RESEARCH

Stable Bagging Feature Selection on Medical Data

Salem Alelyani

Correspondence:
s.alelyani@kku.edu.sa
Department of Computer Science, Center for Artificial Intelligence, King Khalid University, P.O. Box 9004, Abha 61413, Saudi Arabia
Full list of author information is available at the end of the article

Abstract

In the medical field, distinguishing genes that are relevant to a specific disease, let's say colon cancer, is crucial to finding a cure and understanding its causes and subsequent complications. Usually, medical datasets are comprised of immensely complex dimensions with considerably small sample size. Thus, for domain experts, such as biologists, the task of identifying these genes have become a very challenging one, to say the least. Feature selection is a technique that aims to select these genes, or features in machine learning field with respect to the disease. However, learning from a medical dataset to identify relevant features suffers from the *curse-of-dimensionality*. Due to a large number of features with a small sample size, the selection usually returns a different subset each time a new sample is introduced into the dataset. This selection instability is intrinsically related to data variance. We assume that reducing data variance improves selection stability. In this paper, we propose an ensemble approach based on the bagging technique to improve feature selection stability in medical datasets via data variance reduction. We conducted an experiment using four microarray datasets each of which suffers from high dimensionality and relatively small sample size. On each dataset, we applied five well-known feature selection algorithms to select varying number of features. The results of the selection stability and accuracy show the improvement in terms of both the stability and the accuracy with the bagging technique.

Keywords: Feature Selection; Ensemble Technique; Bagging; Dimensionality Reduction; Medical Data; Microarray; Variance; Bias

Introduction

In the growth of data mining and collection technologies, data learning and understanding are a tedious task due to a large number of features present that are known as variables or attributes. Usually, data harvesting is conducted in relation to a specific problem, such as collecting human genomes from patients for a particular disease, gathering social media data for gender identification, or crawling websites for offensive materials to name just a few. In this paper, we call this problem-specific a class. When we know the class of the dataset, the learning is called *supervised learning*. Otherwise, it is called *unsupervised learning* [1], [2], [3].

Most of the collected data suffer from high dimensionality the includes a high number of features. Most of these features are irrelevant and noisy [4], [5]. The noise in the dataset should be eliminated before the learning segment and only depend on a relevant feature set to accomplish reasonable learning accuracy leading to a reduction in the computational cost [6]. Assume $X = \{x_1, x_2, ..., x_m\}$ to be the dataset m-*by*-n matrix where m is the number of samples and the vector x_i is the i^{th}

sample with n features. Therefore, the feature selection algorithm $f()$ with respect to the class y could be represented in the following mathematical equation:

$$f(X, y) \rightarrow X',$$

where X' is a *m-by-r* and $r \leq n$, in real cases $r \ll n$. The set of features that are in X but not in X' are the irrelevant features to y. Moreover, $f()$ maximizes learning performance while minimizing r.

There are few indicators of how good the feature selection algorithm $f()$ is. Learning performance, such as: classification accuracy, is the most utilized indicator of the quality of $f()$. Usually, the learning performance can be described as $\psi(X') \geq \psi(X)$. Another indicator is the computational cost which is markedly smaller in X' than X due to the removal of irrelevant features. The third indicator has gained increasing attention in recent times which is due to selection stability. In machine learning tasks, we assume to build a learning model that is applicable to new samples from the same domain. However, this is not always the case; it reduces occurrence of the domain experts within our learning model. Feature selection stability $\varphi()$ is the measure of how robust the feature selection algorithm $f()$ is with respect to a certain level of permutation on data set X during the selection and learning processes.

In the realm of medicine, machine learning has been widely adopted and applied during the last few decades to expedite diagnosis and to deliver profoundly accurate inferences [7], [8]. Computerized Tomography Scan (CT-Scans), for instance, generate a large number of images for each patient. The physician's decision depends heavily upon analyzing these images and other related results [9]. However, it is time-consuming and difficult to identify diseases in a timely manner, especially, in case of early disease. Therefore, machine learning might play a critical role in supporting a doctor's decision [5]. One form of medical diagnosis and classification stem from *microarrays* which is a representational array of genome expression levels. Microarrays are burdened with vital issues when it comes to machine learning tasks, which we will discuss in more details in the sections below. The most important issue in microarrays is known as *curse-of-dimensionality* [7], [8], [10]. Typically, it has a large number of features, i.e., genes, with a small sample size, i.e., number of patients. Most of these features are irrelevant to the class causing lower learning performance and a sharp rise in computational costs. Thus, we need to select the most relevant subset features.

