Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 832347, 10 pages
http://dx.doi.org/10.1155/2014/832347

Research Article

Predicting Audience Location on the Basis of the *k*-Nearest Neighbor Multilabel Classification

Haitao Wu[1,2] **and Shi Ying**[1]

[1] *State Key Laboratory of Software Engineering, Computer School, Wuhan University, Wuhan 430072, China*
[2] *Software School, Huanghuai University, Zhumadian 463000, China*

Correspondence should be addressed to Haitao Wu; wwwht09@163.com

Received 10 June 2014; Revised 7 December 2014; Accepted 8 December 2014; Published 23 December 2014

Academic Editor: Elmetwally Elabbasy

Understanding audience location information in online social networks is important in designing recommendation systems, improving information dissemination, and so on. In this paper, we focus on predicting the location distribution of audiences on YouTube. And we transform this problem to a multilabel classification problem, while we find there exist three problems when the classical *k*-nearest neighbor based algorithm for multilabel classification (ML-*k*NN) is used to predict location distribution. Firstly, the feature weights are not considered in measuring the similarity degree. Secondly, it consumes considerable computing time in finding similar items by traversing all the training set. Thirdly, the goal of ML-*k*NN is to find relevant labels for every sample which is different from audience location prediction. To solve these problems, we propose the methods of measuring similarity based on weight, quickly finding similar items, and ranking a specific number of labels. On the basis of these methods and the ML-*k*NN, the *k*-nearest neighbor based model for audience location prediction (AL-*k*NN) is proposed for predicting audience location. The experiments based on massive YouTube data show that the proposed model can more accurately predict the location of YouTube video audience than the ML-*k*NN, MLNB, and Rank-SVM methods.

1. Introduction

According to sociology, people often show different characteristics because of their different cultural backgrounds, customs, and traditional sociocultural environments. These factors have a direct effect on the behavior of people in choosing personal information. Studies have shown that people with similar cultural backgrounds are likely to pay attention to information with similar contents [1]. Therefore, grasping the regional background of the user in an online social network can improve the effectiveness of information dissemination. For example, Guha et al. [2] found that user location significantly affects the advertisement content placed by Google and Facebook.

Currently the investigation on user location prediction in a social network mainly focuses on the user friends and the characteristic of information dissemination. Most studies use Facebook and Twitter as examples. On the basis of large-scale Facebook data, Backstrom et al. [3] analyze the relationship between friends and physical distance and find a negative correlation between them. The results show that a larger distance between two users corresponds to the lower probability of them becoming friends. On the basis of this finding, they propose an algorithm to predict the physical location of user. The experiment shows that the accuracy of predicting location by using this algorithm is significantly higher than that of the IP address method.

Several studies on user location prediction employ Twitter as an example. McGee et al. [4] consider not only the user's friends but also the interaction level between users to predict user location in Twitter. They first analyze the user relationship and physical distance in Twitter and find that users who have many fans tend to have a significant distance between each other, whereas users mentioned mutually are separated by a short distance. They then presented a model based on the decision tree for predicting user location. Rout et al. [5] hypothesize about the number of features of the user network in Twitter and transform the locating problem to a classification problem by using the support vector machine (SVM) classifier. Li et al. [6] construct a probability model for predicting user family location by using a microblog written by the user and the user's friends.

Instead of predicting location of the individual, we focus on the issue of predicting the location distribution of audience. We take YouTube, for example, to predict audience location, because YouTube is the largest online video sharing social network in the world. YouTube supports 61 languages, is visited by more than 1 billion visitors per month, and accumulates 80% of website traffic from other countries and regions outside the United States [7]. YouTube is the most representative and influential online social network for video sharing. Thus, the results obtained from YouTube have practical significance.

Given that obtaining the actual viewers of videos is difficult, the number of video comments is used as the number of audiences to predict the audience geographical position because studies have shown that YouTube video views and comments are highly correlated [8], and comments have been widely used to represent views [9, 10]. The countries or regions of audiences are used to represent the audience geographical location. Therefore, the question of this study is how to predict the n countries or regions with the largest number of video audiences. The traditional prediction or classification model can only have one predictive value (e.g., the linear regression and decision-tree classification model assigns only a label for a sample). While our goal is to assign a specific number of countries and regions to a video, the geographical position needs to be sorted according to the number of audiences.

To this end, we transform the location distribution prediction of YouTube video audiences to the question of the multilabel classification. When the classical k-nearest neighbor based algorithm for multilabel classification (ML-kNN) [11] is used to predict audience location, there exist three problems: (1) the difference of features is not considered in the ML-kNN when computing the similarity degree; (2) all objects in the training set are required to be traversed when seeking the k-nearest neighbors of the sample; however, large sizes of the training set will lead to tremendous computing workloads, and a small size will cause misclassification; (3) the goal of the multilabel classification method is not to solve the problem of audience location prediction: the ML-kNN method finds relevant labels for every sample, whereas our goal is to rank a specific number of relevant labels. To solve the above-mentioned problems, this paper provides the method of computing similarity based on weight and presents a method for quickly finding similar videos, and, on the basis of these two methods and the ML-kNN, the k-nearest neighbor based model for audience location prediction (AL-kNN) model is proposed for predicting audience location. Finally, experiments based on massive YouTube data show the performance of the proposed method.

2. Data Description

In this section, we describe the data collected from YouTube for the analyses and experiments in the paper. In order to learn the characters of audience, we need to know the information of videos, their uploaders, and viewers. Given that obtaining the actual viewers of videos is difficult, the commenters are used to represent viewers. The information is downloaded by YouTube APIs, and the details are shown in Table 1.

Table 1: Data description.

Names	Quantity
Videos	144,695
Video comments	51,354,025
Users including uploaders and commenters	15,153,442
Users with location information	14,906,800

Table 2: Selected countries of the videos.

