From Fuzzy-Rough to Crisp Feature Selection

by

© *Javad Rahimipour Anaraki*

A thesis submitted to the

School of Graduate Studies

in partial fulfilment of the

requirements for the degree of

Doctor of Philosophy

Department of *Computer Science*

Memorial University of Newfoundland

May 2019

St. John's Newfoundland

Abstract

A central problem in machine learning and pattern recognition is the process of recognizing the most important features in a dataset. This process plays a decisive role in big data processing by reducing the size of datasets. One major drawback of existing feature selection methods is the high chance of redundant features appearing in the final subset, where in most cases, finding and removing them can greatly improve the resulting classification accuracy. To tackle this problem on two different fronts, we employed fuzzy-rough sets and perturbation theories. On one side, we used three strategies to improve the performance of fuzzy-rough set-based feature selection methods. The first strategy was to code both features and samples in one binary vector and use a shuffled frog leaping algorithm to choose the best combination using fuzzy dependency degree as the fitness function. In the second strategy, we designed a measure to evaluate features based on fuzzy-rough dependency degree in a fashion where redundant features are given less priority to be selected. In the last strategy, we designed a new binary version of the shuffled frog leaping algorithm that employs a fuzzy positive region as its similarity measure to work in complete harmony with the fitness function (i.e. fuzzy-rough dependency degree). To extend the applicability of fuzzy-rough set-based feature selection to multi-party medical datasets, we designed a privacy-preserving version of the original method. In addition, we studied the feasibility and applicability of perturbation theory to feature selection, which to the best of our knowledge has never been researched. We introduced a new feature selection based on perturbation theory that is not only capable of detecting and discarding redundant features but also is very fast and flexible in accommodating the

special needs of the application. It employs a clustering algorithm to group likely-behaved features based on the sensitivity of each feature to perturbation, the angle of each feature to the outcome and the effect of removing each feature to the outcome, and it chooses the closest feature to the centre of each cluster and returns all those features as the final subset. To assess the effectiveness of the proposed methods, we compared the results of each method with well-known feature selection methods against a series of artificially generated datasets, and biological, medical and cancer datasets adopted from the University of California Irvine machine learning repository, Arizona State University repository and Gene Expression Omnibus repository.

Acknowledgements

First and foremost I would like to thank my supervisors Dr. Hamid Usefi, a man of perseverance and determination, Dr. Saeed Samet, committed and caring, and Dr. Wolfgang Banzhaf for their continuous support and believing me. Their guidance shed light on my path toward a better understanding of the question at hand and also about the philosophy of life, itself.

None of this would ever even be dreamt of without enormous unconditional love and non-stop energy and positivity of my family, Ali, Sadiyeh, Shohreh, Nahid, Shadi, Farshid and dearest Kian. They all contributed to who I am and where I am standing.

I have been inspired by many brilliant individuals at the Memorial University of Newfoundland, who I will always be thankful for their encouragement, faith and reassurance behaviour. To name a few, Dr. Daniel Fuller, a true friend and remarkably understanding and supportive supervisor, Dr. Antonina Kolokolova, undoubtedly a genius and treasure in the Department of Computer Science, Dr. Lourdes Peña-Castillo, an example of well-knowledgeable and precise scientist, and Dr. Andrew Smith, a talented engineer and one-of-the-kind physician who asks the best and the most to-the-point questions.

Finally, I would like to thank my friends, who have been extremely motivating and cheerful at the time of disappointment and failure.

Contents

List of Tables

List of Figures

Chapter 1

Introduction and Overview

Selection is a fundamental and vital process in nature that influenced the very first living creatures. Any failure in the selection mechanism would result in the destruction and extinction of all living beings.

The selection mechanism is borrowed from nature and used to design problem solvers, such as evolutionary algorithms (EA). The artificial selection process requires a quality assessment method to evaluate the "goodness" of individuals involved in the process, in which the one with the best quality(ies) is selected. EAs utilize the selection concept to choose the "best" individuals so that the "quality" of the population improves over time. Another area where selection is used is feature selection (FS). FS is the process of selecting the most "important" and "informative" columns of a dataset, which can result in a fewer prediction errors compared to unreduced datasets.

1.1 Definitions

Feature selection is the process of selecting the most important and informative features of a dataset can be applied to supervised, unsupervised and semi-supervised learning problems. Each feature selection method consists of a criterion and a search method. The process of selection is led by the search method by evaluating subsets using the chosen criterion. Features can be divided into three categories:

1. Independent: Features that are not correlated to the other features except for the outcome

2. Redundant: Features that are correlated to other features but not the outcome

3. Irrelevant: Features that are neither correlated to the other features nor the outcome

In real-world datasets, features are not explicitly grouped as presented above, as we have a combination of characteristics for each feature as follows:

1. Independent-relevant: Features that are loosely correlated to the other features and strongly correlated to the outcome

2. Independent-redundant: Features that are strongly correlated to the other features and the outcome

3. Irrelevant-redundant: Features that are strongly correlated to the other features but loosely correlated to the outcome

In this thesis, we aim to select independent-relevant features and remove the rest, since this type of feature provides great distinguishing power to the post-processing phase (i.e. classification).

1.2 Why do we need feature selection?

Noisy features can mislead induction algorithms [39] and significantly disrupt the results. Although some classifiers, such as decision tree and linear regression, are more robust methods for dealing with noise [39], there should be a mechanism available to reduce the effect of noise for methods that are prone to be affected. One solution is to employ a feature selection method to remove noisy/misleading features by calculating the relevancy to the outcome and predictability of a feature. If a feature does not meet the requirements/threshold for being relevant, then it will be removed. The other type of noise is noisy labels, in which the outcome is polluted. Only a small set of feature selection methods and classifiers can handle this type of noise [27, 15].

The time and space complexity of classifiers are quadratic and higher [64]. For instance, support vector machine classifier [67] has a time complexity of $O(m^3)$, where m is the number of samples [64], and by applying extreme improvements to the implementation by Platt [56], it can be reduced to a range from $O(m)$ to $O(m^{2.3})$. In the case of using a decision tree classifier [16], the time complexity is $O(nm^2 \log m)$, where n is the number of features. In most cases, classification is done more than once since we would like to improve our model as more data become available. So, the more often a new set of data becomes available, the more frequently a classifier is re-applied to the data. To make this process feasible, feature selection can be

employed to select a subset of features, in some cases less than 0.01% of all features, to improve the performance of a classifier.

The curse of dimensionality is a phenomenon noted by Bellman [7] that occurs when the number of samples in a dataset is far smaller than the number of features $m \ll n$. This becomes problematic for statistical methods that test the null hypothesis to try to decide on the significance of data. There are two ways to handle this problem: 1- adding more samples or 2- applying feature selection to the dataset. In many applications, adding more data is hard and applying feature selection is the most commonly used option. The number of samples required for a classifier to train over an arbitrary dataset is a vague and under-researched problem. However, Almuallim and Dietterich [3] have stated that to learn a binary concept using a subset p of all features n, through any probably-approximately correct (PAC) learning algorithm that implements MIN-FEATURES bias, $\Theta(\frac{1}{\epsilon}\ln\frac{1}{\delta} + \frac{1}{\epsilon}[2^p + p\ln n])$ samples are required.

1.3 What about deep neural networks?

With the recent advancements in deep neural networks [9, 43, 62, 73, 61, 28], it might be assumed that machine learning methods will soon to be phased out. This, however, is not the case. The fact that deep neural networks have pushed boundaries in artificial intelligence and data mining is inevitable; however, two requirements need to be met for deep neural networks to work in their full capacity. The first requirement is having access to a very large dataset containing millions of samples, and the second is providing significant processing power to handle the dataset and train the networks

against the data. These two requirements are not available in most cases.

The other issue with deep neural networks is that the generated model is a *black-box* where we cannot see explicitly what features are present and how they contribute to the outcome. In many applications, such as DNA microarray data [48, 46, 50], intrusion detection systems [14, 24, 4], document classification [26, 66, 1] and land cover classification [49, 74, 29], researchers would like to know what are the most informative and important features to determining the outcome. Feature selection can likely provide an answer to those questions.

1.4 Feature subset selection or feature ranking?

Feature selection is divided into two subcategories: feature subset selection and feature ranking [13]. The first is the process of selecting a subset of features, and the second is the process of ranking features using a criterion. A simple way to have a subset of features as an outcome of a feature ranking procedure is by either choosing a threshold for the criterion or the subset size of k, where top k features are returned as a subset of features. Although these methods work differently, the output is a subset of features. Therefore, in general, feature subset selection and feature ranking can be considered as feature selection. We will use the same terminology throughout this thesis.

1.5 Feature selection taxonomy

Feature selection methods can be categorized into three groups: 1- filter methods, 2- wrapper methods, and 3- embedded methods [59]. We will go through each method in the following.

Filter methods assess each feature based on its mathematical and statistical significance, where the most important features are selected and returned solely based on a criterion. These methods are fast, scalable and work independently from the induction algorithm. Some well-known filter-based feature selection methods are correlation-based feature selection [31, 75], max-dependency, max-relevance, and min-redundancy feature selection [55, 45], Relief [40] and its successors such as ReliefF [42] and RReliefF [58].

Wrapper methods measure the quality of each feature using an induction algorithm. These methods are slow, highly computational and more accurate compared to filter methods, and the selected subset can vary from one learning algorithm to another. Some of the first attempts were sequential backward selection [51], sequential forward selection [70] and Plus-l-Minus-r $(l - r)$ [63].

Embedded methods select features in the training process of a learning algorithm in response to the resulting classification error [12]. These methods are faster than wrapper methods, but the final result can vary if the learning algorithm changes. Some of the seminal works are recursive feature elimination [30] and decision trees [23].

In this thesis, we will investigate filter-based feature selection methods for supervised learning problems.

Rough set was introduced by Pawlak in 1982 [53] to deal with uncertainty and vagueness in data. The approximation space in the rough set is defined by $(\mathbb{U}, R)$, where $\mathbb{U}$ is the universe of discourse and R is the equivalence relation on $\mathbb{U}$. Let P be a subset of R and X be a subset of $\mathbb{U}$, then approximating X concerning P (see the sample dataset adopted from the UCI repository [21] shown in Table 1.1) is done using *lower* and *upper* approximations. *Lower* approximation, denoted by $\underline{P}X$, contains those objects in X that can be exactly categorized by considering the attributes in P. However, the *upper* approximation, denoted by $\overline{P}X$, contains those objects in X that can be "possibly" categorized using the attributes in P. For interested readers, a comparison of rough sets and fuzzy sets can be found in [54].

Based on the *lower* and *upper* approximations, three regions are defined as positive, negative and boundary regions. The positive region, denoted by $POS_P(Q)$, contains all the objects of different subsets of $\mathbb{U}$ partitioned by Q concerning P. The negative region, denoted by $NEG_P(Q)$, contains all the objects that are in $\mathbb{U}$ but not in the *upper* approximation (see Figure 1.1). The boundary region, denoted by $BND_P(Q)$, contains all the objects that are in the *upper* approximation but not in the positive region. To find the dependency of each feature to the outcome, we use dependency degree (DD), which is the size of the positive region normalized by the total number of samples in a dataset.

In rough set feature subset selection, the evaluation measure is DD, and the search method is usually a simple greedy forward method. However, many researchers have improved the resulting subsets by fusing dependency degree with different search

Table 1.1: A sample dataset

	Condition Attributes (R)					Decision Attribute (Q)
	Object	**Colour**	**Size**	**Act**	**Age**	**Inflated**
Universe of Discourse ($\mathbb{U}$)	x_1	Yellow	Small	Stretch	Adult	T
	x_2	Yellow	Small	Stretch	Child	T
	x_3	Yellow	Small	Dip	Adult	T
	x_4	Yellow	Small	Dip	Child	F
	x_5	Yellow	Small	Dip	Child	F
	x_6	Yellow	Large	Stretch	Adult	T
	x_7	Yellow	Large	Stretch	Child	T
	x_8	Yellow	Large	Dip	Adult	T
	x_9	Yellow	Large	Dip	Child	F
	x_{10}	Yellow	Large	Dip	Child	F
	x_{11}	Purple	Small	Stretch	Adult	T
	x_{12}	Purple	Small	Stretch	Child	T
	x_{13}	Purple	Small	Dip	Adult	T
	x_{14}	Purple	Small	Dip	Child	F
	x_{15}	Purple	Small	Dip	Child	F
	x_{16}	Purple	Large	Stretch	Adult	T
	x_{17}	Purple	Large	Stretch	Child	T
	x_{18}	Purple	Large	Dip	Adult	T
	x_{19}	Purple	Large	Dip	Child	F
	x_{20}	Purple	Large	Dip	Child	F

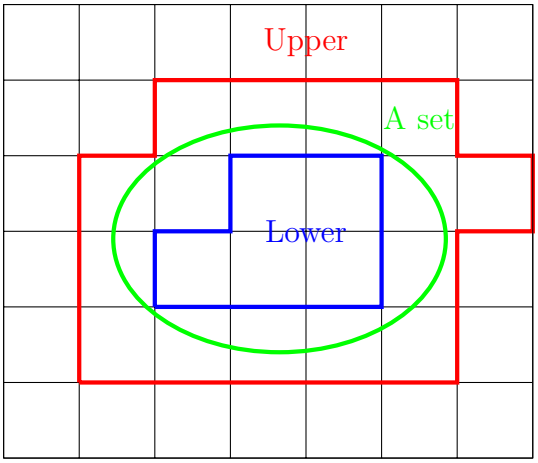


Figure 1.1: A set (green region) and its *upper* (red region) and *lower* (blue) approximations in rough set theory

algorithms, such as particle swarm optimization [69, 33, 71, 65], genetic algorithm [72, 76], ant colony optimization [17, 36, 8], harmony search [34, 5], and tabu search [68, 32].

One main drawback of rough set feature selection is that it can only be applied to datasets with categorical features. To tackle this problem, two solutions are available: 1- the discretization of continuous features [20, 41, 52], and 2- using fuzzy-rough sets to handle continuous features [44, 10, 38]. In the next section, we will provide more details on fuzzy-rough set methods.

Fuzzy-rough set was introduced by Dubois and Prade [22] to handle continuous features where the *lower* and *upper* approximations introduced in the previous section are redefined by Jensen and Shen [37] based on the fuzzy-rough set concept, and different types of fuzzy-rough sets are compared by Radzikowska and Kerre [57]. Jensen and Shen [37] also introduced a simple feature selection method based on fuzzy-rough dependency degree and a greedy forward search method called the FuzzyRough

QuickReduct algorithm. This starts by calculating the dependency degree of each feature and selects a feature with the highest dependency degree and adds it to a subset. Then, it calculates all the combinations with the selected feature and chooses the one with the highest dependency degree. This process continues until either the overall dependency degree becomes one or it stops improving by adding more features.

Using fuzzy-rough dependency degree and a shuffled frog leaping algorithm [25], we proposed a feature-sample selection method in which features and samples are represented as a binary vector. This method simultaneously selects the most important features and samples, so the final subset is smaller in size in both dimensions. For further reading on instance selection based on the fuzzy-rough set, the reader is referred to [35, 19]. The proposed method is applied to a real-world brain signal dataset collected using a functional near-infrared spectroscopy device. More details are provided in Chapter 2.

Due to the deficiency of the original fuzzy-rough dependency degree in removing redundant features, we proposed a new measure for feature selection based on the correlation-based feature selection merit [31] and fuzzy-rough dependency degree. Because of the structure of correlation-based feature selection merit, the number of redundant features in the final set is smaller than the original measure. Therefore, we substituted the Pearson correlation in the merit with the fuzzy-rough dependency degree, and the results are presented in Chapter 3.

To further improve the results of fuzzy-rough feature selection fused with the shuffled frog leaping algorithm, we proposed a binary version of the shuffled frog leaping algorithm as well as a set of improvements. We employed a fuzzy positive region as a similarity measure to find similar high-quality subsets. The results show

that the modifications have improved the results significantly. We will go through the changes in Chapter 4.

