x_i having label l_j and $P(l_j^0|E_{l_j}^\lambda)$ is probability of instance x_i having no label l_j. $P(l_j^1|E_{l_j}^\lambda)$ is defined as

$$P\left(l_j^1|E_{l_j}^\lambda\right) = \begin{cases} 1 & \text{if } |\delta_{l_j}(x_i)| = k \\ 0 & \text{if } |\delta_{l_j}(x_i)| = 0 \\ \xi & \text{otherwise} \end{cases}$$

$|\delta_{l_j}(x_i)|$ denotes the number of instances with label l_j in $\delta_{l_j}(x_i)$ and $P(l_j^0|E_{l_j}^\lambda) = 1 - P(l_j^1|E_{l_j}^\lambda)$. The multi-label neighborhood decision of x_i for label l_j is defined as $ND_{l_j}(x_i) = l_j^1$ if $P(l_j^1|E_{l_j}^\lambda) = \arg\max_{b\in\{0,1\}} P(l_j^b|E_{l_j}^\lambda)$. k is the number of instances in $\delta(x_i)$. $E_{l_j}^\lambda$ means the event that the instances with label l_j in $\delta(x_i)$ account for λ proportion of all kinds of instances in $\delta(x_i)$ and $\lambda = |\delta_{l_j}(x_i)| / \sum_{\eta=1,\ldots,m}(|\delta_{l_\eta}(x_i)|)$. How to calculate ξ when $0 < |\delta_{l_j}(x_i)| < k$ will be described later.

Using the Bayesian rule, the probability for determining whether the label l_j belongs to instance x_i can be written as follows:

$$y_i^j = \arg\max_{b\in\{0,1\}} P\left(l_j^b|E_{l_j}^\lambda\right) = \arg\max_{b\in\{0,1\}} \frac{P(l_j^b)P(E_{l_j}^\lambda|l_j^b)}{P(E_{l_j}^\lambda)}$$

$P(E_{l_j}^\lambda)$ is the same for l_j^1 and l_j^0, so the y_i^j is actually determined by $\arg\max_{b\in\{0,1\}} P(l_j^b)P(E_{l_j}^\lambda|l_j^b)$. The priori probability $P(l_j^b)$ and conditional probability $P(E_{l_j}^\lambda|l_j^b)$ can be directly estimated from the training instances based on frequency counting.

In Definition 4, the multi-label classification task is transferred into m binary classification tasks and the neighborhood decision function is used in each binary classification task respectively. Meanwhile, the correlation and non-exclusion among the labels are considered during the process of classification. Obviously, for each label l_j, $ND_{l_j}(x_i) = l_j^1$ if x_i is located in the positive region of class l_j; $ND_{l_j}(x_i) = l_j^0$ if x_i is located in the negative region of class l_j; Moreover, if x_i is located in the boundary region of class l_j, we should compute the probability of label l_j for x_i according to the statistics information from samples.

Algorithm 1 gives the complete description of MLNRS according to Definition 4:

Algorithm 1 Multi-label classification based on neighborhood rough sets (MLNRS).

Input: T, k, t, s, ϕ
Output: y

// Computing the prior probability $P(l_j^1)$.
1: for each $l_j \in L$
2: $P(l_j^1) = (\sum_{i=1}^n y_i^j + s)/(n + s \times 2)$
3: $P(l_j^0) = 1 - P(l_j^1)$
//Counting the number of training instances with specific λ.
4: for each training instance $x_i \in T$
5: compute the neighborhood $\delta(x_i)$
6: for $l_j = l_1, \ldots, l_m$
7: count proportion λ for l_j
8: if $(y_i^j == 1)$ $w_j^\lambda = w_j^\lambda + 1$
9: else $w'^\lambda_j = w'^\lambda_j + 1$
//Estimating the conditional probability $P(E_l^\lambda|l_j^b)$ for each label.
10: for $l_j = l_1, \ldots, l_m$
11: count the number of the different λ and save it in the α
12: for each λ
13: $P(E_{l_j}^\lambda/l_j^1) = (w_j^\lambda + s)/(\sum w_j^\lambda + s \times m)$
14: $P(E_{l_j}^\lambda/l_j^0) = (w'^\lambda_j + s)/(\sum w'^\lambda_j + s \times m)$
15: get two fitting curves from $P(E_{l_j}^\lambda/l_j^1)$ and $P(E_{l_j}^\lambda/l_j^0)$ with degree ϕ
// Computing the posterior probability of the testing instance t
16: compute the neighborhood $\delta(t)$
17: for $l_j = l_1, \ldots, l_m$
18: count the number β of instances with label l_j in $\delta(t)$
19: if $(\beta = k)$ $ND_{l_j}(t) = l_j^1$
20: else if $(\beta = 0)$ $ND_{l_j}(t) = l_j^0$
21: else count proportion λ for l_j
22: if $P(l_j^1|E_{l_j}^\lambda) = \arg\max_{b\in\{0,1\}} P(l_j^b|E_{l_j}^\lambda)$ $ND_{l_j}(t) = l_j^1$ else $ND_{l_j}(t) = l_j^0$
23: if $ND_{l_j}(t) = l_j^1$ $y_t^j = 1$ else $y_t^j = 0$

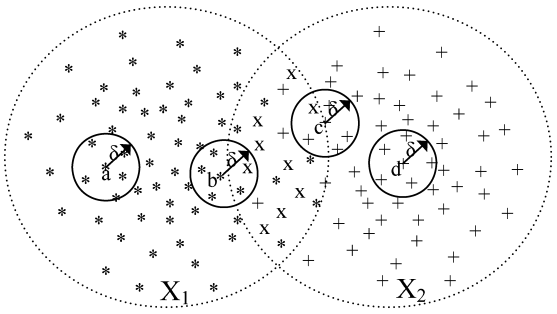

Fig. 4. Illustration of multi-label classification using MLNRS.

