A Comprehensive Study of Eleven Feature Selection Algorithms and Their Impact on Text Classification

Suchi Vora  
Department of Computer Science  
San Francisco State University  
San Francisco, California, USA  
vorasuchi786@gmail.com

Hui Yang  
Department of Computer Science  
San Francisco State University  
San Francisco, California, USA  
huiyang@sfsu.edu

Abstract: Feature selection has been routinely used as a preprocessing step to remove irrelevant features and conquer the “curse of dimensionality”. In contrast to dimensionality reduction techniques such as PCA, the resulting features from feature selection are selected from the original feature space; hence, easy to interpret. A large host of feature selection algorithms has been proposed in the literature. This has created a critical issue: which algorithm should one use? Moreover, how does a feature selection method affect the performance of a given classification algorithm?

This paper addresses these issues by (1) presenting an open source software system that integrates eleven feature selection algorithms and five common classifiers; and (2) systematically comparing and evaluating the selected features and their impact over these five classifiers using five datasets. Specifically, this system includes ten commonly adopted filter-based feature selection algorithms: ChiSquare, Information Gain, Fisher Score, Gini Index, Kruskal-Wallis, Laplacian Score, ReliefF, FCBF, CFS, and mRmR. It also includes one state-of-the-art embedded approach built upon Random Forests. The five classifiers are SVM, Random Forests, Naïve Bayes, kNN and C4.5 Decision Tree.

Comprehensive evaluations consisting of around 1000 experiments were conducted over five text datasets. Several approximately equivalent groups (AEG), where algorithms in the same group select highly similar features, have been identified. Suitable feature-selection-classifier combinations have also been identified. For example, Chi-square and Information Gain form an AEG. Furthermore, Gini Index or Kruskal-Wallis together with SVM often produces classification performance that is comparable with or better than using all the original features. Such results will provide empirical guidelines for the data analytic community. The above software system is available at https://www.dropbox.com/sh/ryw23s52e98uhrv/AAANpcoJU4X6r3Sfv4qB5ERna?dl=0

Keywords: feature selection/ranking algorithms; classification algorithms; comparison and evaluation.

I. INTRODUCTION

Dimensionality reduction has become an essential component in today’s big data analysis to overcome the “curse of dimensionality” and/or to reduce the computational cost. For instance, text and image datasets often contain tens of thousands of dimensions. Coupled with their high volume, this has made it challenging to apply even the most advanced data mining algorithms. Furthermore, many of these dimensions in the original feature space might be irrelevant or noisy to the analytical task at hand.

Dimensionality reduction helps address the above problems by removing redundant and irrelevant features [19]. Dimensionality reduction techniques can be broadly classified into transformative methods such as PCA (Principal Component Analysis) and feature subset selection methods. The former transforms or projects the original feature space into a much smaller space by focusing on some inherent data characteristics (e.g., data variance). The latter works in the original feature space of a dataset and reduces the number of features by applying specific evaluation criteria to measure and then identify the most relevant features. As detailed in Section III, these evaluation criteria often require the input data to be labeled, with Laplacian score [13] being one exception. (Thus, it is more proper to call them ‘supervised feature selection’.) To measure the effectiveness of a dimensionality reduction technique, a common approach is to utilize the resulting features for a specific analytic task, e.g., build a classification model using these features. One can then empirically evaluate the effectiveness of different dimensionality reduction techniques by comparing the classification performance on the same datasets.

This article focuses on the second type of reduction techniques, namely, a host of eleven feature subset selection algorithms. Furthermore, it will evaluate these algorithms in the context of text classification using five different classification algorithms with the following two main goals: (1) given an input dataset, how will the resulting feature subsets from different selection algorithms compare with each other? Can one empirically put these algorithms into approximately equivalent groups, where algorithms in the same group often select the same or approximately same features? And (2) given a classification algorithm, how will its performance be influenced by different feature selection algorithms? Will there be an “optimal” feature-selection-classifier combination?

These are important questions, especially considering the large number of algorithms for both feature selection and classification. Unfortunately, to the best of our knowledge, no concrete studies have been carried out to address these issues.


For the feature selection algorithms, this work considers the following eleven: ChiSquare [21], Information Gain [20], Fisher Score[22], Gini Index [20], Kruskal-Wallis [14], Laplacian Score[13], ReliefF [12], FCBF (Fast Correlation Based Filter) [15], CFS (Correlation-based Feature Selection) [23], mRmR (Minimum-Redundancy-Maximum-Relevance) [24], and a Random Forests-based feature selection algorithm [16]. Except the last algorithm, all the others are filter-based. Generally speaking, filter-based methods explore one or more general characteristics of the input data, such as the correlation between a feature and its class label, to identify the most relevant features with respect to its class label in a dataset. They are often considered as a preprocessing step and not associated with a specific learning algorithm. On the other hand, the Random Forests-based method first makes use of the feature ranking mechanism embedded in the training process of the Random Forests (RF) algorithm [25]. It then employs RF to identify a subset of features that renders the best classification performance. Hence, the RF-based feature selection algorithm is both embedded and wrapper-based [3].