Relevant features set to a specific class is found to drastically differ with an introduction of new samples [11], [12], [13]. The measurement of this change is called *feature selection stability*. The stability of feature selection algorithm $\varphi(f())$ means how sensitive the algorithm is to a perturbation on sample sets [14]. For illustration, assume we have microarray X with n genes for m patients who have been preliminarily diagnosed with colon cancer. Biologically, scientists believe that the number of genes related to this kind of cancer is very small comparison to the total number of genomes in the human DNA; thus, a small subset of n features should only be used to build the learning model. If feature selection algorithms select different subsets of genes each time we introduce new patients, domain experts would have less trust in our algorithms.

In this paper, we aim to propose an ensemble approach based on the bagging technique to improve stability while maintaining or improving learning performance, i.e., classification accuracy. The remainder of the paper is organized as follows: (i) we introduce feature selection algorithms in Section , (ii) we give a literature review of the stability and how to evaluate it in Section , (iii) we provide the proposed ensemble method in Section , (iv) we conduct an experiment on microarray datasets in Section , and (v) we discuss the results and conclude the paper.

Feature Selection Algorithm

Feature selection is one family of dimensionality reduction algorithms [15], [16], [17]. The other family is the feature extraction algorithms. The latter projects features into lower dimensional space; the newly generated features are not in the original space [18], [19]. Therefore, it cannot be interpreted or understood by domain experts. In the medical domain, if we apply Principal Component Analysis PCA, for instance, on a microarray, features in the new lower dimensional dataset are not genes. Hence, biologists will not be able to determine which gene is relevant to the problem and which is not. On the other hand, feature selection keeps the original representation of features, and as a result, is interpreted, justifiable, and understood by domain experts.

A large number of feature selection algorithms have been proposed to improve learning performance while reducing computational cost. These algorithms can broadly be categorized by either according to the utilization of class information and ***supervised or unsupervised algorithms***. [2], [3] reviews these two categories of algorithms. In this paper, we will be using supervised feature selection algorithms since we have datasets with class labels [20]. Another categorization is with respect to utilization of a classifier during the selection process. The ***filter model*** is the most widely used in this category; it is independent of any classifier. For example, Information Gain is a filter model that measures the amount of information of feature f_i given the class label y, $IG(f_i|y)$ [21], [22].

In contrast to the filter model, the ***wrapper model*** is dependent on a given classifier. Generally, a greedy approach is utilized to select different sets of features, and the classifier evaluates the quality of the selected sets with the best being Support Vector Machine Recursive Feature Elimination (SVM-RFE). It is one popular example of this model [23]. From its name, the algorithm starts by selecting the whole set of features and recursively eliminates features that are not effective in predicting the class label using the Support Vector Machine SVM classifier.

There are several benefits for each model. While the filter process is not as computationally costly as the wrapper model, the latter is more accurate in class prediction than the former since it uses the classifier in its selection process. Yet, the wrapper model usually tends to overfit especially with higher dimensionality, whereas filter generalizes easily. Therefore, a ***hybrid model*** was proposed to merge the two models and to benefit from the advantages of each while minimizing and removing the disadvantages. It selects different sets using the filter methodology and then compares the different sets using the wrapper approach. Thus, it is theoretically less complex than the wrapper model while more effective than the filter model.

Feature Selection Stability

Biologists believe that genes related to a specific type of cancer to be a limited set of genes. In other words, a variant of cancer can be caused by permutations of genes, and these genes are almost always the same with different patients. However, feature selections algorithms show instability in selected feature sets with a small perturbation in the data set, such as: introducing new samples. Therefore, instability would lower the confidence of domain experts in the selection algorithm even if it performs well due to a contradiction with a biological fact [24], [25], [26]. Hence, the stability of feature selection algorithms is a highly desired characteristic.

Stability of feature selection algorithms can be defined as how similar the selected feature sets with specific levels of perturbation on the training datasets in a controlled environment [27], [28], [29]. It is impacted mostly by data *variance*. The variance, in terms of feature selection, is the variability of the selection model for a given sample. Assume, we have a microarray dataset X with 100 samples. Then, we randomly generated two sets X_1 and X_2 out of X, where $X_1 = \{x_1, x_2, ..., x_{90}\}$ and $X_2 = \{x_{10}, x_{11}, ..., x_{100}\}$. The overlap between X_1 and X_2 is exactly 80%. The following equations illustrate the stability notion:

