Combining Instance-Based Learning and Logistic Regression for Multilabel Classification

Weiwei Cheng and Eyke Hüllermeier
Department of Mathematics and Computer Science
University of Marburg, Germany
{cheng,eyke}@mathematik.uni-marburg.de

Machine Learning 76(2-3):211-235, 2009

Abstract

Multilabel classification is an extension of conventional classification in which a single instance can be associated with multiple labels. Recent research has shown that, just like for conventional classification, instance-based learning algorithms relying on the nearest neighbor estimation principle can be used quite successfully in this context. However, since hitherto existing algorithms do not take correlations and interdependencies between labels into account, their potential has not yet been fully exploited. In this paper, we propose a new approach to multilabel classification, which is based on a framework that unifies instance-based learning and logistic regression, comprising both methods as special cases. This approach allows one to capture interdependencies between labels and, moreover, to combine model-based and similarity-based inference for multilabel classification. As will be shown by experimental studies, our approach is able to improve predictive accuracy in terms of several evaluation criteria for multilabel prediction.

1 Introduction

In conventional classification, each instance is assumed to belong to exactly one among a finite set of candidate classes. As opposed to this, the setting of multilabel classification allows an instance to belong to several classes simultaneously or, say, to attach more than one label to a single instance. Problems of this type are ubiquitous in everyday life: At IMDb, a movie can be categorized as *action*, *crime*, and *thriller*; a CNN news report can be tagged as *people* and *political* at the same time; in biology, a typical multilabel learning example is the gene functional prediction problem, where a gene can be associated with multiple functional classes, such as *metabolism*, *transcription*, and *protein synthesis*.

Multilabel classification has received increasing attention in machine learning in recent years, not only due to its practical relevance, but also as it is interesting from a theoretical point of view. In fact, even though it is possible to reduce the problem of multilabel classification to conventional classification in one way or the other and, hence, to apply existing methods for the latter to solve the former, straightforward solutions of this type are usually not optimal. In particular, since the presence or absence of the different class labels has to be predicted *simultaneously*, it is obviously important to exploit correlations and interdependencies between these labels. This is usually not accomplished by simple transformations to standard classification.

Even though quite a number of more sophisticated methods for multilabel classification has been proposed in the literature, the application of *instance-based learning* (IBL) has not been studied very deeply in this context so far. This is a bit surprising, given that IBL algorithms based on the nearest neighbor estimation principle have been applied quite successfully in classification and pattern recognition for a long time [1]. A notable exception is the *multilabel k-nearest neighbor* (MLKNN) method that was recently proposed in [23], where it was shown to be competitive to state-of-the-art machine learning methods.

In this paper, we propose a novel approach to multilabel classification, which is based on a framework that unifies instance-based learning and logistic regression, comprising both methods as special cases. This approach overcomes some limitations of existing instance-based multilabel classification methods, including MLKNN. In particular, it allows one to capture interdependencies between the class labels in a proper way.

The rest of this paper is organized as follows: The problem of multilabel classification is introduced in a more formal way in Section 2, and related work is discussed in Section 3. Our novel method is then described in Section 4. Section 5 is devoted to experiments with several benchmark data sets. The paper ends with a summary and some concluding remarks in Section 6.

2 Multilabel Classification

Let $\mathbb{X}$ denote an instance space and let $\mathcal{L} = \{\lambda_1, \lambda_2 \ldots \lambda_m\}$ be a finite set of class labels. Moreover, suppose that each instance $\boldsymbol{x} \in \mathbb{X}$ can be associated with a subset of labels $L \in 2^{\mathcal{L}}$; this subset is often called the set of *relevant* labels, while the complement $\mathcal{L} \setminus L$ is considered as *irrelevant* for $\boldsymbol{x}$. Given training data in the form of a finite set T of observations in the form of tuples $(\boldsymbol{x}, L_{\boldsymbol{x}}) \in \mathbb{X} \times 2^{\mathcal{L}}$, typically assumed to be drawn independently from an (unknown) probability distribution on $\mathbb{X} \times 2^{\mathcal{L}}$, the goal in multilabel classification is to learn a classifier $h : \mathbb{X} \to 2^{\mathcal{L}}$ that generalizes well beyond these observations in the sense of minimizing the expected prediction loss with respect to a specific loss function; commonly used loss functions will be reviewed in Section 5.3.