In some cases where the dataset is not available in a centralized fashion due to privacy concerns, such as in medical data, and the researcher cannot apply feature selection methods directly to data, we need a set of methods called privacy-preserving data mining methods [2, 47, 60]. This class of methods do not reveal any information about the data and only work on the aggregation values of the data. So far, only a small number of researchers [18, 6, 11] have proposed privacy-preserving versions of well-known feature selection methods. This was a motivation for us to start with rough set feature selection and propose a privacy-preserving rough set feature selection. Mathematical concepts and details are presented in Chapter 5.

Generally, feature selection methods tend to not only select redundant features but also appear to be a computational burden for a data processing pipeline. Moreover, the application of a system of equations in feature selection has not been investigated thoroughly. Therefore, we proposed a new feature selection method called perturbation-based feature selection based on the system of equations and method of least square, which has a mechanism to detect redundant features through the phenomenon of *Shaking Minarets* and perturbation theory. *Shaking Minarets* is a historical monument located in Isfahan Iran that has a unique characteristic in which if one of the minarets is shaking, the other starts to shake as well. We employed perturbation theory to mimic the same effect and uncover redundant features through applying a clustering method to group like-behaved features. To refine each cluster, we calculate the angle of each feature to the outcome and the angle of resulting outcome after removing each feature to the original outcome. This idea is discussed in

Chapter 6.

To further investigate the effectiveness of the proposed feature selection method based on perturbation theory, we did a comparison study with two well-known feature selection methods over an Inflammatory Bowel Disease (IBD) dataset. The results are reflected in Chapter 7

Bibliography

[1] D. Agnihotri, K. Verma, and P. Tripathi. Variable global feature selection scheme for automatic classification of text documents. *Expert Systems with Applications*, 81:268–281, 2017.

[2] R. Agrawal and R. Srikant. *Privacy-preserving data mining*, volume 29. ACM, 2000.

[3] H. Almuallim and T. G. Dietterich. Learning with many irrelevant features. In *AAAI*, volume 91, pages 547–552. Citeseer, 1991.

[4] M. A. Ambusaidi, X. He, P. Nanda, and Z. Tan. Building an intrusion detection system using a filter-based feature selection algorithm. *IEEE transactions on computers*, 65(10):2986–2998, 2016.

[5] M. Bagyamathi and H. H. Inbarani. A novel hybridized rough set and improved harmony search based feature selection for protein sequence classification. In *Big data in complex systems*, pages 173–204. Springer, 2015.

[6] M. Banerjee and S. Chakravarty. Privacy preserving feature selection for dis-

tributed data using virtual dimension. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 2281–2284. ACM, 2011.

[7] R. Bellman. Curse of dimensionality. *Adaptive control processes: a guided tour. Princeton, NJ*, 1961.

[8] R. Bello, A. Nowe, Y. Gomezd, and Y. Caballero. Using aco and rough set theory to feature selection. *WSEAS Transactions on Information Science and Applications*, 2(5):512–517, 2005.

[9] Y. Bengio et al. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009.

[10] R. B. Bhatt and M. Gopal. On fuzzy-rough sets approach to feature selection. *Pattern recognition letters*, 26(7):965–975, 2005.

[11] H. K. Bhuyan and N. K. Kamila. Privacy preserving sub-feature selection in distributed data mining. *Applied Soft Computing*, 36:552–569, 2015.

[12] A. L. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial intelligence*, 97(1-2):245–271, 1997.

[13] V. Boln-Canedo, N. Snchez-Maroo, and A. Alonso-Betanzos. *Feature Selection for High-Dimensional Data*. Springer Publishing Company, Incorporated, 1st edition, 2015.

[14] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos. *Application of*

Feature Selection to Real Problems, pages 95–124. Springer International Publishing, Cham, 2015.

[15] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos. Feature selection for high-dimensional data. *Progress in Artificial Intelligence*, 5(2):65–75, 2016.

[16] L. Breiman. *Classification and regression trees*. Routledge, 2017.

[17] Y. Chen, D. Miao, and R. Wang. A rough set approach to feature selection based on ant colony optimization. *Pattern Recognition Letters*, 31(3):226–233, 2010.

[18] K. Das, K. Bhaduri, and H. Kargupta. A local asynchronous distributed privacy preserving feature selection algorithm for large peer-to-peer networks. *Knowledge and information systems*, 24(3):341–367, 2010.

[19] J. Derrac, C. Cornelis, S. García, and F. Herrera. Enhancing evolutionary instance selection algorithms by means of fuzzy rough set based feature selection. *Information Sciences*, 186(1):73–92, 2012.

[20] J. Dougherty, R. Kohavi, and M. Sahami. Supervised and unsupervised discretization of continuous features. In *Machine Learning Proceedings 1995*, pages 194–202. Elsevier, 1995.

[21] D. Dua and E. Karra Taniskidou. UCI machine learning repository, 2017.

[22] D. Dubois and H. Prade. Rough fuzzy sets and fuzzy rough sets. *International Journal of General System*, 17(2-3):191–209, 1990.

[23] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern classification*. John Wiley & Sons, 2012.

[24] A. S. Eesa, Z. Orman, and A. M. A. Brifcani. A novel feature-selection approach based on the cuttlefish optimization algorithm for intrusion detection systems. *Expert Systems with Applications*, 42(5):2670–2679, 2015.

[25] M. M. Eusuff and K. E. Lansey. Optimization of water distribution network design using the shuffled frog leaping algorithm. *Journal of Water Resources planning and management*, 129(3):210–225, 2003.

[26] G. Forman. An extensive empirical study of feature selection metrics for text classification. *Journal of machine learning research*, 3(Mar):1289–1305, 2003.

[27] B. Frénay and M. Verleysen. Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems*, 25(5):845–869, 2014.

[28] A. Graves, A.-r. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, pages 6645–6649. IEEE, 2013.

[29] S. Gupta, S. Kumar, A. Garg, D. Singh, and N. S. Rajput. Class wise optimal feature selection for land cover classification using sar data. In *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 68–71. IEEE, 2016.

[30] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer

classification using support vector machines. *Machine learning*, 46(1-3):389–422, 2002.

[31] M. A. Hall. Correlation-based feature selection for machine learning. 1999.

[32] A.-R. Hedar, J. Wang, and M. Fukushima. Tabu search for attribute reduction in rough set theory. *Soft Computing*, 12(9):909–918, 2008.

[33] H. H. Inbarani, A. T. Azar, and G. Jothi. Supervised hybrid feature selection based on pso and rough sets for medical diagnosis. *Computer methods and programs in biomedicine*, 113(1):175–185, 2014.

[34] H. H. Inbarani, M. Bagyamathi, and A. T. Azar. A novel hybrid feature selection method based on rough set and improved harmony search. *Neural Computing and Applications*, 26(8):1859–1880, 2015.

[35] R. Jensen and C. Cornelis. Fuzzy-rough instance selection. In *Fuzzy Systems (FUZZ), 2010 IEEE International Conference on*, pages 1–7. IEEE, 2010.

[36] R. Jensen and Q. Shen. Finding rough set reducts with ant colony optimization. In *Proceedings of the 2003 UK workshop on computational intelligence*, volume 1, pages 15–22, 2003.

[37] R. Jensen and Q. Shen. New approaches to fuzzy-rough feature selection. *Fuzzy Systems, IEEE Transactions on*, 17(4):824–838, Aug 2009.

[38] R. Jensen, Q. Shen, et al. New approaches to fuzzy-rough feature selection. *IEEE Transactions on Fuzzy Systems*, 17(4):824, 2009.

[39] E. Kalapanidas, N. Avouris, M. Craciun, and D. Neagu. Machine learning algorithms: a study on noise sensitivity. In *Proc. 1st Balcan Conference in Informatics*, pages 356–365, 2003.

[40] K. Kira and L. A. Rendell. The feature selection problem: Traditional methods and a new algorithm. In *Aaai*, volume 2, pages 129–134, 1992.

[41] R. Kohavi and M. Sahami. Error-based and entropy-based discretization of continuous features. In *KDD*, pages 114–119, 1996.

[42] I. Kononenko, E. Šimec, and M. Robnik-Šikonja. Overcoming the myopia of inductive learning algorithms with relieff. *Applied Intelligence*, 7(1):39–55, 1997.

[43] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[44] L. I. Kuncheva. Fuzzy rough sets: application to feature selection. *Fuzzy sets and Systems*, 51(2):147–153, 1992.

[45] O. Kurşun, C. O. ŞAKAR, O. Favorov, N. Aydin, and S. F. GÜRGEN. Using covariates for improving the minimum redundancy maximum relevance feature selection method. *Turkish journal of electrical engineering & computer sciences*, 18(6):975–989, 2010.

[46] M. Lamba, G. Munjal, and Y. Gigras. Feature selection of micro-array expression data (fsm)-a review. *Procedia Computer Science*, 132:1619–1625, 2018.

[47] Y. Lindell and B. Pinkas. Privacy preserving data mining. In *Annual International Cryptology Conference*, pages 36–54. Springer, 2000.

[48] S. Liu, C. Xu, Y. Zhang, J. Liu, B. Yu, X. Liu, and M. Dehmer. Feature selection of gene expression data for cancer classification using double rbf-kernels. *BMC Bioinformatics*, 19(1):396, Oct 2018.

[49] L. Ma, T. Fu, T. Blaschke, M. Li, D. Tiede, Z. Zhou, X. Ma, and D. Chen. Evaluation of feature selection methods for object-based land cover mapping of unmanned aerial vehicle imagery using random forest and support vector machine classifiers. *ISPRS International Journal of Geo-Information*, 6(2):51, 2017.

[50] Z. Manbari, F. A. Tab, and C. Salavati. Hybrid fast unsupervised feature selection for high-dimensional data. *Expert Systems with Applications*, 2019.

[51] T. Marill and D. Green. On the effectiveness of receptors in recognition systems. *IEEE transactions on Information Theory*, 9(1):11–17, 1963.

[52] D. M. Maslove, T. Podchiyska, and H. J. Lowe. Discretization of continuous features in clinical datasets. *Journal of the American Medical Informatics Association*, 20(3):544–553, 2012.

[53] Z. Pawlak. Rough sets. *International Journal of Computer & Information Sciences*, 11(5):341–356, 1982.

[54] Z. Pawlak. Rough sets and fuzzy sets. *Fuzzy sets and Systems*, 17(1):99–102, 1985.

[55] H. Peng, F. Long, and C. Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence*, 27(8):1226–1238, 2005.

[56] J. C. Platt. 12 fast training of support vector machines using sequential minimal optimization. *Advances in kernel methods*, pages 185–208, 1999.

[57] A. M. Radzikowska and E. E. Kerre. A comparative study of fuzzy rough sets. *Fuzzy sets and systems*, 126(2):137–155, 2002.

[58] M. Robnik-Šikonja and I. Kononenko. An adaptation of relief for attribute estimation in regression. In *Machine Learning: Proceedings of the Fourteenth International Conference (ICML97)*, volume 5, pages 296–304, 1997.

[59] Y. Saeys, I. Inza, and P. Larrañaga. A review of feature selection techniques in bioinformatics. *bioinformatics*, 23(19):2507–2517, 2007.

[60] S. Samet. *Privacy-Preserving Data Mining*. PhD thesis, University of Ottawa (Canada), 2010.

[61] J. Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[62] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484, 2016.

[63] S. Strearns. On selecting features for pattern classifiers. *Proc. ICPR, 1976*, pages 71–75, 1976.

[64] I. W. Tsang, J. T. Kwok, and P.-M. Cheung. Core vector machines: Fast svm training on very large data sets. *Journal of Machine Learning Research*, 6(Apr):363–392, 2005.

[65] A. Unler and A. Murat. A discrete particle swarm optimization method for feature selection in binary classification problems. *European Journal of Operational Research*, 206(3):528–539, 2010.

[66] A. K. Uysal. An improved global feature selection scheme for text classification. *Expert systems with Applications*, 43:82–92, 2016.

[67] V. Vapnik, I. Guyon, and T. Hastie. Support vector machines. *Mach. Learn*, 20(3):273–297, 1995.

[68] J. Wang, K. Guo, and S. Wang. Rough set and tabu search based feature selection for credit scoring. *Procedia Computer Science*, 1(1):2425–2432, 2010.

[69] X. Wang, J. Yang, X. Teng, W. Xia, and R. Jensen. Feature selection based on rough sets and particle swarm optimization. *Pattern recognition letters*, 28(4):459–471, 2007.

[70] A. W. Whitney. A direct method of nonparametric measurement selection. *IEEE Transactions on Computers*, 100(9):1100–1103, 1971.

[71] B. Xue, M. Zhang, and W. N. Browne. Particle swarm optimization for feature

selection in classification: A multi-objective approach. *IEEE transactions on cybernetics*, 43(6):1656–1671, 2013.

[72] J. Yang and V. Honavar. Feature subset selection using a genetic algorithm. In *Feature extraction, construction and selection*, pages 117–136. Springer, 1998.

[73] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328, 2014.

[74] L. Yu, H. Fu, B. Wu, N. Clinton, and P. Gong. Exploring the potential role of feature selection in global land-cover mapping. *International journal of remote sensing*, 37(23):5491–5504, 2016.

[75] L. Yu and H. Liu. Feature selection for high-dimensional data: A fast correlation-based filter solution. In *Proceedings of the 20th international conference on machine learning (ICML-03)*, pages 856–863, 2003.

[76] L.-Y. Zhai, L.-P. Khoo, and S.-C. Fok. Feature extraction using rough set theory and genetic algorithmsan application for the simplification of product quality evaluation. *Computers & Industrial Engineering*, 43(4):661–676, 2002.

Chapter 2

SUFFUSE: Simultaneous Fuzzy-Rough Feature-Sample Selection

This paper is accepted in 4th International Conference on Advancements in Information Technology, Toronto, Canada, 2015, and is selected and published in Journal of Advances in Information Technology.

2.1 Abstract

One of the most successful tools for modelling and dealing with uncertainty is rough set theory. Based on this theory several feature selection methods have been proposed. As an extension, fuzzy-rough set has been introduced to deal with vagueness of both discrete and continuous data in feature and sample selection methods. How-

ever, both fuzzy-rough sample selection and simultaneous fuzzy-rough feature-sample selection are investigated by few. This paper proposes a novel Simultaneous fuzzy-rough feature-sample selection method based on Shuffled Frog Leaping Algorithm. The effectiveness of proposed method demonstrated and compared through its performance resulting from nine conventional as well as an improved mGP classifiers over 15 UCI datasets. This work is also applied to a real world classification problem of noisy Functional Near-Infrared Spectroscopy neural signals. Experimental results show meaningful increase in classification accuracy, and decrease in dataset size according to non-parametric statistical analysis.

Index terms— Fuzzy-rough sets, simultaneous fuzzy-rough feature-sample selection, feature selection, sample selection

2.2 Introduction

The amount of raw data produced daily is much higher than the information extracted from them. Therefore, more cost and time are needed to process, save and maintain those data for later processing. Many problems in machine learning, data mining and pattern recognition involve big datasets. A high dimensional data in terms of number of features and samples needs huge effort to be processed. Therefore, feature selection (FS) methods can effectively reduce the size of datasets in one direction by selecting significant columns. These methods select most-informative features which are highly correlated to the outcome and loosely depended on other features in favor of minimizing further processing. Since the size of datasets can also be decreased in terms of samples, sample selection (SS) methods have emerged to reduce size

of datasets by removing irrelevant samples. Therefore, by employing FS and SS methods, datasets' size can be lowered and further processing can be done more efficiently.

Raman and Ioerger [16], proposed a feature selection, and sample selection method. The former eliminates irrelevant features using a sequential search on feature space to maintain a balance between local hypotheses, the concept which dataset is representing, and prediction accuracy. The latter, uses Hamming distance to filter out samples, and naive Bayes classifier to predict class labels based on the selected samples. Then each method has been applied on a same dataset to perform two dimensional selection. Rozsypal and Kubat [17] have introduced simultaneous feature-sample selection based on genetic algorithm with the aim of increasing classification accuracy and decreasing the number of selected features and samples. Chromosome designation have been established to accommodate two subsets of integers, each representing selected features and samples. The fitness function has been designed based on the number of retained features and samples, and also the number of misclassified examples.

