Implementation of feature selection algorithm on high-dimensional datasets

1st Ewan Ross
Dept. of Computer Science
University of Nottingham
Nottingham, UK
psyer1@nottingham.ac.uk

2nd Mathews Roy
Dept. of Computer Science
University of Nottingham
Nottingham, UK
psymr3@nottingham.ac.uk

3rd Soham Talukdar
Dept. of Physics and Astronomy
University of Nottingham
Nottingham, UK
ppxst3@nottingham.ac.uk

4th Srushanth Baride
Dept. of Physics and Astronomy
University of Nottingham
Nottingham, UK
ppxsb5@nottingham.ac.uk

***Abstract*—Univariate feature selection is an efficient and effective feature selection procedure used to determine the important features in a high-dimensional dataset. In this paper, we implemented the Chi2 algorithm on a leukemia dataset to figure out how quickly it can determine the features. A Decision Tree Classifier is then implemented to understand how accurately the new subset of features help to classify unseen data into different classes.**

***Index Terms*—high-dimensionality;feature selection;**

I. Introduction

In the age of big data, machine learning and pattern recognition algorithms face some major issues when learning complex data [1]. With the rapid growth of the internet, data now comes in different formats from different sources and requires new tools and methodologies to extract information from it [2]. This rapid growth of high-dimensional data has led to the development of different techniques to reduce and remove redundant and irrelevant features [3]. The fundamental incentive of feature selection is to tackle the curse of dimensionality because high-dimensional data not only involves high memory utilization and computational power, but also deteriorates the ability to make generalizations [2]. In the case of classification, the number of features becomes important when training with a limited number of samples in order to better predict from unseen data. The number of features determines the size of the hypothesis space. A linear increase in the number of features results in an exponential increase in the size of the hypothesis space. Hence, to make a more accurate prediction, it becomes important to have a small hypothesis space [4].

Understanding a feature's relevance and redundancy is important during feature selection. A highly relevant feature refers to one that, once removed from the training data, can cause a decrease in the model's performance. A feature is said to be weakly relevant if there exists a subset of features such that the performance on the subset is worse than when combined with it, or generally has minimal effect on the model's performance. A feature becomes irrelevant when it is neither strongly nor weakly relevant and becomes completely redundant when it has a high correlation with another feature [5].

Figure 1 represents the categorical flowchart of feature selection. At the superficial level, feature selection can be based on labelled information referring to supervised, semi-supervised and unsupervised techniques [6]. The difference among these is based on the availability of labelled information. It generally becomes difficult when dealing with problems that have little to no labelled data. Labelled information allows the algorithms to differentiate relevant features for individual samples from different classes. For our work, we will focus on supervised learning. Supervised feature selection primarily has three techniques or search strategies: filter, wrapper, and embedded [7]. Filter methods usually take a two-step approach and are implemented before classification and clustering tasks. In the initial step, it evaluates all the features based on a specific set of criteria. Then, based on the evaluation, the features which score the highest are selected. Wrapper methods use learning algorithms by themselves to understand and assess the viability of the functions, whereas embedded models perform feature selection while constructing the model [7], [6]. Feature ranking, also known as univariate feature selection (Figure 2), ranks individual features. Multivariate feature selection, however, evaluates them based on a possible subset of features to distinguish between the different classes [7].