Note that multilabel classification can be reduced to a conventional classification problem in a straightforward way, namely by considering each label subset $L \in 2^{\mathcal{L}}$ as a distinct (meta-)class. This approach is referred to as *label powerset* (LP) in the literature. An obvious drawback of this approach is the potentially large number of classes that one has to deal with in the newly generated problem; obviously, this number is $2^{|\mathcal{L}|}$ (or $2^{|\mathcal{L}|} - 1$ if the empty set is excluded as a prediction). This is the reason why LP typically works well if the original label set $\mathcal{L}$ is small but quickly deteriorates for larger label sets. Nevertheless, LP is often used as a benchmark, and we shall also include it in our experiments later on (cf. Section 5).

Another way of reducing multilabel to conventional classification is offered by the *binary relevance* approach. Here, a separate binary classifier h_i is trained for each label $\lambda_i \in \mathcal{L}$, reducing the supervision to information about the presence or absence of this label while ignoring the other ones. For a query instance $\boldsymbol{x}$, this classifier is supposed to predict whether λ_i is relevant for $\boldsymbol{x}$ ($h_i(\boldsymbol{x}) = 1$) or not ($h_i(\boldsymbol{x}) = 0$). A multilabel prediction for $\boldsymbol{x}$ is then given by $h(\boldsymbol{x}) = \{\lambda_i \in \mathcal{L} \,|\, h_i(\boldsymbol{x}) = 1\}$. Since binary relevance learning treats every label independently of all other labels, an obvious disadvantage of this approach is that it ignores correlations and interdependencies between labels.

Some of the more sophisticated approaches learn a multilabel classifier h in an indirect way via a scoring function $f : \mathbb{X} \times \mathcal{L} \to \mathbb{R}$ that assigns a real number to each instance/label combination. The idea is that a score $f(\boldsymbol{x}, \lambda)$ is in direct correspondence with the probability that λ is relevant for $\boldsymbol{x}$. Given a scoring function of this type, multilabel prediction can be realized via thresholding:

$$h(\boldsymbol{x}) \;=\; \{\lambda \in \mathcal{L} \,|\, f(\boldsymbol{x}, \lambda) \geq t\} \;\;,$$

where $t \in \mathbb{R}$ is a threshold. As a byproduct, a scoring function offers the possibility to produce a ranking of the class labels, simply by ordering them according to their score. Sometimes, this ranking is even more desirable as a prediction, and indeed, there are several evaluation metrics that compare a true label subset with a predicted ranking instead of a predicted label subset (cf. Section 5.3).

3 Related Work

Multilabel classification has received a great deal of attention in machine learning in recent years, and a number of methods has been developed, often motivated by specific types of applications such as text categorization [14, 19, 11, 22], computer vision [2], and bioinformatics [3, 7, 22]. Besides, several well-established methods for conventional classification have been extended to the multi-label case, including support vector machines [10, 7, 2], neural networks [22], and decision trees [20].

In this paper, we are especially interested in instance-based approaches to multilabel classification, i.e., methods based on the nearest neighbor estimation principle [5, 1]. This interest is largely motivated by the *multilabel k-nearest neighbor* (MLKNN) method that has recently been proposed in [23]. In that paper, the authors show that MLKNN performs quite well in practice. In the concrete experiments presented, MLKNN even outperformed some state-of-the-art model-based approaches to multilabel classification, including RankSVM and AdaBoost.MH [7, 4].

MLKNN is a binary relevance learner, i.e., it learns a single classifier h_i for each label $\lambda_i \in \mathcal{L}$. However, instead of using the standard k-nearest neighbor (KNN) classifier as a base learner, it implements the h_i by means of a combination of KNN and Bayesian inference: Given a query instance $\boldsymbol{x}$ with unknown multilabel classification $L \subseteq \mathcal{L}$, it finds the k nearest neighbors of $\boldsymbol{x}$ in the training data and counts the number of occurrences of λ_i among these neighbors. Considering this number, y, as information in the form of a realization of a random variable Y, the posterior probability of $\lambda_i \in L$ is given by