Country	Region ID
United States	US
Great Britain	GB
Germany	DE
Brazil	BR
Canada	CA
Italy	IT
Spain	ES
Mexico	MX
Poland	PL
France	FR

Specifically, we firstly download the most popular or the latest video uploaded from different countries and regions. By this way, we obtain about 1 million videos IDs. And then the information of the uploaders and commenters of these videos is collected. By using the standard two-bit ISO country and area code in the user's profile, the country or region of the user is determined. Because videos with few commenters may not reflect the popularity of videos in every country, videos with less than 20 comments are excluded. Because the experiments in this paper are time-consuming, we further select videos whose uploaders belong to 10 countries (Table 2) with the largest population of users and also select commenters who belong to them. As a result a total of 144,695 videos are selected for the experiments.

3. Modeling Preliminaries

In this section we first define the problem of audience location prediction and explain how this problem is transformed into the problem of multilabel classification. The ML-kNN multilabel classification method is then introduced.

3.1. Audience Location Prediction Problem. In this study, the country or region of YouTube video audiences is used to represent the audience geographical position. Therefore, predicting audience position means predicting the rank of a number of countries or regions with the largest number of video audiences.

The traditional prediction or classification models cannot apply to our question, because they can only assign one label for a given sample, whereas this study needs to assign a number of ranked labels to a video. To solve this problem, we introduce the multilabel classification. Below, we will present how to transform the problem of audience location prediction into the problem of the multilabel classification.

Different from single label classification methods, such as SVM and decision tree, multilabel classification allows a sample to be classified to more than one class. Our goal is to predict a given number of countries and regions with the largest viewers for each video, that is, to assign a number of top countries and regions to each video according to the audience number. Therefore, multilabel classification can be used to predict audience location; that is, the video is considered as a classified sample and the country or region is considered as a label category. And the goal is to rank the countries and regions of the audiences according to the number of audiences and choose first n countries and regions for each video. The formal description of the problem is presented as follows.

Let $X \subseteq R^d$ be the sample space that is defined over the d-dimensional feature space; that is, X is the set of samples (videos). Every sample has d characteristic values, and let $L = \{1, 2, \ldots, M\}$ be the finite set of labels (countries of origin of audiences). Let $Y(x) \subseteq L$ be the first n countries with the largest number of video audiences over $x \in X$. Given the train set $D = \{(x_1, Y_1), (x_2, Y_2), \ldots, (x_m, Y_m)\}$, where $Y_i = Y(x_i)$ $(x_i \in X, Y_i \subseteq L)$, the goal of multilabel classification is to construct a classifier that can effectively predict the labeled set for each unknown sample; that is, the classifier can effectively select the first n countries with the largest number of video audiences from the candidate countries.

The method for solving our problems is the ranking classification method. Multilabel classification based on the ranking classification method would construct a binary real function $f : X \times L \rightarrow R$ in training process according to the train set. All labels are ranked by the value of $f(x_i, y)$ for any sample.

3.2. ML-kNN Method. This paper improves the ML-kNN method to predict audience location; therefore we firstly introduce the ML-kNN. By implementing multilabel classification by improving the kNN algorithm, the ML-kNN method confirms the final label set of training samples from the k-nearest labels by maximizing a posteriori probability. The ML-kNN method is described as follows.

For sample $x \in X$ and the corresponding label set $Y \subseteq L$, if l is the label of x, then $Y_x(l) = 1$; else $Y_x(l) = 0$. Let $N(x)$ be the set of x K-closest neighbors, and $C_x(l)$ is the number of x neighbors that belong to the lth class:

$$C_x(l) = \sum_{a \in N(x)} Y_a(l). \tag{1}$$

For the samples to be classified t, the ML-kNN method first finds the k-closest neighbors. $C_t(l)$ is then computed to predict the category of sample t according to $N(t)$. Let H_1^l indicate that sample t has label l and let H_0^l indicate that sample t does not have label l. E_j^l $(j \in \{0, 1, \ldots, k\})$ denotes that j samples in $N(t)$ have label l. $Y_t(l)$ can be obtained by using the MAP method:

$$Y_t(l) = \underset{b \in \{0,1\}}{\arg\max}\, P\left(H_b^l \mid E_{C_t(l)}^l\right). \tag{2}$$

The equation above can be transformed by using the Bayesian rule:

$$Y_t(l) = \underset{b \in \{0,1\}}{\arg\max}\, P\left(H_b^l\right) P\left(E_{C_t(l)}^l \mid H_b^l\right). \tag{3}$$

The prior probability and posterior probability are calculated according to the statistical frequency of the neighbor category in the training set.

4. The Model of Predicting Audience Location

The ML-kNN, which incorporates kNN and Bayesian rule to conduct multilabel classification, has been widely used. However, when ML-kNN is used in predicting audience location, there exist three problems. To solve these problems, we firstly propose method of similarity measurement based on weight and then present the algorithm of quickly finding similar videos. Finally, we modify select labels method of ML-kNN into ranking a specific number of relevant labels and incorporate the similarity measurement and quickly finding similar videos method into the ML-kNN method to build AL-kNN model for video audience location prediction.

4.1. The Method of Similarity Measurement Based on Weight. For the ML-kNN, the effect of finding similar items with the sample directly influences the accuracy of classification. To find similar items, the feature vector distance is used in general to calculate the similarity of two samples, such as the Euclidean distance and the cosine angle between the vectors. These methods consider all features with equal importance and do not consider the weight of each feature. However, the features of YouTube videos are different from the location distribution of audiences. For example, the data analysis results show that the position of a YouTube user can be closely related to the geographical position distribution of audiences but can be completely irrelevant to the user's gender. Therefore, we propose the method of similarity measurement in which the weight of each feature is considered.

The method of calculating similarity measurement based on weight mainly determines the weight of each feature according to the relationship between features and audience location. Specifically, the local similarity of each feature is firstly computed. Supposing that the ith feature values of feature vectors over two videos (i.e., u and v) are u_i and v_i, respectively, if the ith feature values u_i and v_i are continuous, we normalize u_i and v_i; that is, this feature value is divided by the maximum feature value over all videos. The distance of the ith feature is calculated as follows:

$$\operatorname{dict}(u_i, v_i) = \begin{cases} 0, & \text{if } d_i \text{ is discrete and } u_i = v_i, \\ 1, & \text{if } d_i \text{ is discrete and } u_i \neq v_i, \\ \sqrt{\left(u_i^n - v_i^n\right)^2}, & \text{if } d_i \text{ is continuous.} \end{cases} \tag{4}$$

And then the final similarity, similarity(u, v), between videos u and v can be calculated according to the distance between corresponding features of two videos:

$$\text{similarity}(u, v) = \sum_{i=1}^{d} w_i \cdot \text{dist}(u_i, v_i), \tag{5}$$

where w_i is the weight of each feature and $\sum_{i=1}^{d} w_i = 1$.

