Mr.KNN: Soft Relevance for Multi-label Classification

Xiaotong Lin
Department of Electrical Engineering and Computer Science
The University of Kansas
1520 West 15th Street, Lawrence, Kansas, USA
(785)864-8825
cindylin@ku.edu

Xue-wen Chen
Department of Electrical Engineering and Computer Science
The University of Kansas
1520 West 15th Street, Lawrence, Kansas, USA
(785)864-4559
xwchen@ku.edu

ABSTRACT
Multi-label classification refers to learning tasks with each instance belonging to one or more classes simultaneously. It arose from real-world applications such as information retrieval, text categorization and functional genomics. Currently, most of the multi-label learning methods use the strategy called binary relevance, which constructs a classifier for each unique label by grouping data into positives (examples with this label) and negatives (examples without this label). With binary relevance, an example with multiple labels is considered as a positive data for each label it belongs to. For some classes, this data point may behave like an outlier confusing classifiers, especially in the cases of well-separated classes. In this paper, we first introduce a new strategy called soft relevance, where each multi-label example is assigned a relevance score to the labels it belongs to. This soft relevance is then employed in a voting function used in a *k* nearest neighbor classifier. Furthermore, a voting-margin ratio is introduced to the *k* nearest neighbor classifier for better performance. We compare the proposed method to other multi-label learning methods over three multi-label datasets and demonstrate that the proposed method provides an effective way to multi-label learning.

Categories and Subject Descriptors
I.2.6 [**Artificial Intelligence**]: Learning – *induction;* H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *clustering, information filtering*

General Terms
Algorithms, Experimentation, Theory

Keywords
Multi-label classification, soft relevance, k-nearest neighbors

1. INTRODUCTION
While most of the research efforts in machine learning have been devoted to single-label classification problems, where each instance is restricted to exactly one class, multi-label classification is drawing increasing interest and emerging as a fast-growing research field. Multi-label learning, arising from real-world applications, refers to learning tasks where each instance is assigned to one or more classes (labels). For example, in web search, each returned webpage with a given query may be labeled with more than one categories: consider the following webpage http://www.pbs.org/science/science_health.html, which may be annotated as "Science", "Health", "Education", and "News and Media", four out of 14 top-level categories used by Yahoo! Search. Other applications with multi-label classification include automatic text categorization [1-2], where each free-text document may be assigned to multiple predefined categories; scene classification of images and videos [3-7], where an image may have more than one tags; functional genomics [8-12], where a gene can be annotated with a set of functions [13-14]; music categorization into emotions [15-17] and directed marketing [18].

Over the years, there have been a variety of methods developed for multi-label classifications. These methods are grouped as either problem transformation methods or algorithm adaptation methods [19]. Problem transformation methods first transform multi-label learning tasks into multiple single-label learning tasks, which are then handled by the standard single-label learning algorithms. The first family of the problem transformation methods is based on "copy" or "selection" [3, 20]. Copy transformation converts every instance of *k* labels into *k* instance of one single label (each label is a unique element of the *k* labels). A constant weight of 1/*k* may also be assigned to each of the copied instances ("copy-weight"). Alternatively, one can use a selection strategy by replacing the multiple labels of each instance with a single label that is the most frequent ("select-max") or least frequent ("select-min"), or randomly selected ("select-random"). Another simple transformation is to use the data with a single label only ("ignore"). The second family is called Label Powerset (LP) method and its variants [21-22]. LP methods treat each unique set of labels in the training set as a new (and single) label, and reconstruct the training set with the newly defined labels. To deal with the small sample problems in LP, the pruned problem transformation method simply discards the newly-defined labels with very few training data and reassigns these data into other relevant labels. Alternatively, the random *k*-labelsets method ensembles classifiers constructed from randomly selected LP label sets. The third family is based on binary relevance (BR) [23]. Rather than focusing on reassigning samples with labels, BR uses a popular yet simple strategy commonly-employed in multi-class classification, namely "one-versus-rest". Basically, it constructs a binary classifier for each unique label using training data grouped

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
CIKM'10, October 26-30, 2010, Toronto, Ontario, Canada.
Copyright 2010 ACM 978-1-4503-0099-5/10/10...$10.00.

as positives (instances with this label) and negatives (instances without this label). An alternative strategy for multi-class classification is called "one-versus-one", which is also explored in multi-label learning. Ranking by pairwise comparison (PRC) methods construct binary classifiers for the $\binom{m}{2} = m(m-1)/2$ possible class pairings where m is the total number of unique labels and each classifier is trained on the data belonging to the two classes (but not both) [24]. Furthermore, extended from PRC, the calibrated label ranking method introduces an artificial label and combines the PRC and BR learning strategies [25].

Algorithm adaption methods modify single-label learning algorithms for multi-label data learning. McCallum first proposed a mixture model based on naïve Bayes and EM algorithms for multi-label text classification [1]. About one year later, Schapire and Singer proposed the use of boosting-based systems for text categorization and automatic call-type identification from unconstrained spoken customer responses [2]. They developed two boosting algorithms by maintaining a set of weights over training examples and associated labels: AdaBoost.MH minimizing Hamming loss and AdaBoost.MR ranking the labels such that the top-ranked labels are also correct. Ghamrawi and McCallum investigated a multi-label conditional random field classification method that models dependencies between labels [26]. Kernel-based hierarchical classification methods are also popular in multi-label text classification. Cai and Hofmann extended support vector machines (SVMs) learning and integrated discriminant functions for hierarchical classification of document categorization problems [27]. Rousu et al. used a kernel-based algorithm modified from the Maximum Margin Markov Network framework [28]. Another family of learning methods treat multi-label learning problems as ranking problems, where a score is assigned to every instance-label pair; traditional learning methods such as SVMs [12], neural networks [29], and online ranking methods [30], were then used for classification. Other learning methods such as decision trees [11], probabilistic generative models [31], and Bayesian rule [32] are also adapted for multi-label classification problems.

A number of multi-label learning methods are adapted from a case-based learning called *k* nearest neighbor (*k*NN) method [7, 16, 33-34], which is a simple yet powerful learning approach. *k*NN makes predictions of a test data based on its *k* nearest neighbors and a majority voting rule. *k*NN is easy to implement and appropriate by nature for multi-label classifications. With an infinite number of data, *k*NN is assured of becoming optimal [35]. *k*NN-based approaches have shown great potential in multi-label learning problems [7, 16, 33-34]. The multi-label *k*NN learning method (ML-KNN) applies maximum a posteriori principle for classification and ranking, where the likelihood is estimated using the *k* nearest neighbors of a data [7]. Brinker and Hullermeier demonstrated that *k*NN-based learning approach is competitive with or even better than state-of-the-art-model-based methods [33]. Similar results are also observed by Dimou et al. [36].