The main reason of considering these eleven algorithms is two fold: First, the ten filter-based algorithms are commonly used in data analysis as a preprocessing step, probably because many of them are available in open source machine learning toolkits such as scikit-learn [11] or Weka [10]. Given a dataset, it is however unclear how the resulting feature subsets from different algorithms compare against each other. One prevalent approach is to involve a target learner (e.g., SVM) to evaluate the quality of each and every feature selection algorithm. This is time consuming. This work conducts extensive and rigorous evaluation of these algorithms to empirically put these algorithms into approximately equivalent groups. Second, it is important to compare these filter-based feature selection methods with an embedded and wrapper-based method. The RF-based feature selection algorithm described in [16] is considered as the state-of-art in this category. It is therefore included in this study.

As for the five classification algorithms under consideration, they are Support Vector Machine (SVM), Random Forests (RF), Naïve Bayes Classifier (NBC), k-Nearest Neighbors (kNN), and C4.5 decision tree classifier (C4.5) [19]. These algorithms are chosen after having considered both an algorithm’s popularity for text classification and its underlying principles. At the same time, it is necessary to include some “baseline algorithms” for comparison purposes. SVM and Random Forests are chosen because they (especially SVM) have been routinely applied to classify text data with top performance when comparing with other algorithms. SVM is an analogy-based learner that utilizes the similarities among data objects for classification. kNN is also an analogy-based classifier though with a much simpler concept. It is therefore included as a “baseline approach”. Random Forests is an induction tree-based approach, whereas the C4.5 decision tree classifier is included as a corresponding “baseline approach”.

Finally, Naïve Bayesian classifier is included due to its frequent applications in text analysis as a “baseline”. Additionally, unlike the other four algorithms, its underlying principal is Bayesian Theorem.

To achieve the aforementioned two goals more efficiently, an open source software system is implemented, which consists of the following two components: (1) feature selection that includes the eleven feature selection algorithms described earlier; and (2) classification that includes the five classifiers listed above. The classification component will take the output of the other component as input to build classification models and evaluate these models by performing k-fold cross validation. Next, this software system is applied to five different datasets to identify approximately equivalent groups of feature selection algorithms and to study the interplay between different feature selection algorithms and classifiers. Around 1000 experiments were conducted in the course of our evaluation, which have resulted in valuable observations. For instance, we have identified two approximately equivalent groups of feature selection algorithms. We have also observed that SVM shows the least bias against different feature selection algorithms and works best with Gini Index or Kruskal-Wallis. These observations will provide practical guidelines and also inspire further theoretical studies.

The rest of paper is organized as follows: Section II summarizes the most relevant work. Section III briefly describes the eleven feature selection algorithms considered in this study. Section IV presents the evaluation results. Finally, Section V concludes and identifies future directions.

II. RELATED WORK

A large number of studies have been conducted in the area of feature selection to design new algorithms, to study the limitations of existing algorithms, or to compare different algorithms to find optimal feature selection methods for a task at hand. A detailed study of the feature selection problem can be found in [2][3][4][5][6][7]. An extensive survey done by Forman [6] using various filter based feature selection algorithms found that ‘Bi-Normal Separation’ (BNS) and Information Gain performed substantially better in most situations. In [7], the authors studied the behavior of different feature selection algorithms based on the criteria of relevance, irrelevance and redundancy. Another excellent work is the ASU feature selection repository implemented as a MATLAB package [1]. Finally, many feature selection algorithms can be found in open source machine learning toolkits such as Weka [10] and scikit-learn [11].

In contrast to these previous studies and software toolkits, this study aims to empirically compare the eleven feature selection algorithms towards identifying approximately equivalent groups, and to characterize the interplay between feature selection and classification algorithms in the context of text classification. As a byproduct, the feature-selection-classification software system can be readily deployed to enable more studies originated in other application domains.
III. FEATURE-SELECTION-CLASSIFICATION SYSTEM

As mentioned earlier, an open source feature-selection-classification system was implemented to provide an integrated environment of feature selection and classification. This system consists of a feature selection component and a classification one, where the output of the former component can be seamlessly passed on to the classification component for model building and evaluation. The feature selection component is composed of eleven algorithms. The classification component includes five commonly used classification algorithms: Support Vector Machine (SVM), Random Forests (RF), Naïve Bayes Classifier (NBC), k-Nearest Neighbors (kNN), and C4.5 decision tree classifier (C4.5). Note that in this study, the primary goal of this system is to facilitate fast analysis of and comparison among the five classification algorithms. As for the five classification algorithms, they are the common ones used. They are Random Forests (RF), Naïve Bayes Classifier (NBC), k-Nearest Neighbors (kNN), and C4.5 decision tree classifier (C4.5).