$$\mathbf{P}(\lambda_i \in L \,|\, Y = y) = \frac{\mathbf{P}(Y = y \,|\, \lambda_i \in L) \cdot \mathbf{P}(\lambda_i \in L)}{\mathbf{P}(Y = y)} \ , \tag{1}$$

which leads to the decision rule

$$h_i(\boldsymbol{x}) = \begin{cases} 1 & \text{if } \mathbf{P}(Y = y \,|\, \lambda_i \in L)\mathbf{P}(\lambda_i \in L) \geq \mathbf{P}(Y = y \,|\, \lambda_i \notin L)\mathbf{P}(\lambda_i \notin L) \\ 0 & \text{otherwise} \end{cases}$$

The prior probabilities $\mathbf{P}(\lambda_i \in L)$ and $\mathbf{P}(\lambda_i \notin L)$ as well as the conditional probabilities $\mathbf{P}(Y = y \,|\, \lambda_i \in L)$ and $\mathbf{P}(Y = y \,|\, \lambda_i \notin L)$ are estimated from the training data in terms of corresponding relative frequencies. As an aside, we note that these estimations come with a relatively high computational complexity, since they involve the consideration of all k-neighborhoods of all training instances.

4 Combining IBL and Logistic Regression

In this section, we introduce a machine learning method whose basic idea is to consider the information that derives from examples similar to a query instance as a feature of that instance, thereby blurring the distinction between instance-based and model-based learning to some extent. This idea is put into practice by means of a learning algorithm that realizes instance-based classification as logistic regression.

4.1 KNN Classification

Suppose an instance $\boldsymbol{x}$ to be described in terms of features ϕ_i, $i = 1, 2 \ldots n$, where $\phi_i(\boldsymbol{x})$ denotes the value of the i-th feature for instance $\boldsymbol{x}$. The instance space $\mathbb{X}$ is endowed with a distance measure: $\Delta(\boldsymbol{x}, \boldsymbol{x}')$ is the distance between instances $\boldsymbol{x}$ and $\boldsymbol{x}'$. We shall first focus on the case of binary classification and hence define the set of class labels by $\mathcal{Y} = \{-1, +1\}$. A tuple $(\boldsymbol{x}, y) \in \mathbb{X} \times \mathcal{Y}$ is called a labeled instance or example. $\mathcal{D}$ denotes a sample that consists of N labeled instances $(\boldsymbol{x}_i, y_i)$, $1 \leq i \leq N$. Finally, a new instance $\boldsymbol{x}_0 \in \mathbb{X}$ (a query) is given, whose label $y_0 \in \{-1, +1\}$ is to be estimated.

The nearest neighbor (NN) principle prescribes to estimate the label of the yet unclassified query $\boldsymbol{x}_0$ by the label of the nearest (least distant) sample instance. The KNN approach is a slight generalization, which takes the $k \geq 1$ nearest neighbors of $\boldsymbol{x}_0$ into account. That is, an estimation $\hat{y}_0$ of y_0 is derived from the set $\mathcal{N}_k(\boldsymbol{x}_0)$ of the k nearest neighbors of $\boldsymbol{x}_0$, usually by means of a *majority vote*:

$$\hat{y}_0 = \arg\max_{y \in \mathcal{Y}} \#\{\boldsymbol{x}_i \in \mathcal{N}_k(\boldsymbol{x}_0) \,|\, y_i = y\}. \tag{2}$$

4.2 IBL as Logistic Regression

A key idea of our approach is to consider the labels of neighbored instances as "features" of the query $\boldsymbol{x}_0$ whose label is to be estimated. It is worth mentioning that similar ideas have recently been exploited in relational learning [8] and collective classification [12, 9].