Here, we still use the formal notation introduced in Section 2.3. The input arguments include T, k, t, s and ϕ. T is the training set and t denotes the testing instance. k denotes the number of the nearest neighbors of the given instance. ϕ means the degree of the polynomial fitting function. Furthermore, in order to avoid the zero times in frequency counts, a smoothing parameter s is used to control the strength of uniform prior (s is set to be 1 which yields the Laplace smoothing). $y = [y^1, y^2, \ldots, y^m]$ is the predicted result of testing instance and y^j is the value on label l_j.

As shown in Algorithm 1, based on the training set, steps from 1 to 3 estimate the prior probabilities $P(l_j^1)$ and $P(l_j^0)$. Steps from 4 to 9 count the number of training instances with specific λ, where vector w_j^λ ($j \in [1, m], \lambda \in [0, 1]$) saves the number of instances with label l_j whose neighbors with label l_j exactly account for proportion λ. Corresponding, w'^λ_j saves the number of instances without label l_j whose neighbors with label l_j account for proportion λ. Steps from 10 to 15 estimate the conditional probabilities $P(E_l^\lambda | l_j^b)$ for each label l_j at specific λ. Steps 13 and 14 only get some data points about $P(E^\lambda_{l_j}/l_j^1)$ and $P(E^\lambda_{l_j}/l_j^0)$ with different λ. In order to infer probability values where no data are available, two probability curves are constructed to fit the known data points. Here, Polynomial Fitting is used. Finally, using the Bayesian rule, steps from 16 to 23 compute the outputs for a given instance based on the estimated probabilities.

In order to explain MLNRS clearly, we use an example (Fig. 4) to illustrate how to classify a multi-label instance using MLNRS. In Fig. 4, there are two classes X_1 and X_2 respectively marked by '$*$' and '$+$' in a 2-D space. Instances belonging to both X_1 and X_2 simultaneously are denoted by 'X'. For convenience, we assume that the distribution of two classes is circular. Firstly, we associate a neighborhood to each instance, such as a, b, c, d and the number of instances in neighborhood is fixed at five. Then the labels are predicted by MLNRS. All instances in $\delta(a)$ only have one label X_1, so a is only located in the positive region of class X_1 and it is non-overlapped. The label X_1 should be assigned to a. Instance d is similar and d should be assigned label X_2. As for instances b and c which are in the boundary region for their impure neighborhood. Among the five instances in $\delta(b)$, all of them have label $X_1(|\delta_{X_1}(b)| = 5)$. So b is located in the positive region of X_1 and $ND_{X_1}(t) = X_1^1$, while in $\delta(b)$, there are only one instance belonging to class $X_2(|\delta_{X_2}(b)| = 1$ and $\lambda = 1/6)$. So b is in the boundary region of X_2 and the probability for label X_2 will be estimated according to λ. The analysis of c is similar to that of b.

5. Experiments

In this section, we will empirically evaluate the proposed methods by comparing MLNRS with other multi-label algorithms. In this paper, we will compare the MLNRS with the binary relevance method (BR) [11], the classifier chains algorithm (CC) [18], the random k label-set method for multi-label classification RAkEL [16] and back-propagation multi-label learning (BPMLL) [39] learner, which are all well-known modern high-performing multi-label learning algorithms. Furthermore, we will show the influence of parameters k and ϕ used in MLNRS.

5.1. Datasets

We conducted comparative experiments using these algorithms on three data sets containing multi-label multimedia objects. The first one, *scene* [9] is a benchmark for image classification containing 2407 natural scene images. The second one, *Corel*5*k* [40], is based on 5000 Corel images, which is used for the ECCV 2002 paper. The last one, *Corel*16*k*001 [41], is produced from the first (001) subset of the data accompanying [41] and it contains 13 766 images.

Table 1 shows multi-label datasets and their associated properties including the number of instances, the number of attributes and the number of labels for each data set. Moreover, the label cardinality and density of each data set are also displayed. Label cardinality is a standard measure of the average number of labels of the instances in the given dataset. Label density is the average number of labels of the instances in the given dataset divided by the number of labels.

Table 1
Multi-label datasets used for experiments and associated properties.

Name	Instances	Attribute	Labels	Cardinality	Density	Train	Test
Scene	2407	294	6	1.074	0.179	1211	1196
Corel5k	5000	499	374	3.522	0.009	4500	500
Corel16k001	13 766	500	153	2.867 ± 0.033	0.018 ± 0.001	5188	1744

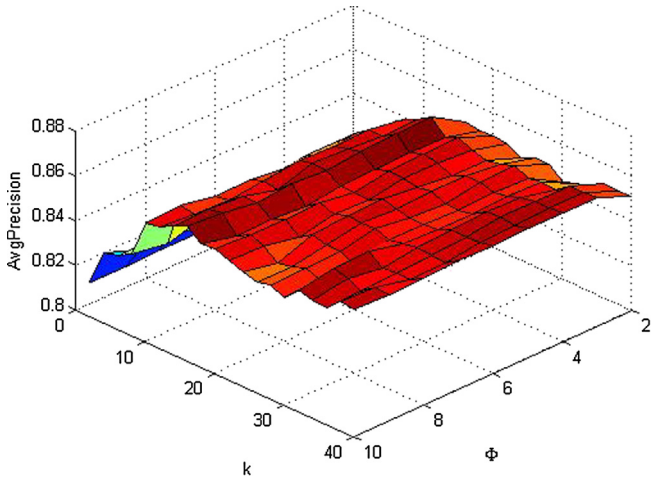

Fig. 5. Average precision varies with k and ϕ over *Scene*.