1. ChiSquare Score

ChiSquare Score [21] is a commonly used statistical measure to test whether two variables are correlated. In the context of feature selection, it is applied to a labeled dataset to test whether the class label is independent of a given feature. The ChiSquare score for a feature with \( k \) different values and \( C \) classes is defined as follows:

\[
\chi^2 = \sum_{i=1}^{k} \sum_{j=1}^{c} \frac{(n_{ij} - \mu_{ij})^2}{\mu_{ij}}
\]  (1)

In (1), \( n_{ij} \) is the number of samples with the \( i^{th} \) value.

\[
\mu_{ij} = \frac{(n_{ij}n_{i}}{n}
\]  (2)

Here, \( n_{i} \) is the number of samples that takes the \( i^{th} \) value of a given feature. \( n_{ij} \) is the number of samples in the \( j^{th} \) class, and \( n \) is the number of samples in the input dataset.

A ChiSquare score is often accompanied with a \( p \)-value. A common practice is to retain all the features with a small \( p \)-value, e.g., \( \leq 0.05 \), which is also used in this study.

2. Information Gain

Information Gain [20] is an information entropy-based measure. In the context of feature selection, it requires a labeled dataset as input and calculates the amount of information gained (the more, the better) to distinguish different classes if a given feature is considered. Specifically, the Information Gain (IG) of a feature \( X \) given the class label \( Y \) is calculated as:

\[
IG(Y|X) = H(Y) - H(Y|X)
\]  (3)

In the above formula, \( H(Y) \) is the entropy of the class label \( Y \) and \( H(Y|X) \) is the conditional entropy of \( Y \) given \( X \). In our evaluation, all the features with a positive information gain are selected as relevant features w.r.t. the class label.

3. Fisher Score

Fisher Score [22] is a supervised feature selection algorithm and has been widely applied to many applications. It is designed to select features whose values are more uniformly distributed for samples in the same class but more dissimilar for samples in different classes. It is formulated as follows:

\[
SCF(f) = \frac{\sum_{j=1}^{c} n_{j}(\mu_{ij} - \mu_{j})^2}{\sum_{j=1}^{c} n_{j} \sigma_{ij}^2}
\]  (4)

In the above formula, \( \mu_{j} \) is the mean of the feature \( f \), \( n_{j} \) is the number of samples in the \( j^{th} \) class, \( \mu_{ij} \) and \( \sigma_{ij} \) are the mean and variance of \( f \) in the \( j \)th class, respectively.

4. Gini Index

Gini Index [20] is another commonly used and supervised measurement to select features that are better at distinguishing samples among different classes. It is used for induction tree-based classifiers such as Random Forests. Given \( C \) classes, the Gini Index of a feature \( f \) can be calculated as follows:

\[
GiniIndex(f) = 1 - \sum_{i=1}^{C} [p(l|f)]^2
\]  (5)

The smaller a feature’s Gini Index is, the more relevant it is to the class label. For a dataset of two classes, the maximal value of Gini Index is 0.5. In our evaluation, we select the top \( k \) features with the smallest Gini Index, where \( k \) is specified by the end-user.

5. Kruskal-Wallis

The Kruskal–Wallis statistic [14] is non-parametric. In the context of feature selection, it follows a similar motivation as Fisher Score. Given a labeled dataset of \( C \) classes, in order to test whether a feature \( f \) is relevant to the class label, one would first partition \( f \)’s values into \( C \) groups according to their class membership. Next, one calculates the Kruskal-Wallis statistics to test whether these \( C \) groups are significantly different. One can select a feature if its Kruskal-Wallis statistics meets a \( p \)-value threshold, e.g., 5%. In our evaluation, we consider the top \( k \) features with the highest Kruskal-Wallis scores.

6. Laplacian Score

Laplacian score [13] is the only one among the eleven-feature selection methods that could handle both labeled and unlabeled data. It evaluates features based on their locality (or class in supervised cases) preserving power. In this sense, it is similar to both Fisher Score and Kruskal-Wallis. (Fisher Score...
has been shown to be a special case of Laplacian score in a supervised context.) Given an affinity matrix \( K \), its corresponding degree matrix \( D \) and Laplacian matrix \( L \), the laplacian score of a feature \( f \) is given as follows:

\[
SC_L(f) = \frac{\bar{f}^T L \bar{f}}{\bar{f}^T D \bar{f}}
\]

where,

\[
\bar{f} = f - \frac{f^T D 1}{1^T D 1} 1
\]

In our evaluation, we select the top \( k \) features with minimal \( SC_L \) values, where \( k \) is a user-specified value.

**A quick summary:** Readers probably have observed a common characteristic of the above six feature selections algorithms--all of them are univariate, i.e., selecting features one at a time. As a result, they can’t avoid selecting redundant yet relevant features. The following algorithms on the other hand have the capability to deal with redundant features.