Denote by p_0 the prior probability of $y_0 = +1$ and by π_0 the corresponding posterior probability. Moreover, let $\delta_i \overset{\text{df}}{=} \Delta(\boldsymbol{x}_0, \boldsymbol{x}_i)$ be the distance between $\boldsymbol{x}_0$ and $\boldsymbol{x}_i$. Taking the known label y_i as information about the unknown label y_0, we can consider the posterior probability

$$\pi_0 \overset{\text{df}}{=} \mathbf{P}(y_0 = +1 \,|\, y_i).$$

More specifically, Bayes' rule yields

$$\begin{aligned} \frac{\pi_0}{1-\pi_0} &= \frac{\mathbf{P}(y_i \,|\, y_0 = +1)}{\mathbf{P}(y_i \,|\, y_0 = -1)} \cdot \frac{p_0}{1-p_0} \\ &= \rho \cdot \frac{p_0}{1-p_0}, \end{aligned}$$

where ρ is the likelihood ratio. Taking logarithms on both sides, we get

$$\log\left(\frac{\pi_0}{1-\pi_0}\right) = \log(\rho) + \omega_0 \tag{3}$$

with $\omega_0 = \log(p_0) - \log(1-p_0)$.

Model (3) still requires the specification of the likelihood ratio ρ. In order to obey the basic principle underlying IBL, the latter should be a function of the distance δ_i. In fact, ρ should become large for $\delta_i \to 0$ if $y_i = +1$ and small if $y_i = -1$: Observing a very close instance $\boldsymbol{x}_i$ with label $y_i = +1$ ($y_i = -1$) makes $y_0 = +1$ more (un)likely in comparison to $y_i = -1$. Moreover, ρ should tend to 1 as $\delta_i \to \infty$: If $\boldsymbol{x}_i$ is too far away, its label does not provide any evidence, neither in favor of $y_0 = +1$ nor in favor of $y_0 = -1$. A parameterized function satisfying these properties is

$$\rho = \rho(\delta) \overset{\text{df}}{=} \exp\left(y_i \cdot \frac{\alpha}{\delta}\right),$$

where $\alpha > 0$ is a constant. Note that the choice of a special functional form for ρ is quite comparable to the specification of the kernel function used in (non-parametric) kernel-based density estimation, as well as to the choice of the weight function in weighted NN estimation. $\rho(\delta)$ actually determines the probability that two instances whose distance is given by $\delta = \Delta(\boldsymbol{x}_0, \boldsymbol{x}_i)$ do have the same label.

Now, taking the complete sample neighborhood $\mathcal{N}(\boldsymbol{x}_0)$ of $\boldsymbol{x}_0$ into account and —as in the naive Bayes approach— making the simplifying assumption of conditional independence, we obtain

$$\begin{aligned}\log\left(\frac{\pi_0}{1-\pi_0}\right) &= \omega_0 + \alpha \sum_{\boldsymbol{x}_i \in \mathcal{N}(\boldsymbol{x}_0)} \frac{y_i}{\delta_i} \\ &= \omega_0 + \alpha \cdot \omega_+(\boldsymbol{x}_0),\end{aligned} \tag{4}$$

where $\omega_+(\boldsymbol{x}_0)$ can be seen as a summary of the evidence in favor of label +1. As can be seen, the latter is simply given by the sum of neighbors with label +1, weighted by their distance, minus the weighted sum of neighbors with label −1.

As concerns the classification of the query $\boldsymbol{x}_0$, the decision is determined by the sign of the right-hand side in (4). From this point of view, (4) does basically realize a weighted NN estimation, or, stated differently, it is a "model-based" version of instance-based learning. Still, it differs from the simple NN scheme in that it includes a bias term ω_0, which plays the same role as the prior probability in Bayesian inference.

From a statistical point of view, (4) is nothing else than a logistic regression equation. In other words, taking a "feature-based" view of instance-based learning and applying a Bayesian approach to inference comes down to realizing IBL as logistic regression.

By introducing a *similarity measure* κ, inversely related to the distance function Δ, (4) can be written in the form

$$\log\left(\frac{\pi_0}{1-\pi_0}\right) = \omega_0 + \alpha \sum_{\boldsymbol{x}_i \in \mathcal{N}(\boldsymbol{x}_0)} \kappa(\boldsymbol{x}_0, \boldsymbol{x}_i) \cdot y_i \ . \tag{5}$$

Note that, as a special case, this approach can mimic the standard KNN classifier (2), namely by setting $\omega_0 = 0$ and defining κ in terms of the (data-dependent) "KNN kernel"

$$\kappa(\boldsymbol{x}_0, \boldsymbol{x}_i) = \begin{cases} 1 & \text{if } \boldsymbol{x}_i \in \mathcal{N}_k(\boldsymbol{x}_0) \\ 0 & \text{otherwise} \end{cases} \ . \tag{6}$$

