Filter Feature Selection Performance Comparison in High-dimensional Data

A theoretical and empirical analysis of most popular algorithms

Carlos Huertas
Department of Computer Science
Autonomous University of Baja California
Tijuana, B.C., Mexico
chuertas@uabc.edu.mx

Reyes Juárez-Ramírez
Department of Computer Science
Autonomous University of Baja California
Tijuana, B.C., Mexico
reyesjua@uabc.edu.mx

***Abstract*— The key idea behind feature selection is to find a subset of features that produce similar or better results as the original set while being more compact. Algorithms in this topic can be grouped in filter, wrapper and hybrid, however for very high dimensional data it has been found that the filter approach is better due to being less computational expensive. In this paper we provide a study about how information explosion has caused an impact on solutions for feature selection. A theoretical analysis is reviewed followed by an empirical comparison of 10 of the most popular filter algorithms with datasets ranging from 2400 up to 100,000 features in order to observe algorithm performance, scalability and detect current open problems. Results suggest that some of the current most popular solutions may become obsolete in the future due to the increase in dataset complexity.**

Keywords—feature selection; filter; algorithm; high dimensional data.

I. INTRODUCTION

Feature Selection dates back since 1960 [1] and is one of the areas in machine learning that has received considerable attention by several researches, at some point there was doubts about the validity of this area [2] nonetheless, the research continued and it had an important boom starting in 1997 with special issues on its relevance [3, 4], however back then only a very minimal set of domains had more than 40 features [5]. The improved technology of recent era has made available measurements with very high resolution, hence there has been an exponential increase in data up to the point it has become unmanageable even for current algorithms as is the case with microarray research [6] where the data is of very high dimension and contains a lot of noise.

In order to determine which part of the information is really useful for the problem domain, different techniques has been proposed such as: automatic pattern recognition & knowledge discovery and data mining; however when the data dimension is too high, it is required to perform a pre-processing step to reduce such complexity and these approaches are known as *Dimensionality Reduction*, the two main techniques can be categorized as *Feature Extraction* and *Feature Selection*. The Feature Extraction approach seeks to transform the high dimensional features into a whole new space of lower dimensionality, an example of these techniques could be Principal Component Analysis (PCA) [7]. In this paper we focus on the other approach which is Feature Selection, and it consist in reducing the dimensionality of the data by removing features that are noisy, redundant or irrelevant for a classifier, examples of most popular algorithms for feature selection will be discussed in section 3. Both approaches: Feature Extraction and Feature Selection, aims to have the smallest number of features [8] as experimentation has proved that a subset of features may work better than the entire set [9], hence reducing computation resources such as memory and processing, however in our opinion the Feature Selection approach is better as it keeps the original values after reduction and there is a perfect relation with the original data, as opposed to Feature Extraction where a transformation occurs and resultant data cannot be directly linked to the source.

Since 1980 we can see the development a lot a different algorithms for feature selection that have been successful in different areas such as: Text categorization [10], pattern recognition [11], image processing [12], bioinformatics [13], etc. As the number of algorithms increases, it becomes more difficult to select the right one for a given application, as researches suggests there is no such thing as best algorithm and the results depends on the characteristics and size of the data itself [14], another complexity to the problem is the called curse of dimensionality [15] which talks about a phenomena that occurs when analyzing data in high-dimensional spaces that does not occur in low-dimensional scenarios, hence the evaluation of large feature sets becomes unmanageable for some algorithms, making the removal of redundant and not relevant features a complex task. The key idea however remains the same for all those algorithms: "*try to keep the most relevant features and remove the rest*". There are several proposals to define what a relevant feature is, however in this research we take the definition of John & Kohavi [16] which defines two different kind of relevance:

- **Strong Relevance:** Happens when the removal of a given feature Fi cause a performance drop of the Bayes Optimum Classifier.

- **Weak Relevance:** Occurs when, for a given subset of features F the performance of F with F_i is better than just F.

Any feature that is not relevant is therefore irrelevant and could be divided under two main groups:

- **Redundant Features:** those that does not provide any unique information about the class and therefore can be substituted by another feature.
- **Noisy Features:** includes the features that are not redundant but, does not provide information about the class neither.

Among the factors that affects a feature selection performance we can find at least 4 groups [3]:

- **Search Direction:** we basically have "forward" where the features are being added as relevant, "backward" where features are being removed as irrelevant, and a combination of both to avoid local minima.
- **Search Strategy:** These are based on corresponding heuristics for each algorithm, one example could be Greedy search.
- **Evaluation Criterion:** Proper of the proposed solution, this is how the features are selected, examples could be Chi2 or Information Gain.
- **Stopping Criterion:** This basically determines if the algorithm will just stop after a given number of iterations or will hold until a given threshold could be reached, this of course has a direct impact on the feature set size, where the optimal size may be defined by the number of training instances [17].

