Multimed Tools Appl
https://doi.org/10.1007/s11042-018-6083-5

Feature selection for text classification: A review

Xuelian Deng[1] · Yuqing Li[1] · Jian Weng[2] · Jilian Zhang[3]

Received: 12 March 2018 / Revised: 14 April 2018 / Accepted: 30 April 2018

Abstract Big multimedia data is heterogeneous in essence, that is, the data may be a mixture of video, audio, text, and images. This is due to the prevalence of novel applications in recent years, such as social media, video sharing, and location based services (LBS), etc. In many multimedia applications, for example, video/image tagging and multimedia recommendation, text classification techniques have been used extensively to facilitate multimedia data processing. In this paper, we give a comprehensive review on feature selection techniques for text classification. We begin by introducing some popular representation schemes for documents, and similarity measures used in text classification. Then, we review the most popular text classifiers, including Nearest Neighbor (NN) method, Naïve Bayes (NB), Support Vector Machine (SVM), Decision Tree (DT), and Neural Networks. Next, we survey four feature selection models, namely the filter, wrapper, embedded and hybrid, discussing pros and cons of the state-of-the-art feature selection approaches. Finally, we conclude the paper and give a brief introduction to some interesting feature selection work that does not belong to the four models.

✉ Jilian Zhang
jilian.z.2007@smu.edu.sg

Xuelian Deng
173213455@qq.com

Yuqing Li
28272306@qq.com

Jian Weng
cryptjweng@gmail.com

[1] College of Public Health and Management, Guangxi University of Chinese Medicine, Guangxi, China

[2] College of Information Science and Technology, Jinan University, Guangzhou, China

[3] College of Cyber Security, Jinan University, Guangzhou, China

Published online: 08 May 2018

Keywords Feature Selection · Text classification · Text classifiers · Multimedia

1 Introduction

With the advancement of information technology over the last decade, many new applications, such as social networks, location-based services, social media and e-Commerce, have emerged as hubs for information gathering and dissemination. The majority of the information generated and shared in the Internet is in the form of text data, for example, news reports on CNN, articles listed on Wall Street Journal website, tweets sent by Twitter users, product reviews published on Amazon, etc. In a typical scenario, people may want to see webpages they are interested in, ignoring those irrelevant ones in a possibly very large corpus. This means that webpages (or documents) that contain text content must be classified by topics or some predefined categories, so as to facilitate efficient indexing and search. It is impossible to process this kind of text data manually, however, due to the volume of text data is simply enormous. To deal with this challenge, many techniques and methods have been proposed for classifying documents automatically, and this process is referred to as *text categorization*.

Text classification has a broad applications in real-world scenarios, such as automatically classifying webpages or documents according to a set of pre-specified labels [113], filing new patents into patent categories, user sentiment analysis for social network multimedia [5], spam email filtering, disseminating information to subscribers, document genre identification, video tagging [7], multimedia recommendation [69], etc. In a typical scenario, a webpage or a text document can contain hundreds or thousands of unique terms. If we use all the terms for text classification, we may get poor result because some terms are not helpful for classification and some terms may mislead the classifiers [60, 82]. In this survey, we aim to provide researchers and practitioners with a comprehensive understanding of feature selection theories, models, and techniques, especially the state-of-the-art feature selection techniques designed for text categorization.

2 Text classification

Text classification, also known as text categorization, is to assign one ore more class labels or categories from a predefined set of labels or categories to a document, according to its content [23, 78, 106, 109]. Formally, given a collection of N documents $\mathcal{D} = \{d_1, d_2, ..., d_N\}$ and a set $\mathcal{C} = \{c_1, c_2, ..., c_k\}$ of k predefined categories, the problem of text categorization can be modeled as finding a mapping $\mathcal{F}$ from the Cartesian product $\mathcal{D} \times \mathcal{C}$ to a set $\{True, False\}$, i.e., $\mathcal{F} : \mathcal{D} \times \mathcal{C} \to \{True, False\}$. The mapping $\mathcal{F}$ is called the *classifier*. Based on this mapping, for a document $d_i \in \mathcal{D}$ and a category $c_j \in \mathcal{C}$, if $\mathcal{F}(d_i, c_j) = True$, then d_i belongs to category c_j, otherwise d_i does not belong to c_j.

In real applications, text classification task is *subjective*, meaning that assignment of a category to a document depends on the judgment of human experts. For a same document, two human experts may have different opinion as to which category the document should be assigned to. On the other hand, in library science a book is classified to a class/category if at least 20% of the content of the book is about that class/category. Hence, a document may be affiliated with more than one categories. For instance, the news '*In 2013 Tiger Woods sold his house for 2.2 million US dollars, where cheating scandal broke out*', may be classified to *sports*, *economics*, and *properties* as well.

A typical text categorization process is illustrated in Fig. 1. Generally, to classify a set $\mathcal{D}$ of documents, we need to build a classifier $\mathcal{T}$ at first. Since $\mathcal{T}$ knows nothing about the relation between content of a document and category of the document, we wish to train $\mathcal{T}$ by feeding to it a set $\mathcal{D}'$ of documents, where each document in $\mathcal{D}'$ has already been assigned a category by human experts in advance. The trained classifier $\mathcal{T}'$ is then applied to classify $\mathcal{D}$, after which each document in $\mathcal{D}$ is assigned to a category by $\mathcal{T}'$. Some important questions arise naturally in the text categorization process, i.e., how do we represent document so that classifiers can access document content efficiently; which document is the most similar one to a given document d_i; what are the available classifiers we can choose from; and how can we know which classifier performs the best. To answer these critical questions, in this section we discuss document representation and indexing, similarity between documents, classifiers for documents, and performance of classifiers, respectively.

2.1 Document representation

Although the topic of document representation is very much related to the information retrieval (IR) community, one may find that document representation can greatly affect performance of text classification algorithms, in terms of computational time, storage overhead, and accuracy.

The most popular document representation scheme is the *bag-of-words* model, which is widely used in natural language processing and information retrieval communities. In this model, each document is regarded as a bag that contains its words, keeping word multiplicity while ignoring grammar and the word ordering. Note that generally in a document there may be many *stop words* like *a*, *the*, and *are*, but they are neither descriptive nor meaningful for the subject of a document. Meanwhile, some words are sharing a same stem and should be treated as a single word, because they are similar in meaning. By removing stop words, applying Porter's stemming algorithm to get word stems, combining synonyms and so on in a preprocessing step, the remaining words (or terms) in the document are both descriptive and thematically unique. In the sequel, when referring to a document we mean that the document has already been preprocessed, and we use *word* and *term* interchangeably.

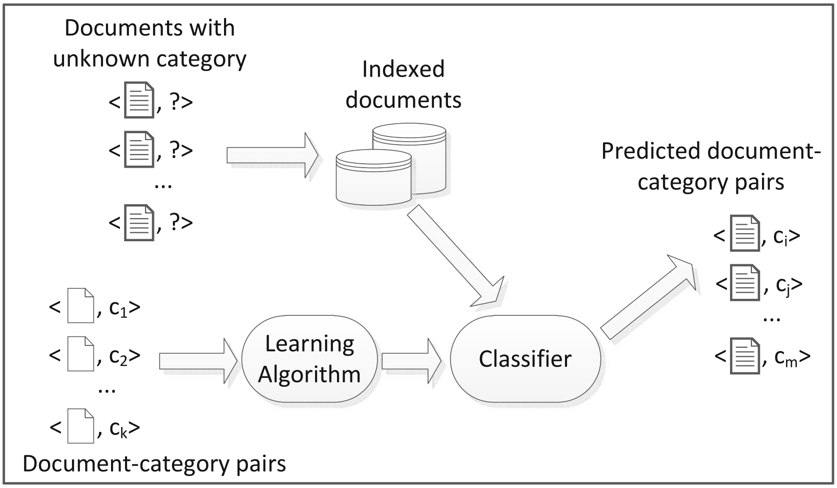


Fig. 1 A Typical Text Classification Process

With the *bag-of-words model*, each document d_i can be represented by a list of pairs (s_j, f_j), where s_j is a term and f_j the frequency of s_j appearing in d_i. If we restrict f_j to take on either 0 or 1, denoting absence and presence of s_j in d_i respectively, then this representation is called the boolean model. To represent a document collection, a natural scheme is to use a document-term matrix [21], where each column of the matrix corresponds to a unique word and each row represents a document. An entry e_{ij} of the matrix reveals how many times word s_j appearing in document d_i. Alternatively, e_{ij} can be the weight of term s_j in document d_i, computed by using the *TF-IDF* weighting scheme.