$$f(X_1, y_1) = X_1' and\ f(X_2, y_2) = X_2'$$

Assume, $\mathcal{F}(X_1')$ is the selected set of features and we call it $\mathcal{F}_1$ for short. Similarly, $\mathcal{F}_2$ is the selected feature set of X_2. Without loss of generality, we are going to use $\mathcal{F}$ to represent the selected features whenever we need to represent dataset X'. The stability $\varphi(f)$ of algorithm $f()$ can be estimated by the evaluation of similarity of $\mathcal{F}_1$ and $\mathcal{F}_2$ as follows:

$$\varphi(f(X_1, y_1), f(X_2, y)) = \Psi(\mathcal{F}_1, \mathcal{F}_2); \tag{1}$$

where Ψ is a similarity measure between $\mathcal{F}_1$ and $\mathcal{F}_2$. There are several measures to evaluate stability [14], [30], [31]. For example, the *Canberra Distance Measure* can be used to find the similarity between two ranked sets of features, while the *Euclidean Distance Measure* could be utilized to measure the distance between two sets of feature weights. Yet in this paper, we used the ***Jaccard Index Measure*** to measure the stability of the selected set, since $\mathcal{F}$ contains a set of feature indexes with no specific order, rank, or weight.

Jaccard Index

To evaluate the similarity between two selected subsets of features, Jaccard Index **Eq**(2) calculates the proportion of intersection of the selected features to the total number of selected features in the two subsets combined.

$$\mathcal{J}(\mathcal{F}_1, \mathcal{F}_2) = \frac{|\mathcal{F}_1 \cap \mathcal{F}_2|}{|\mathcal{F}_1 \cup \mathcal{F}_2|} \tag{2}$$

Jaccard has several advantages as a stability measure. First, it is symmetric, thus, $\mathcal{J}(\mathcal{F}_1, \mathcal{F}_2) = \mathcal{J}(\mathcal{F}_2, \mathcal{F}_1)$. Also, it is monotonic. In other words, the larger the intersection between $\mathcal{F}_1$ and $\mathcal{F}_2$, the larger $\mathcal{J}$ will be. Besides that, Jaccard is always

between 0 and 1.
On the other hand, Jaccard has some disadvantages in some cases. For instance, $\mathcal{J}$ is larger when the size of selected feature subset is close to the original feature space size. In this case, intersection by chance is very likely. Yet, this is seldom in real cases since we usually aim to minimize the selected feature subset size. After all, Jaccard is widely used as a stability metric, and hence, we use it in this paper.

Generally, we evaluate the stability of an algorithm with respect to a data set by either (1) randomly sampling of different samples from the dataset or (2) l-fold cross-validation. **Eq.**(3) illustrates the mathematical form of l sampling time. The larger l is, the more accurate stability estimation will be.

$$S(\mathbf{F}') = \frac{2}{l(l-1)} \sum_{i=1}^{l-1} \sum_{j=i+1}^{l} \mathcal{J}(\mathcal{F}'_i, \mathcal{F}'_j) \tag{3}$$

Where $\mathbf{F}' = \{\mathcal{F}'_1, \mathcal{F}'_2, ..., \mathcal{F}'_l\}$.

Proposed Method: Bagging Feature Selection

In Sections and , we have seen that feature selection algorithms are utilized to reduce dimensionality by removing irrelevant features while maintaining learning performance. In some domains; such as the medical field, they have been used to identify a subset of features, say genes, that are relevant to the problem, let us suppose colon cancer. Biologists believe that sets of genes related to a certain disease are almost always the same. However, when we apply a feature selection algorithm to a dataset for several times with introduced noise and/or perturbation we may end up selecting an entirely different set of features. Consequently, domain experts do no trust algorithms' results since this instability contradicts with a biological assumptions.

Boosting and Bagging are very popular ensemble techniques. Although they are similar in a sense that they both combine weak learners to aggregate a better model, they have several differences. First, Boosting performs data sampling without replacement. Then, it performs another deterministic sampling from the first round including a certain percentage of misclassified samples. This approach leads to lower classification error. In other words, learning bias is better with Boosting while data variance increases due to overfitting. On the other hand, ***Bagging*** (**B**ootstrap **Agg**regat**ing**) samples data with replacement which leads to randomly selecting data points. Unlike Boosting, Bagging is known for the ability to reduce data variance, i.e., the overfitting [32]. In Section , we mentioned errors due to variance which can be seen as a leading cause of selection instability. Therefore, we assume that reducing variance error may help improving stability.It was noted that we need to be careful here due to the fact that a lower variance leads to a higher bias, which is another error that is not preferred. Thus, our aim is to lower the variance error to an ordinarily acceptable level. If our assumption is valid, reducing variance will markedly improve selection stability. We propose to use Bagging technique to reduce data variance, hence, improving the stability. However, improving stability should not greatly impact learning performance. In the Results Section, Bagging selection proved to be stable and reasonably accurate.