When designing a feature selection algorithm there is at least four challenges to address [18]

- **The Curse of Dimensionality:** probably the most important reason that feature selection exists, and this makes reference to a very large number of features, as in microarray where we can have over 20,000 features but only a few can be considered relevant for learning, trying to train a classifier with such large data would not give encouraging results, even the feature selection algorithms have scalability and efficiency problems with such big data.
- **Small sample size:** Usually the data that has so many features is very hard to obtain, therefore the number of samples is often limited, sometime as small as 100 examples with thousand of features each. This relation of feature size and samples cause feature selection to be a non-deterministic polynomial-time hard problem (NP-hard). In order to be able to generalize for unseen data the sample size must be of a suitable size.
- **Data noise:** It is just data that does not really help in the generalization problem, irrelevant and redundant features can be tagged as noise as well. Even the device or methodology to gather the data could increase the problem
- **Selected Features:** Once a feature selection algorithm ranked or selected a subset of features, the following question arise, *how many of those best features select?*, the decision of how many to take has a direct impact in classifier performance.

The rest of the paper is structured as follows; in Section 2 we present an overview of different categories in feature selection solutions. Section 3 presents a theoretical review of algorithms and groups them according to their main characteristics. In Section 4, the empirical comparison is performed using multiple datasets and classifiers. In Section 5 we provide our conclusions and future work.

II. FEATURE SELECTION

Feature selection algorithms can be divided into two major groups: Supervised and Unsupervised. These groups can be further divided in minor groups: Filter, Wrapper and Hybrid as discussed in introduction. In the following section we will review the details between each approach based on the research from Alelyani [18].

A. Supervised Feature Selection

Some datasets are being built to specific label setup, for instance if the task is face recognition a dataset will be built with photos of faces. In most cases as part as the machine learning process, a group of experts will label the data. The labels later became the hypothesis to determine a class vs. the other and this information becomes very important to guide feature selection. The relation between the label and the class makes possible to study the statistical relation and characteristics of the samples of the same class [5,19]. As mentioned before we have three main minor groups which are:

Supervised Filter Model: These algorithms shine by the fact that are completely independent of any classifier, hence the final selection depends only on the characteristics of the data itself and its relation to the class label, e.g. we have Fisher Score[20] to evaluate each feature independently by fisher criterion. One of the most popular approaches are based from Lasso and recent research has shown significant success with them [21]. As a general review for Filter models we can say they are overall very efficient, scalable and their results are usually portable as they do not depend on external components such a guiding classifier, these advantages promotes that most of the proposed methods belong to this model, as a downside this methods may not be as accurate as other models, as explained in next section with Wrapper methods.

Supervised Wrapper Model: One key difference with filter models is that wrappers uses a classifier to evaluate its performance [16, 4], the initial feature selection is usually achieved by a greedy search strategy, the classifier performance is evaluated with the selected features and if the result is satisfactory the search stops, however in the case the results are not the expected, the whole process is repeated again but this time with a different subset. It is clear how expensive and time consuming this approach could be, however it is not hard to infer that the results of these kind of algorithms are usually better than the filter approach.

Supervised Hybrid Model: As we have seen the filter models are more efficient, while the wrapper models are usually more accurate, in order to achieve a balance, the hybrid model is proposed to fill the gap between those approaches [22]. In the search step they employ a filter selection to reduce the number of candidates, later a classifier is used to select the subset that provided the best accuracy. Results produced by these kind of algorithms are usually more accurate than filters at the cost of more computation.

B. *Unsupervised Feature Selection*

There is a big amount of domains where manually getting or labeling the data is not a feasible option, therefore feature selection becomes a very complicated task, without the labels there is not a direct relationship between the data itself and the class it belongs. Different methods have been proposed to handle the problem, a common approach implies automatic label generation that later guide the algorithm in a similar way as supervised feature selection.

The scope of this paper focus on supervised feature selection, but for further review of unsupervised approach, one of the popular methods is Spectral Feature Selection (SPEC) [23].

For low-dimensional data, it is possible to employ different techniques, even the wrapper approach could be used, but once the dimension grows at a considerable scale of thousand of features, the computation complexity becomes a problem and the filter approach being the more efficient becomes most of the time the only feasible option; however, information explosion has caused that even filter approaches start to become unfeasible and such condition is investigated in this work. In the following section we provide a theoretical review of some of the most widely used supervised feature selection filter algorithms.

III. THEORETICAL REVIEW

In order to provide an overall overview of the most popular algorithms currently employed in feature selection we based our search on the Advancing Feature Selection Research Repository built at the Arizona State University by Zhao et. al [24] and Weka[25] project, each of the algorithm is reviewed individually and later a general comparison review is provided.

A. *CFS*

The Correlation-based feature selection (CFS) uses heuristics to evaluate the worth of merit of subsets of features, such heuristics take in consideration how useful is a given feature to classify a class. In order to be able to perform the selection, CFS requires all features to be of the same type, hence a discretization pre-processing step is required [26].

