Knowledge-Based Systems 54 (2013) 298–309

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Feature selection via maximizing global information gain for text classification

Changxing Shang [a,b,c,*], Min Li [a,b], Shengzhong Feng [a], Qingshan Jiang [a], Jianping Fan [a]

[a] Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China
[b] Graduate School of Chinese Academy of Sciences, Beijing 100080, China
[c] Zhengzhou Institute of Information Science and Technology, Zhengzhou 450001, China

ARTICLE INFO

Article history:
Received 25 January 2013
Received in revised form 18 September 2013
Accepted 23 September 2013
Available online 14 October 2013

Keywords:
Feature selection
Text classification
High dimensionality
Distributional clustering
Information bottleneck

ABSTRACT

Feature selection is a vital preprocessing step for text classification task used to solve the curse of dimensionality problem. Most existing metrics (such as information gain) only evaluate features individually but completely ignore the redundancy between them. This can decrease the overall discriminative power because one feature's predictive power is weakened by others. On the other hand, though all higher order algorithms (such as mRMR) take redundancy into account, the high computational complexity renders them improper in the text domain. This paper proposes a novel metric called global information gain (GIG) which can avoid redundancy naturally. An efficient feature selection method called maximizing global information gain (MGIG) is also given. We compare MGIG with four other algorithms on six datasets, the experimental results show that MGIG has better results than others methods in most cases. Moreover, MGIG runs significantly faster than the traditional higher order algorithms, which makes it a proper choice for feature selection in text domain.

1. Introduction

With the rapid growth of online information, the amount of digital documents has drastically increased. The problem of how to organize these resources effectively has gained increasing attention from researchers. Text classification, also known as text categorization, is the key technology to solve this problem. The goal of text classification is to assign a new document automatically to a predefined category. In order to achieve this, classification algorithms, also called as classifiers, firstly use the labeled documents to train a model, and then use the learned model to classify the unlabeled documents to a predefined category. Many classification methods, such as Naïve Bayes (NB) classifier [1,2]or support vector machines (SVM) [3,4], have been extensively explored and widely apply in the text classification field.

Before using classification algorithms, documents have to be transformed into compact representations which can easily feed to classifier. A common text representation model is vector space model (VSM). In VSM model, the content of a document is represented as a vector in the term space, i.e. for jth document, $\mathbf{d}_j = (w_{j1}, w_{j2}, \ldots, w_{jV})$, where V is vocabulary count of terms, and w_{jk} is the weight of term k in document j. There are many term weighting methods which can assign different value to w_{jk}. The simplest method is to assign a binary value to w_{jk}, w_{jk} is 1 if the kth word is present in the jth document, and 0, if it is absent. Term frequency is the second most commonly used method. Term frequency-inverse document frequency (tf-idf), as a slightly more complicated weighting method, is also often used. Although weighting methods are important and have an impact on classification accuracy, they are beyond the scope of our research. In this paper, we only use the binary method to represent documents. Though VSM can effectively represent a document as vector, it also has a drawback. Since taking each distinct term in text collection as a feature, even moderate numbers of documents in a text corpus can easily result in thousands or even tens of thousands of features, causing many learning algorithms intractable to use in practice. To solve this problem, two techniques are often used to reduce the dimension of the raw space: feature selection and feature extraction.

Feature selection for text classification is the task of reducing the dimensionality of raw feature space by identifying discriminative features and decreasing the computational complexity. Feature selection has been extensively studied [5], a more effective and popular method being information gain (IG) [6]. IG measures the number of bits of information obtained for category prediction by knowing the presence or absence of a term in a document. Though IG has been successfully applied in reducing the raw feature space, and shows that it can improve effectiveness comparing with no feature selection [4,7],it also has a drawback. That is, IG

* Corresponding author at: Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China. Tel.: +86 13798229773.

E-mail addresses: cxshang@gmail.com (C. Shang), liminghuadi@hotmail.com (M. Li), sz.feng@siat.ac.cn (S. Feng), qs.jang@siat.ac.cn (Q. Jiang), jp.fan@siat.ac.cn (J. Fan).

0950-7051/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.knosys.2013.09.019

only evaluates features individually, gives a score to each feature without considering the redundancy between them, and selects the predefined number of features with the highest correlation scores. Since in reality data collections are often uneven, major categories have more features to select than minor categories. It is even worse for rare categories because they do not have enough examples for training. As a consequence, features for major categories are more likely to be picked out. Some selected features are highly redundant with each other because the documents they can distinguish are highly overlapping.

To tackle the above shortcoming, feature selection algorithms should take the redundancy factor into account. Indeed, numerous of feature selection algorithms which incorporate relevancy and redundancy factors together have been proposed over the past two decades. Many of them are information theory based, such as mRMR [8], JMI [9], DISR [10,11]. In the paper, we call them higher order algorithms. As compared to IG, higher order algorithms can effectively restrain redundancy between selected features. All these algorithms select features one by one, supposing that k features have already been selected. When selecting $(k+1)$th feature, a redundant score between the $(k+1)$th feature and k selected features must be calculated. This will cost $O(k)$ complexity. If the total number of terms is V, the cost of selecting a feature from $V-k$ unselected candidate features is about $O(Vk)$, and the total cost of selecting K features is $O(VK^2)$. Such a computational complexity is too high to apply if both V and K are large. This is the one reason for which higher order algorithms are seldom used in the text classification field. Though selecting a small number features can reduce computational complexity, too few features may not hold sufficient information required for the high classification accuracy.

