Expert Systems with Applications 39 (2012) 2813–2821

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

FSKNN: Multi-label text categorization based on fuzzy similarity and *k* nearest neighbors ☆

Jung-Yi Jiang, Shian-Chi Tsai, Shie-Jue Lee *

Department of Electrical Engineering, National Sun Yat-Sen University, Kaohsiung 804, Taiwan

ARTICLE INFO

Keywords:
Document classification
Multi-label classification
Fuzzy similarity measure
k-nearest neighbor algorithm
Maximum a posteriori estimate

ABSTRACT

We propose an efficient approach, FSKNN, which employs fuzzy similarity measure (FSM) and *k* nearest neighbors (KNN), for multi-label text classification. One of the problems associated with KNN-like approaches is its demanding computational cost in finding the *k* nearest neighbors from all the training patterns. For FSKNN, FSM is used to group the training patterns into clusters. Then only the training documents in those clusters whose fuzzy similarities to the document exceed a predesignated threshold are considered in finding the *k* nearest neighbors for the document. An unseen document is labeled based on its *k* nearest neighbors using the maximum a posteriori estimate. Experimental results show that our proposed method can work more effectively than other methods.

1. Introduction

Multi-label text classification plays an important role in information retrieval, text processing, and web search (Baeza-Yates & Ribeiro-Neto, 1999; Boutell, Luo, Shen, & Brown, 2004; Elisseeff & Weston, 2002; Salton & McGill, 1983). In multi-label text classification, a document can belong to more than one category. For example, a newspaper article concerning the reactions of the scientific circle to the release of the Da Vinci Code film can be classified to any of the three classes: arts, science, and movies. Most machine learning algorithms, such as Rocchio's method (Rocchio, 1971), *k*-nearest neighbor classifiers (Aha, 1997; Hull, 1994; Mitchell, 1997; Tan, 2005, 2006; Yang, 1997; Yang & Chute, 1994), probabilistic Bayesian models (Good, 1965; Joachims, 1997; Lewis & Ringuette, 1994), decision trees (Fuhr & Buckley, 1991; Quinlan, 1986, 1993), decision rules (Apté, Damerau, & Weiss, 1994; Cohen & Singer, 1999), and support vector machines (Dumais, Platt, Heckerman, & Sahami, 1998; Joachims, 1998), were designed for single-label classification in which a document can only belong to one category. To deal with multi-label text classification, two approaches are adopted (Tsoumakas & Katakis, 2007): (i) problem transformation, by which a multi-label text classification task is transformed into several single-label classification tasks for which single-label classification methods can be applied, and (ii) algorithm adaptation, which concerns extending specific single-label classification algorithms in order to handle multi-label data directly.

A popular problem transformation method, called binary relevance, was proposed in Tsoumakas and Katakis (2007). Binary relevance transforms the original data set into *p* data sets where *p* is the number of categories associated with the original data set. Each resulting data set contains all instances of the original data set with only two labels, 'belonging to' or 'not belonging to' a particular category. Since the resulting data sets are single-labeled, all single-label classification techniques are applicable to them. However, redundant data are generated, which may cause a drop on the efficiency of training. Besides, this kind of methods does not consider the correlations between different labels of each instance and the expressive power of such a system can be weak (Elisseeff & Weston, 2002; McCallum, 1999; Schapire & Singer, 2000).

Several approaches adaptively designed for multi-label classification tasks have been proposed. McCallum (1999) assumed a mixture probabilistic model for each category to generate each document and used the EM (Dempster, Laird, & Rubin, 1977) algorithm to learn the parameters in each model. Schapire and Singer (2000) proposed a boosting-based system, named BoosTexter, which is extended from AdaBoost (Freund & Schapire, 1997) and specifically intended for multi-label data. Comité, Gilleron, and Tommasi (2003) extended the method of Schapire and Singer (2000) and produced sets of rules that can be viewed as alternating decision trees (Freund & Mason, 1999). Zhang and Zhou (2006) proposed a multi-label version of BP neural network, named BP-MLL, which employs an error function to capture the characteristics of multi-label learning. In Zhang and Zhou (2007), a lazy learning algorithm, named MLKNN (multi-label *k*-nearest neighbor), was presented. MLKNN is derived from the *k*-nearest neighbor (KNN)

☆ This work was supported by "Aim for the Top University Plan" of the National Sun Yat-Sen University and Ministry of Education, and the National Science Council under the Grants NSC-97-2221-E-110-048-MY3 and NSC-98-2221-E-110-052.