The core of this algorithm is based on Pearson's correlation coefficient, which is a measure of linear dependency (correlation) between two variables *X* and *Y*, developed by Karl Pearson based on the idea of Francis Galton who discovered it in 1888 [27].

B. *Chi2*

The Chi Square (Chi2) algorithm is based on Kerbers ChiMerge [28] which is used to discretize numeric attributes based on the X2 statistic. The Chi2 approach aims to solve ChiMerge limitation of finding the optimal significance level automatically, the way this is accomplished is by wrapping ChiMerge in a loop that automatically play with different X2 thresholds. A consistency checking is used as the stopping criteria and thus completing the automatic parameter selection which represents Chi2 phase 1.

The phase 2 starts with the results obtained in phase 1, each attribute is associated with a significance level and the merging step begins, if the consistency rate is not archived for the *i-attribute,* the parameters are tuned and the attributes wait for a next merging round. The phase 2 continues until no more attributes can be merged [29].

C. *FCBF*

The Fast Correlation-Based Filter (FCBF) solution tries to find the best feature subset based on goodness of features. In general to FCBF a good feature is one that is relevant to the class concept and not redundant to any other features, hence a correlation between variables is used to determine feature goodness [30]. The two main approaches to measure correlation are based on classical linear correlation and the other is based on information theory, however it is not safe to assume that in real world data there will be always a linear correlation between features, to overcome that problem FCBF uses an approach based on the information theory concept of entropy [30].

D. *Fisher*

The Fisher Score is a univariate algorithm that selects features that assign similar values to the sample of the same class and different values to samples from different classes [31]. The evaluation criteria according to Xe et al. [32] is a special case of Laplacian Score, however since Fisher Score evaluates each feature individually it is unable to handle any redundancy between them.

E. *Gini*

The Gini Index [33] is a univariate algorithm that quantifies how easily a feature alone can distinguish between classes, the smaller the value the more related a feature is related to a class, so in this algorithm the top *k-features* with the smallest values are selected. As with other feature weighting algorithms, the *k* number of features to be selected needs to be set manually. One particular downside of this algorithm is the inability to deal with redundancy.

F. *InfoG*

The Information Gain (InfoG) is one of the most popular techniques as it is very easy to interpret and compute. InfoG measures the reduction in the entropy of X that is caused after observing a random variable Y [34]. Since this evaluation is univarate it does not deal with redundancy.

G. KW

The Kruskal-Wallis algorithm is non-parametric, therefore no assumption about the distribution of the data is being done[35]. The measures are done by comparing the population medians among groups, to do that a ranking initial step is required to merge all groups in a common scale. As with other univariate algorithms, no redundancy reduction is possible.

H. mRmR

The Minimum-Redundancy Maximum-Relevance (mRmR) algorithm is based on mutual information. Given two random variables X and Y, their mutual information is defined in terms of their probabilistic density functions. In Max-Relevance, the idea is to select features X_i with the largest mutual information over the target class C, however it has been recognized that the combination of individual good features do not necessary lead to good classification performance [36]; mRmR deals with this problem incorporating the Minimum-Redundancy factor, where features that have been already selected as relevant, are evaluated to find out the dependency between them, if they are found to be highly dependent the less relevant feature is eliminated. Since mRmR is a feature ranking algorithm, it does not need to select a subset of features but instead the user is required to choose among the best ranked features.

I. RelieF

The original Relief algorithm was proposed by Kira and Rendell [37], which was known for being very efficient, the main idea behind Relief is to estimate features according to how well their values distinguish among instances that are near each other, such instances are called neighbors. Relief looks for the two nearest instances, one of the same class labeled as *nearest-hit* and one of the other class labeled *nearest-miss*. The rationale behind the algorithm is that good features should have the same values for instances of the same class and different values for a different class. For discrete features the differences is either 1 (they are different) or 0 (they are equal) while for continuous data, it is the actual difference normalized to the [0-1] interval.

Relief-F is an extension to the original Relief where, instead of finding one *near-miss M* from a different class, the algorithm finds one *near-miss M(C)* for each class and averaged their contribution, the results are averaged with the prior probability of each class. Besides, to deal with the original Relief noise problems, Relief-F uses a user configured *k-value* for the number of neighbors to search.

J. t-T

The t-test score is usually employed for binary classification problems [38], and comes particularly useful for unequal sample size and unequal variance. The t-test algorithm ranks features according to its capability to separate classes, ranking occurs with single features hence no redundancy reduction is possible.

From the reviewed algorithms we can find some key differences and similitude, as a summary we can say that all seeks for the same idea, which is: "*classes of the same type looks similar, while different classes should look different*". The main difference would be the way each algorithm computes such similarity. We can distinguish too different ways of presenting results, some algorithms only ranks features (feature weighting) and later a further selection needs to be done to select subsets from the ranking result, while other algorithms are capable of completely remove features (feature set) leaving a dataset with only the features that help discriminate the data. In Table 1 we show the algorithms in terms of their output type.