Rough set theory (RST) [14] is one of the most successful mathematical tools in FS [4] which nowadays receives much of attention in SS. This theory has been applied to many real-world applications [18] since it allows minimal representation of the data while sustaining semantic of data with no human provided information. However, RST is only useful to deal with crisp and discrete data; therefore, a combination of RST and Fuzzy Set has been proposed in [5] to overcome this inadequacy. Stand on fuzzy-rough set (FR), some research has been conducted in FS [18, 9] and SS [8], and very few works have been done in simultaneous fuzzy-rough feature-sample selection [12].

Genetic Programming (GP) is capable of finding hidden relations in data and presenting them in terms of mathematical functions [13]. This method has been widely used in tough classification problems and investigated by many researchers to develop classifiers for two- and multi-class problems. In [2], An et al. designed a new multi-tree GP (mGP) classifier by modifying crossover and mutation operators.

In this paper we have proposed a simultaneous fuzzy-rough feature-sample selection method (SUFFUSE) based on Shuffled Frog Leaping Algorithm (SFLA) [6], as well as an improved mGP. The rest of the paper is organized as follows: Section 2.3 describes preliminaries of FR, SFLA and mGP. Section 2.4 presents the proposed methods, SUFFUSE, and improved mGP. In Section 2.5, experimental results are shown. Application to noisy Functional Near-Infra-red Spectroscopy (fNIRS) neural signals dataset and conclusion are placed in Sections 2.6 and 2.7, respectively.

2.3 Preliminaries

Two fundamental components of feature, sample and feature-sample selections are Evaluation Metric and Search Method. In this work the former is based on fuzzy-rough positive region (FRPR), and the latter uses SFLA. Finally, an improved mGP classifier analyzes and builds data models to figure out capabilities of proposed methods. All basics are categorized as follows:

2.3.1 Evaluation metric: Fuzzy-Rough Positive Region (FRPR)

In RST, data are organized in decision table. Let $\mathbb{U}$ be the universe of discourse, R be the equivalence relation on $\mathbb{U}$, so approximation space is shown by $(\mathbb{U}, R)$. Let

X be a subset of $\mathbb{U}$ and P be a subset of A, which is a non-empty set of attributes. Approximating X using RST is done by means of lower and upper approximations. Objects in lower approximation ($\underline{P}X$) are the ones which are surely classified in X regarding the attributes in P. Upper approximation of X with regards to ($\overline{P}X$) contains objects which are possibly classified in X regarding the attributes in P. Based on these approximations, three different regions are defined as positive, negative and boundary that are shown by Equations 4.1, 3.3, and 3.4, respectively [10].

$$POS_P(Q) = \bigcup_{X \in \mathbb{U}/Q} \underline{P}X, \tag{2.1}$$

$$NEG_P(Q) = \mathbb{U} - \bigcup_{X \in \mathbb{U}/Q} \overline{P}X, \tag{2.2}$$

$$BND_P(Q) = \bigcup_{X \in \mathbb{U}/Q} \overline{P}X - \bigcup_{X \in \mathbb{U}/Q} \underline{P}X. \tag{2.3}$$

2.3.2 Search method: Shuffled Frog Leaping Algorithm (SFLA)

SFLA is a meta-heuristic search algorithm which is inspired by real frogs. The search starts by generating population over the search space. Then the population is divided into sub-populations called *memeplexes* which are able to evolve separately. In each *memeplex*, frogs participate in meme evolution due to infection by other frogs. By meme evolution, each frog's performance is increased referring to the best frog in each *memeplex* and poor ideas evolve toward new ideas. The frogs are infected both by best frogs in their *memeplex* and the entire population. After specified number of evolutions, *memeplexes* are mixed together and new *memeplexes* are emerged by

shuffling the population. This process migrates frogs to different regions of the swamp. Therefore they can share their experiences with other frogs. A modified binary form of SFLA has been applied to the problem of simultaneous selection.

2.3.3 Multi-tree genetic programming classifier

In [2], individuals of a problem with c-classes are generated randomly with $c-1$ trees. Then all the individuals are evaluated using fitness function and top N individuals are selected based on τ-wise tournament selection. The classifier continues by applying crossover and mutation for generating new individuals. Then, the worst individuals are substituted with the newly generated best ones and the classifier continues until the stopping criterion is satisfied.

2.4 Proposed Methods

2.4.1 Simultaneous Fuzzy-Rough Feature-Sample Selection (SUFFUSE)

The FRSS [3] is based on FRPR as an evaluation measure, and SFLA as a search method. The length of each frog in population is equal to the number of samples in the dataset where their presence or absence are depicted by one and zero, respectively. As SFLA generates initial population, related dataset formations are constructed referring to each individual frog. Then, fitness of all frogs are calculated using FRPR as shown in Equation 4.1. Each frog's formation is shown in Figure 2.1, where $s_j \in \{0, 1\}$ and j is number of samples of dataset.

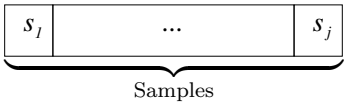


Figure 2.1: Each Frog's Formation in FRSS

Table 2.1: A Decision Table

Samples	**Features**		**Class**
	f_1	f_2	
s_1	0.65	0.59	Yes
s_2	0.93	0.88	No
s_3	0.48	0.73	No
s_4	0.70	0.43	Yes
s_5	0.49	0.76	No
s_6	0.05	0.23	Yes
s_7	0.54	0.60	No

Table 5.3 represents a dataset with two features and seven samples. Based on the table, a possible frog's formation and related dataset is presented in Figure 2.2 and Table 2.2, respectively.

The SFLA continues until the stopping criterion, which is either maximum iteration or gaining the highest FRPR value, is satisfied. Feature and sample selections can be done either in order or simultaneously. Applying either feature or sample selection beforehand might have a huge effect on the final performance. Even if the first operation has a great efficiency, the outcome would be less desirable since each

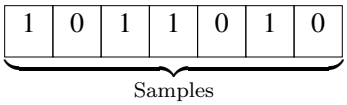


Figure 2.2: A Possible Frog's Formation in FRSS

Table 2.2: Resulting Dataset Referring to Possible Frog's Formation

Samples	**Features**		**Class**
	f_1	f_2	
s_1	0.65	0.59	Yes
s_3	0.48	0.73	No
s_4	0.70	0.43	Yes
s_6	0.05	0.23	Yes

method acts independently. Thus, simultaneous selection would increase the quality of the outcome by considering ongoing two dimensional selection together.

At the starting point, a population consists of frogs with the length proportional to the number of features and samples is generated. Figure 2.3 depicts each frog's formation. In this formation, each bit's value and position show the presence or absence of either a feature or sample that specifies the final structure of the extracted dataset from the original one, where $f_i, s_j \in \{0, 1\}$, and i and j are the number of features and samples in each dataset, respectively.

Figure 2.4 shows the formation of each frog for aforementioned table. Since the first position is equal to one, therefore the proportional feature should participate in the new dataset. Similarly, those samples which corresponding bits are equal to one

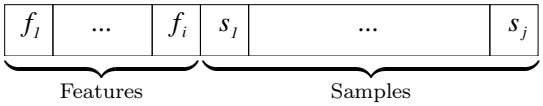


Figure 2.3: Each Frog's Formation with Features and Samples Individuals

will form the output dataset.

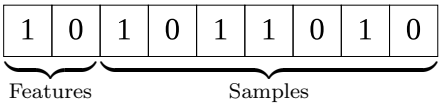


Figure 2.4: Possible Frog's Formation with Features and Samples Individuals

Table 2.3 demonstrates the final dataset formation based on the original dataset in Table 5.3 and by referring to presence and absence of both features and samples in Figure 2.4.

Table 2.3: Resulting Dataset of Possible Frog's Formation with Feature and Samples Individuals

Samples	**Feature**	**Class**
	$\boldsymbol{f_1}$	
s_1	0.65	Yes
s_3	0.48	No
s_4	0.70	Yes
s_6	0.05	Yes

Since Rough Set could not deal with continuous values, the original fuzzy-rough

set has been proposed by Dubois and Prade [5] to elude this lack. Later, a new definition was introduced by Radzikowska and Kerre [15] and then Shen and Jensen [18] modified the original definitions. In [9], final definitions of X-lower and X-upper approximations based on fuzzy-rough sets are presented as in Equations 2.4 and 2.5, where I is Łukasiewicz Fuzzy implicator, which is defined by $min(1 - x + y, 1)$ and T is Łukasiewicz Fuzzy t-norm, which is shown by $max(x + y - 1, 0)$.

$$\mu_{\underline{R_P}X}(x) = \inf_{y \in \mathbb{U}} I\{\eta_{R_P}(x, y), \mu_X(y)\}, \tag{2.4}$$

$$\mu_{\overline{R_P}X}(x) = \sup_{y \in \mathbb{U}} T\{\eta_{R_P}(x, y), \mu_X(y)\}, \tag{2.5}$$

$$\eta_{R_P}(x, y) = \bigcap_{a \in P} \{\eta_{R_a}(x, y)\}. \tag{2.6}$$

In Equation 2.6, R_P is Fuzzy similarity relation and $\eta_{R_a}(x, y)$ is the degree of similarity between objects x and y, considering feature a [9]. A fuzzy similarity relation is shown in Equation 2.7, where σ_a is the variance of feature a. Positive region in RST is defined as a union of lower approximations. Referring to extension principle [9], the membership of object x to a FRPR is defined in Equation 2.8.

$$\eta_{R_P}(x, y) \quad = \quad max\left(min\left(\frac{(a(y) - (a(x) - \sigma_a))}{\sigma_a}, \frac{((a(x) + \sigma_a) - a(y))}{\sigma_a}\right), 0\right), \tag{2.7}$$

$$\mu_{POS_P(Q)}(x) = \sup_{X \in \mathbb{U}/Q} \mu_{\underline{P}X}(x). \tag{2.8}$$

If the equivalence class of which x belongs to, does not belong to the positive region, obviously x will not be a part of the positive region. Equation 2.8 is the

fitness function of the search algorithm which measures the significance of the selected features-samples subset [8]. Finally, SFLA evaluates each final dataset corresponding to each frog by calculating FRPR. The best frog in each *memeplex* infects other frogs, and as a result the whole population moves toward the final goal, which is finding the lowest number of features and samples with the highest fitness value.

In the very first point, dataset is loaded and the number of its features and samples, specifies all parameters of SFLA. In SUFFUSE, SFLA and FRPR collaborate to find the best feature-sample subsets. Then the classification methods, which involve conventional classifiers as well as improved mGP, classify the datasets. The value of division of classification accuracies' mean by summation of the number of selected features and samples is calculated and compared with the results of the FRFS and FRSS. Figure 2.5 shows the overall workflow of SUFFUSE.

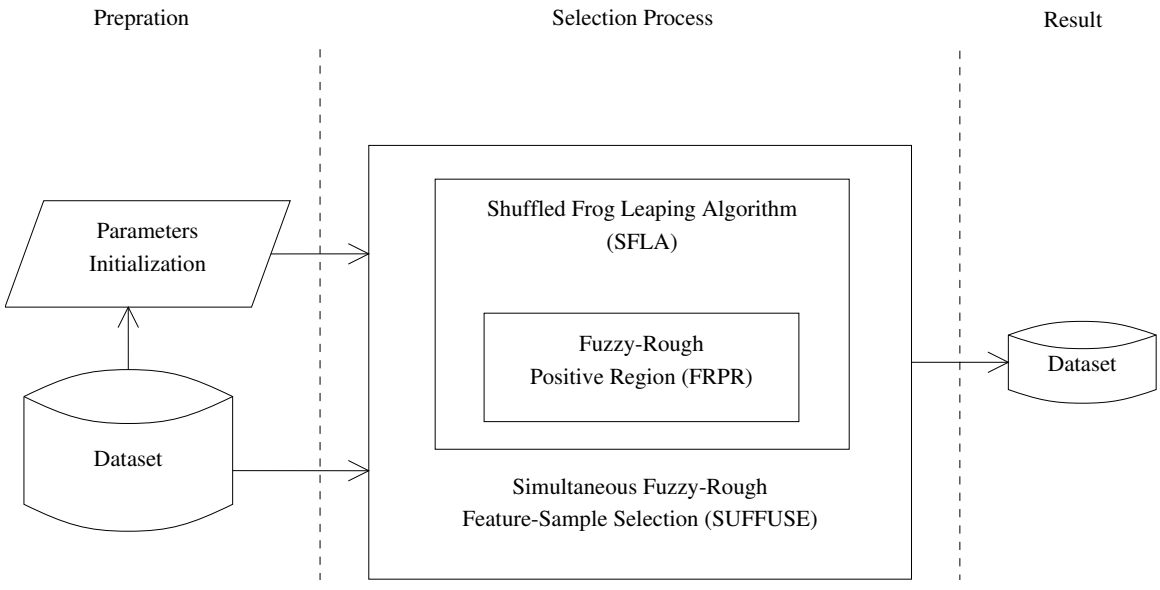


Figure 2.5: Simultaneous Fuzzy-Rough Feature-Sample Selection Workflow

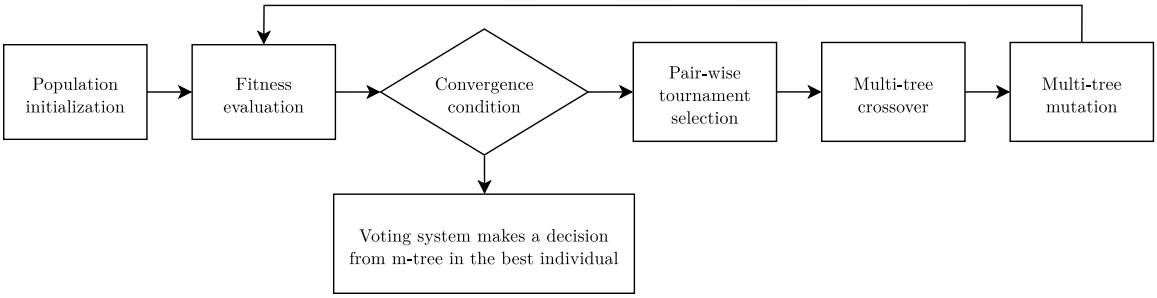


Figure 2.6: Improved Multi-Tree GP

2.4.2 Improved multi-tree GP classifier

This method is robust to noise since the voting system is inspired by honey bee migration that is less sensitive to noise. Figure 2.6 describes the method. Figure 2.7 shows the representation of each individual with its equation referring to the number of trees (m), which is specified by user and number of classes. For instance a three-class dataset would have two classifiers. In the proposed classifier four main parts have been modified as follows:

2.4.2.1 Fitness Function

The new multi-modal fitness function is based on classification accuracy and variance. The goal is to maximize the classification margin, while decreasing intra-class similarities using Equation 2.9. Equation 2.10 calculates the centroid of each class to be used in Equation 2.9. Therefore fitness function is determined by the summation of Classification Accuracy (CA) and distance function as shown in Equation 2.11.

$$Distance = \sum_{i \in classA} \frac{|T_{m,x}^{(i)} - CentroidA|}{|Max(classA) - Min(classA)|}, \tag{2.9}$$