Bagging [33] aggregates selection model by vote using the same algorithm on multiple bootstrap samples of the same dataset. We withdraw samples of dataset X for l times. Different sampling techniques are possible. Yet, leave-one-out, for instance, leads to a very simple model. Hence, a very high bias will occur. In contrast, model training using each data point ends up producing an overly complex model, and therefore, a very high variance and an overfitting model. As a result, a moderated selection is better. In Section , we explain in details the selection model.

As shown in Algorithm(1), bagging algorithm starts by bootstrap sampling of the dataset, assume l samples. Then, we train the model on each bootstrap sample which produces a selected l sets of features. Where **F**' is a two dimensional $l-by-n$ matrix. The i^{th} row represents the selected feature set from the bootstrap sample $(X', y')_i$, while each i^{th} column is the value of the selected feature. This value could be the weight of that feature or simply a binary number representing whether the feature was selected. After that, we aggregate the selected feature set. Each feature's value is summed and averaged over all bagging selection matrix, $\mathbf{F}'$. We aggregate here by voting, where the most occurring feature will be assigned the highest weight. In the experiment, we will repeat this process several times in order to be able to evaluate the stability.

Data: Input dataset (X, y)
Result: 1-Selected Features $\mathsf{F}' = \{\mathcal{F}'_1, \mathcal{F}'_2, ..., \mathcal{F}'_l\}$
2-Aggregated Features $\mathsf{F} = \{\mathcal{F}_1, \mathcal{F}_2, ..., \mathcal{F}_n\}$
for $i = 1 : l$ **do**
  $(X', y')_i$ = Bootstrap Sampling of (X, y);
  $\mathcal{F}'_i = f(X', y')_i$;
end
$\mathsf{F}' = \{\mathcal{F}'_1, \mathcal{F}'_2, ..., \mathcal{F}'_l\}$
for $j = 1 : n$ **do**
  $\mathcal{F}_j = \frac{1}{l} \sum_{i=0}^{l} \mathcal{F}'_{ij}$
end
$\mathsf{F} = \{\mathcal{F}_1, \mathcal{F}_2, ..., \mathcal{F}_n\}$

Algorithm 1: Bagging Algorithm

Without loss of generality, we can apply any feature selection algorithm $f(\cdot)$ in Algorithm(1) due to our assumption that reducing data variance contributes positively to selection stability. In this work, we make use of five well-know feature selection algorithms. More details about the algorithms could be found in Section .

Experiment

In this section, we perform several experiments using a variety of selection algorithms on different publicly available medical datasets, Table 1. To ensure generalization and to make it a comprehensive experiment, the datasets varied in sample size and dimensions. Similarly, the algorithms belong to different families of feature selection algorithms. Additionally, we conducted the experiment with different cardinalities of selected feature sets $\mathcal{F}_i$. In addition, to ensure that the improve of selection stability is not due to random behavior, we used two well-known classifiers, namely: SVM and KNN, as performance evaluation methods. Finally, we used the

Jaccard Index to evaluate selection stability. The baseline to validate our proposed method is the same algorithms without bagging, since we claim that bagging improves selection stability while maintaining learning performance. In this following sub-sections, we are going to discuss our methodology and experiments in details.

Experiment Methodology

As shown in **Fig. 1**, we start by taking each dataset and apply cross-validation l times. This step generates l training datasets and l test sets. After that, bootstrap sampling is applied to each training set. This generates another l-folds of each training sample. Then, we apply feature selection algorithms on each fold to select k number of features for each fold. Without loss of generality, the number of folds l may differ between the cross-validation stage and bagging stage.

Next, we evaluate the stability of the bagging feature selection algorithm. The baseline method that we compare our method with is the normal feature selection without bagging. The final stage is performance and stability evaluation. We evaluate stability using Jaccard Index and accuracy of the selected sets using SVM and KNN. We repeat the aforementioned steps for each data set on each algorithm and each cardinality of the selected feature set from 1 to 100 selected features.

Medical Datasets

In this experiment, we used 4 microarrays listed in Table 1 which are publicly available [1]. All datasets are relatively high-dimensional with a very small sample size, $n \gg m$ [34]. These characteristics are problematic in the machine learning task. Datasets suffer from what is known as the *curse-of-dimensionality*. This problem becomes even more challenging when we have a small number of samples. In such case, the learner will not be able to converge due to these two factors. Since, the more features we have, the more data samples we need.