7. **Relieff**

Relieff [12] is a supervised method that is built upon the Relief [17] algorithm, which was designed to select features from a \( 2 \)-class dataset. Relieff has been shown to handle noise very well and deliver superior performance in many different applications. The main idea is to select features that can distinguish instances originated in different classes. Let \( p \) be the number of randomly sampled instances from a labeled dataset, Relieff is defined as follows:

\[
LL_{\bar{f}}(L_y) = \frac{1}{L} \sum_{t=1}^{L} \left( \frac{1}{L - l} \sum_{l \in L} i_l(t) \right) + \sum_{l \in L} \left( \frac{1}{L - l} \sum_{l \in L} i_l(t) \right)
\]

Here, \( y_{xt} \) is the class label of the instance \( x_t \) and \( P(y) \) is the probability of an instance being from the class \( y \). \( NH(x) \) or \( NM(x, y) \) denotes a set of nearest points to \( x \) with the same class of \( x \), or a different class (the class \( y \), respectively). \( m_y \) and \( m_{x,y} \) are the sizes of the sets \( NH(x_t) \) and \( NM(x_t, y) \), respectively. Usually, the size of both \( NH(x) \) and \( NM(x, y) \), \( \forall y \neq y_{x,t} \) is set to a pre-specified constant \( k \).

8. **Fast Correlation-Based Filter (FCBF)**

FCBF [15] is a supervised method that not only considers the correlation between a feature and the class label, but also the correlation between features. It consists of two major steps: First, it selects features that are highly relevant to the class label based on an entropy-based measure. Second, it utilizes several heuristics to remove redundant features.

9. **Correlation-based Feature Selection (CFS)**

CFS [23] is a supervised feature selection algorithm and shares similarity with FCBF. CFS uses a greedy approach to identify a subset of feature that not only has high predictive power of the class label but also exhibits little redundancy. Both feature-feature correlations and feature-class correlations are considered to heuristically measure the quality of the selected feature subsets. Information entropy-based measurements are also used to determine whether to include a given feature.

10. **Minimum-Redundancy-Maximum-Relevance (mRmR)**

mRmR [24] is a supervised approach aiming to select features that are highly correlated to the class label (i.e., maximum relevance) yet mutually far away from each other (i.e., minimum redundancy). It can be considered as an approximate approach to maximize the dependency between the class label and the joint distribution of all the selected features.

11. **Variable Selection using Random Forest (VSRF)**

VSRF [16] is a two-step feature selection algorithm that is both embedded and wrapper based. It is embedded as it uses Random Forest’s built-in variable importance measure to first identify a preliminary candidate feature subset. It is wrapper-based as it employs RF itself to identify the final subset of features. Below are the two steps required by VSRF:

Step 1. Identify a subset of \( m \) features based on their variable importance score calculated by RF. The value of \( m \) is determined by another sub-process, which determines a minimum threshold of variable importance.

Step 2. Build a sequence of RF models starting with the most important feature, the top two most important features, the top three, and so on, finally all the \( m \) features. The feature subset that leads to the model with the smallest OOB (out-of-bag) error is returned as the selected feature subset. The OOB error measures the prediction error of RF. It is calculated as the mean prediction error on each training example using the trees that were not trained on this example.

Software implementation: This feature-selection-classification software system is implemented primarily in Java. We have implemented FCBF, Gini Index, Kruskal-Wallis, Fisher Score, and the Random Forests-based feature selection algorithm. For the remaining feature selection and all the classification algorithms, we use Weka’s implementation with the following exception: the Laplacian Score implementation is from scikit-learn and Random Forests is from Livingston [9].

IV. **Evaluation**

This section presents the main results from our extensive evaluation that is designed to address the following two main goals stated in Section I:

1) Given an input dataset, how will the resulting feature subsets from different selection algorithms compare with each other? Can one empirically put these algorithms into *approximately equivalent* groups?

2) Given a classification algorithm, how will it be affected by different feature selection algorithms? Will there be an “optimal” feature-selection-classifier combination?
A. Datasets

Five text datasets originated from different application domains are used in our evaluation. Table 1 summarizes their main characteristics. All these datasets are labeled and four of them have three or more classes.

The Polarity and Strength datasets are concerned with sentence-level entity-entity relationships in a biomedical article. They are hand annotated by a group of three people [8]. Their features capture both syntactic and semantic aspects of a sentence, moreover, the sequential arrangement of different semantic components. The Movie Reviews and Webtext datasets are created from the two corresponding corpora included in the Python NLTK toolkit for natural language processing. Both datasets contain features of frequently used nouns, adjectives, adverbs and verbs. Additionally, the Webtext dataset also includes most frequently used bigrams. Finally, the Amazon dataset is obtained from the UCI ML repository [18]. It consists of product reviews on amazon.com from 50 most active users. We consider 5 users in our evaluation. We consider the following features for this dataset: frequently occurring unigrams, bigrams, and trigrams.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#Instances</th>
<th>Dimensionality</th>
<th>#Classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Polarity</td>
<td>757</td>
<td>1230</td>
<td>4</td>
</tr>
<tr>
<td>Strength</td>
<td>757</td>
<td>1230</td>
<td>3</td>
</tr>
<tr>
<td>Movie Reviews</td>
<td>2000</td>
<td>1447</td>
<td>2</td>
</tr>
<tr>
<td>Webtext</td>
<td>1376</td>
<td>1651</td>
<td>5</td>
</tr>
<tr>
<td>Amazon</td>
<td>150</td>
<td>640</td>
<td>5</td>
</tr>
</tbody>
</table>