* Corresponding author.
E-mail address: leesj@mail.ee.nsysu.edu.tw (S.-J. Lee).

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.08.141

algorithm (Aha, 1997). For each unseen instance, its k nearest neighbors in the training set are identified. Then, according to the statistical information obtained from the label sets of these neighboring instances, maximum a posteriori (MAP) principle is utilized to determine the label set for the unseen instance. MLKNN was shown (Zhang & Zhou, 2007) to perform better than some other well-established multi-label classification methods.

Although MLKNN can perform multi-label classification task, the computational power required for finding the k nearest neighbors is prohibitively large. We propose an efficient approach, FSKNN, which employs fuzzy similarity measure (FSM) and k nearest neighbors (KNN), for multi-label text classification. FSM (Saracoğlu, Tütüncü, & Allahverdi, 2008; Widyantoro & Yen, 2000) is used to group the training patterns into clusters. Then only the training documents in those clusters whose fuzzy similarities to the document exceed a predesignated threshold are considered in finding the k nearest neighbors for the document. An unseen document is labeled based on its k nearest neighbors using the maximum a posteriori estimate. Experimental results show that our proposed method can work more effectively than other methods. The rest of this paper is organized as follows. The problem to be solved is stated in Section 2. Our proposed method is described in Section 3. An illustrating example is given in Section 4. Experimental results are presented in Section 5. Finally, concluding remarks are given in Section 6.

2. Learning for multi-label classification

In a multi-label text classification problem, we are given a triplet (D,T,C) where

$$D = \{(\mathbf{d}_1,\mathbf{y}_1),(\mathbf{d}_2,\mathbf{y}_2),\ldots,(\mathbf{d}_\ell,\mathbf{y}_\ell)\} \tag{1}$$

is a set of ℓ training patterns, $T = \{t_1,t_2,\ldots,t_m\}$ is a set of m terms, and $C = \{c_1,c_2,\ldots,c_p\}$ is a set of p categories. T contains all the keywords selected for $\mathbf{d}_1,\mathbf{d}_2,\ldots,\mathbf{d}_\ell$. The ith training pattern consists of two components, document $\mathbf{d}_i$ and label-vector $\mathbf{y}_i$. A document $\mathbf{d}_i$, $1 \leqslant i \leqslant \ell$, is represented as a vector $\langle w_{1i},w_{2i},\ldots,w_{mi}\rangle$ where w_{ji} is the term frequency of term t_j, i.e., the number of occurrences of t_j, in document $\mathbf{d}_i$. Each document can belong to one or more than one category in C. For each $\mathbf{d}_i$, $1 \leqslant i \leqslant \ell$, $\mathbf{y}_i$ is a vector with p components, i.e., $\mathbf{y}_i = \langle y_{1i},y_{2i},\ldots,y_{pi}\rangle$, and

$$y_{ji} = \begin{cases} 1, & \text{if } \mathbf{d}_i \text{ belongs to } c_j \\ 0, & \text{if } \mathbf{d}_i \text{ does not belong to } c_j \end{cases} \tag{2}$$

for $1 \leqslant j \leqslant p$. Note that for single-label classification, only one component in $\mathbf{y}_i$ is 1, but for multi-label classification, several components may be 1 in $\mathbf{y}_i$. For example, $\mathbf{y}_i = \langle 0,1,0,1\rangle$ indicates that $\mathbf{d}_i$ belongs to categories c_2 and c_4 simultaneously.

To classify an unseen document $\mathbf{d}^t$ based on the given (D,T,C), one can create p datasets, $D_1,D_2,\ldots,D_p$, by binary relevance (Tsoumakas & Katakis, 2007) such that

$$\begin{cases} (\mathbf{d}_i,+1) \in D_j, & \text{if } y_{ji} = 1 \\ (\mathbf{d}_i,-1) \in D_j, & \text{if } y_{ji} = 0 \end{cases} \tag{3}$$

for $1 \leqslant i \leqslant \ell$ and $1 \leqslant j \leqslant p$. Then apply any single-label classification algorithm to $D_1,D_2,\ldots,D_p$, respectively. If the value returned from D_j is +1, then $\mathbf{d}^t$ belongs to category c_j. In this way, $\mathbf{d}^t$ can be classified to multiple categories. However, this approach has several disadvantages as mentioned previously. An alternative way to classify $\mathbf{d}^t$ based on the given (D,T,C) is to adapt specific single-label classification algorithms to handle multi-label data directly. MLKNN (Zhang & Zhou, 2007) is one example, which was derived from the KNN algorithm (Aha, 1997).