To determine the weights of each feature, the relationship between the feature and the location of video audiences should be analyzed, that is, to determine which features play key roles in the location distribution of audiences and quantify the relationship. If the audience location distribution of videos with the same value of the feature is similar, this feature is strongly related to the audience position of the video. Accordingly, this feature should have a large proportion in calculating the similarity degree of the video. On the basis of this idea, the specific calculation method of feature weighting is presented as follows.

Given d features, suppose there are H_t feature values for each feature F_t $(1 \leq t \leq d)$. And the video is placed in different sets R_k $(1 \leq k \leq H_t)$ according to the feature value. It should be noted that if the feature value is continuous, the feature value is piecewise processed; that is, all videos with the feature value in the same segment are placed in a set. The video in R_k then composes the video pair (u_t, v_t). Suppose that the first n countries with the largest number of audiences for these two videos are (C_i, C_j); the similarity S_{ij}^k for calculating these two sets is as follows:

$$S_{ij}^k = 2 \cdot \frac{|C_i \cap C_j|}{|C_i| + |C_j|}. \tag{6}$$

On the basis of the above-mentioned equation, the audience similarity M_k for calculating all videos in R_k is as follows:

$$M_k = \frac{1}{|R_k|} \sum_{i,j=1,i\neq j}^{|R_k|} S_{ij}^k. \tag{7}$$

The average similarity of the audience's country of videos with the feature F_t can be calculated as follows:

$$A_t = \frac{1}{H_t} \sum_{k=1}^{H_t} M_k. \tag{8}$$

The weight of the feature F_t is the proportion of its average similarity weight in all feature similarities:

$$w_t = \frac{A_t}{\sum_{p=1}^{d} A_p}. \tag{9}$$

4.2. The Algorithm for Quickly Finding Similar Videos. Before proposing the search algorithm, a corresponding analysis is first conducted to provide reference for designing efficient algorithm. Many characteristics of online social networks have shown a certain degree of homogeneity. For example, Wu et al. [12] find that Twitter users always pay more attention to the same user categories. Thelwall [13] found that MySpace users show obvious homogeneity in religion, nationality, and age. Therefore, the video audience position on YouTube is assumed to also show a certain degree of homogeneity; that is, for a seed user, videos of its closer neighbor have more similar audience position distribution with the seed user than its further neighbor. If this assumption holds, we can only search videos of close neighbors to find similar videos, instead of searching all the neighbors.

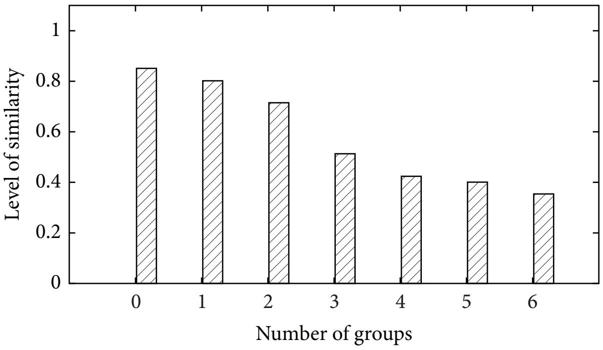


Figure 1: Relationship between the video and distance of uploaders.

To test this hypothesis, n^2 video pairs (V_i, V_j) from n videos are made and are then placed into different video groups according to the distance between the uploader and the viewer. In the zeroth group, the distance of the uploader of two videos is zero; that is, two videos are uploaded by the same user. In the first group, the distance of the uploader of two videos is one; that is, the uploader of a video is the direct neighbor of another video's uploader. In the second group, the distance of the uploader of two videos is two; that is, the uploader of a video is a two-hop neighbor of another video's uploader. The remaining steps are followed by analogy. The average value of the similarity in videos with the same group is then calculated for each group. The results are shown in Figure 1. The x-axis indicates the group number of the videos, and the y-axis is the average similarity value. Figure 1 illustrates that the similarity between video audience positions decreases with the increasing distance between uploaders. For example, the average similarity of the videos in group 0 is higher by nearly 50% than group 6. This result supports our proposed hypothesis; that is, a shorter distance between video uploaders corresponds to a higher similarity degree of video audience position. Therefore, instead of traversing all the videos, the videos possessed by the closer uploader are then searched emphatically when finding similar videos.

The analysis shows that a shorter distance between the user and its neighbor leads to a higher similarity between their videos. Therefore, instead of traversing all the videos, the videos uploaded by closer neighbors are searched emphatically when searching the k-nearest neighbors of the seed video. Generally, online social networks have the characteristic of a small world. Existing research also shows that the average path in online social networks is about 6 [14]. Hence, searching the six-hop friends of the video uploader

```
Input: the seed video and topology of the uploader's neighbors, searching hop number $m$,
        and threshold $p$
Output: the sorting output of the video set according to the similarity degree
(1)  for $i = 0$ to $m$ do
(2)      $S_i = \{the\ ith\ hop\ neighbors\ of\ uploader\}$
(3)      $V_i = \{all\ videos\ uploaded\ by\ users\ in\ S_i\}$
(4)      for each video $j$ in $V_i$ do
(5)          $V_{\text{all}} = V_{\text{all}} \cup \{j\}$
(6)          if $(|V_{\text{all}}| \geq p \cdot k)$
(7)              go to line (8)
(8)  Compute the similarity of each video in $V_{\text{all}}$ and seed video
(9)  Rank videos in $V_{\text{all}}$ based on their similarity
(10) Return $V_{\text{all}}$
```

Algorithm 1: Identifying the k-nearest neighbors.

is as complex as traversing the whole training set, while identifying the k-nearest neighbors is uncertain if only the one-hop friends of the uploader are searched. Therefore, the searching hop number m in designing the algorithm is variable, and this parameter should be determined according to the actual situation. At the same time, the threshold p about the number of searching videos is set up; that is, the searching process is stopped when the number of acquired videos exceeds $p \cdot k$, and the result is determined. The algorithm is described as in Algorithm 1.