4.3 Estimation and Classification

The parameter α in (4) determines the weight of the evidence

$$\omega_+(\boldsymbol{x}_0) = \sum_{\boldsymbol{x}_i \in \mathcal{N}(\boldsymbol{x}_0)} \kappa(\boldsymbol{x}_0, \boldsymbol{x}_i) \cdot y_i \tag{7}$$

and, hence, its influence on the posterior probability estimation π_0. In fact, α plays the role of a smoothing (regularization) parameter. The smaller α is chosen, the smoother an estimated probability function (obtained by applying (5) to all points $\boldsymbol{x}_0 \in \mathcal{X}$) will be. In the extreme case where $\alpha = 0$, one obtains a constant function (equal to ω_0).

An optimal specification of α can be accomplished by adapting this parameter to the data $\mathcal{D}$, using the method of maximum likelihood (ML). For each sample point $\boldsymbol{x}_j$ denote by

$$\omega_+(\boldsymbol{x}_j) \overset{\text{df}}{=} \sum_{\boldsymbol{x}_j \neq \boldsymbol{x}_i \in \mathcal{N}(\boldsymbol{x}_j)} \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) \cdot y_i$$

the sample evidence in favor of $y_j = +1$. The log-likelihood function is then given by the mapping

$$\alpha \mapsto \sum_{j\,:\,y_j=+1} w_0 + \alpha\,\omega_+(\boldsymbol{x}_j) - \sum_{j=1}^{N} \log\big(1 + \exp(w_0 + \alpha\,\omega_+(\boldsymbol{x}_j)\big)\,, \tag{8}$$

and the optimal parameter α^* is the maximizer of (8). The latter can be computed by means of standard methods from logistic regression. The posterior probability π_0 for the query is then given by

$$\pi_0 = \frac{\exp(\omega_0 + \alpha^*\,\omega_+(\boldsymbol{x}_0))}{1 + \exp(\omega_0 + \alpha^*\,\omega_+(\boldsymbol{x}_0))}\ .$$

To classify $\boldsymbol{x}_0$, one applies the decision rule

$$\hat{y}_0 \overset{\text{df}}{=} \begin{cases} +1 & \text{if} \quad \pi_0 \geq 1/2 \\ -1 & \text{if} \quad \pi_0 < 1/2 \end{cases}\ .$$

Subsequently, we shall refer to the method outlined above as IBLR (Instance-Based Learning by Logistic Regression).

4.4 Including Additional Features

In the previous section, instance-based learning has been embedded into logistic regression, using the information coming from the neighbors of a query $\boldsymbol{x}_0$ as a "feature" of that query. In this section, we consider a possible generalization of this approach, namely the idea to extend the model (5) by taking further features of $\boldsymbol{x}_0$ into account:

$$\log\left(\frac{\pi_0}{1-\pi_0}\right) = \alpha\,\omega_+(\boldsymbol{x}_0) + \sum_{\varphi_s \in \mathcal{F}} \beta_s\,\varphi_s(\boldsymbol{x}_0), \tag{9}$$

where $\mathcal{F} = \{\varphi_0, \varphi_1 \ldots \varphi_r\}$ is a subset of the available features $\{\phi_0, \phi_1 \ldots \phi_n\}$ and $\varphi_0 = \phi_0 \equiv 1$, which means that β_0 plays the role of ω_0. Equation (9) is a common logistic regression model, except that $\omega_+(\boldsymbol{x}_0)$ is a "non-standard" feature.

The approach (9), that we shall call IBLR+, integrates instance-based and model-based (attribute-based) learning and, by estimating the regression coefficients in (9), achieves an optimal balance between both approaches. The extended model (9) can be interpreted as a logistic regression model of IBL, as outlined in Section 4.2, where the bias ω_0 is no longer constant:

$$\log\left(\frac{\pi_0}{1-\pi_0}\right) \;=\; \omega_0(\boldsymbol{x}_0) + \alpha\,\omega_+(\boldsymbol{x}_0) \;\;, \tag{10}$$

with $\omega_0(\boldsymbol{x}_0) \overset{\mathrm{df}}{=} \sum \beta_s \varphi_s(\boldsymbol{x}_0)$ being an instance-specific bias determined by the model-based part of (9).

