Pattern Recognition Letters 32 (2011) 375–382

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

Pairwise optimized Rocchio algorithm for text categorization

Yun-Qian Miao [*], Mohamed Kamel

Pattern Analysis and Machine Intelligence (PAMI) Research Group, Department of Electrical and Computer Engineering, University of Waterloo, 200 University Avenue West, Waterloo, Ontario, Canada N2L 3G1

ARTICLE INFO

Article history:
Received 27 August 2009
Available online 25 September 2010
Communicated by R.C. Guido

Keywords:
Text categorization
Rocchio algorithm
Parameter optimization

ABSTRACT

This paper examines the Rocchio algorithm and its application in text categorization. Existing approaches using global parameters optimization of Rocchio algorithm result in choosing one fixed prototype representing each category for multi-category text categorization problems. Therefore, they have limited discriminating power on different category's distribution and their parameter optimization methods are based on weak representation ability of the negative samples consisting of several categories. We present a pairwise optimized Rocchio algorithm, which dynamically adjusts the prototype position between pairs of categories. Experiments were conducted on three benchmark corpora, the 20-Newsgroup, Reuters-21578 and TDT2. The results confirm that our proposed pairwise method achieves encouraging performance improvement over the conventional Rocchio method. A comparative study with the top notch text classifier Support Vector Machine (SVM) also shows the pairwise Rocchio method achieves competitive results.

© 2010 Elsevier B.V. All rights reserved.

1. Introduction

The aim of text categorization (TC) is to assign text documents to appropriate predefined labels that represent their categories. It is one of the fast paced applications of pattern recognition to data mining (Sebastiani, 2002). As the availability of digital format of text documents is ever increasing in recent years, there are many applications employing the techniques of text categorization. For example, news are typically organized by subject topics or geographical code; academic papers are often classified and retrieved by research domains and hierarchical subdomains; patients reports in health care organization are indexed according to disease categories.

Most text categorization algorithms are based on the vector space model (VSM) (Sebastiani, 2002; Salton, 1989; Vinciarelli, 2005; Yang and Liu, 1999), where each document is represented by a list of words that present in the document. Each word is considered as a feature and the feature's value is the weight transformation of the word's frequency in the document. Thus, a document is represented as a feature vector. It is obvious that this VSM leads to high dimensionality of the text categorization problem.

A number of classifiers have been used to classify text documents, including regression models, nearest neighbor classification, Bayesian probabilistic approaches, decision trees, inductive rule learning, neural networks, on-line learning, Support Vector Machines, and combining classifier (Sebastiani, 2002; Guo et al., 2003; Yang and Liu, 1999). In a comparison performed by Sebastiani (2002), Support Vector Machines, example based methods, regression methods and boosting based combining classifier deliver top-notch performance.

The Rocchio method was originally developed in 1971 for interactive document retrieval based on user's feedback of relevant documents and irrelevant documents (Rocchio, 1971). It has been applied to text categorization by Ittner et al. (1995). The Rocchio algorithm is a very efficient text categorization method for applications such as web searching, on-line query, etc., because of its simplicity in both training and testing (Sebastiani, 2002; Vinciarelli, 2005; Guo et al., 2003).

However, most research considers the Rocchio algorithm in TC as an underperformer in term of effectiveness. This paper re-examines the applicable assumptions and parameters optimization method of the Rocchio algorithm, and proposes a pairwise optimized strategy. The proposed enhancement of Rocchio algorithm uses different optimized prototypes to represent one category when building Rocchio classifiers for different pair of classes. We conduct experiments on three common document corpora to compare the categorization performance of the globally optimized Rocchio and pairwise optimized Rocchio method. The results show that our proposed pairwise method outperforms the globally optimized (traditional) Rocchio method in all experiments, especially in the cases that have unbalanced sample distributions among categories. Additionally, the comparison study with the top-notch TC classifier, SVM, reveals that our enhanced version of the Rocchio method achieved relatively close performance.

* Corresponding author. Tel.: +1 519 888 4567x33746; fax: +1 519 746 3077.
E-mail addresses: mikem@pami.uwaterloo.ca (Y.-Q. Miao), mkamel@pami.uwaterloo.ca (M. Kamel).

0167-8655/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2010.09.018

The rest of this paper is organized as follows. Section 2 introduces the architecture of text categorization and reviews the Rocchio algorithm from its first introduction to today's applications. Section 3 explains the details of our proposed pairwise Rocchio method. In Section 4, experimental results and discussions are provided. Section 5 gives conclusions and future directions.

2. Background

In this section, the general architecture of text classification and Rocchio algorithm will be reviewed. First, the structure of a text classification system and the description of each component's functions are explained. Following, we introduce the Rocchio method from its first release to recent applications in text categorization. In these applications, different parameter optimization and setups are explained.