Table 1 shows medical datasets, DNA microarrays or biomarkers. Microarrays are gene expressions collected from tissues and cells usually to diagnose tumors [35]. They are either continuous or discrete feature values. Feature selection in gene expression dataset usually helps removing irrelevant and redundant genes and to find relevant set of genes related to a certain kind of tumor. In this paper, we used different types of data sets with different characteristics to ensure generalization of proposed method.

Table 1 Datasets statistics

	Dataset Name	#Samples n	Dimensionality m
1	BLOOD	89	2759
2	SMK-CAN	187	19993
3	Colon	62	2000
4	Leukemia	72	12582

Algorithms

As we discussed in Section , there are different families of feature selection models. In this paper we used 5 algorithms to conduct the experiment. We consider each algorithm to be a baseline for the proposed bagging approach of the same algorithm

[1]The datasets were obtained from: http://featureselection.asu.edu/

since our assumption is that bagging should reduce variance, and hence, improve selection stability. The chosen algorithms are either filter or wrapper algorithms.

Fisher Score [36], [37] is a supervised filter model that utilizes fisher criteria to evaluate features independently. This leads to selecting a subset of features that maximizes the distance between data points in different classes while minimizing distance within a given class.

ReliefF [38], [37], similarly, is a supervised and filter feature selector. It is a heuristic algorithm that utilizes k-nearest neighbor to select features in multi-classes, noisy, dependent and incomplete data. Therefore, un like the Fisher score, it can detect conditional dependencies between features [39].

Information Gain [21] measures the dependency between feature and class label. It represents the difference between the entropy of the feature and the entropy of the feature given the class label.

Chi-square score (χ^2) is similar in notion to Information Gain. It measures how independent the feature is from class label [15]. All the aforementioned algorithms are filter and supervised models. However, ℓ_1SVM represents the wrapper approach [40], [41], [42]. It finds features that maximize separation between classes using SVM and 1-norm optimization since it utilizes SVM to evaluate the separability of each feature with respect to the classes.

KNN and SVM Classifiers

Stability by itself is a highly desirable feature. However, it could be achieved by a characteristically ad-hoc algorithm that selects the same sets of features each time. It should be noted, that is not the purpose of feature selection. Indeed, we need stability, yet, stable selection should select very relevant features that are highly discriminative in their features. In other words, we are looking for an algorithm that performs well in terms of either accuracy, precision or any performance measure and performs stable selection as well.

In order to evaluate selection quality, we used two well-known accuracy measurements, namely: K-Nearest Neighbor KNN and Support Vector Machine SVM. KNN classifies data points by majority voting out of K nearest neighbors using a distance measure [43]. These distance measures could be Euclidean, Manhattan, Hamming Distance and so forth. On the other hand SVM creates separating hyperplanes between classes. It finds the optimal plane by maximizing the margin between classes [44].

Model Selection

Model selection in its simplest form is choosing the parameters that perform the best on test data. It is still an open problem [17]. In this experiment, **Fig. 1**, we have several models to be selected. To ensure the reliability of our proposed method, we needed to perform model learning several times. Also, we needed to randomly sample test sets of data points each time. l-fold Cross-Validation is proven to be effective in such tasks [45], [17], [46], [47]. Another selection to be chosen is the bootstrap sampling. We generally chose $l = 10$ for both models since we had a small sample size in the original datasets X.

The last model to be selected is the number of selected features k. To the best of our knowledge, there is no rule of thumb of choosing an optimal number of features.

Thus, practitioners consult domain experts in this open problem. In In [48], the authors tried to determine the proper k using false nearest neighbor from chaos theory for clustering. In this experiment, we chose to examine the behavior of the algorithms' stability and accuracy over different cardinalities where k ranges from 1 to 100. In other words, we performed the experiment 100 times for each algorithm on each dataset, and selecting one feature more each time. This number was reasonable in earlier studies, see [3], [49]. The reason behind this model selection is the believe that the number of genes related to a certain disease is small. Also, this range of k shows us the erratic behavior of the selection stability, if any. We will explain in details about the effect of that in the result and discussion section.

Results and Discussion

The conducted experiments showed the effectiveness of our proposed method in both stability and classification accuracy. The figures (**Fig. 2 to Fig. 6**) illustrate the Jaccard stability of each algorithm on each microarray dataset using range of cardinalities of $\mathcal{F}$. As we can see in the figures, the proposed ensemble bagging technique improves the stability on each single case. The improvement mostly exceeded 20%. In some cases, the improvement exceeds 50% **Fig. 3** on **Blood** and **SMK-CAN** datasets. These results means a lot in the medical domain since algorithms produce robust set of genes each time with respect to perturbation in data samples which is equivalent in this context to diagnosing.