Up to date, all the *k*NN-based multi-label learning methods use the popular BR strategy [23]. Although effective, BR may artificially introduce outliers, which tend to degrade the performance of classifiers, as we will discuss in Section 3. Furthermore, the estimation of the posteriori in ML-KNN may be inaccurate, due to the facts that the samples with and without a particular label are typically highly imbalanced and also it is highly possible that only few samples are available for some given number of nearest neighbors with a certain number of labels. To address these problems, we propose a novel *k*NN-based multi-label learning approach called voting Margin-Ratio *k*NN (Mr.KNN), which introduces the voting margin-ratio concept and soft relevance in the vote strategy. We test it on three commonly-used datasets and experimental results show a significant improvement in performance compared to the ML-KNN method.

The major contribution of this paper is summarized as follows: (1) we propose a novel learning algorithm that integrates both problem transformation methods and algorithm adaptation methods. (2) A new concept called soft relevance is introduced for data transformation. Rather than making a hard decision on label assignments, we introduce the use of a modified fuzzy c-means algorithm in a supervised setting, which will provide a relevance score of an instance with respect to a particular label. This score is produced based on real data distribution. (3) We introduce a new voting schema in Mr.KNN, which is based on the distances between a test data and its nearest neighbors and the soft relevance of each nearest neighbors. Furthermore, a margin-ratio of votes is first used to allow the selection of different distance metric, which is a critical issue in *k*NN design. (4) We evaluate the proposed algorithm and provide the comparison with the ML-KNN algorithm. A detailed discussion about the newly proposed method is also presented.

The structure of this paper is organized as follows. In Section 2, we briefly review the ML-KNN algorithm. This is followed by a detailed description of the new algorithm we propose in Section 3. We compare different algorithms with three multi-label datasets and show the results in Section 4. Finally, we conclude with discussions in Section 5.

2. RELATED WORK

In this section, we briefly review problem transformation methods and algorithm adaptation methods [19]. We will then discuss the ML-KNN algorithm. Throughout this paper, we will use the following notations. Let $\{(\vec{x}_i, \vec{y}_i)\}_{i=1}^{n}$ denote a training set of n multi-label examples with input vectors $\vec{x}_i \in \mathfrak{R}^{d}$ and class label vectors $\vec{y}_i \in \{\mathbf{0}, \mathbf{1}\}^{l}$. A "1" in the j-th component of a class label vector indicates that the associated instance belongs to the j-th class.

2.1 Problem Transformation Methods

For each multi-label instance, problem transformation methods convert it into one or multiple instances with a single label. For illustration, we consider a five-label classification problem with a multi-label data set shown in Table 1. This data set consists of two single-label instances ($\vec{x}_2$ and $\vec{x}_4$) and six multi-label instances. Tables 2 and 3 show the new single-label datasets transformed from the original multi-label data set by simply labeling each multi-label instance with the most frequent (select-max) and the least frequent (select-min) labels, respectively. As can be seen, this strategy will most likely create highly imbalanced datasets.

Another popular strategy employed in problem transformation method is the so-called binary relevance, which converts the multi-label learning problem to multiple single-label binary classification problems. For example, for the data set shown in Fig. 1, five new data sets will be generated, each corresponding to a particular class label (Table 4). For each data set in Table 4, an instance with the associated label is marked as positive (+), or

negative (-) otherwise. Standard binary classification algorithms can then be applied to each dataset.

While there are some other similar strategies [19], a common problem in problem transformation methods is that multi-label instances are forced into one single category without considering data distribution. For example, with select-max strategy, many classes would consist of very few positive examples and dominant number of negative examples. In Section 3.1, we will further discuss the potential problems with a binary relevance strategy.

Table 1. A five-class multi-label data set

Instances ($\vec{x}_i$)	Label vectors ($\vec{y}_i$)
$\vec{x}_1$	(0, 1, 1, 0 , 0)
$\vec{x}_2$	(0, 0, 0, 1, 0)
$\vec{x}_3$	(0, 1, 0, 0, 1)
$\vec{x}_4$	(1, 0, 0, 0, 0)
$\vec{x}_5$	(0, 0, 0, 1, 1)
$\vec{x}_6$	(1, 1, 0, 1, 1)
$\vec{x}_7$	(1, 1, 1, 0, 1)
$\vec{x}_8$	(0, 1, 0, 1, 0)

Table 2. Transformed data set using select-max

Instances ($\vec{x}_i$)	Label vectors ($\vec{y}_i$)
$\vec{x}_1$	(0, 1, 0, 0 , 0)
$\vec{x}_2$	(0, 0, 0, 1, 0)
$\vec{x}_3$	(0, 1, 0, 0, 0)
$\vec{x}_4$	(1, 0, 0, 0, 0)
$\vec{x}_5$	(0, 0, 0, 0, 1)
$\vec{x}_6$	(0, 1, 0, 0, 0)
$\vec{x}_7$	(0, 1, 0, 0, 0)
$\vec{x}_8$	(0, 1, 0, 0, 0)

Table 3. Transformed data set using select-min

Instances ($\vec{x}_i$)	Label vectors ($\vec{y}_i$)
$\vec{x}_1$	(0, 0, 1, 0 , 0)
$\vec{x}_2$	(0, 0, 0, 1, 0)
$\vec{x}_3$	(0, 0, 0, 0, 1)
$\vec{x}_4$	(1, 0, 0, 0, 0)
$\vec{x}_5$	(0, 0, 0, 1, 0)
$\vec{x}_6$	(1, 0, 0, 0, 0)
$\vec{x}_7$	(0, 0, 1, 0, 0)
$\vec{x}_8$	(0, 0, 0, 1, 0)

Table 4. Transformed data sets using binary relevance

Instances ($\vec{x}_i$)	Dataset-1 (label 1)	Dataset-2 (label 2)	Dataset-3 (label 3)	Dataset-4 (label 4)	Dataset-5 (label 5)
$\vec{x}_1$	-	+	+	-	-
$\vec{x}_2$	-	-	-	+	-
$\vec{x}_3$	-	+	-	-	+
$\vec{x}_4$	+	-	-	-	-
$\vec{x}_5$	-	-	-	+	+
$\vec{x}_6$	+	+	-	+	+
$\vec{x}_7$	+	+	+	-	+
$\vec{x}_8$	-	+	-	+	-

2.2 Algorithm Adaptation Methods

Algorithm adaptation methods modify standard single-label learning algorithms for multi-label classification. For example, among many others, decision trees [11], AdaBoost [2], and support vector machines [27] are adapted for multi-label learning. In [11], the C4.5 is adapted by allowing leaves of a tree to represent a set of labels. Furthermore, to measure the amount of uncertainty, an entropy-like function is modified as $-\sum_{i=1}^{l} p(c_i)logp(c_i) + q(c_i)logq(c_i)$, where $p(c_i)$ is the probability of class c_i and $q(c_i) = 1 - p(c_i)$.

