Information Processing and Management 53 (2017) 473–489

Contents lists available at ScienceDirect

Information Processing and Management

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

Feature selection based on a normalized difference measure for text classification

Abdur Rehman[a,b,*], Kashif Javed[c], Haroon A. Babri[c]

[a] Department of Computer Science and Information Technology, University of Gujrat, Gujrat, Pakistan
[b] Al-Khawarizmi Institute of Computer Science, University of Engineering and Technology, Lahore, Pakistan
[c] Department of Electrical Engineering, University of Engineering and Technology, Lahore, Pakistan

ARTICLE INFO

Article history:
Received 13 November 2015
Revised 25 October 2016
Accepted 14 December 2016

Keywords:
Text classification
Feature selection
Accuracy measure
Document frequency

ABSTRACT

The goal of feature selection in text classification is to choose highly distinguishing features for improving the performance of a classifier. The well-known text classification feature selection metric named balanced accuracy measure (ACC2) (Forman, 2003) evaluates a term by taking the difference of its document frequency in the positive class (also known as true positives) and its document frequency in the negative class (also known as false positives). This however results in assigning equal ranks to terms having equal difference, ignoring their relative document frequencies in the classes. In this paper we propose a new feature ranking (FR) metric, called normalized difference measure (NDM), which takes into account the relative document frequencies. The performance of NDM is investigated against seven well known feature ranking metrics including odds ratio (OR), chi squared (CHI), information gain (IG), distinguishing feature selector (DFS), gini index (GINI) ,balanced accuracy measure (ACC2) and Poisson ratio (POIS) on seven datasets namely WebACE(WAP,K1a,K1b), Reuters (RE0, RE1),spam email dataset and 20 newsgroups using the multinomial naive Bayes (MNB) and supports vector machines (SVM) classifiers. Our results show that the NDM metric outperforms the seven metrics in 66% cases in terms of macro-F1 measure and in 51% cases in terms of micro F1 measure in our experimental trials on these datasets.

1. Introduction

We are living in an era of fast paced information technology, where large amounts of data are being generated every minute in audio, visual or text form. Twitter users post 300,000 tweets, Google search engine receives more than 4 million queries, email users send 240,000,000 messages in one minute (Data Never Sleeps 2.0, 2014). Significant amount of the data available over Internet is in text form (The Internet, 2009). It is a big challenge to search for information in such a large amount of data in a timely manner. Arranging documents into different categories reduces the search space for a user query (Chen, Schuffels, & Orwig, 1996).

Text classification (TC), or text categorization is the task of assigning one or more than one categories to the documents in a collection from a set of known categories (Sebastiani, 2002). The collection of documents under consideration is called a corpus. Text classification has found a number of applications in a number of domains, such as text mining and information

[*] Corresponding author.
E-mail addresses: a.rehman@uog.edu.pk (A. Rehman), kashif.javed@uet.edu.pk (K. Javed), babri@uet.edu.pk (H.A. Babri).

http://dx.doi.org/10.1016/j.ipm.2016.12.004
0306-4573/© 2016 Elsevier Ltd. All rights reserved.

retrieval (Aggarwal & Zhai, 2012). Separating spam emails from legitimate emails, placing documents in relevant folders, attaching comments with customer complaints and finding user interests based on their comments in social media are some examples (Marin, Holenstein, Sarikaya, & Ostendorf, 2014).

Text classification is a three stage process: feature extraction or preprocessing, feature selection and classification (Marin et al., 2014). Feature extraction generates features also known as terms from documents in a corpus, feature selection selects discriminating features, while classification takes documents containing features selected in feature selection as an input and assigns them labels from a set of known classes. Text data also contains few very frequently occurring terms, and a number of rarely occurring terms (Grimmer & Stewart, 2013). Words like “is”, “the”, “was” etc., which are used for grammatical structure and do not convey any meanings, are called stop words (Joshi, Pareek, Patel, & Chauhan, 2012). Stop words are removed using a list of stop words. Removal of too frequent and in-frequent terms is necessary as a preprocessing step to feature selection (Srividhya & Anitha, 2011). Topic specific frequent terms and rarely occurring terms are removed using a process called pruning (Aggarwal & Zhai, 2012). Pruning removes terms occurring above an upper threshold or below a lower threshold.