TABLE I. ALGORITHMS OUTPUT TYPE

Feature Weighting	Feature Subset
Chi Square	CFS
Fisher Score	FCBF
Gini Index	
Information Gain	
Kruskal-Wallis	
mRmR	
Relief-F	
t-Test	

From Table 1, we can notice that there is a very disproportion between feature subset and feature weighting algorithms, one potential reason for this is that subset algorithms are more complex to design but have the advantage that no necessary additional step is required as with weighting that a very computational intensive step needs to be performed to select the right subset of features.

Another key element to classify algorithms would be the ability to evaluate combinations of multiple features, known as multivariate, grouping is shown in Table 2.

TABLE II. ALGORITHMS VARIABLE HANDLING

Univariate	Multivariate
Chi Square	CFS
Fisher Score	FCBF
Gini Index	mRmR
Information Gain	
Kruskal-Wallis	
Relief-F	
t-Test	

From Table 2 we can see the same pattern, as multivariate design is more complex, the vast majority of algorithms fall in

the univariate category. In theory, multivariate design should produce more compact results as they handle redundancy.

In the following section we present the experiments carried out to determine which algorithm performs the best in terms of efficacy and efficiency as the dimensionality increases.

IV. Empirical Review

In this section we present a study to compare the 10 algorithms reviewed on section 3 to evaluate how their performance behave as the size of features increases, the grow in information is a very important aspect to provide further information about the algorithms since most of them have been presented with very low-dimensional data, e.g. the FCBF algorithm proposed by Lei Yu [30] shows a remarkable comparison between FCBF, CorrSF, Relief-F and ConsSF algorithms evaluated against 10 different datasets, however their mean dimensionality size was 220, having datasets as small as 57 features and being the largest only 650 features. The grow in information allow us to rise the complexity using datasets more than 150 times larger such as microarray and biological datasets. Dataset properties are reviewed below.

A. Datasets

We have selected 6 different high-dimensional datasets from the ASU Feature Selection Repository [24] and UCI Machine Learning Repository [39], focusing on datasets which presents more challenges to classifiers having a very large *feature-to-instance* ratio, such as microarray samples. In order to have comparable execution times, we have setup a fixed number of instances (60) to utilize in feature selection step. In Table 3 we present the dataset main characteristics.

TABLE III. Datasets Characteristics

Dataset	Type	Features	Instances	Classes
Arp	Image	2,400	130	10
Tox	MicroA	5,748	171	4
Cll-sub	MicroA	11,340	111	3
Smk-can	MicroA	19,993	187	2
Gla-bra	MicroA	49,151	180	4
Dorothea	Bio	100,000	800	2

B. Experiment Setup

To test each algorithm we utilize the ASU Feature Selection Algorithm Repository [24] which in turn uses implementations from Weka project [40], besides we have included a basic random feature selection algorithm to have a reference point in datasets that are too complex to process and using the full set of features is not feasible.

Since we are going to test algorithms that produce different type of results as shown in Table 2, we have the following two cases:

Feature Weight: these algorithms rank the features instead of returning useful features. In order to have meaningful results we run an iterative process to find a suitable subset of features. We start with the 10 best ranked features, and constantly increment features by 10 up to reaching 400.

Feature Set: these algorithms provide ready to use results, so we proceed to test the quality of result with the returned sub-set of features.

In order to evaluate the quality of the algorithms, we evaluate the classifiers accuracy obtained after the feature selection has been performed. For each dataset we randomly selected a fixed number of instances (60) for training in order to keep grow in the number of features only. The whole process is repeated 10 times using different parts of dataset each iteration. The classifiers utilized for comparison are SVM, J48 and Naives Bayes, the results of all classifiers are averaged including accuracy and variance.

Analysis in execution time grow rate is provided as well as comparison on the effectiveness to reduce the dimension of the dataset which has a good impact on eliminating redundancy

C. Experiments Results

In Table 4 and Table 5, we presents the results in terms of accuracy for each classifier (in percentage %), the reported number represents the average value from all test and under all classifiers; each value has its corresponding variance included. Values in bold indicate the algorithm surpass the accuracy obtained when the full set of features or the random sub-set selection is used. OM stands for *Out of Memory*, and represents a status where feature selection is a required step as our limit of 16GB of memory is surpassed if training the full number of features is attempted. NF stands for *No Feasible*, and means the given algorithm requires so much time that it falls out of practical usage.