AdaBoost.MH is an extension of AdaBoost for multi-label and multi-class learning tasks [2]. It deals with multi-label learning problems with a divide and conquer strategy and maintains a set of weights as a distribution over both training examples and associated labels.

To overcome the potential overfitting problems in AdaBoost.MH or other adapted methods, a SVM-like optimization strategy is introduced for multi-label learning [11], where a multi-label learning problem is treated as a ranking problem and a linear model that minimizes a ranking loss and maximizes a margin is developed.

2.3 The ML-KNN Method

Let $\mathcal{N}(\vec{x}_i)$ denote the training data subset consisting of the k nearest neighbors of the example $\vec{x}_i$. Let $C_{\vec{x}_i}(j)$ denote the number of neighbors in $\mathcal{N}(\vec{x}_i)$ belonging to the *j*-th class, which can be expressed as follows.

$$C_{\vec{x}_i}(j) = \sum_{\vec{x}_b \in N(\vec{x}_i)} \vec{y}_b(j) \quad (1)$$

ML-KNN assigns the *j*-th label to an instance using the binary relevance strategy. First, k nearest neighbors $\mathcal{N}(\vec{x}_i)$ of the instance $\vec{x}_i$ is identified. Then the *j*–th label is assigned to this instance in terms of the probability: $P\left(\vec{y}(j) = 1 \middle| C_j = C_{\vec{x}_i}(j)\right)$, where C_j is a variable representing the number of nearest neighbors belonging to the *j*-th class. Specifically, let

$$R_j = \frac{P\left(\vec{y}(j) = 1 \middle| C_j = C_{\vec{x}_i}(j)\right)}{P\left(\vec{y}(j) = 0 \middle| C_j = C_{\vec{x}_i}(j)\right)} = \frac{P(C_j = C_{\vec{x}_i}(j) | \vec{y}(j) = 1) P(\vec{y}(j) = 1)}{P(C_j = C_{\vec{x}_i}(j) | \vec{y}(j) = 0) P(\vec{y}(j) = 0)}$$

If $R_j > 1$, then $\vec{y}_i(j) = 1$; otherwise, $\vec{y}_i(j) = 0$. Both likelihood and prior can be estimated from training data using frequency counting for each $j \in \{0,1,\dots,l\}$ (where a smoothing parameter is used to control the strength of uniform prior) [7]. Note that by frequency counting, the ratio $R_j = p/q$, where p and q are the number of positive and negative examples with exactly $C_{\vec{x}_i}(j)$ nearest neighbors belonging to the j-th class, respectively (this ratio is slightly different with a smoothing parameter). With the BR strategy, data distributions for some labels are highly imbalanced (i.e., the number of positive samples is much less than that of negative samples). Consequently, the ratio estimation may not be accurate. Fig. 1 shows the data distribution of each label for the yeast data with 14 labels (more information about this data set can be found in Section 4). The y-axis is the ratio between the number of data with a particular label (x-axis) and total number of samples. While more than 70% of samples have labels 12 and 13, only less than 2% of samples have label 14.

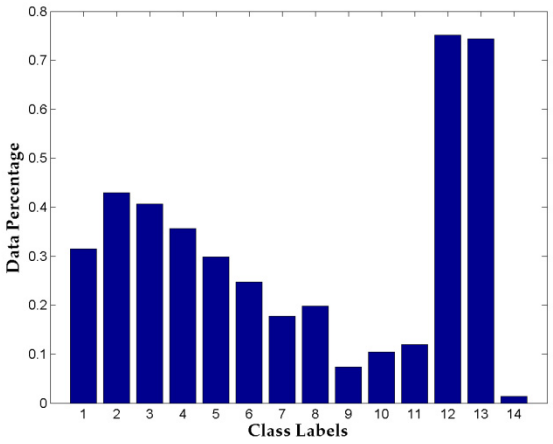

Figure 1. Sample distribution for yeast data with 14 class labels

3. Mr.KNN: METHOD DESRIPTION

Mr.KNN consists of two components: a modified fuzzy c-means (FCM)-based approach to produce soft relevance and a modified *k*NN for multi-label classification.

3.1 Soft Relevance

To see the limitation of the binary relevance strategy, consider an example shown in Fig. 2. In Fig. 2, data points from three classes (represented by triangles, cross symbols, and circles) are plotted in a two-dimensional feature space. The instance marked with a cross inside a circle belongs to two classes, marked by circle and cross symbols. As can be seen, for the class with circles, this instance looks like a typical data. However, it may be an outlier for the class with cross. In binary relevance-based methods, this instance will be used in both classes as positive samples, which may degrade the classification performance. To deal with this problem, we propose the application of an unsupervised learning algorithm in a supervised setting. Specifically, we will adapt the fuzzy c-means (FCM) algorithm [37] to yield a soft relevance value for each instance with respect to each label. This soft relevance indicates the strength of an instance related to a label in the given feature space.

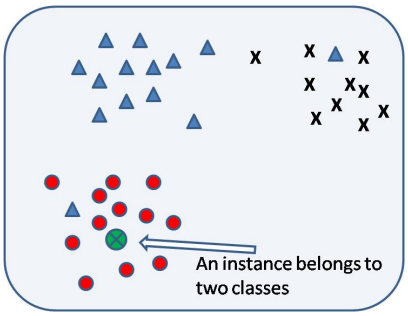

Figure 2. A 2-d feature space with multi-label data.