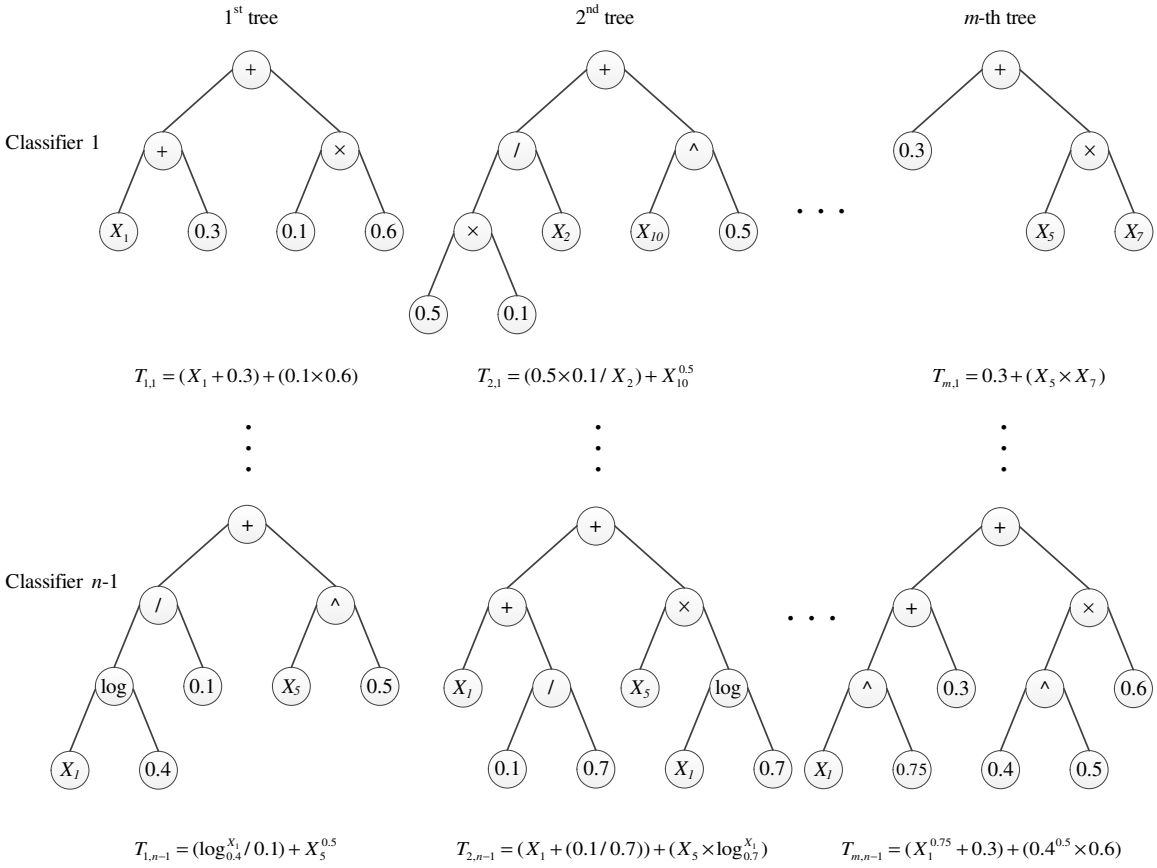


Figure 2.7: An Individual with $m \times (n-1)$ Trees

$$CentroidA = \frac{\sum_{i \in classA} T_{m,x}^{(i)}}{||classA||}, \tag{2.10}$$

$$Fitness = CA + Distance. \tag{2.11}$$

2.4.2.2 Selection Strategy

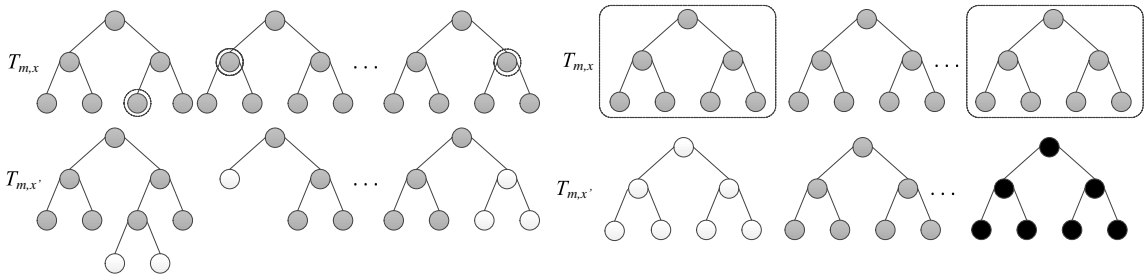
The selection process has three stages. At first top 3% of previous generation is selected to construct new generation, and if there were more than 3% individuals with highest ranking, top 10% will be selected. However, if two or more classifiers have the same fitness value, all of them will be used in the next generation. Then 65% of the new generation is selected based on pair-wise tournament selection. Finally the rest of the individuals will be randomly generated.

2.4.2.3 Mutation

The mutation process contains three policies for the internal mutation and one policy for the external one. In the internal mutation, a node can add, remove or exchange children. Thus the whole tree is reconstructed in the external mutation as Figure 2.8 shows.

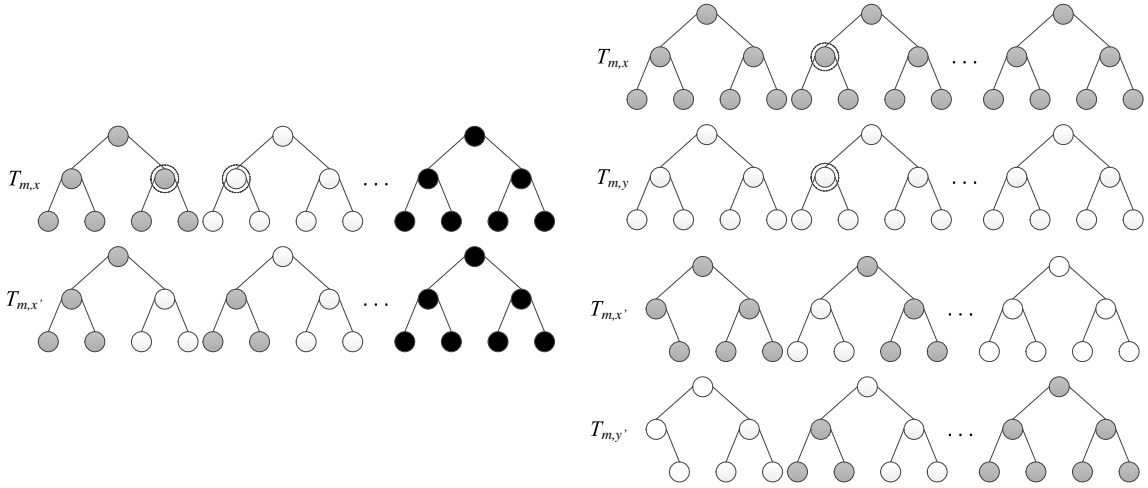
2.4.2.4 Crossover

The crossover is divided into the internal and external crossovers. In the former, trees are selected in each individual based on the internal crossover probability parameter. The latter is based on one-point crossover and it takes place among any trees by considering external crossover probability. Figure 2.9 describes the crossover strategy.



(a) Internal Mutation (b) External Mutation

Figure 2.8: Proposed Mutation Operator



(a) Internal Crossover (b) External Crossover

Figure 2.9: Proposed Crossover Operator

2.5 Experimental Results

Fifteen UCI datasets [11] have been selected to measure the performance of the proposed methods. Parameter selection for SFLA has been formulated based on the number of features $|F|$, samples $|S|$ and feature-samples $|FS|$ using trial and error method. The results are mentioned in Table 2.4, in which m is the number of *memeplexes*, n is the number of frogs in each *memeplex*, N is the number of evolution processes, q is the number of frogs which are selected randomly from n frogs to form a *memeplex* and S_{max} is the maximum step size allowed to be adopted after infection.

Each algorithm runs ten times over the datasets and information-rich features, samples, and features-samples are selected by FRFS, FRSS and SUFFUSE, respectively. The best results over all iterations are chosen and presented in Table 2.5 in terms of the number of selected features and samples and overall model size. The number of samples are fix in the results of FRFS as it only selects features, whereas, the number of features are constant for FRSS since it just affects samples. The mean of ranking for each method is calculated and shown in Table 2.6, in which SUFFUSE performs 51% and 31% better than FRSS and FRFS, respectively.

Table 4.5 shows mean of the classification results for conventional classifiers (such as PART, JRip, Naive Bayes, Bayes Net, J48, BFTree, FT, NBTree and RBFNetwrok, which are implemented in WEKA [7]) as well as improved mGP, and Figure 2.10 presents the classification workflow process. The mean of accuracies of conventional classifiers for our proposed method shows 3.55% increase comparing to unreduced datasets, as well as 2.55% and 1.58% improvement comparing with FRFS and FRSS, respectively. Whereas, the result of improved mGP for SUFFUSE shows 5.58%, 4.10%

and 1.23% increase comparing to the results of improved mGP for unreduced datasets, FRFS and FRSS. As the initial experiment results show, the fusion of SUFFUSE with improved mGP produces the simplest model which leads to the higher classification accuracies.

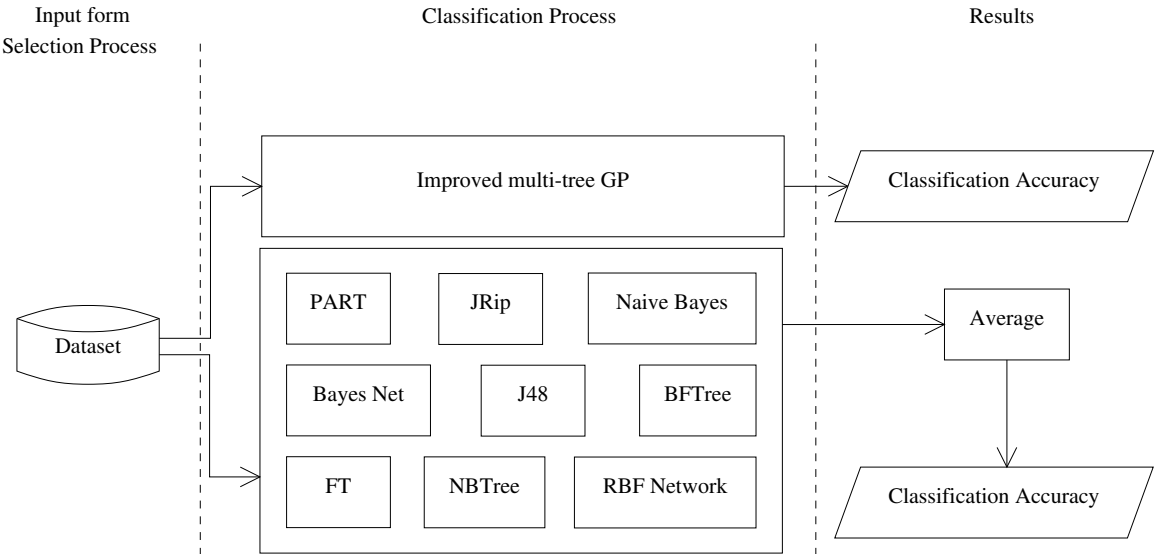


Figure 2.10: Classification Workflow

Table 2.4: SFLA parameters for FRFS, FRSS and SUFFUSE

Method	$\boldsymbol{m}$	$\boldsymbol{n}$	$\boldsymbol{N}$	$\boldsymbol{q}$	$\boldsymbol{S_{max}}$
FRFS	$\|F\| \times 2.20$	$\|F\| \times 0.70$	$\|F\| \times 0.50$	$\|F\| \times 0.45$	$\|F\| \times 0.50$
FRSS	$\|S\| \times 0.02$	$\|S\| \times 0.01$	$\|S\| \times 0.01$	$\|S\| \times 0.50$	$\|S\| \times 0.50$
SUFFUSE	$\|FS\| \times 0.02$	$\|FS\| \times 0.01$	$\|FS\| \times 0.01$	$\|FS\| \times 0.50$	$\|FS\| \times 0.50$

A Non-parametric statistical analysis [1] is employed to compare the overall performance of each method based on the results of improved mGP in Table 4.5. The

Table 2.5: Resulting Reduction and Model Size by FRFS, FRSS, & SUFFUSE

Dataset	**Unreduced**			**FRFS**			**FRSS**			**SUFFUSE**		
	$\|S\|$	$\|F\|$	**Size**	$\|S\|$	$\|F\|$	**Size**	$\|S\|$	$\|F\|$	**Size**	$\|S\|$	$\|F\|$	**Size**
Blood Transfusion	748	4	2992	748	3	2244	264	4	1056	372	2	**744**$^+$
Breast Cancer	683	9	6147	683	7	4781	256	9	2304	357	6	**2142**$^+$
Breast Tissue	106	9	954	106	6	636	70	9	630	51	5	**255**$^+$
Cleveland	297	13	3861	297	7	2079	199	13	2587	108	2	**216**$^+$
Glass	214	9	1926	214	6	1284	144	9	1296	130	7	**910**$^+$
Heart	270	13	3510	270	7	1890	156	13	2028	166	9	**1494**$^+$
Ionosphere	351	33	11583	351	7	2457	115	33	3795	203	12	**2436**$^+$
Lung Cancer	27	56	1512	27	3	**81**$^+$	20	56	1120	10	25	250
Olitos	120	25	3000	120	5	**600**$^+$	81	25	2025	74	12	888
Parkinson	195	22	4290	195	6	1170	130	22	2860	111	10	**1110**$^+$
Pima Indian Diabetes	768	8	6144	768	6	4608	256	8	2048	270	3	**810**$^+$
Sonar	208	60	12480	208	6	**1248**$^+$	140	60	8400	128	34	4352
Soybean	47	35	1645	47	2	**94**$^+$	31	35	1085	30	20	600
SPECTF Heart	80	44	3520	80	6	**480**$^+$	55	44	2420	38	29	1102
Wine	178	13	2314	178	5	890	115	13	1495	97	7	**679**$^+$

Table 2.6: Ranking of FRFS, FRSS and SUFFUSE Based on Model Size

Datasets	**FRFS**	**FRSS**	**SUFFUSE**
Blood Transfusion	3	2	1
Breast Cancer	3	2	1
Breast Tissue	3	2	1
Cleveland	2	3	1
Glass	2	3	1
Heart	2	3	1
Ionosphere	2	3	1
Lung Cancer	1	3	2
Olitos	1	3	2
Parkinson	2	3	1
Pima Indian Diabetes	3	2	1
Sonar	1	3	2
Soybean	1	3	2
SPECTF Heart	1	3	2
Wine	2	3	1
Mean	1.93	2.73	**1.33**$^{+}$

Table 2.7: Average Classification Accuracies (%) of Conventional Classifiers (Part, JRip, Naive Bayes, Bayes Net, J48, BFTree, FT, NBTree and RBFNetwork) and Improved mGP Based on FRFS, FRSS and SUFFUSE Results

Dataset	**Unreduced**		**FRFS**		**FRSS**		**SUFFUSE**	
	Conv.	**mGP**	**Conv.**	**mGP**	**Conv.**	**mGP**	**Conv.**	**mGP**
Blood Transfusion	77.20	79.95	77.30	79.14	78.87	**82.26**$^{+}$	79.24	80.11
Breast Cancer	96.18	96.93	96.40	97.95	95.14	**98.05**$^{+}$	96.70	98.04
Breast Tissue	66.46	69.81	68.66	73.58	65.56	77.14	69.93	**82.35**$^{+}$
Cleveland	50.13	41.28	50.88	52.53	52.26	**57.79**$^{+}$	55.86	57.41
Glass	61.89	53.74	66.87	70.09	66.82	71.53	65.47	**71.54**$^{+}$
Heart	79.55	82.96	73.61	81.85	80.56	84.62	84.54	**86.14**$^{+}$
Ionosphere	89.68	91.74	89.55	91.19	85.70	87.83	90.15	**92.12**$^{+}$
Lung Cancer	55.56	74.07	57.61	77.78	60.56	75.00	64.44	**80.00**$^{+}$
Olitos	69.81	75.00	69.07	72.73	77.23	76.54	73.87	**79.73**$^{+}$
Parkinson	82.34	88.72	85.64	87.69	84.10	90.77	84.99	**92.79**$^{+}$
Pima Indian Diabetes	75.00	75.42	75.61	75.42	75.82	**78.91**$^{+}$	76.01	78.89
Sonar	67.47	78.85	74.73	79.81	74.13	**86.43**$^{+}$	73.61	83.59
Soybean	98.58	**100.00**$^{+}$	90.54	91.49	91.76	**100.00**$^{+}$	92.96	**100.00**$^{+}$
SPECTF Heart	73.06	78.75	73.47	78.75	73.74	80.36	78.65	**84.21**$^{+}$
Wine	85.52	93.82	93.51	93.26	95.70	**99.16**$^{+}$	95.30	97.94
Mean	75.23	78.74	76.23	80.22	77.20	83.09	78.78	**84.32**$^{+}$

Table 2.8: Average Rankings of the Algorithms (Friedman)

Algorithm	**Ranking**
SUFFUSE	$\mathbf{1.4667^{+}}$
FRSS	1.8000
Unreduced	3.3333
FRFS	3.4000

Table 2.9: Post Hoc comparison Table for $\alpha = 0.05$ (Friedman)

$\boldsymbol{i}$	**Algorithm**	$\boldsymbol{z = (r_0 - R_i)/SE}$	$\boldsymbol{p}$
3	FRFS	4.101219	0.000041
2	Unreduced	3.959798	0.000075
1	FRSS	0.707107	0.4795

average ranks obtained by each method in the Friedman test are presented in Table 2.8. As shown, SUFFUSE has gained the lowest ranking, which proves the effectiveness of the proposed method. Friedman statistic (distributed according to chi-square with 3 degrees of freedom) is 27.56, and p-value computed by Friedman test is $4 \times e^{-6}$. By referring to the post hoc comparison results in Table 2.9, the probability of FRFS and Unreduced to perform better than SUFFUSE is less $(5 \times e^{-3})\%$ and $(8 \times e^{-3})\%$, respectively. Also, the probability of FRSS to outrun SUFFUSE is less than 48%.