4.5 Extension to Multilabel Classification

So far, we only considered the case of binary classification. To extend the approach to multilabel classification with a label set $\mathcal{L} = \{\lambda_1, \lambda_2 \ldots \lambda_m\}$, the idea is to train one classifier h_i for each label. For the i-th label λ_i, this classifier is derived from the model

$$\log\left(\frac{\pi_0^{(i)}}{1-\pi_0^{(i)}}\right) \;=\; \omega_0^{(i)} + \sum_{j=1}^{m} \alpha_j^{(i)} \cdot \omega_{+j}^{(i)}(\boldsymbol{x}_0) \;\;, \tag{11}$$

where $\pi_0^{(i)}$ denotes the (posterior) probability that λ_i is relevant for $\boldsymbol{x}_0$, and

$$\omega_{+j}^{(i)}(\boldsymbol{x}_0) = \sum_{\boldsymbol{x}\in\mathcal{N}(\boldsymbol{x}_0)} \kappa(\boldsymbol{x}_0,\boldsymbol{x}) \cdot y_j(\boldsymbol{x}) \tag{12}$$

is a summary of the presence of the j-th label λ_j in the neighborhood of $\boldsymbol{x}_0$; here, $y_j(\boldsymbol{x}) = +1$ if λ_j is present (relevant) for the neighbor $\boldsymbol{x}$, and $y_j(\boldsymbol{x}) = -1$ in case it is absent (non-relevant).

Obviously, the approach (11) is able to take interdependencies between class labels into consideration. More specifically, the estimated coefficient $\alpha_j^{(i)}$ indicates to what extent the relevance of label λ_i is influenced by the relevance of λ_j. A value $\alpha_j^{(i)} \gg 0$ means that the presence of λ_j makes the relevance of λ_i more likely, i.e., there is a positive correlation. Correspondingly, a negative coefficient would indicate a negative correlation.

Note that the estimated probabilities $\pi_0^{(i)}$ can naturally be considered as scores for the labels λ_i. Therefore, a ranking of the labels is simply obtained by sorting them in decreasing order according to their probabilities. Moreover, a pure multilabel prediction for $\boldsymbol{x}_0$ is derived from this ranking via thresholding at $t = 0.5$.

Of course, it is also possible to combine the model (11) with the extension proposed in Section 4.4. This leads to a model

$$\log\left(\frac{\pi_0^{(i)}}{1-\pi_0^{(i)}}\right) \;=\; \sum_{j=1}^{m} \alpha_j^{(i)} \cdot \omega_{+j}^{(i)}(\boldsymbol{x}_0) + \sum_{\varphi_s \in \mathcal{F}} \beta_s^{(i)}\, \varphi_r(\boldsymbol{x}_0) \;\; . \qquad (13)$$

We shall refer to the extensions (11) and (13) of IBLR to multilabel classification as IBLR-ML and IBLR-ML+, respectively.

5 Experimental Results

This section is devoted to experimental studies that we conducted to get a concrete idea of the performance of our method. Before presenting the results of our experiments, we give some information about the learning algorithms and data sets included in the study, as well as the criteria used for evaluation.

5.1 Learning Algorithms

For the reasons mentioned previously, our main interest is focused on MLKNN, which is arguably the state-of-the-art in instance-based multilabel ranking; we used its implementation in the MULAN package [18]. MLKNN is parameterized by the size of the neighborhood, for which we adopted the value $k = 10$. This value is recommended in [23], where it was found to yield the best performance. For the sake of fairness, we use the same neighborhood size for our method, in conjunction with the KNN kernel (6). In both cases, the simple Euclidean metric (on the complete attribute space) was used as a distance function. For our method, we tried both variants, the pure instance-based version (11), and the extended model (13) with $\mathcal{F}$ including all available features. Intuitively, one may expect the latter, IBLR-ML+, to be advantageous to the former, IBLR-ML, as it can use features in a more flexible way. Yet, one should note that, since we simply included all attributes in $\mathcal{F}$, each attribute will essentially be used twice in IBLR-ML+, thus producing a kind of redundancy. Besides, model induction will of course become more difficult, since a larger number of parameters needs to be estimated.