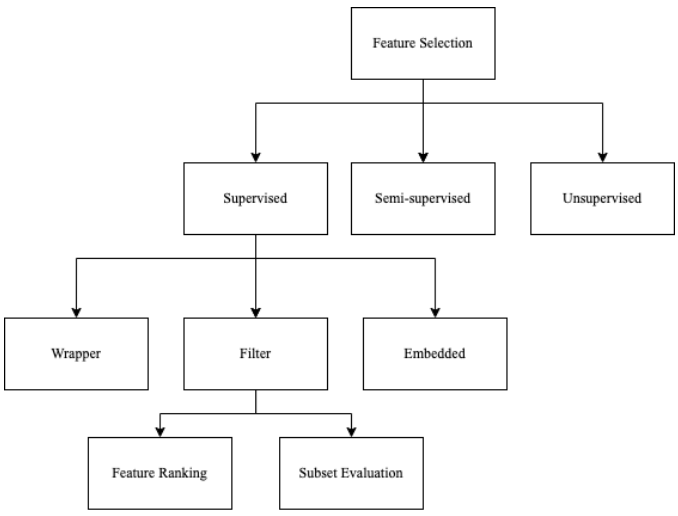

Fig. 1. Category of feature selection

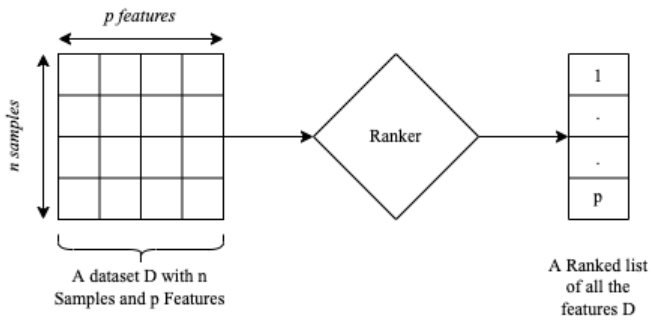

Fig. 2. Ranking of features

TABLE I
TARGET COUNT FOR EACH CLASS IN LEUKEMIA_GSE28497 DATASET

Target	Count
AML	26
Bone_Marrow	10
Bone_Marrow_CD34	8
PB	10
PBSC_CD34	10
Total	64

The primary objective of our study is to experiment an approach that can perform effective feature selection for classification with high accuracy using microarray gene data. The high throughput of biological datasets makes the sample space relatively small, with a large number of features causing the dataset to become much wider. These "wide" datasets pose a challenge in making a reliable prediction of the model's accuracy and in identifying over-fitting [4].

The remainder of this paper is presented as follows. Section II elaborates the rationale of our approach. Section III indicates the detail of our experiment. Section IV discusses the results and findings of our experiment.

II. Methodology and Setup

The last decade has seen a plethora of research being done in dealing with high-dimensional microarray data. The complication in dealing with a significant amount of features with limited sample space combined with the experimental impediments, like noise and variability render, has posed a serious challenge in the scientific community and has made the analysis of microarray data an exciting field to explore [8]. For our work, we will focus on implementing the chi-squared algorithm, a univariate statistical filter-based feature selection algorithm used for classification tasks, on a leukemia dataset [9]. Analyzing high-dimensional data requires a fast and efficient feature selection technique. In 2006, a systematic approach to analyzing microarray data first took place [10]. Since then, there have been comparative analyses of different classification and feature selection techniques but the univariate feature paradigm became the most dominant. The reason for its dominance is because it is computationally efficient as it works without the consideration of the classifiers [11]. The prevalence of this technique [12], apart from being efficient, can be explained by a couple of reasons:

- Intuitive output and easy to interpret
- Comparatively less computation time than multivariate techniques

A similar research has been done on the dataset where Chi2 is implemented [13]. PySpark's UnivariateFeatureSelector determines which selector to use based on the feature and label type. Currently, it supports 3 feature selectors based on univariate statistical tests:

- chi-squared : featureType categorical and labelType categorical
- ANOVA F-test : featureType continuous and labelType categorical
- F-value : featureType continuous and labelType continuous

One way of approaching this task is to provide a sequential solution that uses the sklearn library. The other main approaches for tackling challenges posed by big data can be constructed globally or locally. A global approach consists of a model having knowledge of the entire dataset, whereas a local approach segments the dataset and analyzes each one to understand and estimate the best solution. Due to the nature of a global approach having a holistic view of the dataset, it can take longer than a local approach when the dataset has a large number of features. A local solution might perform better on microarray data since these tend to have a significantly large number of features compared to the sample space. Post the feature selection stage, a Decision Tree Classifier is used to get an idea how good the feature selection algorithm performed in classifying the sample space.

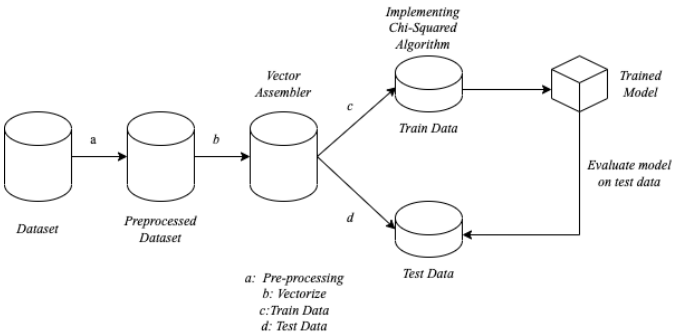

Fig. 3. Flowchart of the Implementation of the algorithm

This study is mainly based on a dataset extracted from NCBI (National Center for Biotechnology Information), consisting of 64 samples, 22,284 genes (features), and seven target classes. Individual rows indicate sample of blood monitored by the microarray. The attribute, "type", in the dataset highlights the individual patients' blood cancer type, while the remaining columns are the predictors for this. Table I gives the target count for each class in the dataset.