TABLE 1. DATASET DESCRIPTION

B. Comparing the feature subsets selected by different algorithms

The main goal for this set of evaluations is to put the eleven feature selection algorithms into approximately equivalent groups (AEGs) where given a dataset, algorithms in the same group select a similar set of features. Let Alg1 and Alg2 be the two feature selection algorithms. Let FS1 and FS2 be the subsets of features selected by Alg1 and Alg2 from a given input dataset respectively. The Jaccard similarity coefficient of FS1 and FS2 is used to quantify the similarity between Alg1 and Alg2.

\[
\text{Similarity}(\text{Alg}_1, \text{Alg}_2) = \frac{|FS_1 \cap FS_2|}{|FS_1 \cup FS_2|}
\]

The range of the above similarity is in [0, 1.00], where 1.00 means a perfect match between FS1 and FS2. Venn diagrams [28] are employed to visualize the similarity of two feature subsets by showing the intersections of FS1 and FS2.

For fair comparison, the eleven algorithms are put into the following two groups: (1) the feature selection group that returns a subset of features, and (2) the feature ranking group that ranks all the features in a dataset. A value k is required to select the top k ranked features. k is set as 25% based on our extensive experiments as described in the following subsection. Among the eleven algorithms, ChiSquare, Information Gain, CFS, FCBF, mRmR, and VSRF are in the selection group; Gini Index, Fisher Score, Laplacian Score, ReliefF, and Kruskal-Wallis are in the ranking group. Below, we first present comparison results among feature selection algorithms, then feature ranking algorithms, and finally between these two groups.

**Definition:** Two feature selection algorithms are considered to be in one *approximately equivalent group* (AEG) if their mean similarity score over the five datasets under study is ≥0.80 with a standard deviation within 0.15.

**Fig. 1** Venn-diagrams of the six feature selections algorithms evaluated on the Movie Reviews dataset. Below each diagram is the similarity score between two algorithms.

**Fig. 2** Venn-diagrams of the six feature selection algorithms evaluated on the Polarity Dataset. Below each diagram is the similarity score between two algorithms.

Shown in Fig. 1 and Fig. 2 are two collections of Venn-diagrams that compare the six feature selection algorithms evaluated on the Movie Reviews and Polarity datasets, respectively. Note that similar evaluations were conducted on all the five datasets. Limited by space, we are not able to include all the five sets of results. Please find the complete set of results at https://db.tt/WUzXAZ3z

From these two figures, one can visually observe that:

1) Information Gain and ChiSquare select identical features from both Movie Reviews and Polarity datasets. Furthermore, when examining their similarity across all
five datasets, their similarity scores fall in a tight interval of [0.93, 1.0] with a mean of 0.97±0.03. Hence, these two algorithms form an AEG.

2) mRmR and CFS are the next pair of algorithms that select highly similar features, followed by mRmR and FCBF, then CFS and FCBF. Specifically, the average similarity of (mRmR, CFS) across all five datasets is 0.84±0.23, that of (mRmR, FCBF) is 0.72±0.25, and of (CFS, FCBF) is 0.65±0.32. This to certain extent verifies our intuition as all three algorithms utilize correlations between all pairs of features in a dataset to select highly relevant yet non-redundant features. They however can’t be put into the same AEG because their similarity scores fluctuate widely when applied to different datasets.

To further investigate whether two ranking algorithms produce similar ranks for the top 25% features, these features are partitioned into two smaller groups; top 12.5% and the next 12.5% ranked features. Fig. 5 and Fig. 6 visualize the pairwise comparisons among the five ranking algorithms using the Amazon and Webtext datasets, respectively. In these two figures, Venn-diagrams on the first row compare the top 12.5% ranked features, those on the second row the next 12.5%. One can see from these figures that the top 12.5% features ranked by Fisher Score and Laplacian Score are largely similar (0.9 and 0.95 in similarity). The same holds for Fisher Score and Gini Index (1.00 and 0.93 in similarity). On the other hand, the similarity between Kruskal-Wallis and the other three—Fisher Score, Laplacian Score and Gini Index—has a similarity of 0.58 or lower for the top 12.5% features except those from the Webtext dataset (0.90 similarity). This suggests that Fisher Score, Laplacian Score, and Gini Index rank the top 12.5% features more similarly than they rank the next 12.5%.