5.2. Experimental setting

The implementation of the compared algorithms comes from the open source Mulan library [30], which is based on the open source Weka library [42]. BR, RA*k*EL and CC use the C4.5 decision tree learning algorithm for training the underlying single-label classifier. As recommended in [39], BPMLL is run with 0.05 learning rate, 100 epochs and the number of hidden units equal to 20% of the input units. MLNRS is implemented by Matlab and the polynomial function is used as a fitting function. Here the neighborhood is defined by the number of neighbors k rather than radius. We evaluate all multi-label learning algorithms based on ten-fold cross-validation on the datasets mentioned above and use the average results to display and test for significance. The exception is that for some parameter analysis, the cross-validation is too intensive. So the analysis of influence of parameters is conducted on the train/test split datasets which will not affect the analysis. These splits are shown in Table 1 where *train* is the training set and *test* is the testing set. They are all provided by the Mulan library.

All experiments were run on a workstation equipped with a 3.0 GHz processor and 6.0 G memory.

5.3. Results and discussion

Firstly, we empirically study the impact of parameters k and ϕ based on split datasets. The number of nearest neighbors k varies from 2 to 40 with step 2 and the degree of the polynomial fitting function ϕ varies from 2 to 10 with step 1.

It can be seen from Figs. 5–19 that the values of the evaluation metrics change along with the change of parameters k and ϕ. However, when the parameters k and ϕ change to a certain range, such as $k > 5$, $9 > \phi > 2$, the performance is relatively stable. This indicates that we can assign the parameters with values in wide arranges. Then we can select relatively smaller values for parameters k and ϕ to reduce the calculation but maintain acceptable performance.

Furthermore, we compare MLNRS with various state-of-art multi-label algorithms including BR, RA*k*EL, BPMLL and CC based on ten-fold cross-validation. Results of ten-fold cross-validation in terms of *Hamming loss*, *average precision*, *coverage*, *one-error* and *ranking loss* are shown in Tables 2–4. The value following "±" gives the standard deviation and the best result on each metric is shown in bold face. The number of the nearest neighbors is set to $k = 10$. The three degree polynomial function is used to construct a fitting curve. The higher degree of the polynomial function maybe result in more exact match, but we actually prefer the effect of averaging out questionable data points in a sample, rather than distorting the curve to fit them exactly. The three datasets are quite different in the dataset properties. We will study the results of the MLNRS algorithm separately for each data set.

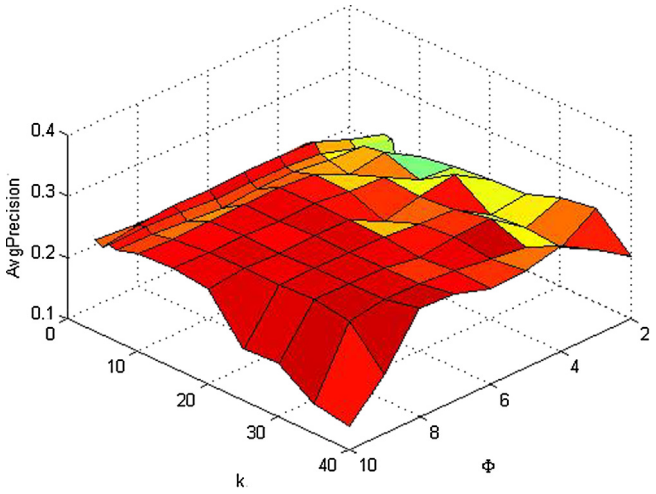

Fig. 6. Average precision varies with k and ϕ over *Corel5k*.

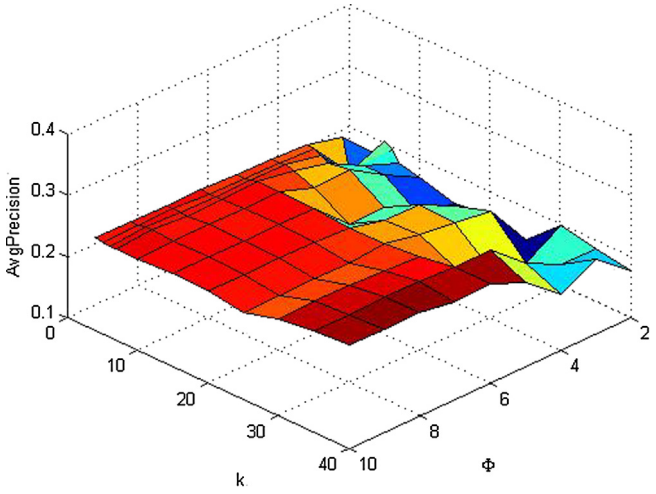

Fig. 7. Average precision varies with k and ϕ over *Corel16k001*.