To modify FCMs for supervised problems, we treat each class as a cluster and denote u_{ki} the membership (relevance) value of an instance $\vec{x}_i$ in class k and $\vec{w}_k \in \mathfrak{R}^d$ the class center (prototype) $(k = 1,2,\dots,l)$. Our goal is to find an optimal fuzzy c-partition by minimizing the following cost function:

$$J_m = \sum_{i=1}^{n}\sum_{k=1}^{l}(u_{ki})^m d(\vec{x}_i, \vec{w}_k) \tag{2}$$

where m is a weighting exponent on each fuzzy membership, called a fuzzifier, and is typically set equal to 2; $d(\vec{x}_i, \vec{w}_k)$ is a distance measure between $\vec{x}_i$ and $\vec{w}_k$. In this study, we use the *Minkowski* distance defined as

$$d(\vec{x}_i, \vec{w}_k) = \left(\sum_{j=1}^{d}\left|x_{ij} - w_{kj}\right|^f\right)^{\frac{1}{f}} \tag{3}$$

where f is a positive integer and $f \geq 1$; x_{ij} and $w_{kj}$are the j-th components of $\vec{x}_i$ and $\vec{w}_k$, respectively. By adjusting f, the *Minkowski* distance can handle different shape of (classes) clusters. For example, $f = 2$ corresponds to an Euclidean distance, which works for clusters with circular shape, while $f = 1$ corresponds to a L_1 norm targeting clusters with a diamond-like shape [38]. One can also introduce a covariance matrix or a weighting vector in the distance measure to reflect the importance of difference features. However, determining the matrix or weights is another research topic and will not be discussed here.

Each membership u_{ki} is between zero and one and satisfies the following equation:

$$\sum_{k=1}^{l} u_{ki} = 1, \quad i = 1,\dots,n \tag{4}$$

Furthermore, unlike an unsupervised learning task where data labels are unknown, the class labels for each training data are known, which can be formulated as follows:

$$\sum_{k=1}^{l} \vec{y}_i(k)u_{ki} = 1, \quad i = 1,\dots,n \tag{5}$$

Eq. (5) reinforce that if an instance does not have a particular label, then the associated membership is zero. To find the membership values, we minimize the cost function J_m with

respect to all the memberships and the prototypes, subject to the constrains in Eqs. (4) and (5). This leads to the following Lagrangian function

$$\mathcal{L} = \sum_{i=1}^{n}\sum_{k=1}^{l}(u_{ki})^2 d(\vec{x}_i, \vec{w}_k) + \sum_{i=1}^{n}\lambda_i\left(1 - \sum_{k=1}^{l}u_{ki}\right) + \sum_{i=1}^{n}\xi_i\left(1 - \sum_{k=1}^{l}\vec{y}_i(k)u_{ki}\right) \quad (6)$$

where λ_i and ξ_i are Lagrangian coefficients. By minimizing Eq. 6 with respect to u_{ki} and using Eqs. (4) and (5), we obtain the class membership as

$$u_{ki} = \frac{\vec{y}_i(k)/d(\vec{x}_i, \vec{w}_k)}{\sum_{j=1}^{l}\vec{y}_i(j)/d(\vec{x}_i, \vec{w}_j)} \quad (7)$$

To find the update for the cluster centers for the fixed u_{ki}, we need to take the gradient of Eq. 6 with respect to $\vec{w}_k$. Normally we cannot get closed form solutions and iterative techniques will be used. If f is even and finite, we have

$$\frac{\partial d(\vec{x}_i, \vec{w}_k)}{\partial w_{kj}} = \frac{(w_{kj} - x_{ij})^{f-1}}{\left(\sum_{r=1}^{d}(w_{kr} - x_{ir})^f\right)^{1-\frac{1}{f}}}, \quad j = 1, \ldots, d \quad (8)$$

Consequently, from $\frac{\partial \mathcal{L}}{\partial w_{kj}} = 0$ and using Eq. (8), we obtain

$$\sum_{i=1}^{n}(u_{ki})^2 \frac{(w_{kj} - x_{ij})^{f-1}}{\left(\sum_{r=1}^{d}(w_{kr} - x_{ir})^f\right)^{1-\frac{1}{f}}} = 0, \quad j = 1, \ldots, d \quad (9)$$

Eq. 9 can be solved by the well-known Gauss-Newton method [39]. In this study, we are also interested in evaluating the L_1 norm, where f = 1 by following the nonlinear constrained optimization procedures described in [40] for center updating.

The algorithm we just describe assigns soft relevance score (the membership) to each instance with respect to a label. Since it is developed for data with known and multiple labels, we name it multi-label FCMs (MLFCM). Like general FCMs, MLFCM consists of two iterative steps: (1) update the class membership u_{ki} based on Eq. 7 for current centers; and (2) update the centers $\vec{w}_k$ for the membership obtained from step 1 using Eq. 9 for even fuzzifiers or the standard optimization procedures for f = 1 [40]. Instead of randomly generating class centers, we take advantage of the known labels by using the sample means of each class as the starting centers, which guarantees that the final membership is not dependent on the initial selection of centers and that the algorithm converges quickly. By running MLFCM, we will obtain the relevance values u_{ki}, which will then be used in the *k*NN as voting factors, as described next.

3.2 Mr.KNN: Voting-Margin Ratio Method

A standard *k*NN method assigns class labels to a test instance based on the majority of its k nearest neighbors. In general, the voting function that relates an instance $\vec{x}_i$ to the j-th class label is defined as follows

$$vote(\vec{y}_i(j) = 1) = \sum_{\vec{x}_b \in N(\vec{x}_i)} \vec{y}_b(j) \quad (10)$$

When applied to our multi-label learning problems, however, two issues need to be addressed. The first issue is the imbalanced data distribution, as seen in Fig. 1. For example, for the yeast dataset, most of the neighbors will have labels 12 and 13. By majority voting, the majority of test data will be assigned to class labels 12 and 13. The second issue is that the voting defined in Eq. 10 does not take into account the distances between a test instance and its k nearest neighbors. To address these problems, we incorporate a distance weighting method [41] and the soft relevance u_{jb} derived from our MLFCMs. The new voting function is defined as

$$vote(\vec{y}_i(j) = 1) = \sum_{\vec{x}_b \in N(\vec{x}_i)} e^{-d(\vec{x}_i, \vec{x}_b)}\, u_{jb} \quad (11)$$

where the distance $d(\vec{x}_i, \vec{x}_b)$ is the *Minkowski* distance defined in Eq. 3.

To determine the optimal values of f in the *Minkowski* distance and k in the *k*NN, we introduce a new evaluation function, which is motivated by the well-known margin concept [42]. Consider a five-class learning problem with an instance belonging to two class labels: labels 2 and 3. Fig. 3 shows the diagram of voting: the instance is in the center (a plus symbol inside a circle); a circle represents a voting value for the label marked by the number inside a circle. For example, Fig. 3(a) indicates that the instance receives largest vote from label 2, followed by labels 3, 5, 4 and 1. While both Figs. (a) and (b) have largest votes for the true labels 2 and 3 the instance belongs to, we prefer f and k that produce Fig. 3(b), as it has a larger voting margin (marked by an arrow in Fig. 3), which is defined as the voting difference between the true label with smallest voting (e.g., label 3) and the false label with the largest voting (e.g., label 5). This can be formulated as follows

$$voting\ margin = \min_{j \in \mathcal{T}_i} vote\,(\vec{y}_i(j) = 1) - \max_{j \in \mathcal{F}_i} vote\,(\vec{y}_i(j) = 1)$$

where $\mathcal{T}_i$ and $\mathcal{F}_i$ represent the true label set and false label set for instance i, respectively. and Fig. 3(c) shows an example where the vote for the true label 3 is smaller than that for both labels 5 and 4, so the margin may be negative.