The approach taken is roughly described:

1) Load the data into the spark DataFrame
2) Minimize the data outliers present in the dataset using the lower and upper boundaries of individual features
3) Vectorized the dataset for the easy processing and accessing of the label and features data
4) Used ChiSqSelector for dimensionality reduction
5) Used DecisionTreeClassifier to classify the type of blood cancer

We also experimented with a sequential model using RFECV to get a comparative understanding of the time required for the execution. RFECV removes the weak and redundant features without impacting the classification outcome [14]. After building a model with all the attributes, it implements a recursive procedure to rank attributes based on their effect on classification [15]. This step takes an iterative approach where after each iteration it will remove the feature that has little to no correlation. RFECV is based upon RFE, while the latter was very much susceptible to training data which rated the attribute with high variation. The cross validation aspect of the algorithm is responsible in selecting the best set of attributes. This makes the model much more stable and reliable [16].

III. Results and Discussion

Table II depicts the number of features and the execution time of the algorithms. Table III highlights the accuracy of the model based on the features selected using our local approach of Chi2. We observed a sharp increase of almost 10% in accuracy when the dataset was split into larger chunks. The sequential model using RFECV took an abnormal amount of time in determining the set of features. To further our understanding, we then made a sequential solution using mutual information regression. Interestingly, it's execution time was a lot quicker than we had assumed. A rationale that can be setup is that mutual information regression describes not only the amount of correlation between paired genes of the given class label, but also investigates the level of dependency between two genes given the class label [17].

$$s_k = 1/(p-1) \sum_{j=1, j\neq k}^{\infty} I(x_{(k)}; x_{(j)}|y)$$

For our understanding, let s_k be the significance of the the kth gene. x_k and x_j denotes the kth and jth gene respectively from a pool of genes, p, in a class condition relation which measures information shared between gene k and remaining genes conditioned on y. This enables us to assess correlations between genes in groups and make more accurate predictions. However, in our case, we have considered the pre-defined standard of figuring out the top 400 most efficient genes (features).

TABLE II
Feature and Execution Time

Model	Features	Execution Time (mins)
Chi2 (Local)	327	9.40
RFECV	3942	260.40
mutual_info_regression	400	1.30

TABLE III
Split and Accuracy for Local Solution

Split	Accuracy
2000*11	84%
222*100	75%

From our findings using the leukemia dataset, it is clear to see that the local solution's execution time fits in between both the RFECV and the mutual info regression sequential solutions. The reason as to why RFECV took the longest is potentially due to it being a wrapper-based solution, hence it takes a lot more time to train compared to filter-based solutions [18].

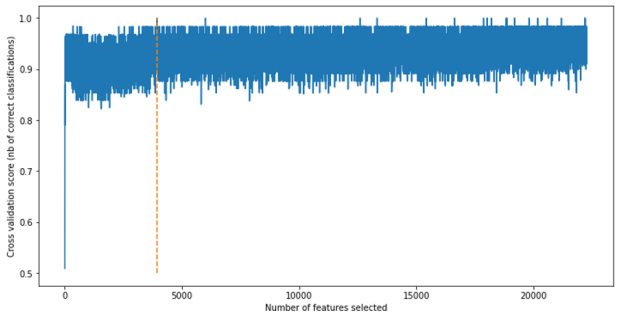

Fig. 4. RFECV feature selection performance

The reason for the mutual info regression sequential solution performing feature selection faster than the local solution may have been caused by the wide dataset used.

In order to see how well the local solution performed, we experimented with the subset size (the number of features passed in) and the number of subsets using the same dataset. From Table III, we see that the local solution's accuracy decreases as more smaller subsets of features are used. We believe that this is the case because when more subsetting is applied to the dataset to be fed into the algorithm, each partition will have a more limited view of the entire dataset, decreasing the model's accuracy of predicting on unseen data.

To further investigate our findings, we have tested these 2 solutions on a different dataset. This dataset is a much more balanced one, having 8992 instances and 1024 features.

TABLE IV
FEATURE AND EXECUTION TIME ON A BALANCED DATASET

Model	Features	Execution Time (mins)
Chi2 (Local)	35	0.98
mutual_info_regression	35	1.34

Table IV clearly shows how much better the local solution performs on this type of dataset. Firstly, we see that the gap between execution times for both of these solutions has drastically reduced after providing it a more balanced dataset. Not only this, but the local solution outperforms the sequential solution. The main difference appears to be the dimensionality of the dataset used. A lot more instances are present in this particular dataset and it is very evident that parallelization has been exploited much better on this dataset. As a result, it performs feature selection faster than the sequential solution.

A lot of novel algorithms are being used in feature selection like Distributed ReliefF which has provided exceptional results with microarray data. However, the generalization of this algorithm with any high-dimensional data still requires a lot of experimentation.