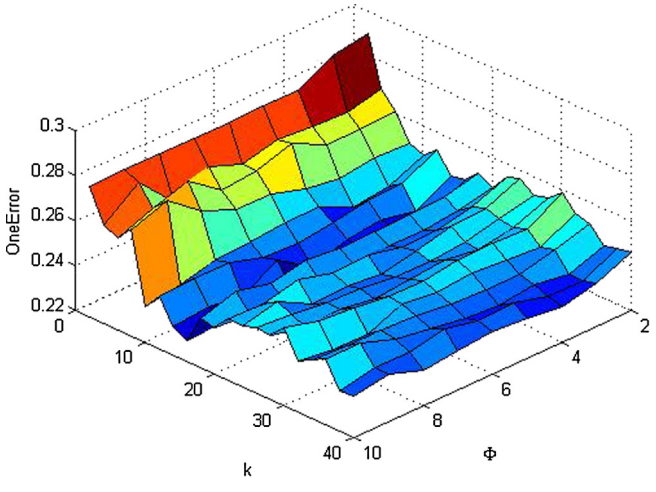

Fig. 8. One error varies with k and ϕ over *Scene*.

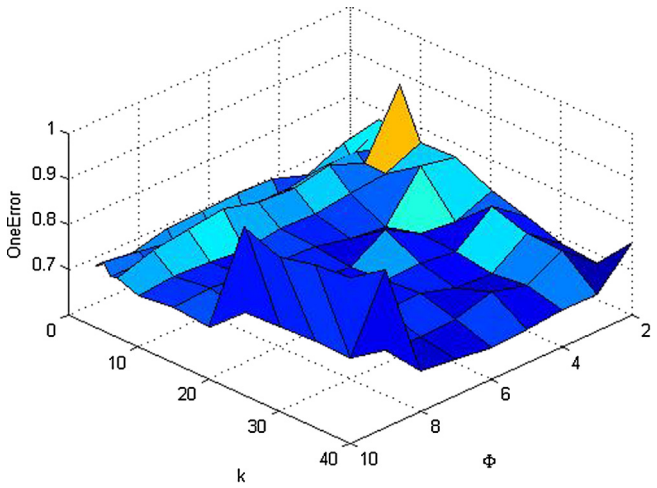

Fig. 9. One error varies with k and ϕ over *Corel5k*.

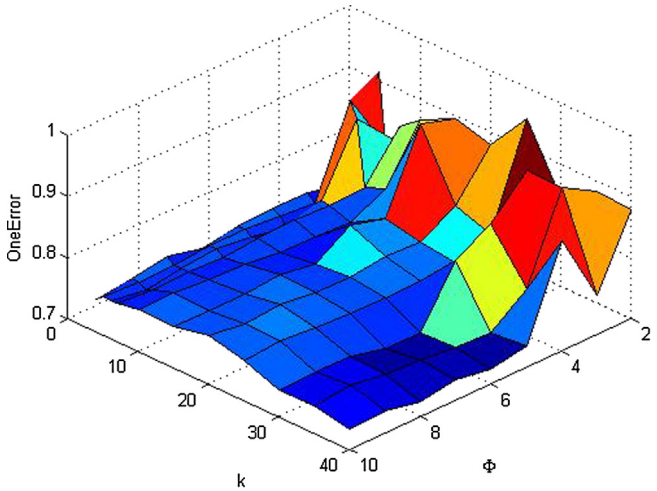

Fig. 10. One error varies with k and ϕ over *Corel16k001*.

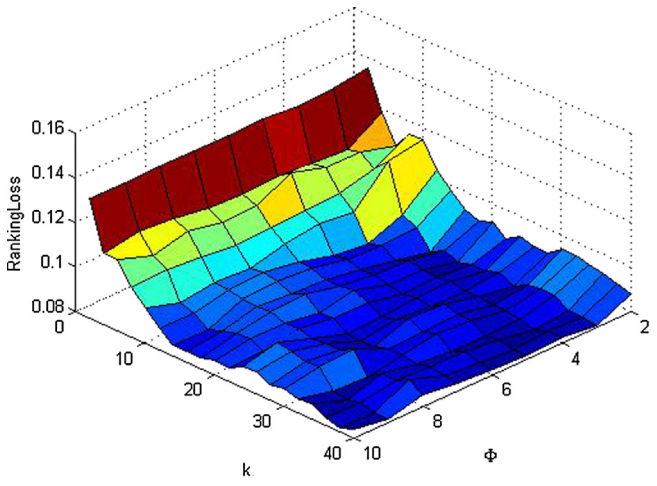

Fig. 11. Ranking loss varies with k and ϕ over *Scene*.

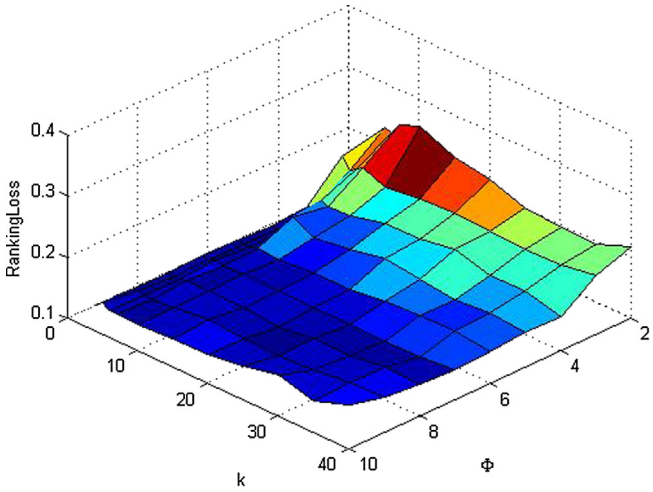

Fig. 12. Ranking loss varies with k and ϕ over *Corel5k*.