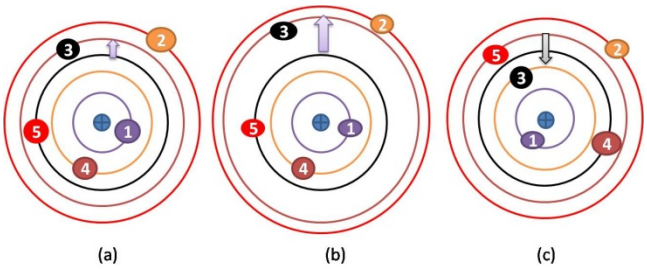

Figure 3. An illustrative diagram of voting margins, where the margins of voting are marked by arrow bars. (a) Correct voting, smaller margin; (b) correct voting, larger margin; and (c) true label 3 is lower than false labels (labels 3 and 5).

Our goal is to seek the combination of *f* and *k* that maximizes the average voting margin ratio, which is defined as the average ratio of voting between the true label with smallest voting (e.g., label 3) and the false label with the largest voting (e.g., label 5). The overall learning method for multi-label learning is called voting Margin Ratio *k*NN, or Mr.KNN.

Mr. KNN consists of two steps: training and test. The procedures are summarized in Fig. 4.

Training (offline):

Input: training data
Output: *f, k*, soft relevance for each training data (u_{jb})
th: threshold of voting to assign a label
for (each combination of *f* and *k*) **do**
{ MLFCM;
for (each training example) **do**
{ Identify *k* nearest neighbors
Compute votes for each label (Eq. 11)
Computer voting margin ratio
};
Computer average voting margin ratio
};

Save *f, k*, and u_{jb} with the largest average voting margin ratio
Leave-one-out-cross-validation to determine the threshold (*th*)

Test (online):

Input: test data
th, f, k, soft relevance for each training data (u_{jb})
Output: labels for each test data

for (each test data) **do**
{
Identify *k* nearest neighbors (in training data set)
Compute votes for each label (Eq. 11)
Label assignment
};

Figure 4. Mr.KNN for multi-label learning

Next, we apply the proposed method to three multi-label classification problems and to evaluate its performance.

4. EXPERIMENTAL RESULTS

Experimental results in [7, 36] showed that ML-KNN outperformed existing multi-label learning methods, such as BoosTexter [2] and Rank-SVM [12]. In this session, we conduct a comparative study between Mr.KNN and ML-KNN.

4.1 Data Description

Three commonly-used multi-label datasets are tested in this study. The first task is to predict gene functions of yeast. Each gene can have several functional categories and is characterized by microarray expression and phylogenetic profiles [12]. The second learning problem is about automatic detection of emotions in music, where each song is labeled using six categories: amazed-surprised, happy-pleased, relaxing-calm, quiet-still, sad-lonely, and angry-fearful [17]. The third problem is about semantic scene classification, where a natural scene may consist of objects from different labels [3].

Table 5 lists the statistics for the three datasets, where label cardinality is defined as the average number of labels per instance and label density is the average number of labels per example divided by the total number of unique labels. As seen, the yeast data have the largest label cardinality.

Table 5. Statistics for three multi-label datasets

Name	**Yeast**	**Emotion**	**Scene**
# of Instances	2417 (1500 training + 917 test)	593 (391 training + 202 test)	2407 (1211 training + 1196 test)
# of Features	103	72	294
# of Labels	14	6	6
Cardinality	4.237	1.869	1.074
Density	0.303	0.311	0.179

4.2 Evaluation Criteria

To evaluate the performance of learning methods, we choose four criteria commonly-used in multi-label classification: Hamming loss, accuracy, precision, and recall. Consider a test dataset $\{(\vec{x}_i, \vec{y}_i, \vec{z}_i)\}_{i=1}^{m}$, where a test instance $\vec{x}_i \in \mathfrak{R}^{\boldsymbol{d}}$, its class label vector $\vec{y}_i \in \{\mathbf{0}, \mathbf{1}\}^{\boldsymbol{l}}$, and the predicted label vector $\vec{z}_i \in \{\mathbf{0}, \mathbf{1}\}^{\boldsymbol{l}}$. Hamming loss is defined as follows:

$$hLoss = \frac{1}{m}\sum_{i=1}^{m}\frac{\sum_{j=1}^{d}\left(\vec{y}_i(j) \oplus \vec{z}_i(j)\right)}{l} \tag{12}$$

where $\oplus$ represents an exclusive OR (XOR) operation in Boolean logic. The other three measures are:

$$Precision = \frac{1}{m}\sum_{i=1}^{m}\frac{< \vec{y}_i, \vec{z}_i >}{< \vec{z}_i, \vec{z}_i >} \tag{13}$$

where $< u_i, v >$ is the inner product between two vectors *u* and *v*.

$$Recall = \frac{1}{m}\sum_{i=1}^{m}\frac{< \vec{y}_i, \vec{z}_i >}{< \vec{y}_i, \vec{y}_i >} \tag{14}$$

and

$$Accuracy = \frac{1}{m}\sum_{i=1}^{m}\frac{< \vec{y}_i, \vec{z}_i >}{\sum_{j=1}^{d}\left(\vec{y}_i(j) \odot \vec{z}_i(j)\right)} \tag{15}$$

where $\odot$ represents a logic OR operation in Boolean logic. In addition, we also introduce a ranking-based measure commonly-

used in learning to rank tasks [43]: normalized discounted cumulative gain (NDCG) [44]. In multi-label learning, we use NDCG to evaluate the final ranking of labels for each instance, not the binary label vector. In other words, for each instance, a label will receive a voting score. Ideally, these true labels will rank higher than false labels. The NDCG of a ranking list of labels at position n is defined as

$$N(n) = Z_n \sum_{i=1}^{n} \begin{cases} 2^{r(i)} - 1, & i = 1 \\ \dfrac{2^{r(i)} - 1}{\log(i)}, & i > 1 \end{cases} \quad (16)$$

where $r(i)$ is the ranked relevance of the label at position i (in our application, it is either zero or one) and the normalization constant Z_n is chosen so that the NDCG of a perfect ranking is 1.