The most commonly used representation for text documents is “Bag of Words” (BoW) representation, which is borrowed from information retrieval (IR) (Lan, Tan, Su, & Low, 2007). BoW completely ignores the order of words in a document and considers only word occurrences (Wallach, 2006) called term count (*tc*) or term frequency (*tf*). A document is represented in the form of a vector D $= \{tw_1, tw_2, tw_3, \ldots, tw_v\}$ (Aggarwal & Zhai, 2012; Lan, Tan, Su, & Lu, 2009), where tw_i is weight of ith term in a vocabulary containing v number of terms.

Text classification is inherently high dimensional where a moderate sized dataset can contain tens of thousands of unique words (Joachims, 2002; Wang, Zhang, Liu, Lv, & Wang, 2014). Training time and classification accuracy of a classifier is greatly affected by high dimensional data (Wang, Zhang, Liu, Liu, & Wang, 2016). Representation in vector form makes text data highly sparse where most of the entries are zero (Su, Shirab, & Matwin, 2011). High dimensional data degrades classification performance in terms of running time and accuracy (Wu & Zhang, 2004). Classifiers should be provided with only relevant features for a classification task to reduce execution time and boost accuracy. The task of choosing only relevant features for a classification task is called feature selection.

The goal of feature selection is to provide data free from irrelevant and redundant features to the classifier. Many feature selection algorithms select features by using a feature ranking metric as a primary or auxiliary mechanism (Guyon & Elisseeff, 2003). Feature ranking algorithms determine the strength of a feature to discriminate instances into different classes (Van Hulse, Khoshgoftaar, & Napolitano, 2011), and choose top ranked features.

Features are ranked according to their values in positive and negative classes. More apart are the values for a feature in positive and negative classes, higher will be its rank. Feature values for text documents are their term frequencies, which are the number of occurrences of a term in a document. Feature ranking metrics use document frequency for the determination of term rank. The document frequency of a term in positive class is the number of true positives (*tp*), while the document frequency in the negative class is the number of false positives (*fp*).

Accuracy (ACC) (Forman, 2003) an intuitively simple feature ranking metric, only considers the difference between true positives and false positives of a term. ACC favors strong positive features. A variant of it termed as balanced accuracy (ACC2) (Forman, 2003) ranks features by taking absolute difference of true positive rate (*tpr*) and false positive rate (*fpr*), where $tpr = \frac{tp}{tp+fn}$ and $fpr = \frac{tn}{tn+fp}$ (Dasgupta, Drineas, Harb, Josifovski, & Mahoney, 2007).

We observe that considering only the difference between *tp* and *fp* can be misleading for text data. Two terms having the same difference between *tp* and *fp* are treated equally by ACC2. We argue that a term whose *tp* or *fp* is close to zero along with a high $|tp - fp|$ value is relatively more important. We illustrate this important behavior through an illustrative example in Section 3.1. In this paper we introduce a new feature ranking measure, namely Normalized Difference Measure (NDM), which elevates the rank of a term having either the *tpr* or *fpr* value closer to zero, among the terms having equal $|tpr - fpr|$ values. We compare NDM with seven well known feature ranking metrics including information gain (IG), odds ratio (OR), chi squared (CHI), Poisson ratio (POIS), gini index (GINI) and distinguishing feature selector (DFS) and ACC2 on seven datasets, using naive Bayes (Stigler, 1983) and SVM (Cortes & Vapnik, 1995) classifiers.

The remainder of this paper is organized in four sections. Section 2 covers the related work. Section 3 explains the working of the newly proposed feature ranking metric. Experimental setup and results are shown in Section 4. Conclusions are drawn in Section 5.

2. Related work

In this section we discuss some existing feature selection methods used for ranking terms in text data. Feature selection methods are divided into three classes: filters, wrappers and embedded methods (Lal, Chapelle, Weston, & Elisseeff, 2006b). Filter methods select features independent of any classification algorithm (Dash & Liu, 1997). Wrappers select features with the support of a learning algorithm or classifier (Kohavi & John, 1997). Embedded methods work as part of a classification algorithm and decide feature ranks during learning phase (Lal, Chapelle, Weston, & Elisseeff, 2006a). Filters are computationally less expensive than wrappers and embedded methods (Guyon & Elisseeff, 2003). Therefore, filters are most widely used for feature selection of text data. We discuss here some commonly used filter methods.

Most feature ranking algorithms are based on document frequency, e.g. information gain, chi squared, odds ratio, which can be represented in terms of document frequency (Forman, 2003). Commonly used document frequency measures can be