Some researcher proposed to use alternative ways to represent documents, in the hope of improving the performance of text categorization algorithms that reply on the *bag-of-words* representation model. Scott et al. [81] proposes two representations for text, the first one is based on phrase whereas the second one is based on synonyms and hypernyms. These two alternative representations are tested on a rule-based classifier RIPPER [15]. Experimental results show that these two alternative representations on their own do not add any classification power to the classifiers, as compared to the bag-of-words representation model. However, combining classifiers based on different representations do bring some benefit in terms of accuracy [81].

To represent a document by a list a words/features, there are two popular schemes, i.e., the *universal* dictionary method and the *local* dictionary method [2]. The former constructs for all the class labels a universal dictionary that stores features in each document, whereas the latter builds a local dictionary for each class label respectively. A word can be put into the dictionary as a feature if it appears more than five times, according to empirical study on this cutoff value [2]. The local dictionary method can yield better results, as observed by some researchers [2, 67].

2.2 Similarity between documents

Similarity computation plays a critical role in various real applications such as document clustering and classification, data mining, and information retrieval [120]. In the context of document processing, a similarity function $Sim()$ computes a similarity score of two documents, whose value is real and within the range [0, 1]. In general, if we have $Sim(d_i, d_j) > Sim(d_i, d_k)$, then we say document d_j is more similar to d_i, as compared to d_k. Currently, various similarity measures have been proposed and we focus on some popular measures for text processing, for example, Euclidean distance, Jaccard coefficient [54], Pearson correlation [40], Cosine Similarity [6], Hamming distance [62], Dice coefficient [85], IT-Sim [3, 56], SMTP [58], Earth mover's distance [29, 96], Kullback-Leibler divergence [50], and BM25 [74]. A detailed survey on similarity and distance measures can be found in [13].

Strehl et al. have conducted experimental comparison between several similarity measures for text categorization and showed that Euclidean distance (L_2 metric) performs the poorest, while Cosine and Jaccard are the best to capture human categorization behavior [84]. It is not surprising that Euclidean distance fails to measure similarity between high-dimensional data objects, since given a data object d_i, the Euclidean distance between d_i and any other distinct object tends to be equal in high-dimensional space [1, 100]. Similar behavior can be observed for L_p distance metrics such as Manhattan distance (L_1), L_3 distance, and general L_k metrics [37].

Cosine similarity is a commonly used measure for text categorization. Since a document can be represented by a high-dimensional vector, where each unique term is a dimension and the value of that dimension corresponds to the number of that term appearing in the document, the Cosine similarity between two documents is defined as $\cos(\theta) = \frac{\overrightarrow{d_i} \cdot \overrightarrow{d_j}}{\|\overrightarrow{d_i}\|_2 \cdot \|\overrightarrow{d_j}\|_2}$, where $\overrightarrow{d_i}$ and $\overrightarrow{d_j}$ are documents in vector form, and θ is the angle between the two vectors. One important property of Cosine similarity is that it depends on the angle between two vectors, instead of on the magnitude of them, meaning that two documents will be treated identically if they have the same term composition but with different total term counts [84]. In practice, document vectors are normalized to a unit length before computation.

The Jaccard coefficient was proposed for evaluation of similarity between ecological species, and is has been widely adopted to measure how similar two sets will be. Mathematically, the Jaccard coefficient of two documents d_i and d_j is defined as $J(d_i, d_j) = \frac{|d_i \cap d_j|}{|d_i \cup d_j|}$, where the numerator is the number of terms appearing in both d_i and d_j, and the denominator is the total number of unique terms in the two documents. A closely related measure to Jaccard coefficient is the Dice coefficient [13, 85], which can be computed by $s(d_i, d_j) = \frac{2\overrightarrow{d_i} \cdot \overrightarrow{d_j}}{\|\overrightarrow{d_i}\|_2^2 + \|\overrightarrow{d_j}\|_2^2}$.

Some researchers propose to measure object similarity from information theory point of view, and designed IT-Sim, i.e., the information-theoretic measure [3, 56]. The main idea of IT-Sim is that similarity between objects may be considered as a question of how much information two objects have in common and how much they have in difference [56]. According to [58], IT-Sim achieves very good performance in measuring document similarities compared to existing popular similarity measures, although it is a bit computationally expensive. SMTP [58] is another information theory-based measure, which takes into account three cases when computing similarity between two documents: (1) the feature considered appears in both documents, (2) the feature considered appears in only one document, and (3) the feature considered appears in none of the two documents. Compared to IT-Sim, SMTP performs better and runs faster [58] .

The document similarity measures we have introduced so far do not consider document structure information, neglecting distribution information of words in documents. In fact, each document has a distribution of words different from that of another document, and this discrepancy can reveal some important information about how similar these two documents will be. Recently, some measures have been proposed, for example EMD-based measure [29, 96] and K-L divergence-based measure [40], for evaluating the similarity between two documents according to the difference between their distributions of words. The EMD-based measure works in two steps: (1) decompose documents into sets of subtopics, and (2) based on the sets of subtopics, calculate document similarity by using the Earth Mover's Distance [77]. A detailed comparison between EMD-based measure and non-distribution-based measures confirms that EMD-based scheme outperforms all the other measures, in terms of MAP, $P@5$ and $P@10$ metrics [29].

K-L divergence-based scheme is another type of distribution-based measure, which computes the divergence between two distributions of words as $D_{KL}(d_i||d_j) = \sum_{t=1}^{m} w_{t,i} \times \log \frac{w_{t,i}}{w_{t,j}}$, where m is the total number of unique words in the document collection, $w_{t,i}$ and $w_{t,j}$ the *tf-idf* weights of word t in document d_i and d_j respectively. A major drawback of the K-L divergence-based measure is that vanilla K-L divergence is not symmetric, i.e., $D_{KL}(d_i||d_j) \neq D_{KL}(d_j||d_i)$, which means that traditional K-L divergence cannot be used directly for measuring document similarity. To solve this problem, Huang et al. proposed an averaged K-L divergence measure, which can produce a symmetric similarity score for any two documents [40]. Similar to the EMD-based measure, experiments show that in most cases the averaged K-L divergence outperforms those non-distribution-based measures [40]. We summarize commonly used similarity measures in Table 1.

Table 1 Similarity measures for documents

MEASURE	TYPE	SOURCE
Euclidean distance	Distance based	[6]
Hamming distance	Distance based	[62]
Cosine similarity	Vector space model based	[6]
Dice coefficient	Vector space model based	[13, 85]
Jaccard coefficient	Vector space model based	[54]
Pearson correlation	Correlation based	[6]
SMTP	Information theory based	[58]
IT-Sim	Information theory based	[3, 56]
EMD-based	distribution based	[29, 96, 97]
K-L divergence-based	distribution based	[40, 50]

2.3 Classifiers for text classification

To automatically classify documents, many statistical and machine learning techniques have been designed in the last decade, such as kNN method [17, 111], Naïve Bayes [22], Rocchio [75], multivariate regression models [80, 107], decision trees [72, 73], Support Vector Machines (SVMs) [94], neural networks [45, 78, 102], graph partitioning-based approach [31], and genetic algorithm-based methods [70, 93]. In this section, we review some of the most frequently used classifiers in text categorization.

kNN classification is a non-parametric method widely used in various fields, including data mining, machine learning, information retrieval, and statistics [106]. Given a document d_i with unknown category, a user defined parameter k, and a set $\mathcal{D}$ of documents where each is associated with a category, kNN method computes k *nearest* documents for d_i according to some similarity measure, such as those in Section 2.2, then kNN method assigns to d_i the category that is the most commonly seemed in the k nearest documents. When deciding which documents is the nearest neighbor to d_i, one needs to evaluate each of the documents in $\mathcal{D}$ according to some distance or similarity measure listed in Section 2.2. Note that when k is set to 1, kNN method degenerates to Nearest Neighbor (NN) method. kNN method is easy to implement and its performance is reasonably good, thus it has been employed in many real-applications since 1970s. Despite its prevalence in many real applications, kNN method has a major drawback, i.e., high computational cost, since it is a lazy learning method and each time an object is given, kNN method needs to examine the whole dataset so as to find the k nearest neighbors for the object.