4.3 Experimental Results

For each dataset, we select the f in the *Minkowski* distance from 1, 2, 4, and 6; and k in the kNN from 10, 15, 20, 25, 30, 35, 40, and 45, which results in 32 combinations of (f, k). The average voting margin ratio is used to choose the optimal parameters as described in Section 3. Fig. 5 shows an example of the average voting margin ratio versus (f, k) obtained from training data set of yeast. Apparently, there is a peak at $(f, k) = (2, 35)$, which is used as the optimal parameters for testing. Table 2 shows the test results for yeast. Mr.KNN outperforms ML-KNN in terms of the Hamming loss, accuracy, precision, and recall. For yeast data, since the cardinality is 4.237 (Table 5), we calculate the NDCG with $n \leq 6$. Fig. 6 shows the difference of NDCG for Mr.KNN and ML-KNN, which conforms to the results shown in Table 6.

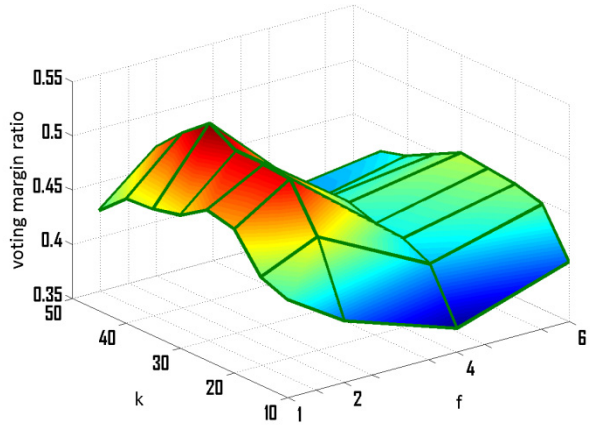

Figure 5. The average voting margin ratio versus f and k obtained from yeast training dat.

Table 6. Comparison of classification performance for yeast

Metric	ML-KNN	Mr.KNN
hLoss	0.229	0.217
Accuracy	0.485	0.533
Precision	0.630	0.660
Recall	0.628	0.695

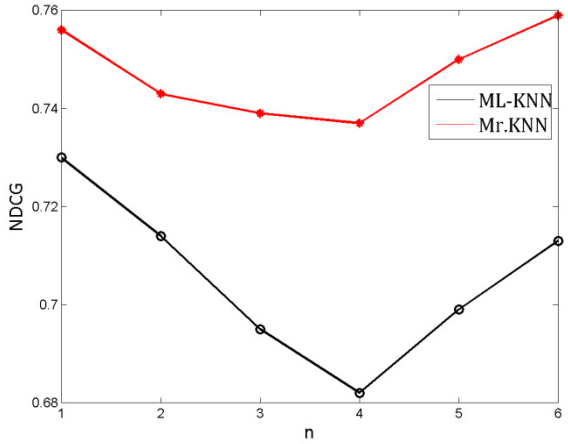

Figure 6. Compare ranking capability: NDCG(n) versus n for yeast data.

We also run the two KNN-based algorithms to emotion and scene datasets, both with smaller cardinality than the yeast data. Tables 7 and 8 show the results for emotion and scene datasets, respectively. Interestingly, both datasets show that ML-KNN produces smaller Hamming loss yet lower accuracy than Mr.KNN. One of the reasons is probably the data distribution. Unlike the yeast data, where few labels are dominant, both emotions and scene data are much balanced and evenly distributed, which is evident from Fig. 7 and Fig. 8. As such, the data imbalance problem is not severe with the binary relevance strategy. Even so, Mr.KNN still yields better accuracy, precision, and recall than ML-KNN, as shown in Tables 7 and 8. We do not compute the NDCG scores for these two datasets as their cardinalities are around one.

Table 7. Comparison of classification performance for emotions

Metric	ML-KNN	Mr.KNN
hLoss	0.218	0.242
Accuracy	0.505	0.562
Precision	0.690	0.625
Recall	0.576	0.795

Table 8. Comparison of classification performance for scene

Metric	ML-KNN	Mr.KNN
hLoss	0.099	0.109
Accuracy	0.668	0.693
Precision	0.700	0.726
Recall	0.704	0.740

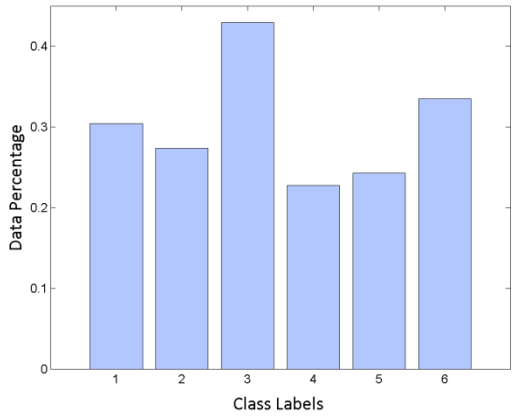

Figure 7. Sample distribution (training) for emotions data with 6 labels

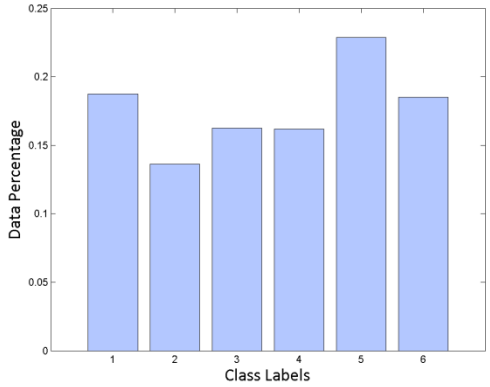

Figure 8. Sample distribution (training) for scene data with 6 class labels

To see the effect of number of labels on the performance of learning methods, we use the mediamill data [5], which are extracted from the generic video indexing problem. The original mediamill data set is highly imbalanced (see the distribution of training data in Fig. 9) with more than 40,000 instances and 101 unique labels. We randomly select 1,500 examples as training data and 500 as test data and evaluate learning of the top (most frequent) 10, 20, 30, 40, and 50 labels. Fig. 10 shows the data distribution of the top 50 labels in the new training set. The cardinality for each dataset is listed in Table 9.