DATA SET	DOMAIN	#INSTANCES	#ATTRIBUTES	#LABELS	CARDINALITY
emotions	music	593	72	6	1.87
image	vision	2000	135	5	1.24
genbase	biology	662	1186*	27	1.25
mediamill	multimedia	5000	120	101	4.27
reuters	text	7119	243	7	1.24
scene	vision	2407	294	6	1.07
yeast	biology	2417	103	14	4.24

Table 1: Statistics for the multilabel data sets used in the experiments. The symbol * indicates that the data set contains binary features; *cardinality* is the average number of labels per instance.

As an additional baseline we used binary relevance learning (BR) with three different base learners: logistic regression, C4.5 (the WEKA [21] implementation J48 in its default setting), and KNN (again with $k = 10$). Finally, we also included label powerset (LP) with C4.5 as a base learner.

5.2 Data Sets

Benchmark data for multi-label classification is not as abundant as for conventional classification, and indeed, experiments in this field are often restricted to a very few or even only a single data set. For our experimental study, we have collected a comparatively large number of seven data sets from different domains; an overview is given in Table 1.[1]

The *emotions* data was created from a selection of songs from 233 musical albums [17]. From each song, a sequence of 30 seconds after the initial 30 seconds was extracted. The resulting sound clips were stored and converted into wave files of 22050 Hz sampling rate, 16-bit per sample and mono. From each wave file, 72 features have been extracted, falling into two categories: rhythmic and timbre. Then, in the emotion labeling process, 6 main emotional clusters are retained corresponding to the Tellegen-Watson-Clark model of mood: amazed-surprised,

[1]All data sets are public available at `http://mlkd.csd.auth.gr/multilabel.html` and `http://lamda.nju.edu.cn/data.htm`.

happy-pleased, relaxing-clam, quiet-still, sad-lonely and angry-aggressive.

Image and *scene* are semantic scene classification data sets proposed, respectively, by [24] and [2], in which a picture can be categorized into one or more classes. In the scene data, for example, pictures can have the following classes: beach, sunset, foliage, field, mountain, and urban. Features of this data set correspond to spatial color moments in the LUV space. Color as well as spatial information have been shown to be fairly effective in distinguishing between certain types of outdoor scenes: bright and warm colors at the top of a picture may correspond to a sunset, while those at the bottom may correspond to a desert rock. Features of the image data set are generated by the SBN method [13] and essentially correspond to attributes in an RGB color space.

From the biological field, we have chosen the two data sets *yeast* and *genbase*. The yeast data set is about predicting the functional classes of genes in the Yeast Saccharomyces cerevisiae. Each gene is described by the concatenation of micro-array expression data and a phylogenetic profile, and is associated with a set of 14 functional classes. The data set contains 2417 genes in total, and each gene is represented by a 103-dimensional feature vector. In the *genbase* data, 27 important protein families are considered, including, for example, PDOC00064 (a class of oxydoreductases) and PDOC00154 (a class of isomerases). During the preprocessing, a training set was exported, consisting of 662 proteins that belong to one or more of these 27 classes.

From the text processing field, we have chosen a subset of the widely studied *Reuters-21578* collection [15]. The seven most frequent categories are considered. After removing documents whose label sets or main texts are empty, 8866 documents are retained where only 3.37% of them are associated with more than one class label. After randomly removing documents with only one label, a text categorization data set containing 2,000 documents is obtained. Each document is represented as a bag of instances using the standard sliding window techniques, where each instance corresponds to a text segment enclosed in one sliding window of size 50 (overlapped with 25 words). “Function words” are removed from the vocabulary and the remaining words are stemmed. Instances in the bags adopt the “bag-of-words” representation based on term frequency. Without loss of effectiveness, dimensionality reduction is performed by retaining the top 2% words with highest document frequency. Thereafter, each instance is represented as a

243-dimensional feature vector.

The *mediamill* data set is from the field of multimedia indexing and originates from the well-known TREC Video Retrieval Evaluation data (TRECVID 2005/2006) initiated by American National Institute of Standards and Technology (NIST), which contains 85 hours of international broadcast news data. The task in this data set is the automated detection of a lexicon of 101 semantic concepts in videos. Every instance of this data set has 120 numeric features including visual, textual, as well as fusion information. The trained classifier should be able to categorize an unseen instance to some of these 101 labels, e.g., face, car, male, soccer, and so on. More details about this data set can be found at [16].