Fig. 3 Venn-diagrams of the top 25% features from the five feature ranking algorithms evaluated on the Movie Reviews dataset. Below each diagram is the similarity score between two algorithms.

Fig. 3 and Fig. 4 compare the top 25% ranked features from running the five feature ranking algorithms on the Movie Reviews and Polarity datasets, respectively. From these two figures, one can visually observe the following:

1) Fisher Score, Laplacian Score and Gini Index form an AEG. Having considered all the five datasets, the similarity score between the first two algorithms lies in [0.81, 0.96] with an average of 0.90±0.07. similarity(Laplacian, Gini) across all five datasets is in [0.66, 0.96] with a mean of 0.81±0.11. Finally, similarity(Fisher Score, Gini Index) is in [0.65, 1.00] with a mean of 0.87±0.13. It is not surprising to see a high similarity between Laplacian Score and Fisher Score, as it has been pointed in literature the latter is a special case of the former [13]. It is however unexpected to observe that Gini Index ranks features similarly as the other two when considering the top 25% ranked features.

2) Kruskal-Wallis statistics ranks the top 25% features similarly when compared to the above group on all the datasets except the Polarity dataset. Specifically, similarity(Kruskal-Wallis, Fisher) over the four datasets is in [0.62, 0.86] with a mean of 0.73±0.11; similarity(Kruskal-Wallis, Laplacian) is in [0.63, 0.86] with a mean of 0.71±0.11; and finally similarity(Kruskal-Wallis, Gini Index) is [0.67, 0.86] with a mean of 0.77±0.08.

Fig. 4 Venn-diagrams of the top 25% features from the five feature ranking algorithms evaluated on the Polarity dataset. Below each diagram is the similarity score between two algorithms.

Fig. 5 Venn-diagrams of the top 12.5% features (1st row) and the next 12.5% (2nd row) from the five feature ranking algorithms evaluated on the Movie dataset. Below each diagram is the similarity score.

Fig. 6 Venn-diagrams of the top 12.5% features (1st row) and the next 12.5% (2nd row) from the five feature ranking algorithms evaluated on the Strength dataset. Below each diagram is the similarity score.
Finally, we compare the six feature selection algorithms with the five ranking algorithms. Shown in Fig. 7 and Fig. 8 are the Venn-diagrams of the pair-wise comparison results on the Movie Reviews and Polarity datasets, respectively. Let k be the number of features selected by the former, these k features are compared with the top k features returned by the ranking algorithm. From Fig. 7 and Fig. 8, one can observe that:

- The AEG (ChiSquare, Information Gain) shares a great deal of similarity with another AEG: (Fisher Score, Gini Index, Laplacian Score) when considering the same number of selected features. Specifically, considering all the five datasets, the similarity score between Chi Square (or Information Gain) and the Fisher Score (or the Gini Index) is in [.51, .98] with a mean of .75 +/- .18. Excluding the Polarity dataset, the similarity score between Chi Square (or Information Gain) and the Laplacian score is in [.66, .96] with a mean of .80 +/- .18.

![Fig. 7 Venn-diagrams of the six feature selection and five ranking algorithms evaluated on the Movie Reviews dataset. Below each diagram is the similarity score.](image)

![Fig. 8 Venn-diagrams of the six feature selection and five ranking algorithms evaluated on the Polarity dataset. Below each diagram is the similarity score.](image)

**Summary**: Based on the extensive and systematic evaluations of the eleven feature selection/ranking algorithms, following two AEGs have been identified: (ChiSquare, Information Gain), and (Fisher Score, Laplacian Score, Gini Index) when considering the top 25%-ranked features. Additionally, even though CFS, FCBF, and mRmR do not form an AEG, their similarity is considerably high. Kruskal-Wallis produces similar results to that of second AEG on four of the five datasets. Finally, the set of top ranked features from the second AEG significantly overlaps with the set of features selected by the first AEG group.

**C. Evaluating the interplay between feature selection and classification algorithms**

Given a classification algorithm, how will it be affected by different feature selection algorithms? Furthermore, will there be an “optimal” feature-selection-classifier combination? To address these questions, we have conducted a series of evaluation. Recall that our feature-selection-classification system (Section III) contains eleven feature selection and five classification algorithms. For each feature selection/ranking algorithm, the selected or top-ranked features are evaluated by each of the five classification algorithms using 10-fold cross validation. The following four measures are used to measure the classification performance: accuracy, precision, recall and F1-measure [27]. Due to space limit, this section will report only the F1-measure-based results. Please visit https://db.tt/WUzXAZ3z for results using the other measures. During our evaluation, we use the linear kernel for SVM. k is set to 5 for the kNN algorithm, For Random Forests, the ntrees parameter (i.e., number of trees) is set to 20. The mtry parameter, i.e., the number of features available for splitting at each tree node, is set to m/3, where m is the total number of features.