Naïve Bayes (NB) is another classification method that has been studied extensively in text categorization [22]. Normally, NB classifiers adopt the assumption that the value of a particular feature is independent of the value of any other feature. In the context of text classification, the naïve Bayes assumption is that the probability of each word appearing in a document is independent of the occurrence of other word in the same document. There are two kinds of NB based text classifiers. The first one is called *multivariate Bernoulli NB*, which uses a binary vector to represent a document, where each component of the vector represent whether a term is present or absent in the document [63, 64]. The second one is *multinomial NB*, which also takes into account term frequencies in the document [64, 109]. In real applications, multinomial NB classifiers usually performs better than multivariate

NB classifiers, especially true on large document collections [64]. Recently, two drawbacks of the multinomial NB classifiers have been identified by some researchers, i.e., its rough parameter estimation and bias against rare classes that contain only a few training documents. Some effective techniques are proposed to further improve prediction accuracy of the multinomial NB classifiers [47].

The Linear Least Squares Fit (LLSF) is a multivariate regression model for text categorization, which can automatically learn the correlation between a set of training documents and their categories [107]. Specifically, the training documents are represented in the form of (*input*,*output*) vector pairs, where *input* vector is a document represented by using the vector space model and *output* vector consists of categories of the corresponding document. A linear least-square fit problem is solved on these training pairs of vectors, resulting in a matrix of word-category regression coefficients. This matrix gives a mapping from an arbitrary document to a vector of weighted categories, through which a ranked list of categories sorted on weights can be obtained for an input document [106, 107].

Decision tree (DT) is a well-known machine learning algorithm that has been used extensively in automatic classification tasks [72, 73]. When applying to text categorization tasks, DT learning algorithms are employed to select informative words according to information gain criterion. Given a document to classify, the constructed decision tree is used to predict which category the document should belong to, according to the occurrence of word combinations in the document. Some researchers showed that in terms of prediction accuracy, decision trees usually outperform Naïve Bayes classifiers and Rocchio's algorithm, but are slightly worse than kNN methods [23, 55, 106].

Support Vector Machines (SVMs) were introduced by Vapnik et al. [94] for classification tasks, which adheres to *structural risk minimization* principle to construct an optimal hyperplane with the widest possible margin to separate a set of data points that consist of positive and negative data examples. SVMs have been widely and successfully used as a powerful classification tool in various applications, including object recognition [79], image classification [57], text categorization [43, 87], etc. Joachims [43] was the first to apply SVMs to text categorization tasks, due to the fact that SVMs are well suited for several critical properties of text data. First, text data normally contains tens of thousands of terms, meaning that text data is very high-dimensional, whereas the ability of SVMs to learn can be independent of the dimensionality of the feature space. Second, although text data intrinsically contains many features (i.e., unique terms), there are few irrelevant features in a document in general. SVMs can take into account all the features, unlike conventional classification methods that must resort to feature selection techniques to reduce the number of features to a manageable level. Third, document vectors are sparse, meaning that each document only contains few non-zero entries. SVMs are well suited for this kind of classification problems with dense concepts and sparse instances. Through extensive experiments, Joachims showed that SVMs consistently outperform conventional classifiers such as Naïve Bayes, Rocchio, decision trees, and kNN [43].

Neural network was first used by Wiener et al. for text classification [102], where in their model a three-layered neural network is used for each category to learn a non-linear mapping from input document (represented by a vector of term weights) to a category. Recently, Johnson et al. proposes to use Convolutional Neural Networks (CNN) for text categorization, by taking into account 1D internal structure of document, i.e., the order of words [45]. In their work, CNN is applied directly to high-dimensional text data, instead of using low-dimensional word vectors as input by most conventional classifiers. Experiments on real datasets show that CNN outperforms SVM and normal neural networks in terms of error rate [45].

3 Feature selection for text classification

Despite many existing classifiers for text categorization, a major challenge of text categorization is high dimensionality of the feature space [108]. A document usually contains hundreds or thousands of distinct words that are regarded as features, however many of them may be noisy, less informative, or redundant with respect to class label. This may mislead the classifiers and degrade their performance in general [60, 82]. Therefore, feature selection must be applied to eliminate noisy, less informative, and redundant features, so as to reduce the feature space to a manageable level, thus improving efficiency and accuracy of the classifiers used.

Generally, a feature selection method involves four basic steps, i.e., feature subset generation, subset evaluation, stopping condition, and classification result validation [89]. In the first step, we use some search strategy to find a candidate feature subset, which is then evaluated by certain goodness criterion in the second step. In the third step, subset generation and evaluation terminate when stopping conditions are met, after which the best feature subset with be chosen from all the candidates. In the last step, the feature subset will be validated using a validation set. Depending on how to generate feature subsets, feature selection methods can be divided into four categories, namely, filter model [10, 83, 110], wrapper model [24, 71], embedded model, and hybrid model [18, 104]. Majority of feature selection methods for text categorization belong to filter-based method, due to its simplicity and efficiency. Detailed investigation and comparison between different feature selection algorithms for generic data can be found in [20, 60, 66, 82]. In the sequel, we focus on feature selection methods that are explicitly designed for text categorization.

3.1 Filter model

Given a set $S = \{s_1, s_2, ..., s_m\}$ of m features (terms), the filter approach evaluates, by employing some scoring function θ, each feature $s_i \in S$ and assigns a real number $\theta(s_i)$ to s_i according to the contribution of s_i to solving the classification task [11, 60, 89]. Among all the features in S, only k (alternatively, a predefined percentage of $|S|$) features with the highest score are retained, where k is pre-specified by the user. The rest ones in S are discarded without consideration, resulting in a reduced feature space. An illustration of how the filter methods work is given in Fig. 2. Some filter methods evaluate goodness of a term based on how frequently it appears in text corpus, such as document frequency (DF), TF-IDF and term strength (TS), while other filter methods reply on information theory, such as mutual information (MI), information gain (IG), χ^2 (CHI), ECCD, PCA, correlation coefficient (CC), t-test, etc. We summarize existing filter methods in Table 2, and discuss some of them in this section.

Document frequency method (DF) : document frequency of a term is defined as the total number of documents in the document collection that contain the term. The basic idea of DF is that rare terms are considered non-informative for classification and they should be removed during feature selection [51, 88]. Specifically, a ranking procedure is performed to evaluate the goodness or importance of each term in the vocabulary of the document collection, and here document frequency is regarded as the goodness measure for terms. The k most important terms are selected as features for classification, and the rest are filtered out. DF is simple and effective, since its time complexity is approximately linear in the number of training documents. However, the drawback of DF is that some selected terms that appear frequently in many documents may not be discriminative, whereas some discarded terms with low document frequency may be relatively informative.