In Algorithm 1, the m-hop neighbors of the uploader are traversed (line 1). The neighbor of each hop is placed in S_i (line 2), and the videos uploaded by each hop are placed in V_i (line 3). The videos achieved according to each hop are then accumulated in V_{all} (lines 4 and 5). If the number of videos exceeds $p \cdot k$, searching is halted; otherwise, the search is continued in the next hop (lines 6 and 7). The similarity degree between the seed video and other videos is calculated (line 8). Finally, the videos in V_{all} are then ranked and returned (lines 8 and 9).

4.3. The Improved Method Based on ML-kNN. On the basis of the above-mentioned method of similarity measurement and the algorithm of searching similar users, the ML-kNN method is improved for proposing audience location prediction based on k-nearest neighbor classification (AL-kNN). The detailed process is as follows.

(1) Calculation of the prior probabilities $P(H_0^l)$ and $P(H_1^l)$ of each label l:

$$P\left(H_1^l\right) = \frac{s + \sum_{i=1}^{n} \gamma_{x_i}(l)}{s \times 2 + n}, \qquad P\left(H_0^l\right) = 1 - P\left(H_1^l\right), \tag{10}$$

where $P(H_1^l)$ denotes the event of the sample containing label l and $P(H_0^l)$ denotes the event of the sample without label l. $\forall l \in L$, s is the preset smoothing exponential. $\gamma_x(l)$ indicates if label l belongs to the label set of sample x, that is, if yes, then $\gamma_x(l) = 1$ or $\gamma_x(l) = 0$.

(2) For the training sample x, the video similarity measurement based on weight and the algorithm of quickly finding similar videos are executed to search for its k-nearest neighbors in the training set, which are placed in set $N(x)$.

(3) Calculation of posterior probabilities $P(E_j^l \mid H_0^l)$ and $P(E_j^l \mid H_1^l)$: $P(E_j^l \mid H_0^l)$ denotes the conditional probability that when the training samples do not contain label l, there are exactly j samples from x's k-nearest neighbors containing label l, and $P(E_j^l \mid H_1^l)$ denotes the conditional probability that when the training samples contain label l, there are exactly j samples from x's k-nearest neighbors containing label l. Their computational formulas are as follows:

$$P\left(E_j^l \mid H_1^l\right) = \frac{s + c[j]}{s \times (k+1) + \sum_{r=0}^{k} c[r]}, \qquad P\left(E_j^l \mid H_0^l\right) = \frac{s + c'[j]}{s \times (k+1) + \sum_{r=0}^{k} c'[r]}, \tag{11}$$

where E_j^l, $j \in \{0, 1, \ldots, k\}$ denotes the event that j samples from the k-nearest neighbors of the training sample exactly contain label l. $c[j]$ denotes the number of the training samples j that exactly contain the label l from its k-nearest neighbors. $c'[j]$ denotes the number of the training samples j that exactly exclude the label l from its k-nearest neighbors.

(4) For the test sample t, the video similarity measurement based on weight and the algorithm of quickly finding similar videos are executed to search for t's k-nearest neighbors in the training set. The k-nearest neighbors are placed into $N(t)$, and the label membership vector p_t is then calculated:

$$p_t(l) = \frac{P\left(H_1^l\right) \cdot P\left(E_{C_t(l)}^l \mid H_1^l\right)}{\sum_{i\in\{0,1\}} P\left(H_1^l\right) \cdot P\left(E_{C_t(l)}^l \mid H_1^l\right)}, \quad \forall l \in L, \tag{12}$$

Table 3: Basic user features.

Feature name	YouTube term	Meaning
The number of subscriptions	Subscriptions	The number of publishers subscribing to other users
The number of videos being subscribed	Subscribers	The number of users subscribing to the publisher
The number of friends	Friends	The number of friends of the publisher
The number of videos	Uploads	The number of videos that are collected by the publisher
The audience population	Views	The number of viewing videos that are uploaded by the publisher
The number of videos being collected	Favorites	The number of videos that are collected by the publisher
Registration time	Published	Registration year of the publisher
Country and area	Country	Country and area wherein the publisher is located

Table 4: Extended uploader features.

Feature name	Meaning
Language	Dominant language of the country of origin of the publisher
The geographical distance	The distance between the capitals of the country of origin of the publisher
Cultural background	The cultural category of the country of origin of the publisher

where $C_t(l)$ records the sample number of $N(t)$ that contains label l. After sorting p_t, the specified number of labels is assigned to the test sample.

5. Feature Selections

To use AL-kNN to predict audience location, it needs extract features for the videos. This section mainly describes the selected features, including the publisher and basic video attributes obtained from YouTube APIs and the language, culture, and physical distance extended based on these basic attributes.

5.1. Basic Publisher Features. The basic features related to the publisher are first provided. The user information that can be downloaded by APIs is used as the features for predicting. The information includes gender, age, and registration time. The features are shown in Table 3.

5.2. The Extended Publisher Features. In addition to the basic features of video uploaders obtained directly from YouTube APIs, other relevant uploader features (e.g., culture background, language, and uploader distance) are described in this section.

According to the cultural background and geographical position [15], the selected 10 countries are divided into 3 groups. The first group is composed of European countries, including Spain, France, Great Britain, Germany, Italy, and Poland. The second group is composed of North American countries, including the United States and Canada. The third group is composed of South American countries, including Mexico and Brazil. On the basis of the language complexity of the different countries, the basic principle for determining the user language is that the official language in the country or region is considered the feature value of the user language. If no designated official language is provided, the most widely used languages in the area are considered the feature value of the user language. The distance between two uploaders is the distance between the two capital cities of their countries. The detailed features of the uploader are shown in Table 4.