IV. CONCLUSION

The Chi2 algorithm is still one of the most efficient and reliable algorithms to work with microarray data. The reliability of the algorithm, backed by numerous research, can set up the stage for experimenting with other algorithms. In our paper, we highlighted that efficiency can drastically change based on how we split the data, irrespective of the execution time. It is also notable that the dimensionality of the dataset provided can have a major effect on the execution time of the local solution. The results obtained seem to imply that if more instances are provided in the dataset, parallelization can be utilized and exploited better by the local solution, whereas a sequential solution cannot take advantage of this. Hence, a lot of experimental ground is still open on that front.

REFERENCES

[1] D. Peralta, S. del Río, S. Ramírez-Gallego, I. Triguero, J. M. Benitez, and F. Herrera, "Evolutionary feature selection for big data classification: A mapreduce approach," *Mathematical Problems in Engineering*, vol. 2015, 2015.

[2] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos, *Feature selection for high-dimensional data*. Springer, 2015.

[3] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," pp. 16–28, 2014, 40th-year commemorative issue. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0045790613003066

[4] H. Liu and H. Motoda, *Computational methods of feature selection*. CRC Press, 2007.

[5] G. H. John, R. Kohavi, and K. Pfleger, "Irrelevant features and the subset selection problem," in *Machine learning proceedings 1994*. Elsevier, 1994, pp. 121–129.

[6] J. Miao and L. Niu, "A survey on feature selection," *Procedia Computer Science*, vol. 91, pp. 919–926, 2016, promoting Business Analytics and Quantitative Management of Technology: 4th International Conference on Information Technology and Quantitative Management (ITQM 2016). [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050916313047

[7] W. Awada, T. M. Khoshgoftaar, D. Dittman, R. Wald, and A. Napolitano, "A review of the stability of feature selection techniques for bioinformatics data," in *2012 IEEE 13th International Conference on Information Reuse Integration (IRI)*, 2012, pp. 356–363.

[8] Y. Saeys, I. Inza, and P. Larrañaga, "A review of feature selection techniques in bioinformatics," *Bioinformatics*, vol. 23, no. 19, pp. 2507–2517, 08 2007. [Online]. Available: https://doi.org/10.1093/bioinformatics/btm344

[9] A. Jović, K. Brkić, and N. Bogunović, "A review of feature selection methods with applications," in *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2015, pp. 1200–1205.

[10] P. Jafari and F. Azuaje, "An assessment of recently published gene expression data analyses: Reporting experimental design and statistical factors," *BMC medical informatics and decision making*, vol. 6, p. 27, 02 2006.

[11] Z. Hira and D. Gillies, "A review of feature selection and feature extraction methods applied on microarray data," *Advances in Bioinformatics*, vol. 2015, pp. 1–13, 06 2015.

[12] A. Statnikov, C. Aliferis, I. Tsamardinos, D. Hardin, and S. Levy, "Statnikov a, aliferis cf, tsamardinos i, hardin d, levy sa comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis. bioinformatics 21: 631-643," *Bioinformatics (Oxford, England)*, vol. 21, pp. 631–43, 04 2005.

[13] V. Rupapara, F. Rustam, W. Aljedaani, H. F. Shahzad, E. Lee, and I. Ashraf, "Blood cancer prediction using leukemia microarray gene data and hybrid logistic vector trees model," *Scientific Reports*, vol. 12, no. 1, p. 1000, 2022. [Online]. Available: https://doi.org/10.1038/s41598-022-04835-6

[14] P. Misra and A. Singh, "Improving the classification accuracy using recursive feature elimination with cross-validation," vol. 11, pp. 659–665, 05 2020.

[15] A. Mustaqim, S. Adi, Y. Pristyanto, and Y. Astuti, "The effect of recursive feature elimination with cross-validation (rfecv) feature selection algorithm toward classifier performance on credit card fraud detection," 06 2021, pp. 270–275.

[16] A. Z. Mustaqim, S. Adi, Y. Pristyanto, and Y. Astuti, "The effect of recursive feature elimination with cross-validation (rfecv) feature selection algorithm toward classifier performance on credit card fraud detection," in *2021 International Conference on Artificial Intelligence and Computer Science Technology (ICAICST)*, 2021, pp. 270–275.

[17] Y. Wang, X.-G. Yang, and Y. Lu, "Informative gene selection for microarray classification via adaptive elastic net with conditional mutual information," *Applied Mathematical Modelling*, vol. 71, pp. 286–297, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0307904X19300745

[18] B. Kumari and T. Swarnkar, "Filter versus wrapper feature subset selection in large dimensionality micro array: A review," *International Journal of Computer Science and Information Technologies*, vol. 2, pp. 1048–1053, 01 2011.