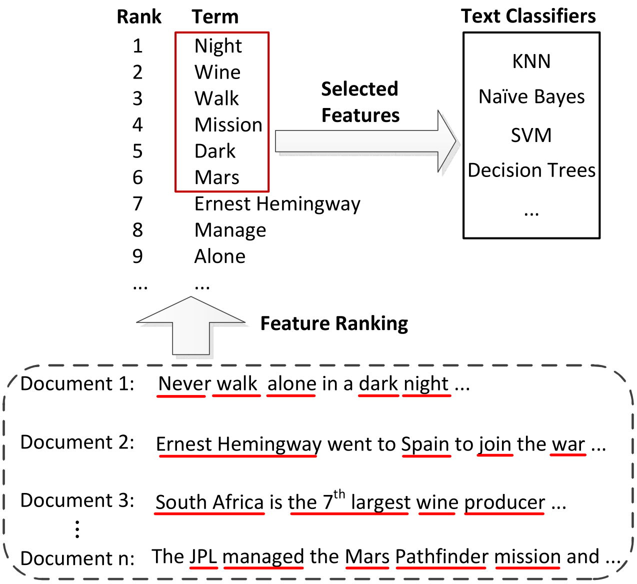

Rank
Term
1
2
3
4
5
6
7
8
9
...
Night
Wine
Walk
Mission
Dark
Mars
Ernest Hemingway
Manage
Alone
...
Selected Features
Text Classifiers
KNN
Naïve Bayes
SVM
Decision Trees
...
Feature Ranking
Document 1: Never walk alone in a dark night ...
Document 2: Ernest Hemingway went to Spain to join the war ...
Document 3: South Africa is the 7th largest wine producer ...
⋮
Document n: The JPL managed the Mars Pathfinder mission and ...

Fig. 2 General framework of filter methods for text Classification

TF-IDF method: this method stems from information retrieval, which takes into account both term frequency (TF) and inverse document frequency (IDF) when measuring the importance of a term [51, 88]. Here, TF is the number of times a term appearing in a document and IDF measures whether the term is common or rare across documents. The importance of a term is then jointly determined by the product of its TF and IDF values.

Information Gain (IG): information gain is one of the most popular metrics used in machine learning for measuring the goodness of attributes, for example, in ID3 [72] and C4.5 [73]. It is used to measure the dependence between features and class labels, as follows

$$IG(s_i, c_j) = H(s_i) - H(s_i|c_j) \tag{1}$$

where s_i and c_j are the j-th term and the class label respectively, $H(s_i)$ the entropy of term s_i, and $H(s_i|c_j)$ the entropy of s_i after observing class label c_j. Here entropy $H(s_i)$ is defined as

$$H(s_i) = -\sum_j p(x_j)\log_2 p(x_j) \tag{2}$$

and entropy $H(s_i|c_j)$ can be computed by

$$H(s_i|c_j) = -\sum_k p(c_k)\sum_j p(x_j|c_k)\log_2 p(x_j|c_k) \tag{3}$$

Basically, if a feature has a larger IG value with respect to a class label, then this feature is more relevant for classification. Zheng et al. [114] proposed iSIG, an improved IG measure, by taking into account of both positive and negative features for imbalanced text data. Some researchers employed IG as a preprocessing component to rank features, which is then followed by a feature reduction technique, such as PCA and genetic algorithm (GA) for feature selection [90].

Table 2 Feature selection methods for text categorization

	Method	Reported best for	Source
Filter	TRL	kNN, SVM	[8]
	ECCD	SVM	[52]
	DF, CF-DF, TFxIDF, PCA	Neural Networks	[51]
	MI, POS	SVM	[87]
	Linear Measure (LM)	SVM	[16]
	t-test based method	kNN, SVM	[98]
	CMFS	NB, SVM	[110]
	DFS	kNN, SVM, DT	[92]
	SpreadFx	SVM	[27]
	BNS	NB	[26]
	IGFSS	NB, SVM	[91]
	Best Terms	NB, SVM	[28]
	Var*IDF, RR(Var*IDF)	SVM-based	[4]
	SS, WS, US	SRLS classifier	[19]
	iSIG, iCC, iOR	NB, LR[a]	[114]
	Correlation Coefficient (CC)	CLASSI	[67]
	χ^2 (CHI)	NB	[76]
	MD, MD-χ^2	NB, SVM	[88]
	PIP	NB	[86]
	IG-PCA, IG-GA	kNN, C4.5	[90]
	Odds ratio	NB	[65]
	MOR, CDM	NB	[12]
	Koller's	NB, C4.5	[49]
Wrapper	Linear Forward Search	NB, C4.5	[34]
	Span-Bound, RW-Bound	SVM	[101]
Hybrid	EGA	NB, AC[1]	[32]
	FSS	NB, C4.5	[9]
	HybridBest, HybridGreedy	SVM, J4.8[1]	[14]

[a] LR: *linear logistic regression*; J4.8: WEKA's implementation of C4.5; AC: associative classification

Another measure called *Term ReLatedness* (TRL) also takes into account occurrences of terms, however it focuses on term absence and presence with respect to categories. Empirical study shows that TRL achieves performance improvement even after removal of 90% unique terms [8]. Some researchers proposes to simultaneously consider the significance of a term both inter-category and intra-category, and based on this idea they designed a comprehensive measure called CMFS for feature selection [110]. CMFS outperforms IG, DF, CHI, when naive Bayes and SVM classifiers are used. Uysal et al. [92] proposed another measure named Distinguishing Feature Selector (DFS), which relies on the same idea of selecting distinctive terms by taking into consideration several rules of how distinctive or irrelevant a term could be. DFS shows competitive performance with respect to some well-known metrics such as CHI and IG.

χ^2 (CHI): CHI can measure the degree of independence between a term and a category and has been widely used for text categorization [76, 108]. Given a term s_i and a category

c_j, a contingency table can be constructed, based on which we can evaluate whether s_i and c_j are independent. The major drawback of CHI is that it is not reliable for low-frequency terms [108]. Similar measures that rely on χ^2-test and contingency table include GSS [30] and ECCD [52], hence they all suffer from the same drawback as CHI. To overcome the drawback of χ^2-based methods, Wang et al. [98] proposed to combine *averaged term frequency* and student t-test for feature selection. Here, the averaged term frequency of term s_i is the average of the term frequencies of s_i across the document corpus. Obviously, the averaged term frequency can capture the low-frequency terms and experiments confirm that their technique is superior to CHI in terms of macro-F_1 and micro-F_1.

Mutual Information (MI): MI is a frequently used metric in information theory to measure the mutual dependency between two variables [87]. Specifically, the MI value between a term s_i and a class label c_j is defined as

$$MI(s_i, c_j) = \log \frac{p(s_i, c_j)}{p(s_i)p(c_j)} \tag{4}$$

MI performs poorly, as compared to other measures such as IG and CHI, due to its bias toward favoring rare terms and its sensitivity to errors in probability estimation [108]. A detailed review on feature selection methods based on mutual information can be found in [95].

Correlation Coefficient (CC): CC is a variant of the CHI measure, and it can be viewed as a one-sided χ^2 metric [67]. The rationale of CC is that it looks for terms that only come from the relevant documents of a category c_j and are indicative of membership in c_j. And terms that come from irrelevant documents or are highly indicative of non-membership in c_j are not regarded as useful. This is the major difference between CC and CHI, which makes CC superior to CHI [67].

Maximum Discrimination (MD): MD is an information theory based method for feature selection, and its basic assumption is that the goodness of a feature can be measured by the discriminative capacity of the feature [88]. Specifically, MD uses a Jeffreys-Multi-Hypothesis divergence (JMH-divergence) to compute discriminative capacity of each feature, and it is designed for naive Bayes classifiers in text categorization.

Linear Measure (LM): LM is a family of linear measures for feature selection in text categorization [16]. A measure is called linear filtering measure if it is in the form of $LM_k(w) = ka_{w,c} - b_{w,c}$, where $a_{w,c}$ represents the number of documents with category label c in which term w appears, $b_{w,c}$ the number of documents containing term w but do not belong to category c, and k is a parameter. The value of LM can reveal the quality of the rule $w \rightarrow c$, that is, *if term w appears in a document, then that document belongs to category c*. LM shows superiority to some entropy-based and TF-IDF based measures when a SVM classifier is adopted for text classification [16].