5.3. Video Content Features. The video content feature mainly describes the relevant information of the video (Table 5). Some features, such as the number of audiences, number of comments, and video rate, can only be obtained after uploading videos, and we predict audience location before videos are published. Hence only three features that can be obtained before videos being published are selected for the video content features.

6. Experiment of Audience Location Prediction

This section presents the performance evaluation of the algorithm for quickly finding similar videos and the AL-kNN method of predicting audience location. The data presented in Section 2 is used for the experiments, and the features in Section 5 are computed and used for the experiments. It should be noted that the performance of AL-kNN can reflect the efficiency of the method of similarity measurement based on weight; therefore we do not evaluate the performance of the method of similarity measurement based on weight separately.

6.1. Evaluating Indicator. The common evaluating indicator of the multilabel classification effect mainly includes Hamming Loss, One Error, Coverage, Ranking Loss, and Average Precision [8]. Among these evaluating indicators, Hamming Loss is calculated according to the predicting label set, and the other four are calculated by using real functions in the corresponding method.

(1) Hamming Loss:

$$f_{\mathrm{HL}}(h) = \frac{1}{|T|}\sum_{i=1}^{|T|}\frac{1}{|L|}\left|h(x_i)\cdot\Delta L_i\right|, \tag{13}$$

Table 5: Basic video features.

Feature name	YouTube term	Meaning
The category	Category	The video is divided into 15 categories in YouTube, such as news and music
Tag	Tag	The tag specified by user video
Duration	Duration	The duration of the video in seconds
Descriptions	Descriptions	The description of the publisher introducing the video

where T denotes the set of training samples, L is the set all labels, $h(x_i)$ denotes the predicting label set of test samples x_i, L_i is the actual label set of x_i, and Δ denotes the symmetric difference of the two sets:

$$h(x_i)\,\Delta L_i = (h(x_i) - L_i) \cup (L_i - h(x_i)). \tag{14}$$

This indicator is used to calculate the inconsistency degree between the predicting label and the actual label of a multilabel classifier. A smaller value of this indicator indicates that the multilabel classifier has a better classification effect.

(2) One Error:

$$\text{One-error}_s(f) = \frac{1}{|T|}\sum_{i=1}^{|T|} H(x_i), \tag{15}$$

where T denotes the set of training samples. Thus,

$$H(x_i) = \begin{cases} 0, & \text{if } \arg\max_{l\in L} f(x_i,l) \in L_i, \\ 1, & \text{otherwise.} \end{cases} \tag{16}$$

This indicator is used to describe the probability of the label with the maximal membership value that is not in the actual label. A smaller value of this indicator also means that the multilabel classifier has a better classification effect.

(3) Coverage:

$$\text{Coverage}_s(f) = \frac{1}{|T|}\sum_{i=1}^{|T|} |C(x_i)| - 1. \tag{17}$$

$C(x_i)$ is defined as follows:

$$C(x_i) = \left\{ l \mid f(x_i,l) \ge f(x_i,l_i'),\, l \in L,\, l_i' = \arg\min_{l_{x_i}\in L_i} f(x_i,l_{x_i}) \right\}. \tag{18}$$

This indicator is used to calculate the average of the number of labels that descend from the label with a maximal membership value in the sorting function. The whole labels possessed by the sample will be covered. A smaller value of this indicator indicates that the multilabel classifier has a better classification effect.

(4) Ranking Loss:

$$\text{RL}_s(f) = \frac{1}{|T|}\sum_{i=1}^{|T|} \frac{1}{|L_i| \cdot |\overline{L_i}|} |R(x_i)|. \tag{19}$$

Here, $R(x_i) = \{(l_1,l_0) \mid f(x_i,l_1) \le f(x_i,l_0), (l_1,l_0) \in L_i \times \overline{L_i}\}$. This indicator is used to describe the probability of the membership value of the sample below the membership value of not being the sample. A smaller value of this indicator indicates that the multilabel classifier has a better classification effect.

(5) Average Precision:

$$\text{Aver-prec}_s(f) = \frac{1}{|T|}\sum_{i=1}^{|T|} \frac{1}{|L_i|} P(x_i). \tag{20}$$

$P(x_i)$ is defined as follows:

$$P(x_i) = \sum_{l_{x_i}\in L_i} \frac{\{l \mid f(x_i,l) \ge f(x_i,l_{x_i}), l \in L_i\}}{\{l \mid f(x_i,l) \ge f(x_i,l_{x_i}), l \in L\}}. \tag{21}$$

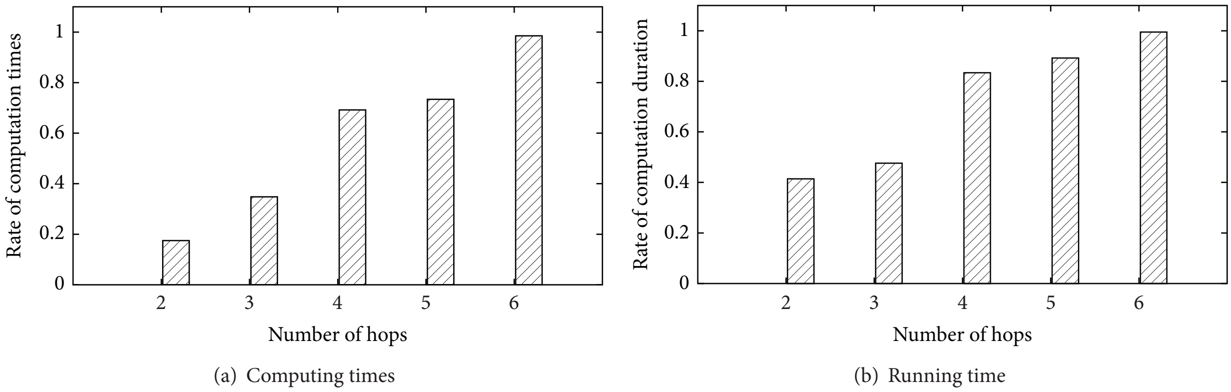
This indicator is used to calculate the average proportion of the label obtained by predicting the actual label after implementing the multilabel classification algorithm. In contrast to the four aforementioned indicators, a larger value of this indicator indicates that the multilabel classifier has a better classification effect.