TABLE IV. Accuracy Mean and Variance Part #1

Dataset	Full	Rand	CFS	Chi2	FCBF	Fisher
Arp	64.3 ±11.1	53.0 ±9.5	63.9 ±9.4	**67.7** ±8.3	60.1 ±4.9	**68.4** ±9.1
Tox	62.3 ±15.9	52.4 ±17.6	**63.3** ±8.5	60.6 ±11.1	**63.6** ±7.0	61.4 ±11.2
Cllsub	69.8 ±5.6	61.1 ±1.6	NF	67.6 ±5.2	69.8 ±10.0	62.4 ±3.5
Smkcan	OM	61.0 ±9.5	NF	**65.2** ±4.7	**61.4** ±3.6	**66.3** ±2.8
Glabra	OM	62.0 ±5.0	NF	**64.3** ±1.9	**63.2** ±6.4	**63.6** ±5.1
Doro	OM	90.7 ±0.6	NF	**92.9** ±0.7	**92.0** ±1.0	**92.9** ±0.7
Average	65.5 ±10.9	63.4 ±7.3	63.6 ±9.0	69.7 ±5.3	68.4 ±5.5	69.2 ±5.4

TABLE V. ACCURACY MEAN AND VARIANCE PART #2

Dataset	Gini	InfoG	K-W	Mrmr	RelieF	t-T
Arp	45.8 ±14.9	**68.1** ±9.6	49.2 ±9.7	54.3 ±9.7	**69.0** ±7.6	**69.3** ±8.6
Tox	55.4 ±15.9	59.7 ±11.9	60.1 ±14.7	59.9 ±12.9	**62.7** ±14.8	61.9 ±12.3
Cllsub	65.4 ±5.0	67.5 ±6.0	65.5 ±8.1	67.2 ±7.3	**73.3** ±5.6	63.8 ±3.7
Smkcan	**62.5** ±6.3	**64.4** ±5.2	**64.0** ±2.8	**61.5** ±3.3	**66.3** ±5.4	**63.3** ±4.2
Glabra	60.0 ±6.2	**63.6** ±2.2	**62.9** ±3.5	**63.1** ±4.1	**64.9** ±4.2	61.9 ±5.8
Doro	NF	**92.9** ±0.7	90.0 ±0.2	**93.1** ±0.3	NF	90.2 ±0.0
Average	57.8 ±9.7	69.4 ±5.9	65.3 ±6.5	66.5 ±6.3	67.2 ±7.5	68.4 ±5.8

From accuracy results in Table 4 and 5 we can notice the necessity of feature selection. In low-dimensional problems it can be seen as a method to improve accuracy, but in high-dimensional it serves another purpose as well, which is reduce the computation complexity of a problem that otherwise would be unfeasible to manage, we can notice this effect as it was not possible to train the classifiers using the full set of features once we get closer to the 20,000 mark.

Some of the algorithms showed their scalability weakness as the number of features dramatically increased to 100,000; for instance CFS, Gini and Relief-F were unable to complete the experiments. On the other hand algorithms such as KW and mRmR were able to complete the task but did not get much encouraging results, sometimes loosing even more than 10.0% accuracy compared with the full dataset.

Results can be divided in two groups, *full-set feasible* and *full-set not feasible*, being the first 3 data sets in the feasible group, this section is of particular interest as it can be seen that while algorithms can easy beat the random selection, when compared with the full number of features they seem to be filtering too much of information causing not only a null improvement but actually decreasing classifier accuracy, e.g. with the Arp dataset, in 50% of cases, the usage of an algorithm result in a drop in accuracy, with the Tox dataset the problem rise to 70%, up to reaching the point where in the Cllsub dataset only one algorithm managed to improve accuracy.

Notable remark is that the top performing 3 algorithms (Chi2, InfoG, Fisher) are of feature weighting type, leaving FCBF to 5th place and the only remaining feature subset algorithm (CFS) not even completing the test. However feature weighting algorithms carry a computational expensive process to select a suitable group of features.

In Figure 1 & 2, we provide a summary of execution times composed by feature select/rank time, plus in the case of feature weighting algorithms, the extra step of finding feature subset to understand more clearly the algorithm grow rate. The quality of the algorithm programming is out of the scope of this paper; however implementations are done by Weka project which is a very well known tool in machine learning. (*Note: since execution times for Chi2 and InfoG diverged less than 1% we have merged their results in C2&IG*)