Some researchers propose to use a round-robin strategy called SpreadFx to rank features with respect to the class, and experiments show that SpreadFx achieves substantial improvements compared to IG and CHI [27].

Bi-Normal Separation (BNS)is a distribution-based metric that relies on normal distribution. For intuition, suppose the occurrence of a given feature in each document is modeled by the event of a random normal variable exceeding a hypothetical threshold. The prevalence rate of the feature corresponds to the area under the curve past the threshold. If the feture is more prevalent in the positive class, then its threshold is further from the tail of the distribution than that of the negative class. The BNS measures the separation between these two thresholds. Eyheramendy et al. proposed to use Bayesian posterior probability based on Bernoulli distribution for feature ranking, and designed a Posterior Inclusion Probability (PIP) method for feature selection [86].

IGFSS [91] is an ensemble feature selection method for text categorization, which aims to improve the performance of classification by modifying the last step of filter-based feature selection algorithms. Conventional filter-based feature selection algorithms rank the features and choose the top-N features for classification, where N is an empirical parameter specified by the user, whereas IGFSS can create a set of features that represent all classes nearly equally, hence these features can improve the performance of classifiers.

Different from existing filter-based algorithms, Dasgupta et al. [19] proposed three sampling-based methods, namely Subspace Sample (SS), Weight-based Sampling (WS), and Uniform Sampling (US), for feature selection. These sampling-based methods randomly sample a small proportion of features, where these sampled features are independent of the total number of features, but dependent on the number of documents and an error parameter. Both theoretical and experimental result show that the proposed sampling methods perform well compared with other popular filter-based feature selection methods [19].

Fragoudis et al. proposed a filter method called Best Terms (BT) for text categorization [28]. Specifically, BT uses a two-step procedure to select the best features. In the first step, BT collects all documents that belong to a given category c_j and selects a set of features that yield the highest prediction accuracy for these documents with respect to c_j. In the second step, BT chooses documents that do not belong to c_j and contain at least one of the features obtained in the first step. Then another set of features are computed, which best classify these documents with respect to $\bar{c_j}$ where $\bar{c_j}$ is the compliment of c_j. The union of the two sets of features obtained during the two steps is the final result. BT improved the accuracy of NB and SVM, as compared to filter methods such as DF, IG, MI, CHI and GSS [28].

Filter-based methods are prevalent in feature selection for text categorization, in that documents usually contain tens of thousands of features and filter methods are generally very efficient to pick single feature among others. However, some researchers also explored the possibility of applying more sophisticated and accurate techniques for text categorization, as described below.

3.2 Wrapper model

The wrapper approach utilizes some search strategy to evaluate each possible subset $S' \subseteq S$, by feeding S' to the chosen classifier and then evaluating the performance of the classifier. These two steps are repeated until the desired quality of feature subset is reached. The wrapper approach achieves better classification accuracy than filter methods, however the computational complexity of wrapper approaches is very high [66, 89]. The number of subset to consider is exponential when the cardinality of S is very large, meaning that the wrapper approach is inadequate for text categorization task, due to the fact that the feature space is usually in the order of hundreds even thousands. Hence, the wrapper approach is only feasible when the number of features is relatively small [24, 71, 89]. In practice, heuristics can be used to restrict the search space, so as to speed up the evaluation process.

There are some search strategies for generating feature subsets, such as hill-climbing, best-first search, branch-and-bound, and genetic algorithms [35, 48]. The hill-climbing expands a feature subset and turns to the subset with the highest accuracy. Hill-climbing terminates when there is no subset improved over current subset. The best-first search is to select the most promising subset that has not been explored before. In general, best-first search is more robust than hill-climbing. Greedy search is a computationally efficient strategy to find the optimal feature subset, which contains forward selection and backward elimination methods. The forward selection method starts with an empty set, and features are added into this set progressively according to some goodness measure. In contrast, backward elimination begins with the whole set of features and less promising features are removed from this set progressively.

Although there are quite a lot wrapper methods for generic data classification, such as LVM [59], FSSEM [24], SFFS [41], backward elimination and forward selection [44], very few are explicitly designed for text categorization purpose. The reason might be that wrapper methods are not suitable for text classification scenario due to high dimensionality, as claimed in [14]. Gutlein et al. focused on forward selection wrapper methods and proposed linear forward selection (LFS) method to reduce the number of feature expansions in each forward selection step [34]. Specifically, two strategies, namely *fixed set* and *fixed width*, are designed to limit the number of features considered during forward selection. The main drawback of LFS, however, is that it only takes into account the top k features (obtained by using some filter method or ranking function) during forward selection, failing to utilize the remaining features. Some researchers focused on SVM classifiers and proposed to increase efficiency of SVMs by desiging wrapper-based feature method for text categorization [101].

3.3 Embedded model

Different from the above two approaches, the embedded approach does not perform the feature selection phase explicitly before learning task begins. Instead, it embeds feature selection operation into the learning process [66]. Some argues that decision trees (DT) such as ID3 and C4.5 are examples of embedded method for feature selection, since while constructing the classifier, DT selects the best features (attributes) that may give the best discriminative power. However, to the best of our knowledge, there is no embedded feature selection method dedicated to text categorization.

3.4 Hybrid model

Hybrid methods are different from the embedded ones, in that the former combine a filter method with a wrapper method during the feature selection process, whereas the latter embed feature selection operation into the learning process of a classifier [9, 66]. Most hybrid methods employ some sort of filter methods to select promising features at first, then apply wrapper methods to the obtained features, for example those methods in [104] and [68], which are designed for generic data though instead of for text corpus.

Günal proposed a simple hybrid method, named HYBRID, that combines filter method and wrapper feature selection steps to select promising combination of features for text categorization [33]. HYBRID consists of two stages. In the first stage each of the four filter methods DF, MI, CHI and IG are employed to select a subset of the features with the top k highest scores, where k is a parameter specified by the user. Then those subsets of selected features are merged together, by eliminating duplicate features. In the second stage, based on this subset of features a generic algorithm is utilized to find the final solution. The major finding of G unal's work is that a combination of features selected by various filter methods is more effective than the features selected by a single filter-based method. A similar idea is proposed by Chou et al., who employ filter method first and then apply wrapper method to the selected features [14].

The above hybrid methods for text categorization suffer from several problems: (1) even though lots of irrelevant features are pruned by the filter methods, the number of wrapper evaluations can still be very large, and (2) the hybrid methods ignore interactions between the selected features and the pruned ones. Aiming to solve these problems, Bermejio et al. proposed a novel iterative hybrid strategy by combing re-ranking method and wrapper evaluation [9]. Specifically, a filter method is used to rank all the features, and then a wrapper method is performed on the first *k* features of the ranked list, resulting in a subset of features selected. Then, this subset of features and those remaining ones are re-ranked again, producing another ranked list of features, which are fed into the wrapper method in the next run. This process repeats until there is no change in the selected features.

4 Discussion and conclusion

In this paper, we have given a detailed review of the state-of-the-art feature selection methods for text classification. Although there is overwhelmingly large number of feature selection techniques, a relatively small portion of them are dedicated to text classification purpose. Feature selection methods are generally divided into four categories, namely the filter model, wrapper model, embedded model, and hybrid model. Filter model is the most efficient one and has also been investigated extensively in text categorization. However, there are very few wrapper and embedded methods for text categorization at present, due to the fact that these two models are very computationally expensive when facing thousands of features contained in a normal text document. To overcome this challenge, researchers have proposed hybrid model that employ filter methods to eliminate redundant and irrelevant features, and the selected features are fed to wrapper methods for further refinement.

We also notice that there are some work targeted at novel applications of feature selection in recently years, for example, multi-label feature selection [42, 61, 82, 117, 118, 122], feature selection with streaming features [103], online feature selection [99], filter-based locality preserving feature selection [36], similarity preserving feature selection [112], feature selection with optimization techniques [105], regularization based feature selection methods [121], feature selection with machine learning techniques [25, 38, 39, 53, 115, 119, 123], stability measures for feature selection algorithms [46]. However, these work mainly focus on generic data, and it is not clear whether they can be applied to text data. With the proliferation of text applications, we may see a trend that these feature selection techniques will be applied to text categorization, and interesting problems may arise, for example, feature selection for text categorization when there are missing values in documents [116].