2.6 Application to Functional Near-Infrared Spectroscopy (fNIRS) neural signals

To show the appropriateness of the proposed methods, a real world dataset called Neural Signal is used as a benchmark dataset. The neural signal acquisition has been done by a multi channel optical brain imaging system (fNIR-300) and the levels of oxy-, deoxy- and total-haemoglobin have been specified using 16 signal channels at 2 Hz sampling rate. The signals are collected through the optical fibers, which are attached to the pre-frontal cortex. As Figure 2.11 shows, two cognitive activities of rest→right imagery movement and rest→left imagery movement have been sampled in a dataset with three classes, rest, right and left. The dataset has 280 samples and 45 features. Table 2.10 shows the average classification accuracies of applying FRFS, FRSS and SUFFUSE. It can be seen that SUFFUSE ends to higher classification accuracy comparing to unreduced, FRFS and FRSS, both by using conventional and improved mGP. The proposed classification system results 5.83% higher than the other classifiers.

2.7 Conclusion

This paper proposes a novel simultaneous fuzzy-rough feature-sample selection (SUFFUSE), and an improved multi-tree GP (mGP). The SUFFUSE selects features and samples simultaneously by coding both in a single frog of SFLA, and use fuzzy-rough positive region (FRPR) as fitness function to evaluate selected subsets. An improved mGP classifier, classifies the results of proposed methods based on the new selection

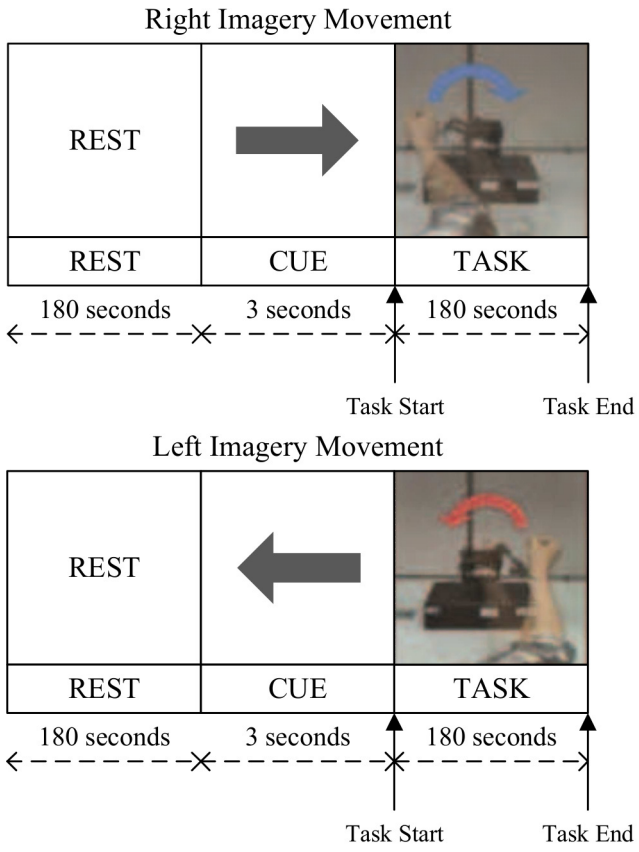


Figure 2.11: Experimental Scenarios for Acquiring fNIRS Neural Signals

Table 2.10: Average Classification Accuracies (%) of Conventional Classifiers (Part, JRip, Naive Bayes, Bayes Net, J48, BFTree, FT, NBTree and RBFNetwork) & Improved mGP for Unreduced & Reduced Neural Signal Dataset Using FRFS, FRSS & SUFFUSE

Dataset	**Unreduced**		**FRFS**		**FRSS**		**SUFFUSE**	
	Conv.	**mGP**	**Conv.**	**mGP**	**Conv.**	**mGP**	**Conv.**	**mGP**
Neural Signal	75.40	82.86	78.02	81.79	75.04	81.34	83.68	**89.51**$^{+}$

strategy, fitness function, mutation and crossover operators. Finally, the experimental results of SUFFUSE, fuzzy-rough feature selection (FRFS) and fuzzy-rough feature selection (FRSS) on fifteen UCI datasets show the effectiveness of the proposed methods, both in terms of classification accuracy and models size. As a real-world application, the proposed methods handle fNIRS neural signal dataset. It can be seen from the results that SUFFUSE and mGP have a great impact on classification accuracy comparing to independent feature and sample selections. As a future work, we are so excited to apply improved version of SFLA, and perform broad comparisons among different evolutionary algorithms.

Acknowledgment

This work has been partially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Research & Development Corporation of Newfoundland and Labrador (RDC).

Bibliography

[1] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera. Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic & Soft Computing*, 17, 2011.

[2] J. An, J. Lee, and C. Ahn. An efficient gp approach to recognizing cognitive

tasks from fnirs neural signals. *Science China Information Sciences*, 56(10):1–7, 2013.

[3] J. R. Anaraki and C. Ahn. Fuzzy-rough sample selection. In *Computer Science and Engineering (WCSE), 2014 The 4th International Workshop on*, in press.

[4] J. R. Anaraki and M. Eftekhari. Rough set based feature selection: A review. In *Information and Knowledge Technology (IKT), 2013 5th Conference on*, pages 301–306, May 2013.

[5] D. Dubois and H. Prade. Putting rough sets and fuzzy sets together. In R. Slowinski, editor, *Intelligent Decision Support*, volume 11 of *Theory and Decision Library*, pages 203–232. Springer Netherlands, 1992.

[6] M. Eusuff, K. Lansey, and F. Pasha. Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization. *Engineering Optimization*, 38(2):129–154, 2006.

[7] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18, Nov. 2009.

[8] R. Jensen and C. Cornelis. Fuzzy-rough instance selection. In *Fuzzy Systems (FUZZ), 2010 IEEE International Conference on*, pages 1–7, July 2010.

[9] R. Jensen and Q. Shen. New approaches to fuzzy-rough feature selection. *Fuzzy Systems, IEEE Transactions on*, 17(4):824–838, Aug 2009.

[10] J. Komorowski, Z. Pawlak, L. Polkowski, and A. Skowron. Rough sets: A tutorial. In S. K. Pal and A. Skowron, editors, *Rough-Fuzzy Hybridization: A New Trend in Decision Making*, pages 3–98. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1998.

[11] M. Lichman. UCI machine learning repository, 2013.

[12] N. Mac Parthaláin and R. Jensen. Simultaneous feature and instance selection using fuzzy-rough bireducts. In *Fuzzy Systems (FUZZ), 2013 IEEE International Conference on*, pages 1–8, July 2013.

[13] D. P. Muni, N. R. Pal, and J. Das. A novel approach to design classifiers using genetic programming. *Evolutionary Computation, IEEE Transactions on*, 8(2):183–196, April 2004.

[14] Z. Pawlak. Rough sets. *International Journal of Computer & Information Sciences*, 11(5):341–356, 1982.

[15] A. M. Radzikowska and E. E. Kerre. A comparative study of fuzzy rough sets. *Fuzzy Sets and Systems*, 126(2):137 – 155, 2002.

[16] B. Raman and T. R. Ioerger. Enhancing learning using feature and example selection. *Texas A&M University, College Station, TX, USA*, 2003.

[17] A. Rozsypal and M. Kubat. Selecting representative examples and attributes by a genetic algorithm. *Intelligent Data Analysis*, 7(4):291–304, 2003.

[18] Q. Shen and R. Jensen. Selecting informative features with fuzzy-rough sets and

its application for complex systems monitoring. *Pattern Recognition*, 37(7):1351 – 1363, 2004.

Chapter 3

A New Fuzzy-Rough Hybrid Merit to Feature Selection

This paper is published in Transactions on Rough Sets XX, part of the Lecture Notes in Computer Science book series, 2016.

3.1 Abstract

Feature selecting is considered as one of the most important pre-process methods in machine learning, data mining and bioinformatics. By applying pre-process techniques, we can defy the curse of dimensionality by reducing computational and storage costs, facilitate data understanding and visualization, and diminish training and testing times, leading to overall performance improvement, especially when dealing with large datasets. Correlation feature selection method uses a conventional merit to evaluate different feature subsets. In this paper, we propose a new merit by adapting and employing of correlation feature selection in conjunction with fuzzy-rough fea-

ture selection, to improve the effectiveness and quality of the conventional methods. It also outperforms the newly introduced gradient boosted feature selection, by selecting more relevant and less redundant features. The two-step experimental results show the applicability and efficiency of our proposed method over some well known and mostly used datasets, as well as newly introduced ones, especially from the UCI collection with various sizes from small to large numbers of features and samples.
Index terms— Feature selection, fuzzy-rough dependency degree, correlation merit

3.2 Introduction

Each year the amount of generated data increases dramatically. This expansion needs to be handled to minimize the time and space complexities as well as the comprehensibility challenges inherent in big datasets. Machine learning methods tend to sacrifice some accuracy to decrease running time, and to increase the clarity of the results [14].

Datasets may contain hundreds of thousand of samples with thousands of features that make further processing on data a tedious job. Reduction can be done on either features or on samples. However, due to the high cost of sample gathering and their undoubted utility, such as in bioinformatics and health systems, data owners usually prefer to keep only the useful and informative features and remove the rest, by applying Feature Selection (FS) techniques that are usually considered as a preprocessing step to further processing (such as classification). These methods lead to less classification errors or at least to minimal diminishing of performance [15].

In terms of data usability, each dataset contains three types of features: 1- infor-

mative, 2- redundant, and 3- irrelevant. Informative features are those that contain enough information on the classification outcome. In other words, they are non-redundant, relevant features. Redundant features contain identical information compared to other features, whereas irrelevant features have no information about the outcome. The ideal goal of FS methods is to remove the last two types of features [14].

FS methods can generally be divided into two main categories [21]. One approach is *wrapper* based, in which a learning algorithm estimates the accuracy of the subset of features. This approach is computationally intensive and slow due to the large number of executions over selected subsets of features, that make it impractical for large datasets. The second approach is *filter* based, in which features are selected based on their quality regardless of the results of learning algorithm. As a result, it is fast but less accurate. Also, a combinational approach of both methods called *embedded* has been proposed to accurately handle big datasets [9]. In the methods based on this approach, feature subset selection is done while classifier structure is being built.

One of the very first feature selection methods for binary classification datasets is Relief [20]. This method constructs and updates a weight vector of a feature, based on the nearest feature vector of the same and different classes using Euclidean distance. After a predefined number of iterations l, relevant vector is calculated by dividing the weight vector by l, and the features with relevancy higher than a specific threshold will be selected. Hall [14] has proposed a merit based on the average intra-correlation of features and inter-correlation of features and the outcome. Those features with higher correlation to the outcome and lower correlation to other features are selected.

Jensen et al. [16] have introduced a novel feature selection method based on lower approximation of the fuzzy-rough set, in which features and outcome dependencies are calculated using a merit called Dependency Degree (DD). In [3], two modifications of the fuzzy-rough feature selection have been introduced to improve the performance of the conventional method: 1- Encompassing the selection process in *equal* situations, where more than one feature result in an identical fitness value by using correlation merit [14] and 2- Combining the first improvement with the stopping criterion [2]. Qian et al. [27], have proposed an accelerator to perform sample and feature selection simultaneously in order to improve the time complexity of fuzzy-rough feature selection. Jensen et al. [17] have designed a new version of fuzzy-rough feature selection to deal with semi-supervised datasets, in which class feature is partially labelled. Shang et al. [29] have introduced a hybrid system for Mars images based on conjunction of fuzzy-rough feature selection and support vector machines. The behaviour of k-nearest neighbours classifier has been improved by Derrac et al. [10], using fuzzy rough feature selection and steady-state genetic algorithm for both feature and prototype selection. Dai et al. [8], have designed a system using fuzzy information gain ratio based on fuzzy rough feature selection structure to classify tumor data in gene expression.

Xu et al. [33] have proposed a non-linear feature selection method based on gradient boosting of limited depth trees. This method combines classification and feature selection processes into one by using gradient boosted regression trees resulting from the greedy CART algorithm.

In this paper, we propose a new merit, which is not only capable of effectively removing redundant features, selecting relevant ones, and enhancing the classification

accuracy, but it also outperforms when applied to large datasets, compared to the other existing methods.

In Section 2, background and preliminaries of correlation based and fuzzy-rough feature selection methods are described in detail. Our proposed method is discussed in Section 3. Section 4 is dedicated to experimental results and discussion on the performance and effectiveness of the new approach comparing with previously introduced methods. Conclusions and future directions are explained in Section 5.

3.3 Preliminaries

In this section, the idea and explanation of the Correlation-based Feature Selection (CFS) method will be presented in 3.3.1. Subsection 3.3.2 illustrates the rough set theory and the rough set based feature selection approach.

3.3.1 Correlation based Feature Selection (CFS)

In the feature selection process, selecting those features that are highly correlated with the class attribute while loosely correlated with the rest of the features, is the ultimate goal. One of the most successful feature selection methods based on this is CFS [14]. The evaluation measure of CFS is designed in such a way that it selects predictive and low level inter-correlated features on the class and other features, respectively. Equation 3.1 shows the merit.

$$Merit_S = \frac{k\overline{r}_{cf}}{\sqrt{k + k(k-1)\overline{r}_{ff}}}, \tag{3.1}$$

where S is a subset of features, k is the number of selected features, $\overline{r}_{cf}$ is the mean of the correlations of the selected features to the class attribute, and $\overline{r}_{ff}$ is the average of inter-correlations of features. The enumerator calculates how much the selected subset is correlated with the class, and the denominator controls the redundancy of selected features within the subsets. At the heart of the merit, correlation undeniably plays the most important role. Therefore, maximizing merit requires the most relevant features (to maximize the numerator) and the least redundant ones (to minimize the denominator) to be included in the subset. The relevancy and non-redundancy are two important factors in feature selection that are handled in CFS. However, correlation is only capable of measuring linear relationships of two vectors [34]; therefore, in the case of non-linear relationships, the result will be inaccurate.

3.3.2 Rough Set Feature Selection

The rough set theory has been proposed by Pawlak that is a mathematical tool to handle vagueness in effective way [25]. Suppose U and A to be the universe of discourse and a nonempty set of attributes, respectively, and the information system is presented by $I = (U, A)$. Consider X as a subset of U, and P and Q as subsets of A; approximating a subset in rough set theory is done through the lower and upper approximations. The lower approximation of X, $(\underline{P}X)$ involves those objects which are surely classified in X with regarding to attributes in P. Whereas, upper approximation of X, $(\overline{P}X)$ accommodates those objects which can possibly classified in X considering attributes of P. By defining the lower and upper approximations, a rough set is shown using an ordered pair $(\underline{P}X, \overline{P}X)$. Based on these approximations,

different regions in rough set theory is illustrated by Equations 4.1, 3.3 and 3.4.