From different angles, these five indicators evaluate the performance of the classifier constructed with different multilabel classification algorithms. Achieving the optimal effect for a classifier over these five indicators is difficult because the emphasis is different for each classifier, and the angle concentrated by each indicator is also different.

6.2. Performance Evaluation of the Searching Algorithm. In this section, the performance of the searching algorithm is evaluated by comparing the algorithm proposed in this paper and the algorithm of traversing in all videos from the angle of computing times, running time, and search result accuracy.

The number of the searching hops changes from two to six to evaluate the algorithm performance. Figure 2 shows the compared results of the computing times and running time, where the x-axis is the number of the searching hops m and the y-axis indicates the ratio of the computing times between our algorithm and the algorithm of traversing. The ratio increases with increasing m because the searching scope expands with increasing m. However, the computing times and running time of our algorithm significantly decrease when $m \le 3$. The ratio is only 27% when $m = 3$.

The effect of searching videos with similar audience location is compared in Figure 3. The x-axis is the number of the searching hops m, and the y-axis indicates the ratio of the number of elements in the set $U \cap V$ to the number of all the similar videos. The set U is the video set obtained by using



(a) Computing times

(b) Running time

Figure 2: Performance ratio of the proposed algorithm to traversing method.

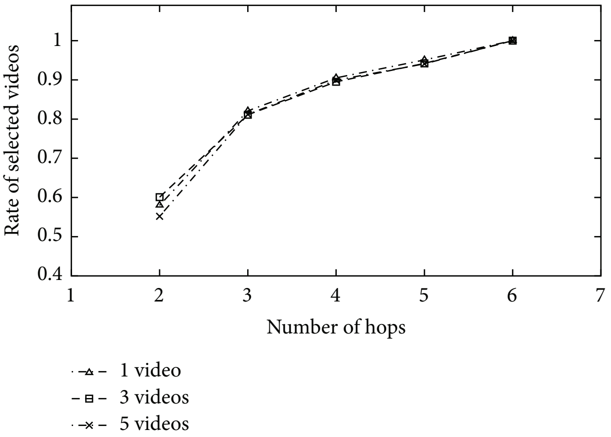


Figure 3: Effect ratio of the proposed algorithm to traversing method.

the proposed algorithm, and set V is the video set obtained by using traversing algorithm; that is, the y-axis denotes the following value:

$$\text{rate} = \frac{|U \cap V|}{|V|}. \tag{22}$$

Figure 3 illustrates that three curves almost overlap when selecting different numbers of similar videos, thus indicating that the proposed algorithm is capable of achieving a similar search performance when selecting different number of similar videos. The ratio of three different numbers of videos is relatively low only when $m = 2$. However, the ratio exceeds 80% when $m \geq 3$. Figures 2 and 3 show that when $m = 3$ our proposed algorithm can significantly reduce computing times and obtain the expected searching performance as the traversing algorithm. Therefore, subsequent experiments are made under the condition $m = 3$.

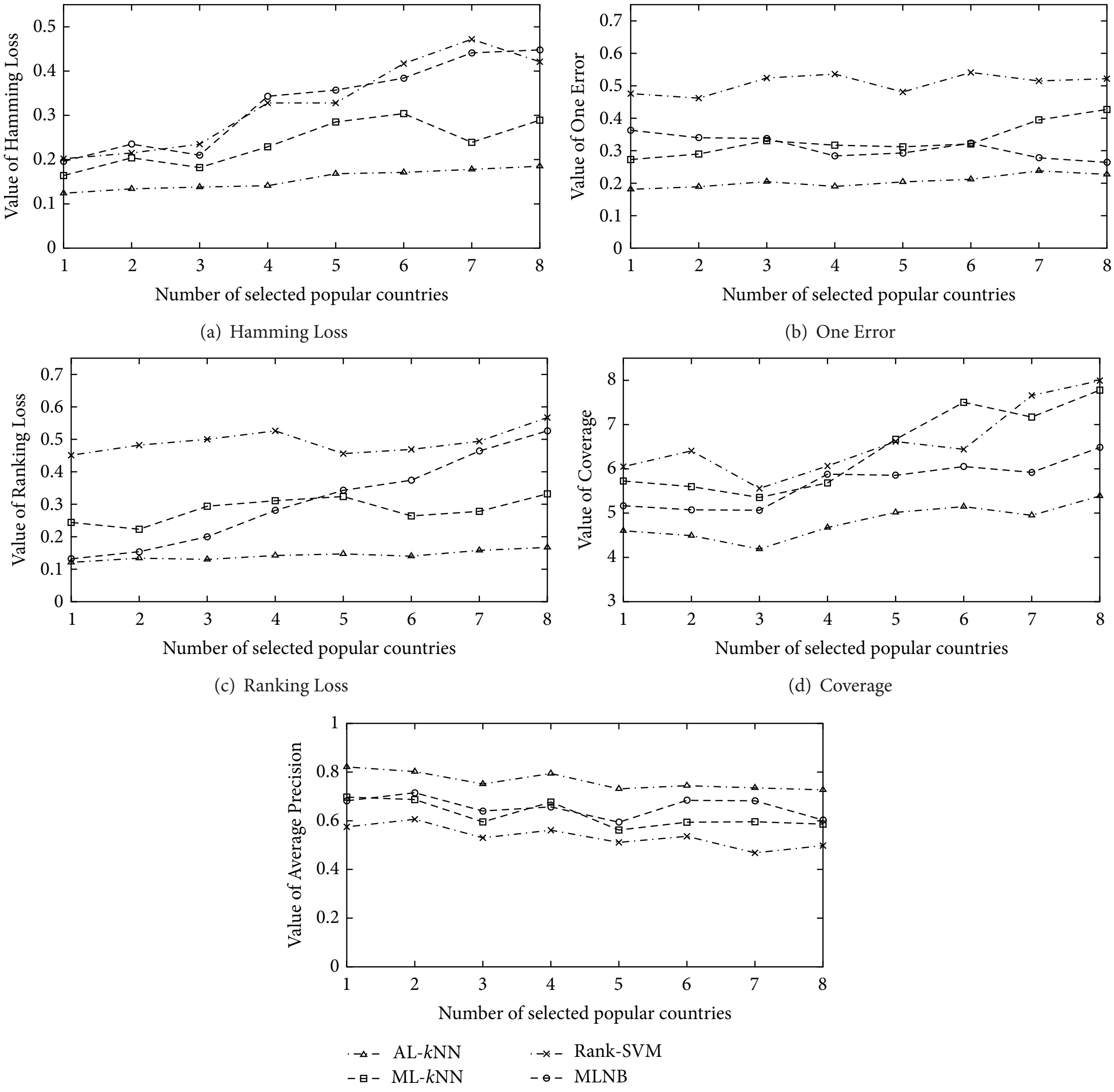
6.3. Predicting Performance with Different Number of Neighbors. In this section, the experiments are conducted to evaluate the performance of AL-kNN when the number of the selected closest neighbors (k) varies. The first 5 countries are chosen; that is, each video is assigned to 5 labels. Experiments are conducted when k varies from 1 to 20, and a part of better results is given in Table 6. Less performance difference occurs when the k value varies, and no one value achieves the maximum performance for all indicators. After comprehensive comparison, the overall performance is relatively better when $k = 7$. Therefore, subsequent experiments are made under $k = 7$.