Acknowledgments This work was supported by National Key R&D Plan of China (Grant No. 2017YFB0802203 and 2018YFB100013), National Natural Science Foundation of China (Grant Number U1736203, 61732021, 61472165, 61373158, and 61363009), Guangdong Provincial Engineering Technology Research Center on Network Security Detection and Defense (Grant No. 2014B090904067), Guangdong Provincial Special Funds for Applied Technology Research and Development and Transformation of Important Scientific and Technological Achieve (Grant No. 2016B010124009), the Zhuhai Top Discipline–Information Security, Guangzhou Key Laboratory of Data Security and Privacy Preserving, Guangdong Key Laboratory of Data Security and Privacy Preserving, National Joint Engineering Research Center of Network Security Detection and Protection Technology.

References

1. Aggarwal CC, Hinneburg A, Keim DA (2001) On the surprising behavior of distance metrics in high dimensional spaces. In: ICDT, vol 1. Springer, pp 420–434

2. Apté C, Damerau F, Weiss SM (1994) Automated learning of decision rules for text categorization. ACM Trans Inf Syst 12(3):233–251
3. Aslam JAMF (2003) An information-theoretic measure for document similarity. In: Proceedings of ACM SIGIR, pp 449–450
4. Baccianella S, Esuli A, Sebastiani F (2014) Feature selection for ordinal text classification. Neural Comput 26(3):557–591
5. Baecchi C, Uricchio T, Bertini M, Del Bimbo A (2016) A multimodal feature learning approach for sentiment analysis of social network multimedia. Multimed Tool Appl 75(5):2507–2525
6. Baeza-Yates R, Ribeiro-Neto B (1999) Modern information retrieval. ACM, New York
7. Ballan L, Bertini M, Uricchio T, Del Bimbo A (2015) Data-driven approaches for social image and video tagging. Multimed Tool Appl 74(4):1443–1468
8. Basu T, Murthy C (2016) A supervised term selection technique for effective text categorization. Int J Mach Learn Cybern 7(5):877–892
9. Bermejo P, de la Ossa L, Gámez JA, Puerta JM (2012) Fast wrapper feature subset selection in high-dimensional datasets by means of filter re-ranking. Knowl-Based Syst 25(1):35–44
10. Brown G (2009) A new perspective for information theoretic feature selection. In: Artificial intelligence and statistics, pp 49–56
11. Chandrashekar G, Sahin F (2014) A survey on feature selection methods. Comput Electr Eng 40(1):16–28
12. Chen J, Huang H, Tian S, Qu Y (2009) Feature selection for text classification with naïve bayes. Expert Syst Appl 3(36):5432–5435
13. Choi SS, Cha SH, Tappert CC (2010) A survey of binary similarity and distance measures. J Syst Cybern Inform 8(1):43–48
14. Chou CH, Sinha AP, Zhao H (2010) A hybrid attribute selection approach for text classification. J Assoc Inf Syst 11(9):491
15. Cohen WW (1995) Fast effective rule induction. In: Proceedings of the twelfth international conference on machine learning, pp 115–123
16. Combarro EF, Montanes E, Diaz I, Ranilla J, Mones R (2005) Introducing a family of linear measures for feature selection in text categorization. IEEE Trans Knowl Data Eng 17(9):1223–1232
17. Cunningham P, Delany SJ (2007) k-nearest neighbour classifiers. Multiple Class Syst 34:1–17
18. Das S (2001) Filters, wrappers and a boosting-based hybrid for feature selection. In: ICML, vol 1, pp 74–81
19. Dasgupta A, Drineas P, Harb B, Josifovski V, Mahoney MW (2007) Feature selection methods for text classification. In: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, mining. ACM, pp 230–239
20. Dash M, Liu H (1997) Feature selection for classification. Intell Data Anal 1(1-4):131–156
21. Deerwester S, Dumais ST, Furnas GW, Landauer TK, Harshman R (1990) Indexing by latent semantic analysis. J Amer Soc Inform Sci 41(6):391
22. Domingos P, Pazzani M (1997) On the optimality of the simple bayesian classifier under zero-one loss. Mach Learn 29(2/3):103–130
23. Dumais S, Platt J, Heckerman D, Sahami M (1998) Inductive learning algorithms and representations for text categorization. In: Proceedings of the seventh international conference on Information and knowledge management. ACM, pp 148–155
24. Dy JG, Brodley CE (2004) Feature selection for unsupervised learning. J Mach Learn Res 5:845–889
25. Fang Y, Zhang J, Zhang S, Lei C, Hu X (2017) Supervised feature selection algorithm based on low-rank and manifold learning. In: Proceedings of the 13th international conference on advanced data mining and applications, ADMA 2017. Singapore, pp 273–286
26. Forman G (2003) An extensive empirical study of feature selection metrics for text classification. J Mach Learn Res 3:1289–1305
27. Forman G (2004) A pitfall and solution in multi-class feature selection for text classification. In: Proceedings of the 21st international conference on machine learning. ACM, p 38
28. Fragoudis D, Meretakis D, Likothanassis S (2005) Best terms: an efficient feature-selection algorithm for text categorization. Knowl Inf Syst 8(1):16–33
29. Fu AY, Wenyin L, Deng X (2006) Detecting phishing web pages with visual similarity assessment based on earth mover's distance (emd). IEEE Trans Dependable Secure Comput 3(4)
30. Galavotti L, Sebastiani F, Simi M (2000) Experiments on the use of feature selection and negative evidence in automated text categorization. In: International conference on theory and practice of digital libraries. Springer, pp 59–68