Before presenting the results, one needs to first decide the number of top-ranked features, denoted as k, to use for classification if the algorithm is one of the following five feature ranking algorithms: Gini Index, Fisher Score, Laplacian Score, ReliefF, and Kruskal-Wallis.

Since there are no guidelines in the literature on choosing the value k, k is set to the following four values for comparison: (i) the maximal number of features selected by the six feature selection algorithms for a given dataset; (ii) the minimal number of features returned by the six feature selection algorithms for a given dataset; (iii) 25% of the total number of original features in a given dataset, and (iv) 50% of the total number of original features in a given dataset. We compare the classification performance of the five classifiers on all the five datasets by setting k to each of the above four values. The third case, choosing the top 25% ranked features dominantly delivers the best overall classification performance. Hence, k is set to this value for all the evaluation results presented below.

Shown in Fig. 9 through Fig. 13 are the average F1-measures of the five classification algorithms—SVM, Naïve Bayesian, Random Forests, C4.5 Decision tree, and kNN—being evaluated on the five datasets—Polarity, Strength, Movie Reviews, Webtext and Amazon, respectively, after each dataset is pre-processed by each of the eleven feature selection algorithms. For brevity, these feature selection algorithms are coded and shown in Table 2. Note that the last code corresponds
to “All original features”, that is, all the features in the input dataset will be considered for classification.

<table>
<thead>
<tr>
<th>Code</th>
<th>Algorithm</th>
<th>Code</th>
<th>Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ChiSquare</td>
<td>2</td>
<td>Information Gain</td>
</tr>
<tr>
<td>3</td>
<td>ReliefF</td>
<td>4</td>
<td>VSRF</td>
</tr>
<tr>
<td>5</td>
<td>Fisher Score</td>
<td>6</td>
<td>Laplacian Score</td>
</tr>
<tr>
<td>7</td>
<td>FCBF</td>
<td>8</td>
<td>CFS</td>
</tr>
<tr>
<td>9</td>
<td>mRmR</td>
<td>10</td>
<td>Kruskal-Wallis</td>
</tr>
<tr>
<td>11</td>
<td>Gini Index</td>
<td>12</td>
<td>All original features</td>
</tr>
</tbody>
</table>

**TABLE 2. CODING SCHEMA OF THE ELEVEN FEATURE SELECTION ALGORITHMS IN Fig. 9 THROUGH Fig. 13.**

Observing the classification performance data presented in Fig. 9 through Fig. 13, one can clearly discern the interplay between feature selection and classification. Specifically,

- Among the five classifiers, SVM delivers the most stable and superior performance across all five datasets with respect to all eleven feature-selection methods. Comparing to the full-feature model (the rearmost bar of the SVM bar-group in Fig. 9 through Fig. 13), it is noticed that some feature selection algorithms lead to comparable or even better performance of SVM. Among all the feature selection algorithms, Kruskal-Wallis and Gini Index consistently result in the highest F1-measure except in the case of the Webtext dataset, where ReliefF and Random Forests-based methods lead to the two highest SVM performances. These suggest that (SVM, Gini Index) and (SVM, Kruskal-Wallis) are two “optimal” combinations.

- In contrast to SVM, kNN on the hand exhibits the widest range of F1 values w.r.t. the different feature selection algorithms across the five datasets. This, we believe, is because some of these feature selection algorithms do not preserve the neighborhoods formed in the original feature space. Out of the eleven feature selection algorithms, one can clearly observe that the following algorithms work best with kNN: Random Forests-based variable selection (VSRF), FCBF, CFS and mRmR. In such cases, kNN
delivers better performance than using all the original features for all the five datasets. Considering that kNN relies on the labels of \( k \) nearest neighbors to classify new data objects, this suggests that these four feature selection algorithms are capable to preserve locality among objects in the newly reduced space. It is surprising to see the absence of Laplacian Score given that its design principle is to preserve locality.

- **Naïve Bayesian classifier (NB)** delivers the second best overall F1 values across the five datasets. We also notice that Gini Index and Kruskal-Wallis lead to the best performance of NB, which is better than or comparable with using all the original features in all the datasets except Strength. On the other hand, except the Strength dataset, NB obtains one of the top F1 measures when using all the features. Considering the simplicity of Naïve Bayesian and that it is robust against involving irrelevant features, it may not be worthwhile going through a feature selection step.

- **Besides Naïve Bayesian, Random Forests is another classifier** that is probably best without including feature selection in the preprocess phase. First, Random Forests contain an inherent component for feature selection. Second, as shown in Fig. 9 through Fig. 13, with the exception of the Strength dataset, Random Forests using all the original features delivers one of the best performances. When compared with SVM, it is however a bit disappointing to see that Random Forests is not as stable in our evaluation.

- **Compared to Random Forests, the decision tree classifier behaves quite differently towards the eleven feature selection algorithms.** It works the best with the following four algorithms across all the five datasets: Fisher Score, Laplacian Score, Kruskal-Wallis, and Gini Index. One can observe that even though only 25% of the original features are used, decision tree classifier is able to achieve equally competitive F1-measures as using all the original features for all the five datasets.