3. Our method

As mentioned, MLKNN (Zhang & Zhou, 2007) is an adaptation of the KNN algorithm (Aha, 1997) for multi-label text classification. One of the problems associated with KNN-like approaches is its demanding computational cost in finding the k nearest neighbors from all training patterns. We employ fuzzy similarity measure (Saracoğlu et al., 2008; Widyantoro & Yen, 2000) to calculate the similarity between a document and each category of training documents, and group the training patterns into clusters. Then only the training documents in those clusters whose fuzzy similarities to the document exceed a predesignated threshold are considered in finding the k nearest neighbors for the document. An unseen document is labeled based on its k nearest neighbors using the maximum a posteriori estimate. As a result, the performance of classification is greatly improved.

Let $N^t = \{\mathbf{d}_{v_1},\mathbf{d}_{v_2},\ldots,\mathbf{d}_{v_k}\}$ be the set of the k nearest neighbors for the unseen document $\mathbf{d}^t$, and $\mathbf{n}^t = \left\langle n_1^t,n_2^t,\ldots,n_p^t\right\rangle$ be the *label-count vector* for $\mathbf{d}^t$ where

$$n_j^t = \sum_{r=v_1}^{v_k} y_{jr} \tag{4}$$

for $1 \leqslant j \leqslant p$. We determine which categories $\mathbf{d}^t$ belongs to by calculating the label-vector $\mathbf{y}^t = \left\langle y_1^t,y_2^t,\ldots,y_p^t\right\rangle$ of $\mathbf{d}^t$ using maximum a posteriori (MAP) estimate as follows:

$$y_j^t = \begin{cases} 1, & \text{if } P\left(H_j = 1|E = n_j^t\right) > P\left(H_j = 0|E = n_j^t\right) \\ 0, & \text{if } P\left(H_j = 0|E = n_j^t\right) > P\left(H_j = 1|E = n_j^t\right) \\ R[0,1], & \text{otherwise} \end{cases} \tag{5}$$

for $1 \leqslant j \leqslant p$, where H_j is the random variable for belonging to category c_j or not ($H_j = 1$ for yes and $H_j = 0$ for no), E is the variable for the number of documents in N^t belonging to category c_j, and $R[0,1]$ indicates 0 or 1 chosen by random. By Bayes' rule (Alpaydin, 2004), we have

$$P\left(H_j = b|E = n_j^t\right) = \frac{P(H_j = b)P\left(E = n_j^t|H_j = b\right)}{P\left(E = n_j^t\right)} \tag{6}$$

for $b = 0, 1$. Therefore, Eq. (5) becomes

$$y_j^t = \begin{cases} 1, & \text{if } P(H_j=1)P\left(E=n_j^t|H_j=1\right) > P(H_j=0)P\left(E=n_j^t|H_j=0\right) \\ 0, & \text{if } P(H_j=0)P\left(E=n_j^t|H_j=0\right) > P(H_j=1)P\left(E=n_j^t|H_j=1\right) \\ R[0,1], & \text{otherwise} \end{cases} \tag{7}$$

for $1 \leqslant j \leqslant p$. Obviously, to calculate y_j^t we have to find N^t, and compute $P(H_j)$ and $P(E|H_j)$ for $1 \leqslant j \leqslant p$. Those that are independent of $\mathbf{d}^t$ can be done in the training stage and the others will be done in the testing stage, as explained below.

3.1. Training stage

Two things are done in this stage. One is to group the training documents into clusters. The other is to compute the priors and likelihoods.

3.1.1. Grouping training patterns into clusters

We group training documents $\mathbf{d}_1,\mathbf{d}_2,\ldots,\mathbf{d}_\ell$ into p clusters based on fuzzy similarity measure (Saracoğlu et al., 2008; Widyantoro & Yen, 2000). Let $dt(t_i,c_j)$ and $dd(t_i,c_j)$ be the distributions of term t_i over category c_j, defined as