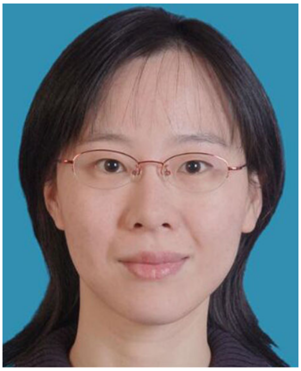
31. Gao B, Liu TY, Feng G, Qin T, Cheng QS, Ma WY (2005) Hierarchical taxonomy preparation for text categorization using consistent bipartite spectral graph copartitioning. IEEE Trans Knowl Data Eng 17(9):1263–1273
32. Ghareb AS, Bakar AA, Hamdan AR (2016) Hybrid feature selection based on enhanced genetic algorithm for text categorization. Expert Syst Appl 49:31–47
33. Günal S (2012) Hybrid feature selection for text classification. Turkish J Electr Eng Comput Sci 20(2):1296–1311
34. Gutlein M, Frank E, Hall M, Karwath A (2009) Large-scale attribute selection using wrappers. In: IEEE symposium on computational intelligence and data mining, 2009. CIDM'09. IEEE, pp 332–339
35. Guyon I, Elisseeff A (2003) An introduction to variable and feature selection. J Mach Learn Res 3(3):1157–1182
36. He X, Cai D, Niyogi P (2006) Laplacian score for feature selection. In: Advances in neural information processing systems, pp 507–514
37. Hinneburg A, Aggarwal CC, Keim DA (2000) What is the nearest neighbor in high dimensional spaces? In: 26th international conference on very large databases, pp 506–515
38. Hu R, Cheng D, He W, Wen G, Zhu Y, Zhang J, Zhang S (2017) Low-rank feature selection for multi-view regression. Multimed Tool Appl 76(16):17,479–17,495
39. Hu R, Zhu X, Cheng D, He W, Yan Y, Song J, Zhang S (2017) Graph self-representation method for unsupervised feature selection. Neurocomputing 220:130–137
40. Huang A (2008) Similarity measures for text document clustering. In: NZCSRSC, pp 49–56
41. Jain A, Zongker D (1997) Feature selection: evaluation, application, and small sample performance. IEEE Trans Pattern Anal Mach Intell 19(2):153–158
42. Jian L, Li J, Shu K, Liu H (2016) Multi-label informed feature selection. In: IJCAI, pp 1627–1633
43. Joachims T (1998) Text categorization with support vector machines: Learning with many relevant features. Mach Learn: ECML-98:137–142
44. John GH, Kohavi R, Pfleger K et al (1994) Irrelevant features and the subset selection problem. In: Machine learning: proceedings of the eleventh international conference, pp 121–129
45. Johnson R, Zhang T (2014) Effective use of word order for text categorization with convolutional neural networks. arXiv:1412.1058
46. Kalousis A, Prados J, Hilario M (2007) Stability of feature selection algorithms: a study on high-dimensional spaces. Knowl Inf Syst 12(1):95–116
47. Kim SB, Han KS, Rim HC, Myaeng SH (2006) Some effective techniques for naive bayes text classification. IEEE Trans Knowl Data Eng 18(11):1457–1466
48. Kohavi R, John GH (1997) Wrappers for feature subset selection. Artif Intell 97(1-2):273–324
49. Koller D, Sahami M (1996) Toward optimal feature selection. Tech. rep., Stanford InfoLab
50. Kullback S, Leibler RA (1951) On information and sufficiency. Ann Math Stat 22(1):79–86
51. Lam SL, Lee DL (1999) Feature reduction for neural network based text categorization. In: Proceedings of the 6th international conference on database systems for advanced applications, 1999. IEEE, pp 195–202
52. Largeron C, Moulin C, Géry M. (2011) Entropy based feature selection for text categorization. In: Proceedings of the 2011 ACM symposium on applied computing. ACM, pp 924–928
53. Lei C, Zhu X (2017) Unsupervised feature selection via local structure learning and sparse learning. https://doi.org/10.1007/s11,042–017–5381–7
54. Levandowsky M, Winter D (1971) Distance between sets. Nature 234(5323):34–35
55. Lewis DD, Ringuette M (1994) A comparison of two learning algorithms for text categorization. In: 3rd annual symposium on document analysis and information retrieval, vol 33, pp 81–93
56. Lin D (1998) An information-theoretic definition of similarity. In: Proceedings of international conference on machine learning, vol 98, pp 29,633–304
57. Lin Y, Lv F, Zhu S, Yang M, Cour T, Yu K, Cao L, Huang T (2011) Large-scale image classification: fast feature extraction and svm training. In: 2011 IEEE conference on computer vision and pattern recognition (CVPR). IEEE, pp 1689–1696
58. Lin YS, Jiang JY, Lee SJ (2014) A similarity measure for text classification and clustering. IEEE Trans Knowl Data Eng 26(7):1575–1590
59. Liu H, Setiono R (1997) Feature selection and classification-a probabilistic wrapper approach. In: Proceedings of the 9th international conference on industrial and engineering applications of AI and ES, pp 419–424
60. Liu H, Yu L (2005) Toward integrating feature selection algorithms for classification and clustering. IEEE Trans Knowl Data Eng 17(4):491–502
61. Ma Z, Nie F, Yang Y, Uijlings JR, Sebe N (2012) Web image annotation via subspace-sparsity collaborated feature selection. IEEE Trans Multim 14(4):1021–1030

62. Manku GS, Jain A, Das Sarma A (2007) Detecting near-duplicates for web crawling. In: Proceedings of the 16th international conference on World Wide Web. ACM, pp 141–150
63. McCallum A, Nigam K (1998) Employing em in poll-based active learning for text classification. In: Proceedings of the 15th international conference on machine learning, pp 350–358
64. McCallum A, Nigam K et al (1998) A comparison of event models for naive bayes text classification. In: AAAI-98 workshop on learning for text categorization, vol 752. Madison, WI, pp 41–48
65. Mladenić D (1998) Feature subset selection in text-learning. In: European conference on machine learning, pp 95–100. Springer
66. Molina LC, Belanche L, Nebot À (2002) Feature selection algorithms: a survey and experimental evaluation. In: Proceedings of 2002 IEEE international conference on data mining, 2002. ICDM 2003. IEEE, pp 306–313
67. Ng HT, Goh WB, Low KL (1997) Feature selection, perceptron learning, and a usability case study for text categorization. In: ACM SIGIR forum, vol 31. ACM, pp 67–73
68. Oh IS, Lee JS, Moon BR (2004) Hybrid genetic algorithms for feature selection. IEEE Trans Pattern Anal Mach Intell 26(11):1424–1437
69. Pappas N, Popescu-Belis A (2015) Combining content with user preferences for non-fiction multimedia recommendation: a study on ted lectures. Multi Tools Appl 74(4):1175–1197
70. Pietramala A, Policicchio VL, Rullo P, Sidhu I (2008) A genetic algorithm for text classification rule induction. In: Joint european conference on machine learning and knowledge discovery in databases. Springer, pp 188–203
71. Pudil P, Novovičová J, Kittler J (1994) Floating search methods in feature selection. Pattern Recogn Lett 15(11):1119–1125
72. Quinlan JR (1986) Induction of decision trees. Mach Learn 1(1):81–106
73. Quinlan JR (2014) C4. 5: programs for machine learning. Elsevier
74. Robertson SE, Walker S (1994) Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In: Proceedings of the 17th ACM SIGIR conference. ACM, pp 232–241
75. Rocchio JJ (1971) Relevance feedback in information retrieval. The Smart retrieval system-experiments in automatic document processing
76. Rogati M, Yang Y (2002) High-performing feature selection for text classification. In: Proceedings of the eleventh international conference on information and knowledge management. ACM, pp 659–661
77. Rubner Y, Tomasi C, Guibas LJ (2000) The earth mover's distance as a metric for image retrieval. Int J Comput Vis 40(2):99–121
78. Ruiz ME, Srinivasan P (2002) Hierarchical text categorization using neural networks. Inf Retr 5(1):87–118
79. Schuldt C, Laptev I, Caputo B (2004) Recognizing human actions: a local svm approach. In: Proceedings of the 17th international conference on pattern recognition, 2004. ICPR 2004, vol 3. IEEE, pp 32–36
80. Schütze H, Hull DA, Pedersen JO (1995) A comparison of classifiers and document representations for the routing problem. In: Proceedings of the 18th annual international ACM SIGIR conference on research and development in information retrieval. ACM, pp 229–237
81. Scott S, Matwin S (1999) Feature engineering for text classification. In: ICML, vol 99, pp 379–388
82. Sebastiani F (2002) Machine learning in automated text cateogirzation. ACM Comput Surv 34(1):1–47
83. Song Q, Ni J, Wang G (2013) A fast clustering-based feature subset selection algorithm for high-dimensional data. IEEE Trans Knowl Data Eng 25(1):1–14
84. Strehl A, Ghosh J, Mooney R (2000) Impact of similarity measures on web-page clustering. In: Workshop on artificial intelligence for web search (AAAI 2000), vol 58, p 64
85. Strehl AJG (2000) Value-based customer grouping from large retail data-sets. In: Proceedings of SPIE, vol 4057, pp 33–42
86. Susana E, David M (2005) A novel feature selection score for text categorization. In: Proceedings of the workshop on feature selection for data mining, in conjunction with the 2005 SIAM international conference on data mining. SIAM, pp 1–8
87. Taira H, Haruno M (1999) Feature selection in svm text categorization. In: AAAI/ IAAI, pp 480–486
88. Tang B, Kay S, He H (2016) Toward optimal feature selection in naive bayes for text categorization. IEEE Trans Knowl Data Eng 28(9):2508–2521
89. Tang J, Alelyani S, Liu H (2014) Feature selection for classification: a review. Data Classification: Algorithms and Applications, p 37
90. Uuz H (2011) A two-stage feature selection method for text categorization by using information gain, principal component analysis and genetic algorithm. Knowl-Based Syst 24(7):1024–1032
91. Uysal AK (2016) An improved global feature selection scheme for text classification. Expert Syst Appl 43:82–92