Table 6: Performance comparison with different k value.

k	Hamming Loss	One Error	Ranking Loss	Coverage	Average Precision
5	0.187	0.221	0.169	5.714	0.69
6	0.179	0.231	0.152	5.170	0.702
7	0.168	0.204	0.147	5.015	0.731
8	0.174	0.212	0.159	5.102	0.724
9	0.191	0.201	0.701	5.014	0.721

6.4. Predicting Performance with Different Number of Countries. In this section, the performance of the classification model over five different indicators is examined when the number of countries that will be assigned to videos varies. For each video, predicting its audience position means selecting the first n countries with the largest number of audiences from the candidate countries. Here we want to observe the performance when n changes from 1 to 8. We evaluate AL-kNN by comparing with three common multilabel classification methods rank support vector machine (Rank-SVM) [16], multilabel naive Bayes (MLNB) [17], and ML-kNN. To conduct the predictive experiments, the videos are divided into 50% training set and 50% test set in the experiment.

The results of evaluation are shown in Figure 4 where x-axis is the number of countries which is assigned to videos and y-axis indicates the predictive performance. It shows different methods differ over the performance. For example, when Hamming Loss is used, the performance of the AL-kNN method is close to the ML-kNN method, and the ML-kNN method exceeds the Rank-SVM. By contrast, when Coverage is used, the AL-kNN method exceeds the ML-kNN.



(a) Hamming Loss

(b) One Error

(c) Ranking Loss

(d) Coverage

(e) Average Precision

Figure 4: Prediction performance with different numbers of countries.

The ML-*k*NN method is close to Rank-SVM. However, with regard to the use of these five indicators, the overall prediction performance of the AL-*k*NN method is superior to the Rank-SVM, ML-*k*NN, and MLNB methods. Therefore the experiment shows AL-*k*NN can achieve better performance in predicting audience location.

7. Conclusions

On the basis of the ML-*k*NN, the model of predicting audience location is proposed in this paper. The problem of predicting audience location distribution of YouTube video is transformed as a multilabel classification problem. First, in terms of the problem that feature weight is not considered for measuring the similarity degree in ML-*k*NN, the method of measuring the video similarity degree on the basis of weight is introduced. And then a method to calculate feature weight is also presented. In terms of the problem that the ML-*k*NN method takes more time to find similar items, the algorithm of quickly finding similar videos based on friend relationship of video owners is proposed. Finally, based on these two methods, the ML-*k*NN method was improved to solve the problem of audience location prediction. The experiments based on massive YouTube data show that the method introduced in this paper can more accurately predict the audience location of YouTube video, compared with the Rank-SVM, ML-*k*NN, and MLNB methods.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] C. Xiao, L. Su, J. Bi, Y. Xue, and A. Kuzmanovic, "Selective behavior in online social networks," in *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI '12)*, pp. 206–213, December 2012.

[2] S. Guha, B. Cheng, and P. Francis, "Challenges in measuring online advertising systems," in *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement (IMC '10)*, pp. 81–87, ACM, New York, NY, USA, November 2010.

[3] L. Backstrom, E. Sun, and C. Marlow, "Find me if you can: improving geographical prediction with social and spatial proximity," in *Proceedings of the 19th International World Wide Web Conference (WWW '10)*, pp. 61–70, ACM, Raleigh, NC, USA, April 2010.

[4] J. McGee, J. Caverlee, and Z. Cheng, "Location prediction in social media based on tie strength," in *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management (CIKM '13)*, pp. 459–468, ACM Press, San Francisco, Calif, USA, November 2013.

[5] D. Rout, K. Bontcheva, D. Preoţiuc-Pietro, and T. Cohn, "Where's @wally?: a classification approach to geolocating users based on their social ties," in *Proceedings of the 24th ACM Conference on Hypertext and Social Media (HT '13)*, pp. 11–20, Paris, France, May 2013.

[6] R. Li, S. Wang, H. Deng, R. Wang, and K. C.-C. Chang, "Towards social user profiling: unified and discriminative influence model for inferring home locations," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '12)*, pp. 1023–1031, ACM, Beijing, China, August 2012.

[7] YouTube, Statistics, http://www.youtube.com/yt/press/statistics.html.

[8] G. Chatzopoulou, C. Sheng, and M. Faloutsos, "A first step towards understanding popularity in YouTube," in *Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM '10)*, pp. 1–6, IEEE, San Diego, Calif, USA, March 2010.

[9] C. Xiao, F. Zhou, and Y. Wu, "Predicting audience gender in online content-sharing social networks," *Journal of the American Society for Information Science and Technology*, vol. 64, no. 6, pp. 1284–1297, 2013.