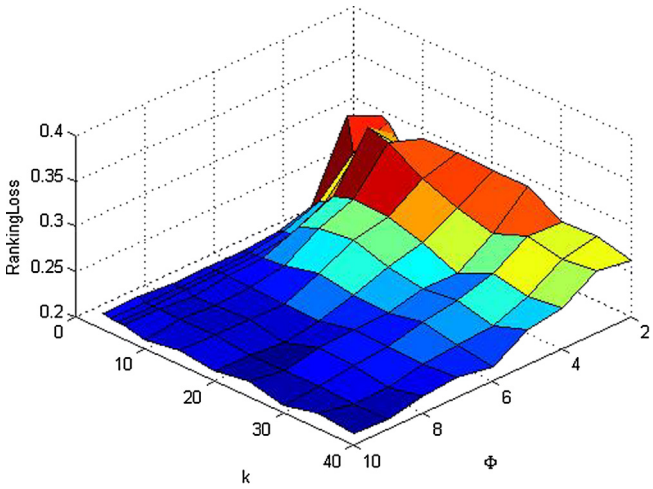

Fig. 13. Ranking Loss varies with k and ϕ over *Corel16k001*.

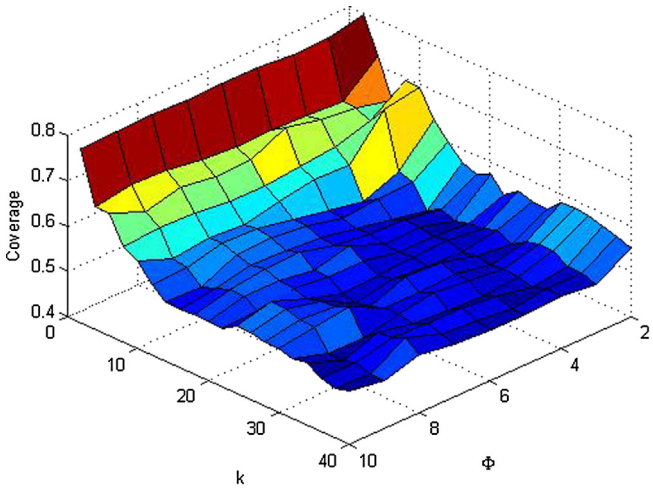

Fig. 14. Coverage varies with k and ϕ over *Scene*.

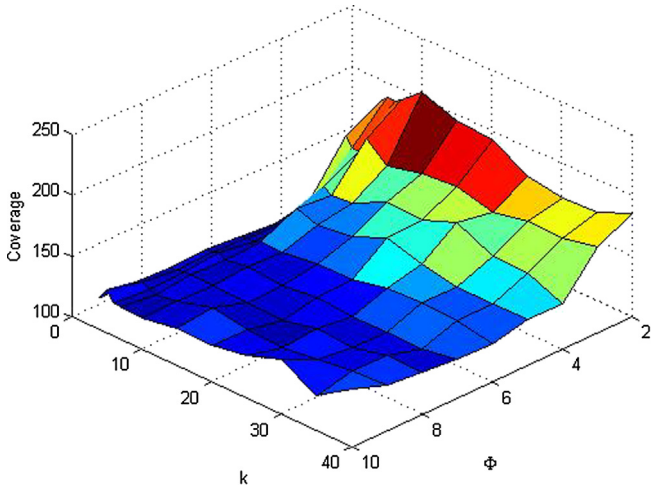

Fig. 15. Coverage varies with k and ϕ over *Corel5kx*.

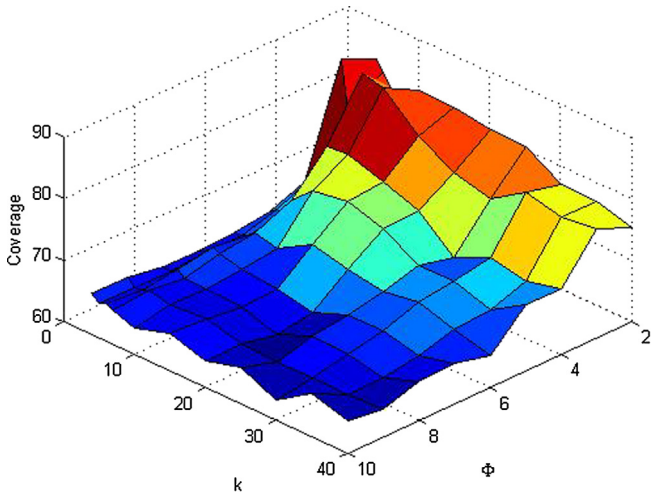

Fig. 16. Coverage varies with k and ϕ over *Corel16k001*.

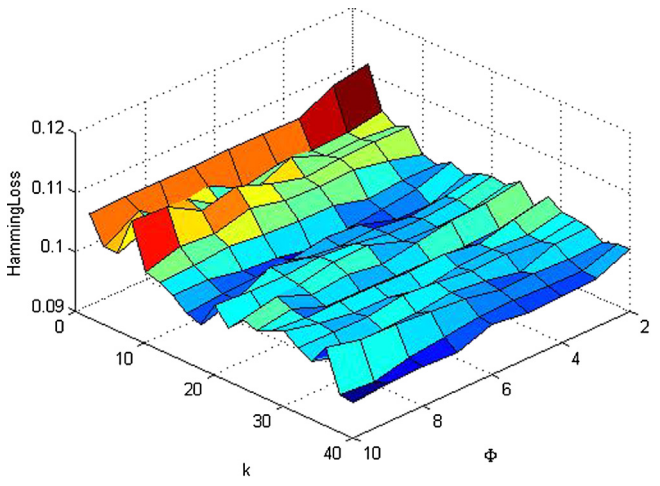

Fig. 17. Hamming loss varies with k and ϕ over *Scene*.