The feature extraction refers to the process of generating a small set of new features by combining or transforming the original ones. In text classification domain, a way to reduce dimensionality is distributional clustering of words which was first proposed by Pereira et al. [12]. Tishby et al. generalized [13] distributional clustering technique and proposed the information bottleneck (IB) principle. IB tries to compress the features/terms while preserving the information about target labels. Methods applying information bottleneck principle for dimensionality reduction can be divided into three categories: hierarchical agglomerative methods, hierarchical divisive methods and partitioning methods. Information bottleneck method (AIB) is a hierarchical agglomerative method proposed by Slonim [14]. Algorithms like AIB treat each feature as singleton initially. Then they iteratively merge two clusters which cause the mutual information loss between the data and the class labels to be as little as possible. Such greedy agglomeration steps stopped until the desired number of clusters, which we denote by K, is achieved. Lastly, each cluster is treated as a feature for classification. Such algorithms can reduce the dimensionality by two orders of magnitude, almost without sacrificing classification accuracy. However, because V is much greater than K, this aggregation process from down to top is particularly long ($V-K$ iterations) and leads to a high computational cost ($O(V^3)$ for AIB). In addition, the greedy nature of agglomerating algorithms above will yield sub-optimal term clusters as compared with splitting algorithms, because splitting algorithms explore collective information to generate term clusters, while agglomerating algorithms merely exploit two clusters information at each agglomeration step [15]. In [16], Bekkerman proposed a method which can be classified into hierarchical divisive categories. [16] resorts the simulated annealing strategy and splits a cluster apart when "temperature" decreases. As for partitioning methods, Slonim and Dhillon et al. separately introduce their divisive clustering algorithm resemblance to the k-means [17,18]. Hierarchical divisive and partitioning methods alleviate two drawbacks of hierarchical agglomerative methods, but they are still feature extraction algorithms and applying them in a feature selection scenario is inappropriate.

An important characteristic of AIB is that it treats all terms in vocabulary as a single random variable. Each term is considered as an elementary event. Different from AIB, in all the above higher order algorithms, the term is treated as a random variable. In AIB, we can merge two terms as a virtual term just as two elementary events can be regarded as a compound event. Moreover, the degree of redundancy between two terms can be learned by observing how much information is lost during the merging step. This paper exploits this technique heavily.

The contributions of this paper are as follows: firstly, we propose a novel higher order feature selection metric called global information gain (GIG), and give a theoretical explanation as to why this metric can be used for selecting features for text classification. Secondly, an efficient algorithm called maximizing global information gain (MGIG) which reduces the computational complexity from $O(VK^2)$ to $O(VK)$ is developed. Thirdly, we conduct a thorough experiment by comparing our proposed approach to four other algorithms on six text collections. The results of the experiment show that MGIG performs better than others in most cases.

The organization of the paper is as follows: Section 2 introduces the basic information theory. Section 3 reviews briefly previous related work on feature selection algorithms. Our new algorithm is proposed in Section 4. In Section 5, we introduce the experimental setup. Experiment results are evaluated and analyzed in Section 6. Finally, conclusions and suggestions for future research are provided.

2. Preliminaries

In this section, we explain some basic concepts from information theory which will be frequently used in this paper. To learn more, see Cover and Thomas's [19] works.

Let X be a discrete random variable that takes on values from set $\mathbb{X}$, and its probability distribution is $\mathbf{p}(X)$. Its uncertainty can be measured by entropy $H(X)$, which is defined as:

$$H(X) = -\sum_{x\in\mathbb{X}} p(x)\log p(x), \tag{1}$$

for keeping a consistent representation with the following formulas, we also call (1) as the entropy of $\mathbf{p}$ and define it as:

$$H(\mathbf{p}) = -\sum_{x\in\mathbb{X}} p(x)\log p(x). \tag{2}$$

Let Y be a discrete random variable that takes on values from set $\mathbb{Y}$, $p(x,y)$ is the joint probability distribution of X and Y, then the mutual information of two discrete random variables X and Y is defined as:

$$I(X;Y) = \sum_{x\in\mathbb{X}}\sum_{y\in\mathbb{Y}} p(x,y)\ \log\ \frac{p(x,y)}{p(x)p(y)}. \tag{3}$$

The mutual information is a quantity that measures the mutual dependence of the two random variables, and can be interpreted as the amount of information shared by the two variables. $I(X;Y)$ will be high if X and Y are closely related; while $I(X;Y) = 0$ means X and Y are independent with each other.

Similarly to $I(X;Y)$, the conditional mutual information $I(X;Y|Z)$ measures the quantity of information shared by X and Y given Z. $I(X;Y|Z)$ may be either greater or less than $I(X;Y)$ [20,21]. When $I(X;Y|Z) \geqslant I(X;Y)$, X and Y are called complementary w.r.t Z [11].

For a specific $x \in \mathbb{X}$, we can define point mutual information as:

$$I_p(x;Y) = \sum_{y\in\mathbb{Y}} p(x,y)\ \log\ \frac{p(x,y)}{p(x)p(y)} = p(x)\sum_{y\in\mathbb{Y}} p(y|x)\ \log\ \frac{p(y|x)}{p(y)}, \tag{4}$$