The union of all objects in different regions of $\mathbb{U}$ partitioned by Q with regarding to P is called positive region $POS_P(Q)$.

$$POS_P(Q) = \bigcup_{X \in \mathbb{U}/Q} \underline{P}X \tag{3.2}$$

The negative region is collection of object that are in $\mathbb{U}$ but not in $POS_P(Q)$, and is shown by $NEG_P(Q)$ [22].

$$NEG_P(Q) = \mathbb{U} - \bigcup_{X \in \mathbb{U}/Q} \overline{P}X \tag{3.3}$$

The boundary region has determinative role in specifying the type of a set. If the region is a non-empty set, it is called a rough set, otherwise, it is a crisp set.

$$BND_P(Q) = \bigcup_{X \in \mathbb{U}/Q} \overline{P}X - \bigcup_{X \in \mathbb{U}/Q} \underline{P}X \tag{3.4}$$

The rough set theory can be used to measure the magnitude of dependency between attributes. The dependency of attributes in Q on attribute(s) in P is shown by $P \Rightarrow_k Q$, in which k equals to $\gamma_P(Q)$ and it is labeled Dependency Degree (DD) [22]. If $0 < k < 1$, then Q partially depends on P, otherwise if $k = 1$ then Q completely depends on P. Equation 4.2 calculates the DD of Q on P.

$$k = \gamma_P(Q) = \frac{|POS_P(Q)|}{|\mathbb{U}|}, \tag{3.5}$$

where notation $|.|$ is number of objects in a set.

The reduct set is a subset of features which has identical DD as considering all features. The members of the reduct set are the most informative features which fea-

ture outcome is highly dependent on them, while non-members are irrelevant and/or redundant ones.

The most important drawback of rough set based FS methods is their incapability of handing continuous data. One way to govern this imperfection is to discretize continuous data in advance that is necessary but not enough, as long as the amount of similarity between discretized data is unspecified. The ultimate way to handle continuous data using rough set theory is fuzzy-rough set. To begin with, the definition of the X-lower and X-upper approximations and the degree of fuzzy similarity [16] are given by Equations 4.3 to 4.4, respectively

$$\mu_{\underline{P}X}(x) = \inf_{y \in \mathbb{U}} I\{\mu_{R_P}(x, y), \mu_X(y)\}, \tag{3.6}$$

$$\mu_{\overline{P}X}(x) = \sup_{y \in \mathbb{U}} T\{\mu_{R_P}(x, y), \mu_X(y)\}, \tag{3.7}$$

$$\mu_{R_P}(x, y) = \bigcap_{a \in P} \{\mu_{R_a}(x, y)\}, \tag{3.8}$$

where I is a Łukasiewicz fuzzy *implicator*, which is defined by $min(1 - x + y, 1)$, and T is a Łukasiewicz fuzzy t-norm, which is defined by $max(x + y - 1, 0)$. In [28], three classes of fuzzy-rough sets based on three different classes of implicators, namely S-, R-, and QL-implicators, and their properties have been investigated. Here, R_P is the fuzzy similarity relation considering the set of features in P, and $\mu_{R_P}(x, y)$ is the degree of similarity between objects x and y over all features in P. Also, $\mu_X(y)$ is the membership degree of y to X. One of the best fuzzy similarity relations as suggested

in [16] is given by Equation 4.5.

$$\mu_{R_a}(x,y) = max\left\{min\left\{\frac{(a(y)-(a(x)-\sigma_a))}{\sigma_a},\ \frac{((a(x)+\sigma_a)-a(y))}{\sigma_a}\right\},0\right\} \quad (3.9)$$

where σ_a is variance of feature a. Definitions of fuzzy lower and upper approximations are the same as rough lower and upper approximations, except the fact that fuzzy approximations deal with fuzzy values, operators, and output; however, rough approximations deal with discrete and categorical values.

The positive region in the rough set theory is defined as a union of lower approximations. By referring to the extension principle [16], the membership of object x to a fuzzy positive region is given by Equation 4.6.

$$\mu_{POS_{P(Q)}}(x) = \sup_{X\in\mathbb{U}/Q} \mu_{\underline{P}X}(x) \quad (3.10)$$

where supremum of lower approximations of all partitions resulting from U/Q construct positive region.

If the equivalence class that includes x does not belong to a positive region, clearly x will not be part of a positive region. Using the definition of positive region, the FRDD function is defined as:

$$\gamma'_P(Q) = \frac{|\mu_{POS_{P(Q)}}(x)|}{|\mathbb{U}|} = \frac{\sum_{x\in\mathbb{U}} \mu_{POS_P(Q)}(x)}{|\mathbb{U}|} \quad (3.11)$$

where notation $|.|$ is number of objects in a set; however, in numerator we are dealing with fuzzy values and cardinality can be calculated using summation. For denominator $|\mathbb{U}|$ is size of samples in dataset.

The Lower approximation Fuzzy-Rough Feature Selection (L-FRFS) as shown in Algorithm 3.1 is based on FRDD as shown in Equation 4.7, and greedy forward search algorithm, which is capable of being applied to real-valued datasets. The L-FRFS algorithm finds a reduct set without finding all the subsets [16]. It begins with an empty set and each time selects the feature that causes the greatest increase in the FRDD. The algorithm stops when adding more features does not increase the FRDD. Since it employs a greedy algorithm, it does not guarantee that the minimal reduct set will be found. For this reason, a new feature selection merit presented in this section.

3.4 Proposed method

On the one hand, FRDD is capable and effective in uncovering the dependency of a feature to another, and the feature selection method based on the merit has shown remarkable performance on resulting classification accuracies [16]. The L-FRFS algorithm evaluates every feature to find the one with the highest dependency, and continues the search by considering every features combination to asset the most dependent features subset to the outcome. However, tracking and finding highly dependent features to the class might end in selecting redundant features.

On the other hand, CFS merit, as shown in Equation 3.1, has the potentiality of selecting less redundant features due to the structure of the denominator, in which the square root of the mean of the correlation of the features to each other has a positive impact on the number of redundant features being selected.

By considering capabilities of CFS merit, substituting the correlation with Fuzzy-

Algorithm 3.1: Lower approximation Fuzzy-Rough Feature Selection

C, the set of all conditional attributes;

D, the set of decision attributes;

$R \leftarrow \{\}; \gamma'_{best} = 0; \gamma'_{prev} = 0;$

do

  $T \leftarrow R;$

  $\gamma'_{prev} \leftarrow \gamma'_{best};$

  for $x \in (C - R)$ **do**

    if $\gamma'_{R \cup \{x\}}(D) > \gamma'_{T}(D)$ **then**

      $T \leftarrow R \cup \{x\};$

      $\gamma'_{best} \leftarrow \gamma'_{T}(D);$

    $R \leftarrow T;$

while $\gamma'_{best} = \gamma'_{prev};$

return $R;$

Rough Dependency Degree (FRDD) that is fuzzy version of DD could take advantage of both criteria to construct a more powerful merit. In this section, the proposed approach is defined based on the two main concepts of feature selection: 1- Evaluation measure, and 2- Search method. The evaluation measure is the new hybrid merit and the search method is hill-climbing.

3.4.1 A New Hybrid Merit

Based on the concepts of the FRDD and CFS, we have developed a new hybrid merit by substituting the correlation in CFS with FRDD to benefit from both merits. Equation 3.12 shows the proposed merit.

$$\delta = \frac{\sum_{i=1}^{k} \gamma'_i(c)}{\sqrt{k \times \left(1 + \sum_{j=1}^{k-1} \gamma'_j(f)\right)}}, \tag{3.12}$$

where $\gamma'_i(c)$ is the FRDD of already selected feature i to the class c, and $\gamma'_j(f)$ is the FRDD of selected feature j to the new under consideration candidate feature f. The numerator is summation of the FRDD of already selected $k-1$ features as well as newly selected k's feature to the outcome, while the summation in denominator is aggregation of the FRDD of all features except currently under consideration one k's, to itself. It is worth to mention that k in denominator controls the number of selected features. We call the feature selection method based on our proposed merit, Delta Feature Selection (DFS). The numerator can vary from zero to one for each k (since $\gamma'_i \in [0,1]$), so we have interval of $[0,k]$ in the numerator. However, summation in the denominator varies from zero to $k-1$ for each k, and the whole portion changes in interval of $[\sqrt{k},k]$ since k is always positive.

The search algorithm of our proposed, that is a greedy forward search method shown in Algorithm 3.2. The QuickReduct algorithm starts from an empty subset and each time selects one feature to be added to the subset, if the selected feature causes the highest increase in δ; therefore, it will be added to the subset, otherwise, the algorithm evaluates next feature. This process will be continued until no more

feature can improve the δ.

Algorithm 3.2: Delta QuickReduct (DQR)

Result: S_f: best subset of features

δ'_{curr}: current DFS;

δ'_{prev}: previous DFS;

nF: number of features ;

bF: best feature ;

$S_f = \{\}$;

$\delta_{curr}, \delta_{prev} = 0$;

do

  $\delta_{prev} = \delta_{curr}$;

  for $i = 1$ *to* $i \leq nF$ **do**

    if $\Big((f_i \notin S_f)\ AND\ (\delta_{S_f \cup \{f_i\}} > \delta_{prev})\Big)$ **then**

      $\delta_{curr} = \delta_{S_f \cup \{f_i\}}$;;

      $bF = f_i$;;

  $S_f = S_f \cup bF$;;

while $(\delta_{curr} != \delta_{prev})$;

return S_f;;

To evaluate the applicability of the proposed merit to different types of datasets, a series of criteria have been considered as follows [5]:

1. Correlated and redundant features

2. Non-linearity of data

3. Noisy input

4. Noisy target

5. Small ratio of samples/features

6. Complex datasets

Based on each criteria, thirteen datasets have been adopted from different papers as mentioned in [5] to examine the appropriateness of DFS. Datasts are shown in Table 3.1. The last column depicts corresponding criterion to the current dataset.

CorrAL dataset has six features, and features one to four are relevant and they generate the outcome by calculating $(f_1 \wedge f_2) \vee (f_3 \wedge f_4)$, feature five is irrelevant, and feature six has 75% of correlation to the outcome. CorrAL-100 has 99 features that the first six are exactly the same as CorrAL, and the rest are irrelevant and randomly assigned. For both datasets, DFS was able to uncover all four relevant features and also the correlated one.

XOR-100 dataset is a non-linear dataset with two relevant features that compute the output by calculating $(f_1 \oplus f_2)$, and the other 97 features are irrelevant. Again, DFS was able to find two relevant features.

Led-25 dataset is composed of seven relevant features and 17 irrelevant ones. Each dataset, contains the amount of noise (i.e. replacing zero with one or vice versa) that is mentioned in parenthesis in front of dataset. Based on the resulting subsets containing two relevant features for all cases, of applying DFS it can be understood that DFS cannot perform well for datasets with noisy inputs.

Table 3.1: Sample datasets to probe different capabilities of a feature selection method

Dataset	**#Relevant**	**#Irrelevant**	**#Correlated**	**Criteria**
CorrAL [18]	4	1	6	1
CorrAL-100 [19]	4	94	1	1
XOR-100 [19]	2	97	-	2
Led-25 [6] (2%)	7	17	-	3
Led-25 [6] (6%)	7	17	-	3
Led-25 [6] (10%)	7	17	-	3
Led-25 [6] (15%)	7	17	-	3
Led-25 [6] (20%)	7	17	-	3
Monk3 [32]	3	3	-	4
SD1 [35]	FCR = 20	4000	-	5
SD2 [35]	FCR = 10, PCR = 30	4000	-	5
SD3 [35]	PCR = 60	4000	-	5
Madelon	5	480	15	6

Monk3 dataset has 5% of misclassification values as a dataset with noisy target. The DFS has selected features one and five that are irrelevant and relevant, respectively. Therefore, DFS was not able to find all relevant features and also has been misled by noisy target.

SD1, SD2 and SD3 datasets each has three classes, and 75 samples, containing both relevant and irrelevant features. Relevant ones are generated based on a normal distribution, and irrelevant features have been generated based on two distributions

namely, normal distribution with mean zero and variance one, and uniform distribution in interval of $[-1, 1]$, each 2000 features. All cancer types can be distinguished by using some genes (or features) called full class relevant (FCR). However, the other genes that are helpful in contrasting some portion of cancer types are called partial class relevant (PCR). Table 3.2 shows the optimal subset for each dataset, in which nine features out of 10 are redundant features.

Table 3.2: Optimal features and subsets of SD1, SD2, and SD3

Dataset	**#Optimal features/subset**	**Optimal subsets**
SD1 [35]	2	{1-10} {11-20}
SD2 [35]	4	{1-10} {11-20} {21-30} {31-40}
SD3 [35]	6	{1-10} {11-20} {21-30} {31-40} {41-50} {51-60}

The DFS has selected 2, 11, and 2 features for SD1, SD2, and SD3, respectively. For SD1, the DFS has selected one feature from the second optimal subset, and one feature from 4000 irrelevant features. For SD2, 11 features have been selected, in which, 10 of them are from the second optimal subset and one feature from 4000 irrelevant features. Finally, two features have been selected from SD3 that one of them is from the third optimal subset and the other one is from irrelevant features.

Madelon dataset has five relevant, 15 linearly correlated to relevant features, and 480 distractor features that are noisy, flipped and shifted [5]. The DFS was able to find five features, in which none of them were among relevant features.

Based on the resulting subsets, our proposed method is capable of dealing with

datasets having characteristics mentioned in Table 3.3.

Table 3.3: DFS capabilities

Dataset	**DFS capability**
Correlated and redundant features	✓
Nonlinearity of data	✓
Noisy input	depends on data
Noisy target	depends on data
Small ratio of samples/features	✓
Complex datasets	×

For datasets with noisy input and target, the DFS was capable of finding a subset of relevant features; however, for complex datasets such as Madelon, finding relevant features is very challenging for DFS and many state-of-art feature selection methods [5].

3.4.2 Performance Measures

In order to evaluate the applicability and performance of FS methods, we define three *Performance* measures to underline classification accuracy and/or reducibility power. The *Reduction* ratio is the value of reduction of total number of features resulting from applying a feature selection method to a datasets, and it is shown in Equation

3.13.

$$Reduction = \frac{all\_F - sel\_F}{all\_F}, \tag{3.13}$$

where all_F is the number of all features, and sel_F is the number of selected features using a feature selection algorithm.

The *Performance* measure is a metric to evaluate the effectiveness of a feature selection algorithm in selecting the smallest subset of features as well as improving the classification accuracy, and is shown by Equation 3.14.

$$Performance = CA \times Reduction, \tag{3.14}$$

where CA is the classification accuracy.

Since the primary aim of FS is to select the smallest meaningful subset of features, we propose a revision of *Performance* measure that emphasizes on the *Reduction* capability of each method and it is presented by Equation 3.15.

$$Performance' = CA \times e^{Reduction}, \tag{3.15}$$

In some cases, data owners prefer those FS methods that lead to higher accuracies; therefore, another revision of Equation 3.14 with the aforementioned preference is depicted by Equation 3.16.

$$Performance'' = e^{CA} \times Reduction. \tag{3.16}$$

3.5 Experimental Results

To validate the proposed method, we have conducted a number of experiments in two steps over 25 UCI [4] traditional as well as newly introduced datasets from three different size categories; Small (S), Medium (M) and Large (L) sizes, in which the number of selected features, *Reduction* ratio, classification accuracy and *Performance* measures are compared. The small size category contains datasets with model size, i.e. $|Features| \times |Samples|$, less than 5 000, the medium size category contains 5 000 to 50 000 cells, and each dataset in the large size category has more than 50 000 cells.

In our experiments the L-FRFS, CFS , and DFS use the same search method called greedy froward search algorithm, and the GBFS uses gradient decent search method.

Computational facilities are provided by ACENET, the regional high performance computing consortium for universities in Atlantic Canada. ACEnet is funded by the Canada Foundation for Innovation (CFI), the Atlantic Canada Opportunities Agency (ACOA), and the provinces of Newfoundland and Labrador, Nova Scotia, and New Brunswick.