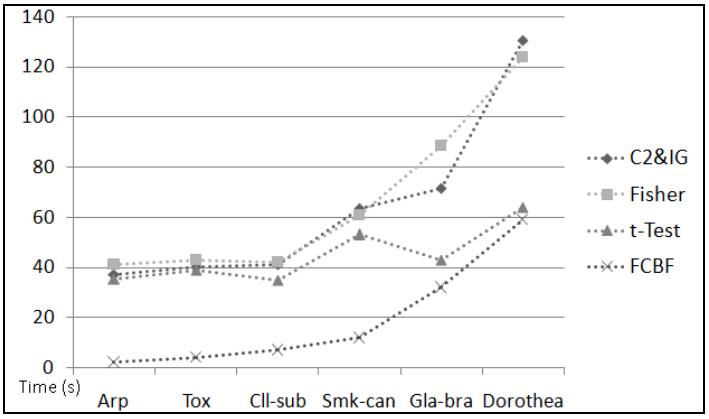

Fig 1. SCALABILITY BEHAVIOR FOR TOP 5 ALGORITHMS

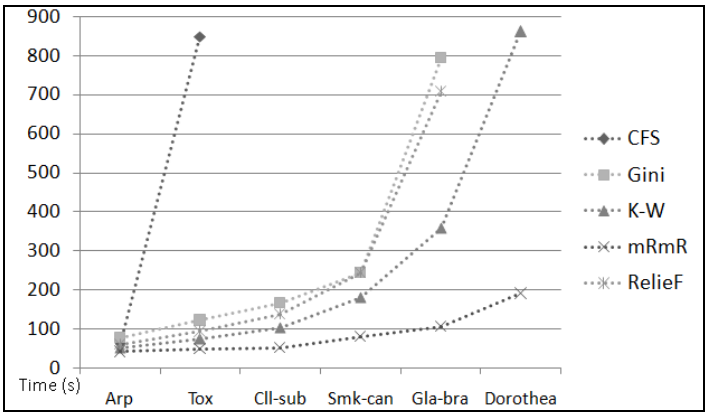

Fig 2. SCALABILITY BEHAVIOR FOR BOTTOM 5 ALGORITHMS

From Fig. 1 & 2 we can notice that in terms of performance, feature weighting have a penalty, such penalty is in function on how large we define the subset seek space, as mentioned in section IV-B we stopped at 400 features, but the larger the space the more pronounced the computation complexity. Particularly from Fig. 2 we can notice how unfeasible can become the usage of some of the algorithms as the dimension increases, for instance the K-W algorithm that in the first dataset perform just in average time, in the last test it required more than 14 times the processing power compared with the fastest algorithm overall (FCBF). Not included in figures but as a reference point training the classifiers with the full number of features took on average 9.4, 26.2 and 58.8 seconds for the first three datasets while providing very good results as shown in Table 4.

One particular problem with feature weighting is that there is no easy way to determine the appropriate feature subset space, we could have just stopped at best 200 ranked features however that would have caused a drop in accuracy, nevertheless we cannot know if a potential best accuracy

could have been achieved with any greater number of features as seeking among all possibilities is just not feasible.

To understand how the algorithms helped to reduce dimensionality we show the average number of features selected that archived best overall accuracy among the 3 classifiers (SVM, Bayes, J48) for each dataset. Results are provided in Figure 3 & 4.

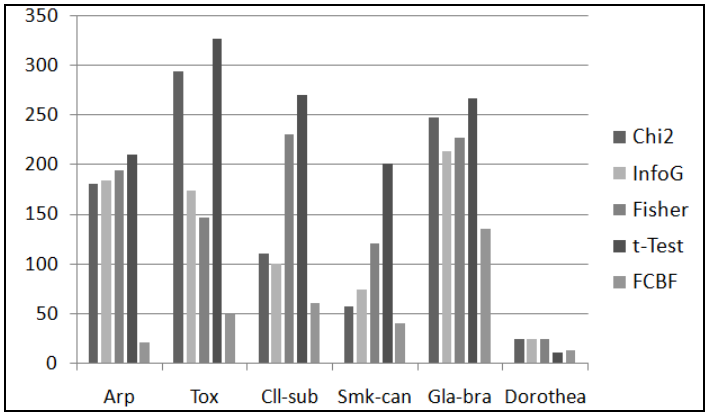

Fig 3. AVERAGE FEATURES SELECTED WITH TOP 5 ALGORITHMS

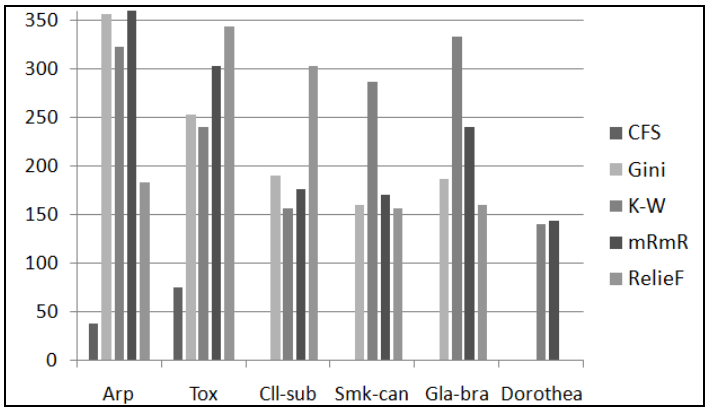

Fig 4. AVERAGE FEATURES SELECTED WITH BOTTOM 5 ALGORITHMS

After analyzing the number of selected features it is clear that the subset algorithms (CFS and FCBF) produces much more compact results and such results are portable to different classifiers. A different story applies with the feature ranking as results are notably less compact, e.g., the best overall algorithm was Chi2 which average features across all dataset was around 152 while FCBF achieved a very similar accuracy ratio with a third of such dimension at 52 features average.