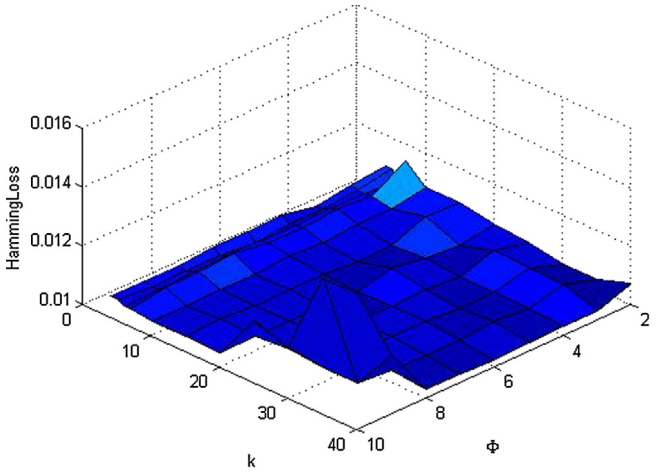

Fig. 18. Hamming loss varies with k and ϕ over *Corel5k*.

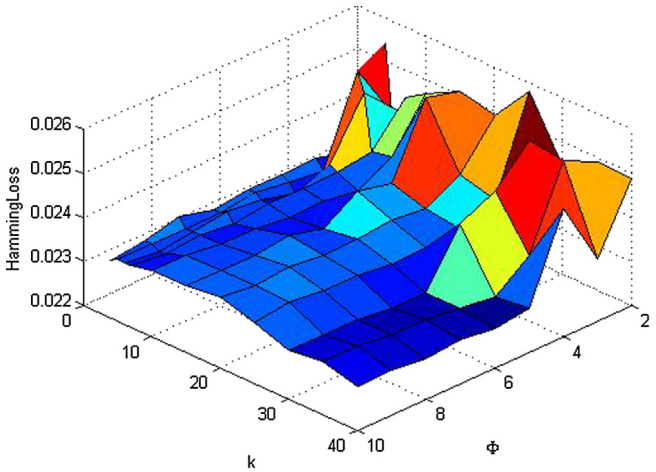

Fig. 19. Hamming loss varies with k and ϕ over *Corel16k001*.

Table 2
MLNRS vs. other multi-label algorithms-predictive performance over the dataset *Scene*.

Performance	BR	RA*k*EL	BPMLL	CC	MLNRS
hloss	0.1368 ± 0.0078	0.1012 ± 0.0075	0.2667 ± 0.0508	0.1444 ± 0.0164	$\mathbf{0.0905 \pm 0.0053}$
avgprec	0.7109 ± 0.0283	0.8379 ± 0.0156	0.6852 ± 0.0235	0.7176 ± 0.0354	$\mathbf{0.8676 \pm 0.0144}$
cov	1.3345 ± 0.1501	0.5862 ± 0.0593	0.9405 ± 0.0855	1.3504 ± 0.2002	$\mathbf{0.4926 \pm 0.0548}$
one-error	0.4138 ± 0.0402	0.2663 ± 0.0258	0.5450 ± 0.0381	0.3914 ± 0.0453	$\mathbf{0.2293 \pm 0.0246}$
rloss	0.2465 ± 0.0296	0.0999 ± 0.0121	0.1714 ± 0.0165	0.3914 ± 0.0453	$\mathbf{0.0809 \pm 0.0108}$

Table 3
MLNRS vs. other multi-label algorithms-predictive performance over the dataset *Corel5k*.

Performance	BR	RA*k*EL	BPMLL	CC	MLNRS
hloss	0.0098 ± 0.0001	$\mathbf{0.0097 \pm 0.0001}$	0.5547 ± 0.0213	0.0099 ± 0.0001	0.0107 ± 0.0001
avgprec	0.2494 ± 0.0093	0.1075 ± 0.0080	0.0563 ± 0.0097	0.2364 ± 0.0102	$\mathbf{0.2456 \pm 0.0096}$
cov	$\mathbf{126.9872 \pm 3.945}$	336.0374 ± 2.6687	169.0732 ± 4.6338	165.3946 ± 5.8193	131.5248 ± 5.5082
one-error	$\mathbf{0.6964 \pm 0.0174}$	0.7734 ± 0.0201	0.9974 ± 0.0025	0.7076 ± 0.0172	0.7442 ± 0.0154
rloss	$\mathbf{0.1472 \pm 0.0043}$	0.6565 ± 0.0116	0.2273 ± 0.0096	0.1869 ± 0.0083	0.1530 ± 0.0052

Table 2 shows that MLNRS performs fairly well over *Scene* where no algorithm outperforms MLNRS on all the evaluation criteria. Table 3 shows the performance comparison over *Corel5k*. With the enormous increasing of the amount of instances and labels, MLNRS still can performs well compared to other multi-label algorithms. It shows that MLNRS has some scalability. Table 4 shows the performance comparison over *Corel16k001*. We can see that the MLNRS performs well

Table 4
MLNRS vs. other multi-label algorithms-predictive performance over the dataset *Corel16k001*.