3.5.1 Step One

In this step, we consider all the 25 datasets in our experiment. Table 7.1 shows the number of samples, features and the size category that each dataset belongs to, and it is sorted based on the model size.

Based on the number of selected features and Equation 3.13, the *Reduction* ratio of each method has been calculated and illustrated in Table 3.5. The cells with zero

Table 3.4: Datasets Specifications

Dataset	**Sample**	**Feature**	**Size**
BLOGGER	100	5	S
Breast Tissue	122	9	S
Qualitative Bankr.	250	6	S
Soybean	47	35	S
Glass	214	9	S
Wine	178	13	S
MONK1	124	6	S
MONK2	169	6	S
MONK3	122	6	S
Olitus	120	26	S
Heart	270	13	S
Cleveland	297	13	S
Pima Indian Diab.	768	8	M
Breast Cancer	699	9	M
Thoracic Surgery [36]	470	17	M
Climate Model [23]	540	18	M
Ionosphere	351	33	M
Sonar	208	60	M
Wine Quality (Red) [7]	1599	11	M
LSVT Voice Rehab. [31]	126	310	M
Seismic Bumps [30]	2584	18	M
Arrhythmia	452	279	L
Molecular Biology	3190	60	L
COIL 2000 [26]	5822	85	L
Madelon	2000	500	L

indicate that the feature selection method could not remove any feature; therefore, all of the features remain untouched.

The bold, superscripted numbers specify the best method in improving the *Reduction* ratio. L-FRFS and GBFS reaches the highest reduction ratio for four datasets, CFS for five datasets, and DFS outperforms the others by gaining the highest *Reduction* values for sixteen datasets. Based on the categories and number of successes of each method, L-FRFS and GBFS result almost similar on the small size category with two and one out of 12 datasets, respectively. However, DFS highly achieves the best results in both medium and large datasets, by having six out of nine best reduction ratios in medium size category compare to two out of nine for L-FRFS and GBFS methods, and one out of all for CFS. For large datasets, DFS gains 100% domination. Table 3.6, shows the number of wins of each method in three categories.

Arithmetic mean has some disadvantages, such as high sensitivity to outliers and also inappropriateness in measuring central tendency of skewed distribution [24], we have conducted the Friedman test that is a non-parametric statistical analysis [1] on the results of Tables 3.8, 3.11, 3.14, and 3.17 to make the comparison fare enough.

The nine classifiers are PART, Jrip, Naïve Bayes, Bayes Net, J48, BFTree, FT, NBTree, and RBFNetwork that have been selected from different classifier categories to evaluate the performance of each method by applying 10-fold cross validation (10CV). These classifiers have been implemented in Weka [13], and mean of resulting classification accuracies of all selected classifiers have been used through out the paper. By considering selected features for each dataset, the resulting average of classification accuracies have been shown in Table 4.8.

By referring to the results in Table 3.5, Table 4.8, and applying Equation 3.14, the

Table 3.5: *Reduction* ratio of L-FRFS, CFS, DFS & GBFS

Datasets	**L-FRFS**	**CFS**	**GBFS**	**DFS**	**Size**
BLOGGER	0.000	0.400	0.400	**0.600**$^+$	S
Breast Tissue	0.000	0.333	**0.444**$^+$	0.111	S
Qualitative Bankr.	0.500	0.333	0.167	**0.667**$^+$	S
Soybean	**0.943**$^+$	0.743	0.886	**0.943**$^+$	S
Glass	0.000	0.111	0.333	**0.556**$^+$	S
Wine	0.615	0.154	0.692	**0.846**$^+$	S
Monk1	0.500	**0.833**$^+$	0.333	0.667	S
Monk2	0.000	**0.833**$^+$	0.167	0.667	S
Monk3	0.500	**0.833**$^+$	0.333	0.667	S
Olitus	**0.808**$^+$	0.346	0.731	0.231	S
Heart	0.462	0.462	0.538	**0.846**$^+$	S
Cleveland	0.154	**0.923**$^+$	0.538	0.846	S
Pima Indian Diab.	0.000	**0.500**$^+$	0.250	**0.500**$^+$	M
Breast Cancer	0.222	0.000	**0.444**$^+$	**0.444**$^+$	M
Thoracic Surgery	0.176	0.706	0.588	**0.882**$^+$	M
ClimateModel	0.667	0.833	**0.944**$^+$	0.889	M
Ionosphere	0.788	0.576	0.818	**0.909**$^+$	M
Sonar	**0.917**$^+$	0.683	0.900	0.050	M
Wine Quality (Red)	0.000	0.636	0.636	**0.727**$^+$	M
LSVT Voice Rehab.	**0.984**$^+$	0.900	0.977	0.923	M
Seismic Bumps	0.278	0.667	0.778	**0.889**$^+$	M
Arrhythmia	0.975	0.910	0.907	**0.993**$^+$	L
Molecular Biology	0.000	0.617	0.000	**0.950**$^+$	L
COIL 2000	0.659	0.882	0.941	**0.965**$^+$	L
Madelon	0.986	0.982	**0.990**$^+$	**0.990**$^+$	L

Table 3.6: Number of wins in achieving the lowest *Reduction* ratio for L-FRFS, CFS, GBFS, and DFS in each category

Algorithm/Category	**Small**	**Medium**	**Large**	**Overall**
L-FRFS	2	2	0	4
CFS	4	1	0	5
GBFS	1	2	1	4
DFS	6	6	4	16

Performance measure of each method has been computed and shown in Table 3.8. The cells that contain zero are the ones with *Reduction* ratio equal to zero. Based on the results shown in Tables 3.5 and 3.8, DFS outperforms the other methods by having the the best results for 10 datasets compared to that by GBFS with seven, CFS with six, and L-FRFS with only three cases. The best performance for small sized datasets has been achieved by DFS and CFS, for medium datasets by DFS and GBFS and for large datasets by DFS. Table 3.9 evaluates the results of Table 3.8, and Friedman statistic (distributed according to chi-square with 3 degrees of freedom) is 11.772, and p-value computed by Friedman Test is 0.008206. Based on the rankings, the DFS has gained the best ranking among others; however, its distinction has been examined by post-hoc experiment. The post-hoc procedure as depicted in Table 3.10 rejects those hypotheses with p-value ≤ 0.030983. So, as shown, DFS and GBFS perform nearly identical. Since performances of DFS and GBFS are not statistically significant, the one with the lowest reduction ratio is selected [12]. Here, based on Table 3.6, the DFS is ranked the best method among others.

Table 3.7: Mean of classification accuracies in % resulting from PART, Jrip, Naïve Bayes, Bayes Net, J48, BFTree, FT, NBTree, and RBFNetwork based on L-FRFS, CFS, GBFS, DFS performance comparing with unreduced datasets

Datasets	**L-FRFS**	**CFS**	**GBFS**	**DFS**	**Unre.**
BLOGGER	**74.22**$^+$	73.78	73.78	73.56	**74.22**$^+$
Breast Tissue	**66.46**$^+$	66.35	64.88	65.72	**66.46**$^+$
Qualitative Bankr.	98.44	98.04	98.31	98.40	**98.49**$^+$
Soybean	**100.00**$^+$	75.48	97.87	95.98	98.58
Glass	**67.29**$^+$	66.93	65.42	59.71	61.89
Wine	**95.63**$^+$	95.44	94.63	74.22	85.52
Monk1	**83.13**$^+$	74.07	81.94	73.53	78.32
Monk2	-	67.13	71.89	67.13	**76.62**$^+$
Monk3	98.15	76.23	**98.28**$^+$	75.62	97.92
Olitus	66.39	**75.65**$^+$	53.8	72.69	69.81
Heart	78.48	**81.48**$^+$	81.4	71.32	79.55
Cleveland	49.76	**54.88**$^+$	52.19	54.55	50.13
Pima Indian Diab.	75.00	**75.20**$^+$	**75.20**$^+$	**75.20**$^+$	75.00
Breast Cancer	**96.23**$^+$	96.18	96.23	95.31	96.18
Thoracic Surgery	83.03	84.54	83.95	**85.11**$^+$	83.10
Climate Model	93.25	90.74	91.38	91.36	**93.54**$^+$
Ionosphere	**91.39**$^+$	90.85	89.97	84.96	89.68
Sonar	69.82	**75.48**$^+$	74.89	74.36	67.47
Wine Quality (Red)	58.59	**59.22**$^+$	58.59	56.54	58.59
LSVT Voice Rehab.	**80.60**$^+$	79.37	75.84	72.57	74.69
Seismic Bumps	91.16	91.96	**92.59**$^+$	51.87	91.13
Arrhythmia	53.74	**70.48**$^+$	63.20	59.41	65.46
Molecular Biology	-	73.66	-	51.69	**94.58**$^+$
COIL 2000	92.79	93.65	93.97	**94.02**$^+$	90.61
Madelon	65.79	69.57	**71.27**$^+$	55.26	66.32

Table 3.8: *Performance* measure resulting from Classification Accuracy × *Reduction*

Datasets	**L-FRFS**	**CFS**	**GBFS**	**DFS**
BLOGGER	0.000	0.295	0.295	**0.441**$^{+}$
Breast Tissue	0.000	0.221	**0.288**$^{+}$	0.073
Qualitative Bankr.	0.492	0.327	0.164	**0.656**$^{+}$
Soybean	**0.943**$^{+}$	0.561	0.867	0.905
Glass	0.000	0.074	0.218	**0.332**$^{+}$
Wine	0.588	0.147	**0.655**$^{+}$	0.628
Monk1	0.416	**0.617**$^{+}$	0.273	0.490
Monk2	0.000	**0.559**$^{+}$	0.120	0.448
Monk3	0.491	**0.635**$^{+}$	0.328	0.504
Olitus	**0.536**$^{+}$	0.262	0.393	0.168
Heart	0.362	0.376	0.438	**0.603**$^{+}$
Cleveland	0.077	**0.507**$^{+}$	0.281	0.462
Pima Indian Diab.	0.000	**0.376**$^{+}$	0.188	**0.376**$^{+}$
Breast Cancer	0.214	0.000	**0.428**$^{+}$	0.424
Thoracic Surgery	0.147	0.597	0.494	**0.751**$^{+}$
ClimateModel	0.622	0.756	**0.863**$^{+}$	0.812
Ionosphere	0.720	0.523	0.736	**0.772**$^{+}$
Sonar	0.640	0.516	**0.674**$^{+}$	0.037
Wine Quality (Red)	0.000	0.377	0.373	**0.411**$^{+}$
LSVT Voice Rehab.	**0.793**$^{+}$	0.714	0.741	0.670
Seismic Bumps	0.253	0.613	**0.720**$^{+}$	0.461
Arrhythmia	0.524	**0.642**$^{+}$	0.573	0.590
Molecular Biology	0.000	0.454	0.000	**0.491**$^{+}$
COIL 2000	0.611	0.826	0.884	**0.907**$^{+}$
Madelon	0.649	0.683	**0.706**$^{+}$	0.547

Table 3.9: Average rankings of the algorithms based on the *Performance* measure over all datasets (Friedman)

Algorithm	**Ranking**
L-FRFS	3.220
CFS	2.440
GBFS	2.320
DFS	**2.020**$^{+}$

Table 3.10: Post Hoc comparison over the results of Friedman procedure of *Performance* measure

i	**Algorithm**	$z = (R_0 - R_i)/SE$	p	**Li**
3	L-FRFS	3.286335	0.001015	0.030983
2	CFS	1.150217	0.250054	0.030983
1	GBFS	0.821584	0.411314	0.05

3.5.2 Step Two

Since the CFS has chosen only one feature for MONK1, MONK2, MONK3 and Cleveland, and also GBFS has selected one out of 18 of Climate Model as the most important feature, further investigations is vital on these suspicious results. The Cleveland dataset has 75 features whereas 13 features out of 75 have been suggested to be used by the published experiments [11]; therefore, all of these 13 features are important from the clinical perspective. By referring to the result of CFS, feature "sex" has been selected as the only important feature due to its highest correlation with the outcome. Neither experts in medical science nor in computer science would arrive at the point that one feature (regardless of type of the feature) out of 13 is enough to predict the outcome. Although selecting "sex" results in the highest classification accuracy, the interpretability of selecting one feature is questionable. Therefore, although "sex" might be an important factor in predicting heart diseases, it is not the only one. For MONK1, MONK2, MONK3 and Climate Model datasets, the only characteristic of the selected feature is its high correlation with the outcome, and very low correlation with the other features.

By removing Cleveland, MONK1, MONK2, MONK3 and Climate Model from Table 3.8, we form Table 3.11 and Figure 3.1 in which DFS gains the best performance. The GBFS works slightly better than the L-FRFS and CFS for medium datasets, but identical in small datasets. While DFS performance surpasses the GBFS, CFS, and L-FRFS for all three categories. The overall effectiveness and capability of DFS is supported by both Table 3.11, and the statistical analysis in Table 3.12. The Friedman statistic (distributed according to chi-square with 3 degrees of freedom) is

9.345, and the p-value computed by Friedman Test is 0.025039. The Li's procedure rejects those hypotheses with p-value ≤ 0.01266, and the results are shown in Table 3.13. The *Performance* measures resulting form Equation 3.15 and 3.16 are shown in Tables 3.14 and 3.17 and also in Figures 3.2 and 3.3, respectively. The Friedman test results are shown in Table 3.15 and 3.18. For *Performance*$'$, those hypotheses with p-value ≤ 0.00257 are rejected based on Li's procedure, and the results are depicted in Table 3.16. For *Performance*$''$ as Table 3.19 shows, those hypotheses with p-value ≤ 0.01266 are rejected based on Li's procedure. Figures 1, 2 and 3 depict *Performance*, *Performance*$'$ and *Performance*$''$ measures values for each dataset, respectively.

3.6 Conclusions and future work

This paper introduces a new hybrid merit based on conjunction of correlation feature selection and fuzzy-rough feature selection. It takes advantages of both methods by integrating them into a new hybrid merit to improve the quality of the selected subsets as well as resulting reasonable classification accuracies. The new merit selects less number of redundant features, and finds the most relevant ones to the outcome.