[10] A. Tatar, J. Leguay, P. Antoniadis, A. Limbourg, M. D. de Amorim, and S. Fdida, "Predicting the popularity of online articles based on user comments," in *Proceedings of the International Conference on Web Intelligence, Mining and Semantics (WIMS '11)*, pp. 67–75, 2011.

[11] M.-L. Zhang and Z.-H. Zhou, "ML-KNN: a lazy learning approach to multi-label learning," *Pattern Recognition*, vol. 40, no. 7, pp. 2038–2048, 2007.

[12] S. Wu, J. M. Hofman, W. A. Mason, and D. J. Watts, "Who says what to whom on twitter," in *Proceedings of the 20th International Conference on World Wide Web (WWW '11)*, pp. 705–714, April 2011.

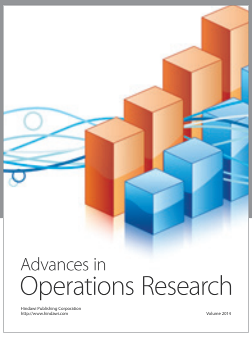
[13] M. Thelwall, "Homophily in myspace," *Journal of the American Society for Information Science and Technology*, vol. 60, no. 2, pp. 219–231, 2009.

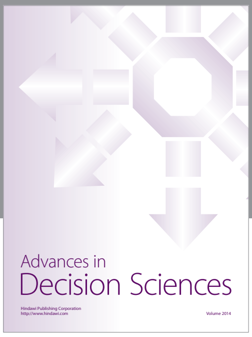
[14] S. Lattanzi, A. Panconesi, and D. Sivakumar, "Milgram-routing in social networks," in *Proceedings of the 20th International Conference on World Wide Web (WWW '11)*, pp. 725–734, ACM, New York, NY, USA, April 2011.

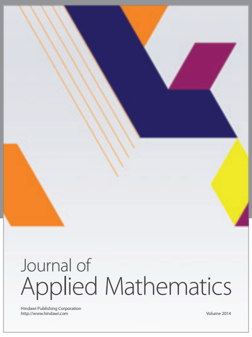
[15] S. Sundqvist, L. Frank, and K. Puumalainen, "The effects of country characteristics, cultural similarity and adoption timing on the diffusion of wireless communications," *Journal of Business Research*, vol. 58, no. 1, pp. 107–110, 2005.

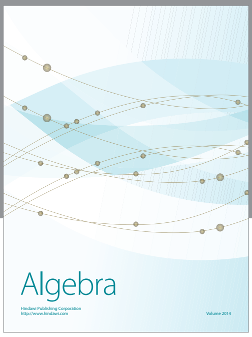
[16] A. Jiang, C. Wang, and Y. Zhu, "Calibrated rank-SVM for multi-label image categorization," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '08)*, pp. 1450–1455, Hong Kong, June 2008.

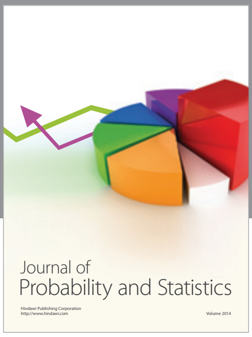
[17] M.-L. Zhang, J. M. Peña, and V. Robles, "Feature selection for multi-label naive Bayes classification," *Information Sciences*, vol. 179, no. 19, pp. 3218–3229, 2009.


Advances in
Operations Research

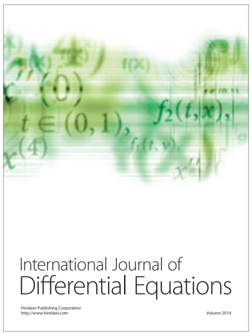

Advances in
Decision Sciences


Journal of
Applied Mathematics

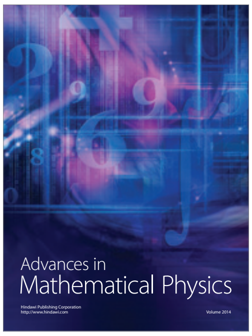

Algebra

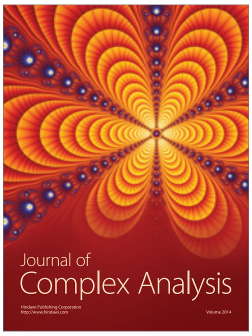

Journal of
Probability and Statistics

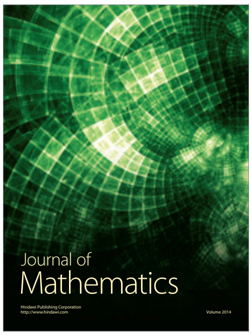
The Scientific
World Journal

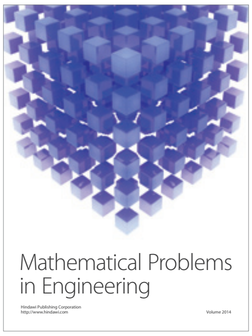

International Journal of
Differential Equations

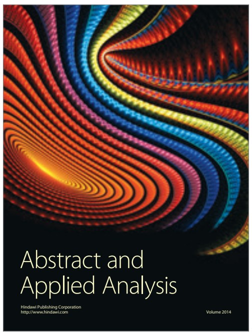
International Journal of
Combinatorics

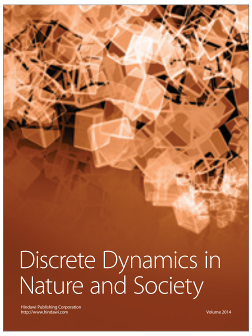

Advances in
Mathematical Physics

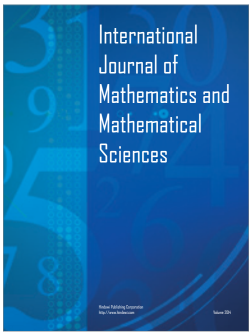

Journal of
Complex Analysis


Journal of
Mathematics


Mathematical Problems
in Engineering

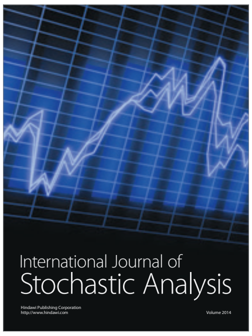

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


International Journal of
Stochastic Analysis

Journal of
Optimization