Performance	BR	RA*k*EL	BPMLL	CC	MLNRS
hloss	0.0197 ± 0.0001	0.0196 ± 0.0001	0.6730 ± 0.1085	0.0206 ± 0.0003	$\mathbf{0.0187 \pm 0.0002}$
avgprec	$\mathbf{0.2892 \pm 0.0045}$	0.1656 ± 0.0036	0.0444 ± 0.0045	0.2677 ± 0.0068	0.2788 ± 0.0069
cov	56.1436 ± 1.0587	100.6580 ± 0.8441	92.9266 ± 2.4173	64.1887 ± 0.8631	$\mathbf{51.2341 \pm 1.4363}$
one-error	$\mathbf{0.7035 \pm 0.0098}$	0.7582 ± 0.0101	0.9935 ± 0.0031	0.7473 ± 0.0182	0.7269 ± 0.0138
rloss	0.1883 ± 0.0042	0.3963 ± 0.0038	0.4029 ± 0.0171	0.2194 ± 0.0036	$\mathbf{0.1727 \pm 0.0047}$

on the evaluation measures except for average precision and *one-error*. It is may be due to the sparse distribution of instances which affects the estimation of conditional probability. In a word, compared with other multi-label algorithms, MLNRS yields superior predictive performance overall. It performs best on more evaluation measures than any other methods. The reason for its good performance is that by introducing the upper and lower approximation of neighborhood rough sets to estimate the location of testing object, MLNRS can eliminate the disturbance of irrelevant labels and give confidence to the absolutely relevant labels. As for the instance located in the boundary region, MLNRS predicts the probabilities of its labels by considering the correlation among labels, which is closer to the reality. On the other hand, we also find that MLNRS performs worse than the simplest multi-label method BR on some datasets, such as *Corel5k*. It is because MLNRS is based on the neighborhood and the process of finding nearest neighbors is affected by the high dimensional feature space. So in order to improve the performance of MLNRS, the future work will focus on the dimensionality reduction of multi-label dataset.

As for the time complexity, it can be calculated in two stages. During the process of training, we need to calculate the distance between arbitrary pair of instances and find the k nearest neighbors for each training instance. So the time complexity of training is $O(n^2 \times k)$. During the process of testing, a testing instance need to find its k nearest neighbors among n training instances and the time complexity is $O(n \times k)$.

6. Conclusion

This paper presented a novel method for multi-label classification algorithm. We derived this method from the neighborhood rough set model, which we argued has many advantages over more sophisticated methods currently in use. By introducing the lower approximation and upper approximation of rough sets, our method obtains high predictive performance compared to a variety of complex methods while maintaining low computational complexity.

Acknowledgements

The authors would like to thank the Editors for their kindly help and the anonymous referees for their valuable comments and helpful suggestions. The work is partially supported by the National Natural Science Foundation of China (Serial Nos. 61075056, 61273304, 61075056, 61103067, 61202170), the Fundamental Research Funds for the Central Universities and the State Scholarship Fund of China (File No. 201206260047).

References

[1] D. Zhang, M.M. Islam, G. Lu, A review on automatic image annotation techniques, Pattern Recognition 45 (1) (2012) 346–362.
[2] M.S. Lew, N. Sebe, C. Djeraba, R. Jain, Content-based multimedia information retrieval: State of the art and challenges, ACM Transaction on Multimedia Computing, Communications and Applications 2 (1) (2006) 1–19.
[3] C. Wang, L. Zhang, H.J. Zhang, Learning to reduce the semantic gap in web image retrieval and annotation, in: Proceedings of the 31st International ACM SIGIR Conference on Research and Development in Information Retrieval, Singapore, 2008, pp. 355–362.
[4] Y. Liu, D. Zhang, G. Lu, Region-based image retrieval with high-level semantics using decision tree learning, Pattern Recognition 41 (8) (2008) 2554–2570.
[5] P. Duygulu, K. Barnard, J.F.G. de Fretias, D.A. Forsythetc, Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary, in: Proceedings of the European Conference on Computer Vision, 2005, pp. 97–112.
[6] F. Kang, F. Jin, Symmetric statistical translation models for automatic image annotation, in: MProc. of SIAM Conf. on Data Mining, New Port, Beach, CA, April, 2005, pp. 21–23.
[7] V. Lavrenko, R. Manmatha, J. Jeon, A model for learning the semantics of pictures, in: Proceedings of the 16th Conference on Advances in Neural Information Processing Systems, NIPS, 2003.
[8] J. Liu, B. Wang, M. Li, Z. Li, W. Ma, H. Lu, S. Ma, Dual Cross-Media Relevance Model for Image Annotation, ACM SIGMM, 2007, pp. 605–614.
[9] M.R. Boutell, J. Luo, X. Shen, C.M. Brown, Learning multi-label scene classification, Pattern Recognition 37 (9) (2004) 1757–1771.
[10] G. Nasierding, G. Tsoumakas, Clustering based multi-label classification for image annotation and retrieval, in: IEEE International Conference on Systems, Man and Cybernetics, SMC, 2009, pp. 4514–4519.
[11] G. Tsoumakas, I. Katakis, Multi label classification: An overview, International Journal of Data Warehousing and Mining 3 (3) (2007) 1–13.
[12] R. Duda, R. Hart, D. Stork, Pattern Classification, 2nd edition, Wiley, New York, 2001.
[13] S. Godbole, S. Sarawagi, Discriminative methods for multi-labeled classification, in: Eighth Pacific–Asia Conference on Knowledge Discovery and Data Mining, PAKDD '04, Springer, Berlin, 2004, pp. 22–30.
[14] M.L. Zhang, Z.H. Zhou, A k-nearest neighbor based algorithm for multi-label classification, in: IEEE International Conference on Granular Computing, GnC '05, IEEE, New York, 2005, pp. 718–721.
[15] J. Fürnkranz, E. Hüllermeier, E. Loza Mencía, K. Brinker, Multilabel classification via calibrated label ranking, Machine Learning 73 (2) (2008) 133–153.