The results show that a very small k is very sensitive. Yet, it becomes more and more steady when k gets greater. Also, this pattern is consistent in all cases. Therefore, machine learning practitioners may need to consider this when they intend to build a learning model.

On the other hand, learning performance, i.e. classification accuracy, improves as well. We evaluate the average learning performance over the same range of k. Then, we plotted the results using two well known and highly powerful classification algorithms, namely: KNN and SVM. Figures (**Fig. 7 to Fig. 11**) illustrate the results of the baseline method and the proposed bagging ensemble method.

It is good enough to improve stability while maintaining accuracy. However, the proposed method improved both stability and accuracy. In almost all cases the proposed method beats the maximum quartile among the baseline. Yet, there are few cases where the accuracy is maintained, i.e. in the range of the baseline accuracy. **Fiq. 8** with Leukemia and Colon datasets are good examples of such cases.

Ultimately, the proposed bagging algorithm is experimentally proven to be very effective with medical data in selecting relevant genes and maintaining, or even improving, classification accuracy. We believe that bagging approach is a very stable in terms of feature selection due to the intrinsic power of reducing learning variance. As consequence, the influence of random sampling would be mitigated.

Conclusion and Future Work

In conclusion, this paper tackles the problem of feature selection instability. One important factor that impacts selection stability is data variance. Therefore, we assumed that Bootstrap Aggregating (Bagging) might mitigate the variance influence during the selection process. The problem of instability becomes even more

critical in medical gene selection, where scientists believe that a set of genes related to a certain disease is robust regardless of the patient and it is almost always a very limited number of genes. Our proposed method improved selection stability and maintained (or, in fact, in most cases improved) classification performance on different microarrays data sets and with different well-known feature selection algorithms and classification accuracy methods.

A further investigation and improvement is possible. Different models may be selected. In addition, this method might be tested on unsupervised learning, e.g. clustering. Also, different aggregation mechanisms could be experimented.

Figures

Figure 1 Experiment Methodology Diagram

1 List of Abbreviation

CT-Scans: Computerized Tomography Scan
PCA: Principal Component Analysis
SVM: Support Vector Machine
SVM-RFE: Support Vector Machine Recursive Feature Elimination
Bagging: Bootstrap Aggregating

2 Declaration

2.1 Availability of Data and Materials

All datasets used in this paper are available in ASU feature selection dataset repository: http://featureselection.asu.edu/.

2.2 Competing Interests

Not applicable.

2.3 Funding

This research was supported in part by Deanship of Scientific Research at King Khalid University research grant number (R.G.P2/100/41).

2.4 Authors' contributions

Since it is only one author, the author contributed 100% to this paper.

2.5 Acknowledgements

This work would not have been possible without the financial support King Khalid University. I would like to express my deepest gratitude to their generous support.

Figure 2 Jaccard Stability of Fisher Score.

Figure 3 Jaccard Stability of Chi Square.