Using Fig. 9 through Fig. 13, one can also identify the feature selection algorithms that result in the best overall classification performances. Specifically, for each of the five classification algorithms, we identify the feature selection algorithms that lead to the top three F1-measures on a given dataset. Ties are allowed. We then aggregate the total number of times that a feature selection algorithm is associated with a top-three classification performance. Note that the maximal number is 25 since there are a total of 25 such classification-dataset combinations. Table 3 summarizes the above results. Gini Index, Kruskal-Wallis and VSRF are the most successful at leading to one of the best F1-measures for all the five classifiers across all the five datasets. Laplacian Score, ChiSquare and Information Gain are the least successful. Finally, it is a bit disappointing to observe that Fisher Score and ReliefF, two of the most applied algorithms, fall short of expectation.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>#(Top 3)</th>
<th>Algorithm</th>
<th>#(Top 3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ChiSquare</td>
<td>5</td>
<td>Information Gain</td>
<td>6</td>
</tr>
<tr>
<td>ReliefF</td>
<td>8</td>
<td>VSRF</td>
<td>16</td>
</tr>
<tr>
<td>Fisher Score</td>
<td>8</td>
<td>Laplacian Score</td>
<td>4</td>
</tr>
<tr>
<td>FCBF</td>
<td>11</td>
<td>CFS</td>
<td>11</td>
</tr>
<tr>
<td>mRmR</td>
<td>14</td>
<td>Kruskal-Wallis</td>
<td>18</td>
</tr>
<tr>
<td>Gini Index</td>
<td>21</td>
<td>All original features</td>
<td>15</td>
</tr>
</tbody>
</table>

**Table 3.** Total number of times a feature selection algorithm has led to one of the top three F1-measures. The maximal number is 25.

**Summary:** From the above analysis of the interplay between feature selection and classification algorithms, following characteristics have been observed: (1) SVM exhibits the least bias towards different feature selection algorithms. It however works best with Gini Index or Kruskal-Wallis; (2) kNN works the best with VSRF, FCBF, CFS and mRmR; (3) C4.5 decision tree classifier works best with Fisher Score, Laplacian Score, Kruskal-Wallis, and Gini Index; (4) Naïve Bayesian and Random Forests are best to run with all the original features; and finally, (5) Gini Index, Kruskal-Wallis, and VSRF are the most successful at resulting top classification performance. Fisher score and Laplacian score exhibited inferior performance to Gini Index.

**D. Runtime analysis**

Table 4 tabulates the runtime (in seconds) of each algorithm on the five datasets on a personal computer with 2.3GHz processor (4 cores) and 8GB memory. VSRF is the most costly algorithm with a runtime several orders of magnitude higher than all the other filter-based methods. Combining this with the results shown in the previous subsection, one could conclude that for VSRF, the tradeoff between the quality of the selected features and its runtime is not proportional, hence not worthwhile.

**V. Conclusion and Discussion**

This article presents a comprehensive study to understand and characterize the similarity among eleven feature selection algorithms. Furthermore, a careful investigation is conducted to study how these different feature selection algorithms interact with five different classifiers in the context of text classification. To facilitate this study, an open source, feature-selection-classification software system, is developed...
With respect to the eleven feature selection algorithms, the following approximately equivalent groups have been empirically identified: (ChiSquare, Information Gain) and (Fisher Score, Gini Index, Laplacian Score) when considering the top 25%-ranked features. Additionally, features selected by the first group significantly overlap with the top 25% features ranked by the second group. Finally, the features selected by FBCF, CFS, and mRmR tend to be significantly similar as well, even though these three algorithms do not meet the criteria to form an AEG.

Out of the five classifiers, SVM is the most reliable and shows the least bias against a given feature selection method. kNN coupled with VSRF, FCBF, CFS and mRmR often produces top performances. This suggests the locality-preserving capability of such feature selection algorithms. The features selected by Fisher Score, Laplacian Score, Kruskal-Wallis, and Gini Index tend to lead to top classification performance of C4.5. Finally, the results suggest that it is not worthwhile to perform feature selection for either Naïve Bayes or Random Forests.

Overall, Gini Index, Kruskal-Wallis and Random Forests-based feature (VSRF) selection are the top algorithms that frequently lead to top classification performances in our evaluation. However, VSRF takes inordinately long time (in days) to finish when compared with the other two (in seconds).

The above results are drawn from extensive empirical evaluations. Further experimental studies are required on datasets with even higher dimensionality. This study however lays the foundation for researchers to pursue the following two fundamental questions: (1) can one theoretically justify all the approximately equivalent groups identified in this study? And (2) given a collection of feature selection algorithms and another collection of classifiers, what are the underlying criteria one can follow to obtain an “optimal” combination? Answers to such questions will provide the much-needed guidance in today’s largely “trial-and-error” style of data analysis.

REFERENCES