[16] G. Tsoumakas, I.P. Vlahavas, Random k-labelsets: An ensemble method for multilabel classification, in: 18th European Conference on Machine Learning, ECML '07, Springer, Berlin, 2007, pp. 406–417.
[17] J. Read, B. Pfahringerand, G. Holmes, Multi-label classification using ensembles of pruned sets, in: Eighth IEEE International Conference on Data Mining, ICDM'08, IEEE, New York, 2008, pp. 995–1000.
[18] J. Read, B. Pfahringerand, G. Holmes, E. Frank, Classifier chains for multi-label classification, in: 20th European Conference on Machine Learning, ECML '09, Berlin, 2009, pp. 254–269.
[19] R.E. Schapire, Y. Singer, Boostexter: A boosting-based system for text categorization, Machine Learning 39 (2/3) (2000) 135–168.
[20] F.D. Comité, R. Gilleron, M. Tommasi, Learning multi-label alternating decision trees from texts and data, in: Lecture Notes in Computer Science, vol. 2734, 2003, pp. 251–274.
[21] M.L. Zhang, Z.H. Zhou, ML-*k*NN: A lazy learning approach to multi-label learning, Pattern Recognition 40 (7) (2007) 2038–2048.
[22] Z. Younes, F. Abdallah, T. Denoeux, H. Snoussi, A dependent multi-label classification method derived from the *k*-nearest neighbor rule, EURASIP Journal on Advances in Signal Processing 2011 (2011) 14 pp, http://dx.doi.org/10.1155/2011/645964, Article ID 645964.
[23] Z. Younes, F. Abdallah, T. Denoeux, An evidence-theoretic *k*-nearest neighbor rule for multi-label classification, in: Proceedings of the 3rd International Conference on Scalable Uncertainty Management, in: LNAI, vol. 5785, Springer-Verlag, Washington, DC, USA, 2009, pp. 297–308.
[24] T. Denoeux, Z. Younes, F. Abdallah, Representing uncertainty on set-valued variables using belief functions, Artificial Intelligence 174 (7–8) (2010) 479–499.
[25] J.C. Wang, H.M. Wang, S.D. Lin, Cost-sensitive multi-label learning for audio tag annotation and retrieval, IEEE Transactions on Multimedia 13 (2011) 518–528.
[26] T. Li, S. Yan, Hidden-concept driven multilabel image annotation and label ranking, IEEE Transactions on Multimedia 14 (1) (2012) 199–210.
[27] M. Wang, X. Zhou, T.S. Chua, Automatic image annotation via local multi-label classification, in: Proceedings of the 2008 International Conference on Content-Based Image and Video Retrieval, CIVR '08, 2008, pp. 17–26.
[28] J. Tang, Z.J. Zha, D. Tao, T.S. Chua, Semantic-gap-oriented active learning for multilabel image annotation, IEEE Transactions on Image Processing 21 (4) (2012) 2354–2360.
[29] C. Wang, S. Yan, L. Zhang, H.J. Zhang, Multi-label sparse coding for automatic image annotation, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2009, pp. 1643–1650.
[30] G. Tsoumakas, I. Katakis, I. Vlahavas, Mining multi-label data, in: O. Maimon, L. Rokach (Eds.), Data Mining and Knowledge Discovery Handbook, 2nd edition, Springer, 2010.
[31] Z. Pawlak, Rough Sets: Theoretical Aspects of Reasoning about Data, vol. 9, Kluwer Academic Publishers, Dordrecht, 1991.
[32] H. Zhang, J. Zhou, D. Miao, C. Gao, Bayesian rough set model: A further investigation, International Journal of Approximate Reasoning 53 (4) (2012) 541–557.
[33] K. Kaneiwa, Y. Kudo, A sequential pattern mining algorithm using rough set theory, International Journal of Approximate Reasoning 52 (6) (2011) 881–893.
[34] P. Maji, S. Pau, Rough set based maximum relevance-maximum significance criterion and gene selection from microarray data, International Journal of Approximate Reasoning 52 (3) (2011) 408–426.
[35] Y.Y. Yao, Relational interpretations of neighborhood operators and rough set approximation operators, Information Sciences 111 (1998) 239–259.
[36] Q. Hu, J. Liu, D. Yu, Mixed feature selection based on granulation and approximation, Knowledge-Based Systems 21 (2008) 294–304.
[37] Q. Hu, W. Pedrycz, D. Yu, et al., Selecting discrete and continuous features based on neighborhood decision error minimization, IEEE Transactions on Systems, Man, and Cybernetics, Part B 40 (1) (2010) 137–150.
[38] Q. Hu, D. Yu, J. Liu, et al., Neighborhood rough set based heterogeneous feature subset selection[J], Information Sciences 178 (18) (2008) 3577–3594.
[39] M.L. Zhang, Z.H. Zhou, Multi-label neural networks with applications to functional genomics and text categorization, IEEE Transactions on Knowledge and Data Engineering 18 (10) (2006) 1338–1351.
[40] P. Duygulu, K. Barnard, N. de Freitas, D. Forsyth, Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary, in: 7th European Conference on Computer Vision, 2002, pp. 97–112.
[41] K. Barnard, P. Duygulu, N. de Freitas, D. Forsyth, D. Blei, M.I. Jordan, Matching words and pictures, Journal of Machine Learning Research 3 (2003) 1107–1135.
[42] I.H. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann, 2005.