V. CONCLUSIONS AND FUTURE WORK

The feature selection problem is far from solved even when there has been work on it for more than 50 years. One particular problem in this area is that the term "*high-dimensional*" is very subjective and in constant change, according to our results the fastest algorithm is FCBF, however when first introduced [30] it was tested on high-dimensional data that these days could be considered *very-low*. Information explosion has caused that the perception of high-dimension is constantly updated, in this work we have a mean of more than 30,000 features but a quick review in machine learning repositories such as UCI [39] reveal that the grow in information is advancing at a very high rate where we can now find datasets with millions of features, which makes no hard to believe that 100,000 features be considered low-dimension in a few years.

We can notice that 80% of our reviewed algorithms are of weighting type and that 70% of them perform univariate analysis, such trend in design seems to be simpler, but comes at a cost in performance as can be seen specially in Fig. 1 with the first three datasets where FCBF (a subset type) performed more than 10 times faster than the best and fastest weighting-type algorithms, however FCBF being a multivariate algorithm showed one of the poorest scalability behaviors matching times with fastest weighting-type algorithm (t-Test) in last dataset. Nevertheless Fig. 2 shows an even less encouraging results, where 3 algorithms (CFS, Gini and ReliefF) were not even able to complete the experiments, a notable mention would be CFS (the only remaining subset type) which processing time increased 14 times when features increased only at double.

It is clear that there is no such thing as the final algorithm that always performs the best, one particular algorithm could perform better on some dataset than others and vice versa, two notable examples would be Chi2, which managed to have the highest average accuracy but did not performed the best on any of the datasets, and mRmR which fall to the 6th place but in Dorothea dataset got the best result. A variety of algorithms is a good thing as it increases the possibilities to improve accuracy by trying different techniques, however for filter algorithms there is still a gap to fill as most algorithms fall in the feature weighting scheme and experimentation shows that subset provides much more compact and reusable results that can be easily utilized with different classifiers.

Our future work focus on the development of a new algorithm in the supervised filter-subset category as it is currently the best suitable option for high dimensional spaces, the current challenges we are focusing are:

Linear Computational Complexity: while FCBF proves to be fast, its time grow could make it hard to utilize in a few years, just as CFS proved to be barely usable now, we plan to overcome this issue with a restricted search in the multivariate space.

Feature Selection Stability: none of the reviewed algorithms that completed the experiments managed to improve all datasets, actually all of them achieved less accuracy in at least 2 datasets, a notable example in the top 5, the t-Test got worst results in 4 of the 6 datasets when compared with the full dataset or the random selection.

Human-in-the-loop (HITL): We are currently studying the benefits of including human feedback to enhance algorithm performance, this approach has been proposed before [41] but we have yet to see a successful development with enough stability to become popular.

REFERENCES

[1] G. F. Hughes, "On the mean accuracy of statistical pattern recognizers". IEEE Transactions on Information Theory, IT-14(1):55–63. 1968.

[2] A. J. Miller, "Selection of subsets of regression variables". Journal of the Royal Statistical Society. Series A (General), 147(3):389–425. 1984.

[3] A. Blum, and P. Langley. "Selection of relevant features and examples in machine learning". In Artificial Intelligence. 1997.

[4] R. Kohavi, and G. H. John. "Wrappers for feature subset selection". 1997.

[5] I. Guyon, and A. Elisseeff. "An introduction to variable and feature selection". In Journal of Machine Learning Research, volume 3, pages 1157–1182. 2003.

[6] X. Wang, and O. Gotoh. "Accurate molecular classification of cancer using simple rules". BMC Medical Genomics, 64(2). 2009.

[7] M. Suganthy, and P. Ramamoorthy, "Principal component analysis based feature extraction", morphological edge detection and localization for fast iris recognition. J. Comput. Sci., 8: 1428-1433. 2012.

[8] M. Dash, and H. Liu. "Feature Selection for Classification", Intelligent Data Analysis, I, pp. 131-156, 1997.

[9] A. K. Jain and B. Chandrasekaran. "Dimensionality and sample size considerations in pattern recognition practice". In P. R. Krishnaiah and L. N. Kanal, editors, Handbook of Statistics, pages 835–855. North Holland, 1982.

[10] G. Forman. "An extensive empirical study of feature selection metrics for text classification". Journal of Machine Learning Research, 3:1289–1305, 2003.

[11] P. Mitra, S. Member, C. A. Murthy, and S. K. Pal. "Unsupervised feature selection using feature similarity". IEEE Transactions on Pattern Analysis and Machine Intelligence, 24:301–312, 2002.

[12] Y. Rui and T. S. Huang. "Image retrieval: Current techniques, promising directions and open issues". Journal of Visual Communication and Image Representation, 10:39–62, 1999.

[13] Y. Saeys, I. Inza, and P. Larranaga. "A review of feature selection techniques in bioinformatics". Bioinformatics, 23(19):2507–2517, Oct 2007.