5.3 Evaluation Measures

To evaluate the performance of multilabel classification methods, a number of criteria and metrics have been proposed in the literature. For a classifier h, let $h(\boldsymbol{x}) \subseteq \mathcal{L}$ denote its multilabel prediction for an instance $\boldsymbol{x}$, and let $L_{\boldsymbol{x}}$ denote the true set of relevant labels. Moreover, in case a related scoring function f is also defined, let $f(\boldsymbol{x}, \lambda)$ denote the score assigned to label λ for instance $\boldsymbol{x}$. The most commonly used evaluation measures are defined as follows:

- *Hamming loss* computes the percentage of labels whose relevance is predicted incorrectly:

$$\textsc{HamLoss}(h) = \frac{1}{|\mathcal{L}|} \big| h(\boldsymbol{x}) \, \Delta \, L_{\boldsymbol{x}} \big|, \tag{14}$$

  where Δ is the symmetric difference between two sets.

- *One error* computes how many times the top-ranked label is not relevant:

$$\textsc{OneError}(f) = \begin{cases} 1 & \text{if } \arg\max_{\lambda \in \mathcal{L}} f(\boldsymbol{x}, \lambda) \notin L_{\boldsymbol{x}} \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

- *Coverage* determines how far one needs to go in the list of labels to cover all the relevant labels of an instance. This measure is loosely related to the precision at the level of perfect recall:

$$\textsc{Coverage}(f) = \max_{\lambda \in L_{\boldsymbol{x}}} rank_f(\boldsymbol{x}, \lambda) - 1 \ , \tag{16}$$

where $rank_f(\boldsymbol{x}, \lambda)$ denotes the position of label $\boldsymbol{x}$ in the ordering induced by f.

- *Rank loss* computes the average fraction of label pairs that are not correctly ordered:

$$\textsc{RankLoss}(f) = \frac{\#\{(\lambda, \lambda') \,|\, f(\boldsymbol{x}, \lambda) \leq f(\boldsymbol{x}, \lambda'), (\lambda, \lambda') \in L_{\boldsymbol{x}} \times \overline{L_{\boldsymbol{x}}}\}}{|L_{\boldsymbol{x}}||\overline{L_{\boldsymbol{x}}}|}, \quad (17)$$

  where $\overline{L_{\boldsymbol{x}}} = \mathcal{L} \setminus L_{\boldsymbol{x}}$ is the set of irrelevant labels.

- *Average precision* determines for each relevant label $\lambda \in L_{\boldsymbol{x}}$ the percentage of relevant labels among all labels that are ranked above it, and averages these percentages over all relevant labels:

$$\textsc{AvePrec}(f) = \frac{1}{|L_{\boldsymbol{x}}|} \sum_{\lambda \in L_{\boldsymbol{x}}} \frac{|\{\lambda' \,|\, rank_f(\boldsymbol{x}, \lambda') \leq rank_f(\boldsymbol{x}, \lambda), \lambda' \in L_{\boldsymbol{x}}\}|}{rank_f(\boldsymbol{x}, \lambda)}. \quad (18)$$

Notice that only Hamming loss evaluates mere multilabel predictions (i.e., the multilabel classifier h), while the others metrics evaluate the underlying ranking function f. Moreover, smaller values indicate better performance for all measures except average precision. Finally, except for coverage, all measures are normalized and assume values between 0 and 1.

5.4 Results and Discussion

The results of a cross validation study (10-fold, 5 repeats) are summarized in Table 2 at the end of the paper. As can be seen, the baseline methods BR and LP are in general not competitive. Looking at the average ranks, IBLR-ML consistently outperforms all other methods, regardless of the evaluation metric, indicating that it is the strongest method overall. The ranking among the three instance-based methods is IBLR-ML $\succ$ IBLR-ML+ $\succ$ MLKNN for all measures except OneError, for which the latter two change the position.

To analyze the results more thoroughly, we followed the two-step statistical test procedure recommended in [6], consisting of a Friedman test of the null hypothesis that all learners have equal performance and, in case this hypothesis is rejected, a