References

1. Dy, J.G., Brodley, C.E.: Feature selection for unsupervised learning. J. Mach. Learn. Res. **5**, 845–889 (2004)
2. Tang, J., Alelyani, S., Liu, H.: Feature selection for classification: A review. Data Classification: Algorithms and Applications, 37 (2014)
3. Alelyani, S., Tang, J., Liu, H.: Feature selection for clustering: A review. Data Clustering: Algorithms and Applications **29**, 110–121 (2013)
4. Yu, L., Liu, H.: Feature selection for high-dimensional data: A fast correlation-based filter solution. In: Fawcett, T., Mishra, N. (eds.) Proceedings of the 20th International Conference on Machine Learning (ICML-03),, pp. 856–863. Morgan Kaufmann, Washington, D.C. (2003)
5. Leung, Y.Y., Chang, C.Q., Hung, Y.S., Fung, P.C.W.: Gene selection for brain cancer classification. Conf Proc IEEE Eng Med Biol Soc **1**, 5846–5849 (2006). doi:10.1109/IEMBS.2006.260197
6. Alelyani, S., Liu, H.: Supervised low rank matrix approximation for stable feature selection. In: Machine Learning and Applications (ICMLA), 2012 11th International Conference On, vol. 1, pp. 324–329 (2012). IEEE
7. Zhang, M., Zhang, L., Zou, J., Yao, C., Xiao, H., Liu, Q., Wang, J., Wang, D., Wang, C., Guo, Z.: Evaluating reproducibility of differential expression discoveries in microarray studies by considering correlated molecular changes. Bioinformatics **25**(13), 1662–1668 (2009). doi:10.1093/bioinformatics/btp295
8. Saeys, Y., Inza, I., Larraaga, P.: A review of feature selection techniques in bioinformatics. Bioinformatics **23**(19), 2507–2517 (2007). doi:10.1093/bioinformatics/btm344
9. Han, C., Tao, X., Duan, Y., Liu, X., Lu, J.: A cnn based framework for stable image feature selection. In: Signal and Information Processing (GlobalSIP), 2017 IEEE Global Conference On, pp. 1402–1406 (2017). IEEE
10. Boulesteix, A.-L., Slawski, M.: Stability and aggregation of ranked gene lists. Brief Bioinform **10**(5), 556–568 (2009). doi:10.1093/bib/bbp034. http://bib.oxfordjournals.org/cgi/reprint/10/5/556.pdf
11. Kuncheva, L.I.: A stability index for feature selection. In: Proceedings of the 25th Conference on Proceedings of the 25th IASTED International Multi-Conference: Artificial Intelligence and Applications, pp. 390–395. ACTA Press, Anaheim, CA, USA (2007). http://portal.acm.org/citation.cfm?id=1295303.1295370
12. Jurman, G., Merler, S., Barla, A., Paoli, S., Galea, A., Furlanello, C.: Algebraic stability indicators for ranked lists in molecular profiling. Bioinformatics **24**(2), 258–264 (2008). doi:10.1093/bioinformatics/btm550
13. Kalousis, A., Prados, J., Hilario, M.: Stability of feature selection algorithms: a study on high-dimensional spaces. Knowledge and Information Systems **12**(1), 95–116 (2007)
14. Alelyani, S.: On feature selection stability: A data perspective. PhD thesis, Arizona State University (2013)
15. Liu, H., Setiono, R.: Chi2: Feature selection and discretization of numeric attributes. In: Vassilopoulos, J.F. (ed.) Proceedings of the Seventh IEEE International Conference on Tools with Artificial Intelligence, November 5-8, 1995, pp. 388–391. IEEE Computer Society, Herndon, Virginia (1995)
16. Das, S.: Filters, wrappers and a boosting-based hybrid for feature selection. In: ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning, pp. 74–81. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2001). http://www.cs.rpi.edu/~sanmay/papers/icml01.pdf
17. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. Journal of Machine Learning Research **3**, 1157–1182 (2003)
18. Guyon, I., Elisseeff, A.: An introduction to feature extraction. Feature extraction, 1–25 (2006)
19. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. science **290**(5500), 2323–2326 (2000)
20. Song, L., Smola, A., Gretton, A., Borgwardt, K., Bedo, J.: Supervised feature selection via dependence estimation. In: International Conference on Machine Learning (2007)
21. Cover, T.M., Thomas, J.A.: Elements of Information Theory. Wiley, ??? (1991)
22. Meier, L., Van De Geer, S., Bühlmann, P.: The group lasso for logistic regression. Journal of the Royal Statistical Society: Series B (Statistical Methodology) **70**(1), 53–71 (2008)
23. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. Machine learning **46**(1), 389–422 (2002)
24. He, Z., Yu, W.: Stable Feature Selection for Biomarker Discovery (2010). http://www.citebase.org/abstract?id=oai:arXiv.org:1001.0887
25. Abeel, T., Helleputte, T., de Peer, Y.V., Dupont, P., Saeys, Y.: Robust biomarker identification for cancer diagnosis with ensemble feature selection methods. Bioinformatics **26**(3), 392–398 (2010). doi:10.1093/bioinformatics/btp630
26. Saeys, Y., Abeel, T., Van de Peer, Y.: Robust Feature Selection Using Ensemble Feature Selection Techniques., 2008. http://dx.doi.org/10.1007/978-3-540-87481-2$_2$1
27. Kalousis, A., Prados, J., Hilario, M.: Stability of feature selection algorithms, p. 8 (2005). doi:10.1109/ICDM.2005.135. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=arnumber=1565682
28. Gulgezen, G., Cataltepe, Z., Yu, L.: Stable and accurate feature selection. In: ECML/PKDD (1), pp. 455–468 (2009). http://www.springerlink.com/content/q752v051wgj141g3/fulltext.pdf
29. Yu, L., Ding, C., Loscalzo, S.: Stable feature selection via dense feature groups. In: KDD '08: Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 803–811. ACM,

Figure 4 Jaccard Stability of Information Gain.

Figure 5 Jaccard Stability of ReliefF.