[14] P. Refaeilzadeh, L. Tang, H. Liu. "On Comparison of Feature Selection Algorithms". In AAAI Workshop on Evaluation Methods for Machine Learning II, pp. 34-39, 2007.

[15] R. Bellman, R. Dynamic programming. In Princeton University Press

[16] G. H. John, R. Kohavi, and K. Pfleger. "Irrelevant features and the subset selection problem". International Conference on Machine Learning, pages 121–129. 1994.

[17] A. Navot, R. Gilad-Bachrach, Y. Navot, and N. Tishby. "Is feature selection still necessary?" In Saunders, C., Grobelnik, M., Gunn, S. R., and Shawe-Taylor, J., editors, SLSFS, volume 3940 of Lecture Notes in Computer Science, pages 127–138. Springer. 2005.

[18] S. Alelyani, "On Feature Selection Stability: A Data Perspective", PhD Thesis, Arizona State University, May 2013

[19] Y. Y. Leung, C. Q. Chang, Y. S. Hung, and P. C. W. Fung. "Gene selection for brain cancer classification". Conf Proc IEEE Eng Med Biol Soc, 2006.

[20] Q. Gu, Z. Li, and J. Han. "Generalized fisher score for feature selection". arXiv preprint arXiv:1202.3725, 2012.

[21] J. Zhou, J. Liu, V. A. Narayan, and J. Ye. "Modeling disease progression via fused sparse group lasso". In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 1095–1103. ACM, 2012.

[22] S. Das. "Filters, wrappers and a boosting-based hybrid for feature selection". In ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning, pages 74–81, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc. 2001.

[23] Z. Zhao and H. Liu. "Spectral feature selection for supervised and unsupervised learning". In ICML '07: Proceedings of the 24th international conference on Machine learning, pages 1151–1157, New York, NY, USA, ACM. 2007.

[24] Z. Zhao, F. Morstatter, S. Sharma, S. Alelyani, A. Anand, and H. Liu. "Advancing Feature Selection Research - ASU Feature Selection Repository", TR-10-007, School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, AZ 85287, 2010

[25] I.H. Witten and E. Frank. Data Mining - Practical Machine Learning Tools and Techniques with JAVA Implementations. 2nd Edition, Morgan Kaufmann Publishers, 2005.

[26] M. A. Hall. Correlation-based feature selection for machine learning. Technical report, 1999.

[27] S. M. Stigler. "Francis Galton's Account of the Invention of Correlation". Statistical Science 4 (2): 73–79. 1989.

[28] R. Kerber. "Chimerge: Discretization of numeric attributes". In AAAI-92, Preceedings Ninth National Conference on Artificial Intelligence, pages 123-128. AAAI Press/The MIT Press, 1992.

[29] H. Liu and R. Setiono. "Chi2: Feature selection and discretization of numeric attributes". In J. Vassilopoulos, editor, Proceedings of the Seventh IEEE International Conference on Tools with Artificial Intelligence, pp. 388–391, IEEE Computer Society. 1995.

[30] L. Yu and H. Liu. "Feature selection for high-dimensional data: A fast correlation-based filter solution". In T. Fawcett and N. Mishra, editors, Proceedings of the 20th International Conference on Machine Learning (ICML-03), pp. 856–863, Washington, D.C., Morgan Kaufmann. 2003.

[31] R.O. Duda, P.E. Hart, and D.G. Stork. Pattern Classification. John Wiley & Sons, New York, 2nd edition, 2001

[32] X. He, D. Cai, and P. Niyogi. "Laplacian score for feature selection". Advances in Neural Information Processing Systems, 18:507, 2006.

[33] C. Gini. Variabilite e mutabilita. Memorie di metodologia statistica, 1912.

[34] T. M. Cover and J. A. Thomas. Elements of Information Theory. Wiley, 1991.

[35] L. J. Wei. "Asymptotic conservativeness and efficiency of kruskal-wallis test for k dependent samples". Journal of the American Statistical Association, 76(376):1006-1009, December 1981.

[36] A.K. Jain, R.P.W. Duin, and J. Mao, "Statistical Pattern Recognition: A Review," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 22, no. 1, pp. 4-37, 2000.

[37] K. Kira and L.A. Rendell. A practical approach to feature selection. In Sleeman and P. Edwards, editors, Proceedings of the Ninth International Conference on Machine Learning (ICML-92), pp. 249-256. Morgan Kaufmann, 1992.

[38] R. Montgomery and Hubele. Engineering Statistics. John Wiley & Sons, Hoboken, NJ, 2007.

[39] C. Blake, and C. Merz. (1998). UCI repository of machine learning databases. http://archive.ics.uci.edu/ml/datasets.html.

[40] I. Witten and E. Frank. Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann Pub, 2005.

[41] H. Raghavan, O. Madani and R. Jones. "InterActive Feature Selection", Proceedings of the 19th International Joint Conference on Artificial Intelligence, pp. 841-846, 2005.