92. Uysal AK, Gunal S (2012) A novel probabilistic feature selection method for text classification. Knowl-Based Syst 36:226–235
93. Uysal AK, Gunal S (2014) Text classification using genetic algorithm oriented latent semantic features. Expert Syst Appl 41(13):5938–5947
94. Vapnik VN, Vapnik V (1998) Statistical learning theory, vol 1. Wiley, New York
95. Vergara JR, Estévez PA (2014) A review of feature selection methods based on mutual information. Neural Comput Appl 24(1):175–186
96. Wan X (2007) A novel document similarity measure based on earth mover's distance. Inf Sci 177(18):3718–3730
97. Wan X, Peng Y (2005) The earth mover's distance as a semantic measure for document similarity. In: Proceedings of the 14th ACM international conference on information and knowledge management. ACM, pp 301–302
98. Wang D, Zhang H, Liu R, Lv W, Wang D (2014) t-test feature selection approach based on term frequency for text categorization. Pattern Recogn Lett 45:1–10
99. Wang J, Zhao P, Hoi SC, Jin R (2014) Online feature selection and its applications. IEEE Trans Knowl Data Eng 26(3):698–710
100. Weber R, Schek HJ, Blott S (1998) A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In: VLDB, vol 98, pp 194–205
101. Weston J, Mukherjee S, Chapelle O, Pontil M, Poggio T, Vapnik V (2001) Feature selection for svms. In: Advances in neural information processing systems, pp 668–674
102. Wiener E, Pedersen JO, Weigend AS et al (1995) A neural network approach to topic spotting. In: Proceedings of SDAIR-95, 4th annual symposium on document analysis and information retrieval, vol 317. Las Vegas, NV, p 332
103. Wu X, Yu K, Ding W, Wang H, Zhu X (2013) Online feature selection with streaming features. IEEE Trans Pattern Anal Mach Intell 35(5):1178–1192
104. Xing EP, Jordan MI, Karp RM et al (2001) Feature selection for high-dimensional genomic microarray data. In: ICML, vol 1, pp 601–608
105. Yan J, Liu N, Zhang B, Yan S, Chen Z, Cheng Q, Fan W, Ma WY (2005) Ocfs: optimal orthogonal centroid feature selection for text categorization. In: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, pp 122–129
106. Yang Y (1999) An evaluation of statistical approaches to text categorization. Inf Retr 1(1):69–90
107. Yang Y, Chute CG (1994) An example-based mapping method for text categorization and retrieval. ACM Trans Inf Syst 12(3):252–277
108. Yang Y, Pedersen JO (1997) A comparative study on feature selection in text categorization. In: ICML, vol 97, pp 412–420
109. Yang Y, Liu X (1999) A re-examination of text categorization methods. In: Proceedings of the 22nd ACM SIGIR, pp 42–49
110. Yang J, Liu Y, Zhu X, Liu Z, Zhang X (2012) A new feature selection based on comprehensive measurement both in inter-category and intra-category for text categorization. Inf Process Manag 48(4):741–754
111. Zhang S, Li X, Zong M, Zhu X, Wang R (2017) Efficient knn classification with different numbers of nearest neighbors. IEEE transactions on neural networks and learning systems. https://doi.org/10.1109/TNNLS.2017.2673241
112. Zhao Z, Wang L, Liu H, Ye J (2013) On similarity preserving feature selection. IEEE Trans Knowl Data Eng 25(3):619–632
113. Zhao S, Yao H, Zhao S, Jiang X, Jiang X (2016) Multi-modal microblog classification via multi-task learning. Multimed Tools Appl 75(15):8921–8938
114. Zheng Z, Wu X, Srihari R (2004) Feature selection for text categorization on imbalanced data. ACM Sigkdd Explorations Newsletter 6(1):80–89
115. Zheng W, Zhu X, Zhu Y, Hu R, Lei C (2017) Dynamic graph learning for spectral feature selection. Multimedia Tools and Applications. https://doi.org/10.1007/s11,042–017–5272–y
116. Zhu X, Zhang S, Jin Z, Zhang Z, Xu Z (2011) Missing value estimation for mixed-attribute data sets. IEEE Trans Knowl Data Eng 23(1):110–121
117. Zhu X, Zhang L, Huang Z (2014) A sparse embedding and least variance encoding approach to hashing. IEEE Trans Image Process 23(9):3737–3750
118. Zhu X, Li X, Zhang S (2016) Block-row sparse multiview multilabel learning for image classification. IEEE Trans Cybern 46(2):450–461
119. Zhu X, Li X, Zhang S, Ju C, Wu X (2017) Robust joint graph sparse coding for unsupervised spectral feature selection. IEEE Trans Neural Netw Learn Syst 28(6):1263–1275

120. Zhu X, Li X, Zhang S, Xu Z, Yu L, Wang C (2017) Graph pca hashing for similarity search. IEEE Trans Multimed 19(9):2033–2044
121. Zhu X, Suk H, Wang L, Lee S, Shen D (2017) A novel relational regularization feature selection method for joint regression and classification in AD diagnosis. Med Image Anal 38:205–214
122. Zhu X, Suk HI, Huang H, Shen D (2017) Low-rank graph-regularized structured sparse regression for identifying genetic biomarkers. IEEE Trans Big Data 3(4):405–414
123. Zhu X, Zhang S, Hu R, Zhu Y et al (2018) Local and global structure preservation for robust unsupervised spectral feature selection. IEEE Trans Knowl Data Eng 30(3):517–529

Xuelian Deng is currently a faculty member with the College of Public Health and Management, Guangxi University of Chinese Medicine, Guangxi China. Her research interests include feature selection, complex networks, multimedia systems and data mining.



Yuqing Li is a faculty member with the College of Public Health and Management, Guangxi University of Chinese Medicine, Guangxi China. Her research interests include multimedia systems, online education, and data mining.

Jian Weng received B.S. and M.S. degrees in computer science and engineering from South China University of Technology, in 2000 and 2004, respectively, and the Ph.D. degree in computer science and engineering from Shanghai Jiao Tong University, in 2008. From April 2008 to March 2010, he was a postdoc in the School of Information Systems, Singapore Management University. Jian Weng is currently a professor and dean with the School of Information Science and Technology, Jinan University, Guangzhou China. He has published more than 80 papers in cryptography and machine learning conferences and journals, such as CRYPTO, EUROCRYPT, ASIACRYPT, TCC, PKC, CT-RSA, IEEE TDSC, IEEE TIFS, IEEE TPAMI, etc. He served as PC co-chairs or PC members for more than 30 international conferences. Jian Weng is currently on the editor board of IEEE Transactions on Vehicular Technology. He has won the 2014 cryptographic innovation award from Chinese Association for Cryptographic Research, the best paper award from the 28th Symposium on Cryptography and Information Security (SCIS 2011), the best student paper award from the 8th International Conference on Provable Security (ProvSec 2014), and the best student paper award from the 37th IEEE International Conference on Distributed Computing Systems (ICDCS 2017).

Jilian Zhang obtained his PhD degree in computer science at Singapore Management University. He is currently an associate professor with the College of Cyber Security, Jinan University, Guangzhou China. His research interests include data management, databases, query processing, and data privacy protection. He has published more than 30 papers on refereed journals and conferences, including IEEE TKDE, IEEE Transactions on Cybernetics, IEEE Transactions on Multimedia, Information Systems, ACM SIGMOD, VLDB, IJCAI etc. Jilian Zhang serves as guest-editor for Pattern Recognition Letters and WWW Journal. He has also been served as PC members for IJCAI and ACM CIKM, and reviewers for VLDB Journal, IEEE TKDE, IEEE Transactions on Computers, IEEE Transactions on Cybernetics, IEEE Transactions on Service Computing, KAIS, Information Sciences, Neurocomputing, Multimedia Systems, and reviewer for ACM SIGMOD, VLDB, ICDE, DASFAA, ICDM, ACM GIS, PAKDD, SSTD, PRICAI, and DaWaK.