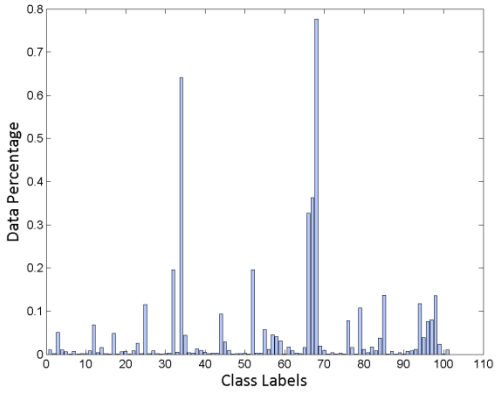

Figure 9. Sample distribution for mediamill data with 101 class labels

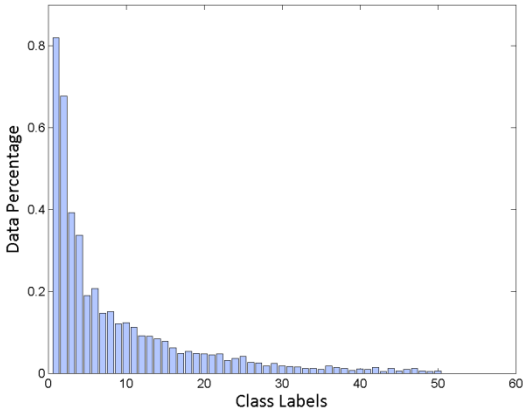

Figure 10. Sample distribution for randomly selected mediamill data with 50 class labels that are most frequent in the original dataset

Table 9. Label cardinality for the new mediamill datasets

Label	10	20	30	40	50
Cardinality	3.1627	3.8773	4.1873	4.3127	4.3940

Tables 10 and 11 list the classification results for ML-KNN and Mr.KNN, respectively. Mr.KNN consistently outperforms ML-KNN. As the number of labels increases, learning performance tends to decrease. We also plot the NDCG with $n = 6$ for 10 label cases in Fig. 11. Similar results are observed for 20-50 class labels.

As of computational complexity, Mr.KNN requires more time to train the model (mainly, time to compute the margin ratio) than ML-KNN. However, the training can be conducted offline and the online test time needed for both Mr.KNN and ML-KNN are almost the same.

Table 10. Classification performance for ML-KNN

label	10	20	30	40	50
hLoss	0.216	0.168	0.132	0.121	0.109
Accuracy	0.501	0.418	0.407	0.365	0.359
Precision	0.665	0.566	0.538	0.453	0.426
Recall	0.676	0.649	0.652	0.679	0.725

Table 11. Classification performance for Mr.KNN

label	10	20	30	40	50
hLoss	0.206	0.143	0.120	0.104	0.087
Accuracy	0.531	0.462	0.428	0.412	0.405
Precision	0.701	0.628	0.589	0.532	0.524
Recall	0.682	0.639	0.615	0.663	0.654

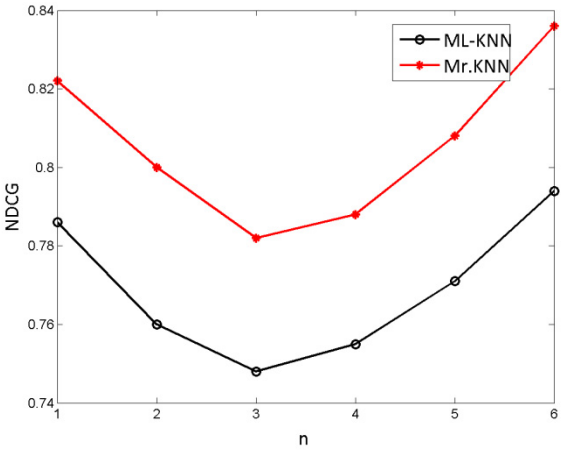

Figure 11. Compare ranking capability: NDCG(n) versus n.

5. CONCLUSION

Multi-label learning is attracting growing interest in information retrieval and data mining societies. Currently, most methods use the binary relevance strategy, which deals with one class label at a time. As analyzed in our study, an instance with multiple labels may be an outlier for some classes, which will degrade the performance of a classifier. In this paper, we introduce the soft relevance strategy, in which each instance is assigned a relevance score with respect to a label. This relevance score is related to the distance of the instance to centers of classes. Furthermore, it is used as a voting factor in a modified *k*NN algorithm. Evaluated over three commonly-used multi-label datasets, the proposed method outperforms ML-KNN (in terms of accuracy, precision, and recall).

Another factor that is not well studied in multi-label learning is the effect of number of unique labels on learning performance. In this paper, we investigate the learning of mediamill data with different number of unique labels. We show that when the number of unique labels increases, performance of the binary relevance-based ML-KNN method decreases. On the contrary, the soft relevance-based Mr.KNN produces similar results for different number of labels. Thus, the soft relevance strategy is appropriate for multi-label learning problems, especially for problems with a large number of labels and large cardinality.

Our future work will explore an efficient way for training with large scale data sets and also evaluate on different distance metrics including these for nominal data.

6. ACKNOWLEDGMENTS

The material is based upon work supported by the US National Science Foundation Award IIS-0644366.

7. REFERENCES

[1] McCallum, A. Multi-label text classification with a mixture model trained by EM. *AAAI 99 Workshop on Text Mining*, 1999.

[2] Schapire, R. and Singer, Y. BoosTexter: A boosting-based system for text categorization. *Machine learning*, 39, 2 2000), 135-168.

[3] Boutell, M., Luo, J., Shen, X. and Brown, C. Learning multi-label scene classification. *Pattern Recognition*, 37, 9(2004), 1757-1771.

[4] Qi, G., Hua, X., Rui, Y., Tang, J., Mei, T. and Zhang, H. Correlative multi-label video annotation. in *Proceedings of ACM Multimedia*, Augsburg, Bavaria, Germany, 2007.

[5] Snoek, C., Worring, M., Van Gemert, J., Geusebroek, J. and Smeulders, A. The challenge problem for automated detection of 101 semantic concepts in multimedia. in *Proceedings of ACM Multimedia*, 421-430, Santa Barbara, USA, 2006.

[6] Yang, S., Kim, S. and Ro, Y. Semantic home photo categorization. *IEEE Transactions on Circuits and Systems for Video Technology*, 17, 2007, 324-335.

[7] Zhang, M. and Zhou, Z. ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognition*, 40, 2007, 2038-2048.

[8] Barutcuoglu, Z., Schapire, R. and Troyanskaya, O. Hierarchical multi-label prediction of gene function. *Bioinformatics*, 22, 2006, 830.

[9] Blockeel, H., Schietgat, L., Struyf, J., Džeroski, S. and Clare, A. Decision trees for hierarchical multilabel classification: A case study in functional genomics. *Knowledge Discovery in Databases: PKDD 2006*, 18-29.