New York, NY, USA (2008). doi:10.1145/1401890.1401986
30. Nogueira, S., Brown, G.: Measuring the stability of feature selection. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 442–457 (2016). Springer
31. Chelvan, P.M., Perumal, K.: A comparative analysis of feature selection stability measures. In: Trends in Electronics and Informatics (ICEI), 2017 International Conference On, pp. 124–128 (2017). IEEE
32. Breiman, L.: Bias, variance, and arcing classifiers (1996)
33. Breiman, L.: Bagging predictors. Machine learning **24**(2), 123–140 (1996)
34. Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R.P., Tang, J., Liu, H.: Feature selection: A data perspective. arXiv preprint arXiv:1601.07996 (2017)
35. Bolon-Canedo, V., Sanchez-Marono, N., Alonso-Betanzos, A., Benitez, J.M., Herrera, F.: A review of microarray datasets and applied feature selection methods. Information Sciences **282**, 111–135 (2014)
36. Gu, Q., Li, Z., Han, J.: Generalized fisher score for feature selection. arXiv preprint arXiv:1202.3725 (2012)
37. Zhao, Z., Morstatter, F., Sharma, S., Alelyani, S., Anand, A., Liu, H.: Advancing feature selection research. ASU feature selection repository, 1–28 (2010)
38. Kononenko, I.: Estimating attributes: analysis and extensions of relief. In: European Conference on Machine Learning, pp. 171–182 (1994). Springer
39. Sikonja, M.R., Kononenko, I.: Theoretical and empirical analysis of relief and relieff. Machine Learning **53**, 23–69 (2003)
40. Bi, J., Bennett, K., Embrechts, M., Breneman, C., Song, M.: Dimensionality reduction via sparse support vector machines. Journal of Machine Learning Research **3**(Mar), 1229–1243 (2003)
41. Bradley, P.S., Mangasarian, O.L.: Feature selection via concave minimization and support vector machines. In: Machine Learning Proceedings of the Fifteenth International Conference(ICML 98, pp. 82–90. Morgan Kaufmann, ??? (1998)
42. Joachims, T., Informatik, F., Informatik, F., Informatik, F., Informatik, F., Viii, L.: Text Categorization with Support Vector Machines: Learning with Many Relevant Features (1997). http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.11.6124rep=rep1type=pdf
43. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann Pub, ??? (2005)
44. Suykens, J., Vandewalle, J.: Least squares support vector machine classifiers. Neural Processing Letters **9**(3), 293–300 (1999)
45. Kohavi, R., *et al.*: A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Ijcai, vol. 14, pp. 1137–1145 (1995). Stanford, CA
46. John, G.H., Kohavi, R., Pfleger, K.: Irrelevant features and the subset selection problem, pp. 121–129. Morgan Kaufmann, ??? (1994). http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.30.3875rep=rep1type=pdf
47. Ng, A.Y.: On feature selection: Learning with exponentially many irrelevant features as training examples. In: Proceedings of the Fifteenth International Conference on Machine Learning, pp. 404–412. Morgan Kaufmann, ??? (1998). http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.48.5103rep=rep1type=pdf
48. Andrade Filho, J.A., Carvalho, A.C., Mello, R.F., Alelyani, S., Liu, H.: Quantifying features using false nearest neighbors: An unsupervised approach. In: Tools with Artificial Intelligence (ICTAI), 2011 23rd IEEE International Conference On, pp. 994–997 (2011). IEEE
49. Alelyani, S., Liu, H., Wang, L.: The effect of the characteristics of the dataset on the selection stability. In: Tools with Artificial Intelligence (ICTAI), 2011 23rd IEEE International Conference On, pp. 970–977 (2011). IEEE

Figure 6 Jaccard Stability of ℓ_1SVM.

Figure 7 Accuracy evaluation of Fisher Score using KNN and SVM. Green Line represents Bagging while Black Lines, Red Line and Blue Box represent maximum, minimum, median and quartile of baseline performance respectively.

Figure 8 Accuracy evaluation of Chi Square using KNN and SVM. Green Line represents Bagging while Black Lines, Red Line and Blue Box represent maximum, minimum, median and quartile of baseline performance respectively.

Figure 9 Accuracy evaluation of Information Gain using KNN and SVM. Green Line represents Bagging while Black Lines, Red Line and Blue Box represent maximum, minimum, median and quartile of baseline performance respectively.

Figure 10 Accuracy evaluation of ReliefF using KNN and SVM. Green Line represents Bagging while Black Lines, Red Line and Blue Box represent maximum, minimum, median and quartile of baseline performance respectively.

Figure 11 Accuracy evaluation of ℓ_1SVM using KNN and SVM. Green Line represents Bagging while Black Lines, Red Line and Blue Box represent maximum, minimum, median and quartile of baseline performance respectively.