2.1. The architecture of text categorization

Text categorization is a supervised learning paradigm where categorization methods assign a document to a set of predefined categories, based on learning from a set of human-labeled training documents (Shehata et al., 2008; Sebastiani, 2002; Yang and Liu, 1999). It can be divided by two parts. The first part is to construct a classifier through the training samples with their reference labels. This step is also called learning. The second part is to apply the constructed classifier to unlabeled data. This can be a test process to examine the classifier, or to put it into an on-line application. In general, the text categorization system comprises of three key functional components: data pre-processing, classifier construction, and on-line classification. Fig. 1 shows the framework of a typical text categorization system.

2.1.1. Data pre-processing

In order to convert the natural language document to the feature space, there are five steps employed for data pre-processing. Their functions are described as below:

- Function word removal: to remove the functional words that are used to construct nature language documents but not related to any specific topics, such as "a", "an", "the","in", "of", "to", etc.
- Word stemming: to group together those words that are in different forms but with the same root. For example, counting these words "buy", "buys", "buying", "bought" under the same feature identification.
- Feature selection: to further reduce the dimensionality of the data space by removing irrelevant features that have no contribution to category discrimination. Feature selection through information theory has been well studied by Yang and Pedersen (1997). In their research, five commonly used feature selection methods were studied: document frequency (DF), information gain (IG), mutual information (MI), χ^2-test (CHI) and term strength (TS). According to their experiments, IG and CHI are most effective in terms of feature removal aggressiveness and classification accuracy improvement. The DF thresholding approach performs similarly while it is also the simplest technique with lowest computation cost. In this paper, the DF thresholding feature selection method will be adopted. It gains favor because of its simplicity and good performance when the feature removal is not as aggressive as filtering out more than 90% of the features (Yang and Pedersen, 1997).
- Feature weighting: to compute each feature's weight using a transforming function. The common feature weighting function is called *tfidf (Term Frequency/Inverse Document Frequency)* using the formula as:

$$tfidf(t_k, d_i) = \#(t_k, d_i) \cdot log(|T_r|/\#T_r(t_k)) \tag{1}$$

In (1), the $\#(t_k,d_i)$ denotes the number of times term t_k occurs in document d_i, $|T_r|$ denotes the number of all training documents, and $\#T_r\,(t_k)$ denotes the document frequency of term t_k, that is, the number of documents in T_r in which term t_k occurs. It is noted that this function embodies the assumptions that (i) the more often a term occurs in a document, the more it is representative of its content, and (ii) the more documents, the term occurs in, the less discriminating it is.

- Normalization: The cosine normalization using (2) is a process to make each vector of the same length (Salton, 1989). This reduces the similarity between document vectors to measuring the cosine value between them. For example, if two documents features are approximately in linear ratio but have different length, the similarity score, the cosine of vector angle, is near 1. Thus, these two documents are considered to be very similar.

$$w_{ik} = tfidf(t_k, d_i) \Bigg/ \sqrt{\sum_{s=1}^{|T|} (tfidf(t_s, d_i))^2} \tag{2}$$

where w_{ik} is the normalization weight of term t_k in document d_i. While $tfidf(t_k,d_i)$ follows the notation from formula (1), the notation $|T|$ refers to the length of features, i.e., the total number of reserved terms after feature selection process.

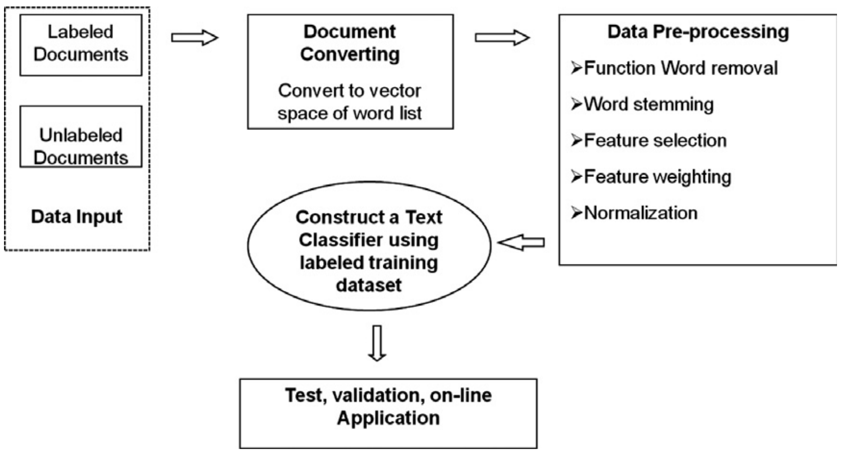

Fig. 1. The framework of text categorization system.

Download English Version:

https://daneshyari.com/en/article/534873

Download Persian Version:

https://daneshyari.com/article/534873

Daneshyari.com