[10] Cesa-Bianchi, N., Gentile, C. and Zaniboni, L. Hierarchical classification: combining Bayes with SVM. in *Proceedings of the 23rd International Conference on Machine Learning*, Pittsburgh, PA, USA, 2006.

[11] Clare, A. and King, R. Knowledge discovery in multi-label phenotype data. *Lecture Notes in Cmputer Science*, 42-53, 2001, Springer.

[12] Elisseeff, A. and Weston, J. Kernel methods for multi-labelled classification and categorical regression problems. In *Advances in Neural Information Processing Systems* 14, 2001.

[13] Chen, X., Liu, M. and Ward, R. Protein function assignment through mining cross-species protein-protein interactions. *PLoS ONE*, 3, 2 2008).

[14] Liu, M., Chen, X. and Jothi, R. Knowledge-guided inference of domain-domain interactions from incomplete protein-protein interaction networks. *Bioinformatics*, 25, 19, 2009, 2492.

[15] Li, T. and Ogihara, M. Toward intelligent music information retrieval. *IEEE Transactions on Multimedia*, 8, 3 (2006), 564-574.

[16] Wieczorkowska, A., Synak, P. and Ra , Z. Multi-label classification of emotions in music. *Intelligent Information Processing and Web Mining,* 2006, 307-315.

[17] Trohidis, K., Tsoumakas, G., Kalliris, G. and Vlahavas, I. *Multilabel classification of music into emotions*. City, 2008.

[18] Zhang, Y., Burer, S. and Street, W. Ensemble pruning via semi-definite programming. *The Journal of Machine Learning Research*, 7, 2006, 1338.

[19] Tsoumakas, G., Katakis, I. and Vlahavas, I. Mining multi-label data. *Data Mining and Knowldege Discovery Handbook*, O. Maimon and L. Rokach (Ed.) , Sringer, 2nd edition, 20010.

[20] Chen, W., Yan, J., Zhang, B., Chen, Z. and Yang, Q. Document transformation for multi-label feature selection in text categorization. in *Proceedings of the Seventh IEEE International Conference on Data Mining*, Omaha, NE, 451-456, 2007.

[21] Read, J. A pruned problem transformation method for multi-label classification. in *Proceedings of the New Zealand Computer Science Research Student Conference*, Christchurch, New Zeaand, 2008.

[22] Tsoumakas, G. and Vlahavas, I. Random k-labelsets: An ensemble method for multilabel classification. *Machine Learning: ECML 2007*, 406-417.

[23] Vembu, S. and Gärtner, T. Label ranking algorithms: A survey. *Preference Learning. Springer,* 2009.

[24] Hüllermeier, E., Fürnkranz, J., Cheng, W. and Brinker, K. Label ranking by learning pairwise preferences. *Artificial Intelligence*, 172, 16-17 (2008), 1897-1916.

[25] Fürnkranz, J., Hüllermeier, E., Loza Mencía, E. and Brinker, K. Multilabel classification via calibrated label ranking. *Machine learning*, 73, 2 (2008), 133-153.

[26] Ghamrawi, N. and McCallum, A. *Collective multi-label classification*. in *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, 195-200, Bremen, Germany, 2005.

[27] Cai, L. and Hofmann, T. Hierarchical document categorization with support vector machines. in *Proceedings of the 13rd ACM International Conference on Information and Knowledge Management*, 78-87, Washington, DC, USA, 2004.

[28] Rousu, J., Saunders, C., Szedmak, S. and Shawe-Taylor, J. Kernel-based learning of hierarchical multilabel classification models. *The Journal of Machine Learning Research*, 7(2006), 1626.

[29] Zhang, M. and Zhou, Z. Multilabel neural networks with applications to functional genomics and text categorization. *IEEE Transactions on Knowledge and Data Engineering*, 18, 10 (2006), 1338-1351.

[30] Crammer, K. and Singer, Y. A family of additive online algorithms for category ranking. *The Journal of Machine Learning Research*, 3(2003), 1058.

[31] Ueda, N. and Saito, K. Parametric mixture models for multi-labeled text. *Advances in Neural Information Processing Systems*, 737-744, 2003.

[32] Gao, S., Wu, W., Lee, C. and Chua, T. A maximal figure-of-merit (MFoM)-learning approach to robust classifier design for text categorization. *ACM Transactions on Information Systems (TOIS)*, 24, 2, 218, 2006.

[33] Brinker, K. and Hüllermeier, E. Case-based multilabel ranking. in *Proceedings of the 20th International Conference on Artificial Intelligence* (IJCAI '07), 702-707, 2007.

[34] Spyromitros, E., Tsoumakas, G. and Vlahavas, I. An empirical study of lazy multilabel classification algorithms. in *Proceedings of 5th Hellenic Conference on Artificial Intelligence: Theories, Models and Applications*, 401-406, 2008.

[35] Duda, R., Hart, P. and Stork, D. *Pattern classification*. Wiley-Interscience, 2nd edition, 2001.

[36] Dimou, A., Tsoumakas, G., Mezaris, V., Kompatsiaris, I. and Vlahavas, I. An empirical study of multi-label learning methods for video annotation. *The 7th International Workshop on Content-Based Multimedia Indexing*, IEEE, Chania, Crete, 2009.

[37] Bezdek, J. *Pattern recognition with fuzzy objective function algorithms*. Kluwer Academic Publishers Norwell, MA, USA, 1981.

[38] Groenen, P. and Jajuga, K. Fuzzy clustering with squared Minkowski distances. *Fuzzy Sets and Systems*, 120, 2 (2001), 227-237.

[39] Fletcher, R. Practical Methods of Optimization: Vol. 2: Constrained Optimization. *JOHN WILEY & SONS, INC., ONE WILEY DR., SOMERSET, N. J. 08873, 1981, 224*1981).

[40] Bobrowski, L. and Bezdek, J. c-means clustering with the l_1 and l_∞ norms. *IEEE Transactions on Systems, Man, and Cybernetics*, 21, 3 (1991), 545-554.

[41] Shepard, R. N. Toward a universal law of generalization for psychological science. *Science*, 237, 4820 (Sep 11 1987), 1317-1323.

[42] Vapnik, V. *The nature of statistical learning theory*. Springer Verlag, 2000.

[43] Chen, X., Wang, H. and Lin, X. *Learning to rank with a novel kernel perceptron method*. in *Proceedings of the 18th ACM Conference on Information and Knowledge Management,* 505-512, 2009, Hong Kong, China.

[44] Järvelin, K. and Kekäläinen, J. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)*, 20, 4 (2002), 446.