The performance of the proposed merit is examined with a variety of different datasets with diverse number of features and samples, that have been chosen because of their predominance as well as recently introduced in the literature. The two-step experimental results show the effectiveness of our new hybrid merit over divergent UCI datasets, especially on medium and large ones. We have also proposed three measures to thoroughly figure out and compare the performance of feature selection

Table 3.11: *Performance* measure resulting from classification accuracy × *reduction* after removing Cleveland, MONK1, MONK2, MONK3 & Climate Model

Datasets	**L-FRFS**	**CFS**	**GBFS**	**DFS**
BLOGGER	0.000	0.295	0.295	**0.441**$^{+}$
Breast Tissue	0.000	0.221	**0.288**$^{+}$	0.073
Qualitative Bankr.	0.492	0.327	0.164	**0.656**$^{+}$
Soybean	**0.943**$^{+}$	0.561	0.867	0.905
Glass	0.000	0.074	0.218	**0.332**$^{+}$
Wine	0.588	0.147	**0.655**$^{+}$	0.628
Olitus	**0.536**$^{+}$	0.262	0.393	0.168
Heart	0.362	0.376	0.438	**0.603**$^{+}$
Pima Indian Diab.	0.000	**0.376**$^{+}$	0.188	**0.376**$^{+}$
Breast Cancer	0.214	0.000	**0.428**$^{+}$	0.424
Thoracic Surgery	0.147	0.597	0.494	**0.751**$^{+}$
Ionosphere	0.720	0.523	0.736	**0.772**$^{+}$
Sonar	0.640	0.516	**0.674**$^{+}$	0.037
Wine Quality (Red)	0.000	0.377	0.373	**0.411**$^{+}$
LSVT Voice Rehab.	**0.793**$^{+}$	0.714	0.741	0.670
Seismic Bumps	0.253	0.613	**0.720**$^{+}$	0.461
Arrhythmia	0.524	**0.642**$^{+}$	0.573	0.590
Molecular Biology	0.000	0.454	0.000	**0.491**$^{+}$
COIL 2000	0.611	0.826	0.884	**0.907**$^{+}$
Madelon	0.649	0.683	**0.706**$^{+}$	0.547

Table 3.12: Average rankings of the algorithms based on the *Performance* measure after removing Cleveland, MONK1, MONK2, MONK3 & Climate Model (Friedman)

Algorithm	**Ranking**
L-FRFS	3.125
CFS	2.700
GBFS	2.150
DFS	**2.025**$^{+}$

Table 3.13: Post Hoc comparison over the results of Friedman procedure of *Performance* measure after removing Cleveland, MONK1, MONK2, MONK3 & Climate Model

i	**Algorithm**	$z = (R_0 - R_i)/SE$	p	**Li**
3	L-FRFS	2.694439	0.007051	0.01266
2	CFS	1.653406	0.098248	0.01266
1	GBFS	0.306186	0.759463	0.05

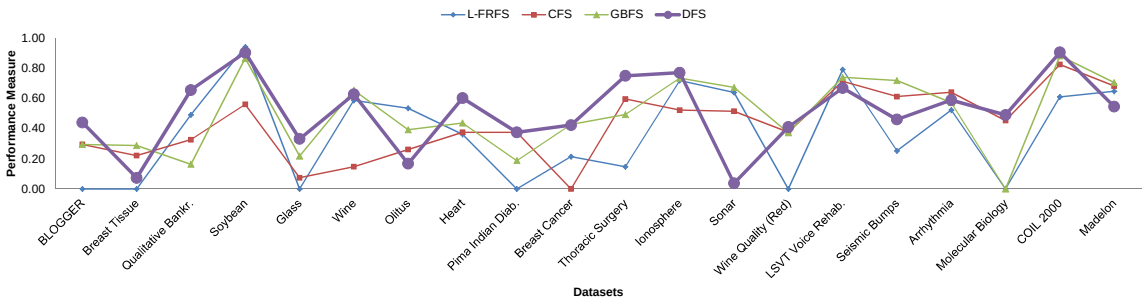


Figure 3.1: *Performance* measure (Classification Accuracy × *Reduction*)

Table 3.14: *Performance′* measure resulting from Classification Accuracy $\times e^{Reduction}$

Datasets	**L-FRFS**	**CFS**	**GBFS**	**DFS**
BLOGGER	0.742	1.101	1.101	**1.340**$^{+}$
Breast Tissue	0.665	0.926	**1.012**$^{+}$	0.734
Qualitative Bankr.	1.623	1.368	1.161	**1.917**$^{+}$
Soybean	**2.567**$^{+}$	1.587	2.373	2.464
Glass	0.673	0.748	0.913	**1.041**$^{+}$
Wine	1.770	1.113	**1.891**$^{+}$	1.730
Olitus	**1.489**$^{+}$	1.069	1.117	0.916
Heart	1.245	1.293	1.395	**1.662**$^{+}$
Pima Indian Diab.	0.750	**1.240**$^{+}$	0.966	**1.240**$^{+}$
Breast Cancer	1.202	0.962	**1.501**$^{+}$	1.487
Thoracic Surgery	0.990	1.712	1.512	**2.057**$^{+}$
Ionosphere	2.009	1.616	2.039	**2.109**$^{+}$
Sonar	1.746	1.495	**1.842**$^{+}$	0.782
Wine Quality (Red)	0.586	1.119	1.107	**1.170**$^{+}$
LSVT Voice Rehab.	**2.156**$^{+}$	1.952	2.015	1.826
Seismic Bumps	1.204	1.791	**2.015**$^{+}$	1.262
Arrhythmia	1.425	**1.752**$^{+}$	1.565	1.604
Molecular Biology	0.000	**1.365**$^{+}$	0.000	1.337
COIL 2000	1.793	2.263	2.408	**2.467**$^{+}$
Madelon	1.763	1.857	**1.918**$^{+}$	1.487

Table 3.15: Average rankings of the algorithms based on the *Performance′* measure after removing Cleveland, MONK1, MONK2, MONK3 & Climate Model (Friedman)

Algorithm	**Ranking**
L-FRFS	3.075
CFS	2.650
DFS	2.150
GBFS	**2.125**$^{+}$

Table 3.16: Post Hoc comparison over the results of Friedman procedure of *Performance′* measure after removing Cleveland, MONK1, MONK2, MONK3 & Climate Model

i	**Algorithm**	$z = (R_0 - R_i)/SE$	p	**Li**
3	L-FRFS	2.327015	0.019964	0.00257
2	CFS	1.285982	0.198449	0.00257
1	GBFS	0.061237	0.95117	0.05

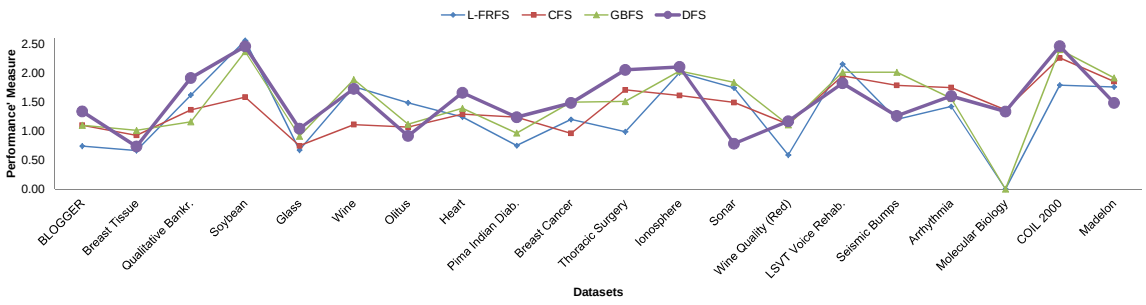


Figure 3.2: *Performance′* measure (Classification Accuracy $\times e^{Reduction}$)

Table 3.17: *Performance″* measure resulting from $e^{ClassificationAccuracy} \times$ *Reduction*

Datasets	**L-FRFS**	**CFS**	**GBFS**	**DFS**
BLOGGER	0.000	0.837	0.837	**1.252**$^{+}$
Breast Tissue	0.000	0.647	**0.850**$^{+}$	0.214
Qualitative Bankr.	1.338	0.889	0.445	**1.783**$^{+}$
Soybean	**2.563**$^{+}$	1.580	2.357	2.462
Glass	0.000	0.217	0.641	**1.009**$^{+}$
Wine	1.601	0.400	**1.784**$^{+}$	1.777
Olitus	**1.569**$^{+}$	0.738	1.251	0.477
Heart	1.012	1.043	1.215	**1.727**$^{+}$
Pima Indian Diab.	0.000	**1.061**$^{+}$	0.530	**1.061**$^{+}$
Breast Cancer	0.582	0.000	**1.163**$^{+}$	1.153
Thoracic Surgery	0.405	1.644	1.362	**2.067**$^{+}$
Ionosphere	1.965	1.428	2.012	**2.126**$^{+}$
Sonar	1.843	1.454	**1.903**$^{+}$	0.105
Wine Quality (Red)	0.000	1.151	1.143	**1.280**$^{+}$
LSVT Voice Rehab.	**2.203**$^{+}$	1.990	2.087	1.906
Seismic Bumps	0.691	1.672	**1.963**$^{+}$	1.493
Arrhythmia	1.669	**1.842**$^{+}$	1.706	1.799
Molecular Biology	0.000	1.288	0.000	**1.593**$^{+}$
COIL 2000	1.666	2.251	2.409	**2.470**$^{+}$
Madelon	1.904	1.969	**2.019**$^{+}$	1.720

Table 3.18: Average rankings of the algorithms based on the *Performance″* measure after removing Cleveland, MONK1, MONK2, MONK3 & Climate Model (Friedman)

Algorithm	**Ranking**
L-FRFS	3.125
CFS	2.700
GBFS	2.150
DFS	**2.025**$^{+}$

Table 3.19: Post Hoc comparison over the results of Friedman procedure of *Performance″* measure after removing Cleveland, MONK1, MONK2, MONK3 & Climate Model

i	**Algorithm**	$z = (R_0 - R_i)/SE$	p	**Li**
3	L-FRFS	2.694439	0.007051	0.01266
2	CFS	1.653406	0.098248	0.01266
1	GBFS	0.306186	0.759463	0.05

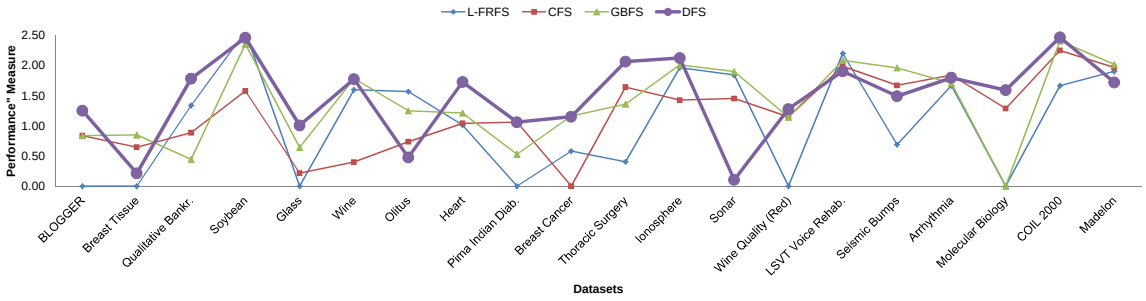


Figure 3.3: *Performance″* measure ($e^{Classification\ Accuracy} \times\ Reduction$)

methods.

Based on the results, we conclude that proposing a universal feature selection method might not be suitable due to the high variety of datasets and applications. Therefore, each and every newly proposed method can be "localized" to a subject and type of the data as well as the purpose of the data. In such a way, data owners can save huge amounts of processing expenses based on a set of categorized methods. As future work, we are excited to perform such categorization for the existing merits on feature selection methods. Also, we are conducting some experiments on Big Data in order to evaluate the performance of the proposed hybrid merit.

Our ongoing task is to prepare an online, web-based application for the new hybrid merit that will be available to the researchers working on datasets in various field of studies.

Acknowledgments

This work has been partially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Research & Development Corporation

of Newfoundland and Labrador (RDC).

Bibliography

[1] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera. Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic & Soft Computing*, 17, 2011.

[2] J. R. Anaraki and M. Eftekhari. Improving fuzzy-rough quick reduct for feature selection. In *Electrical Engineering (ICEE), 2011 19th Iranian Conference on*, pages 1502–1506, May 2011.

[3] J. R. Anaraki, M. Eftekhari, and C. W. Ahn. Novel improvements on the fuzzy-rough quickreduct algorithm. *IEICE Transactions on Information and Systems*, E98.D(2):453–456, 2015.

[4] K. Bache and M. Lichman. UCI machine learning repository, 2013.

[5] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos. Feature selection for high-dimensional data. *Progress in Artificial Intelligence*, 5(2):65–75, 2016.

[6] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and regression trees*. CRC press, 1984.

[7] P. Cortez, A. Cerdeira, F. Almeida, T. Matos, and J. Reis. Modeling wine

preferences by data mining from physicochemical properties. *Decision Support Systems*, 47(4):547–553, 2009.

[8] J. Dai and Q. Xu. Attribute selection based on information gain ratio in fuzzy rough set theory with application to tumor classification. *Applied Soft Computing*, 13(1):211 – 221, 2013.

[9] S. Das. Filters, wrappers and a boosting-based hybrid for feature selection. In *ICML*, volume 1, pages 74–81. Citeseer, 2001.

[10] J. Derrac, N. Verbiest, S. García, C. Cornelis, and F. Herrera. On the use of evolutionary feature selection for improving fuzzy rough set based prototype selection. *Soft Computing*, 17(2):223–238, 2012.

[11] R. Detrano, A. Janosi, W. Steinbrunn, M. Pfisterer, J.-J. Schmid, S. Sandhu, K. H. Guppy, S. Lee, and V. Froelicher. International application of a new probability algorithm for the diagnosis of coronary artery disease. *The American journal of cardiology*, 64(5):304–310, 1989.

[12] I. Guyon, S. Gunn, A. Ben-Hur, and G. Dror. Result analysis of the nips 2003 feature selection challenge. In *Advances in Neural Information Processing Systems*, pages 545–552, 2004.

[13] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18, Nov. 2009.

[14] M. A. Hall and L. A. Smith. Feature subset selection: a correlation based filter

approach. In *Proceedings of the 1997 International Conference on Neural Information Processing and Intelligent Information Systems: New Zealand*, pages 855–858, 1997.

[15] K. Javed, H. A. Babri, and M. Saeed. Feature selection based on class-dependent densities for high-dimensional binary data. *Knowledge and Data Engineering, IEEE Transactions on*, 24(3):465–477, 2012.

[16] R. Jensen and Q. Shen. New approaches to fuzzy-rough feature selection. *Fuzzy Systems, IEEE Transactions on*, 17(4):824–838, Aug 2009.

[17] R. Jensen, S. Vluymans, N. M. Parthaláin, C. Cornelis, and Y. Saeys. *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing: 15th International Conference, RSFDGrC 2015, Tianjin, China, November 20-23, 2015, Proceedings*, chapter Semi-Supervised Fuzzy-Rough Feature Selection, pages 185–195. Springer International Publishing, Cham, 2015.

[18] G. H. John, R. Kohavi, K. Pfleger, et al. Irrelevant features and the subset selection problem. In *Machine learning: proceedings of the eleventh international conference*, pages 121–129, 1994.

[19] G. Kim, Y. Kim, H. Lim, and H. Kim. An mlp-based feature subset selection for hiv-1 protease cleavage site analysis. *Artificial Intelligence in Medicine*, 48(23):83 – 89, 2010. Artificial Intelligence in Biomedical Engineering and Informatics.

[20] K. Kira and L. A. Rendell. The feature selection problem: Traditional methods and a new algorithm. In *AAAI*, pages 129–134, 1992.

[21] R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial intelligence*, 97(1):273–324, 1997.

[22] J. Komorowski, Z. Pawlak, L. Polkowski, and A. Skowron. Rough sets: A tutorial. In S. K. Pal and A. Skowron, editors, *Rough-Fuzzy Hybridization: A New Trend in Decision Making*, pages 3–98. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1998.

[23] D. D. Lucas, R. Klein, J. Tannahill, D. Ivanova, S. Brandon, D. Domyancic, and Y. Zhang. Failure analysis of parameter-induced simulation crashes in climate models. *Geoscientific Model Development*, 6(4):1157–1171, 2013.

[24] S. Manikandan. Measures of central tendency: The mean. *Journal of pharmacology & pharmacotherapeutics*, 2(2):140, 2011.

[25] Z. Pawlak. Rough sets. *International Journal of Computer & Information Sciences*, 11(5):341–356, 1982.

[26] P. V. D. Putten and M. V. Someren. Coil challenge 2000: The insurance company case. Technical report 2000-09, Leiden Institute of Advanced Computer Science, Universiteit van Leiden, June 2000.

[27] Y. Qian, Q. Wang, H. Cheng, J. Liang, and C. Dang. Fuzzy-rough feature selection accelerator. *Fuzzy Sets and Systems*, 258:61 – 78, 2015. Special issue: Uncertainty in Learning from Big Data.

[28] A. M. Radzikowska and E. E. Kerre. A comparative study of fuzzy rough sets. *Fuzzy Sets and Systems*, 126(2):137 – 155, 2002.

[29] C. Shang and D. Barnes. Fuzzy-rough feature selection aided support vector machines for mars image classification. *Computer Vision and Image Understanding*, 117(3):202 – 213, 2013.

[30] M. Sikora and Ł. Wróbel. Application of rule induction algorithms for analysis of data collected by seismic hazard monitoring systems in coal mines. *Archives of Mining Sciences*, 55:91–114, 2010.

[31] A. Tsanas, M. Little, C. Fox, and L. Ramig. Objective automatic assessment of rehabilitative speech treatment in parkinson's disease. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 22(1):181–190, Jan 2014.

[32] J. Wnek and R. S. Michalski. Comparing symbolic and subsymbolic learning: Three studies. *Machine learning: A multistrategy approach*, 4:318–362, 1994.

[33] Z. Xu, G. Huang, K. Q. Weinberger, and A. X. Zheng. Gradient boosted feature selection. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 522–531. ACM, 2014.

[34] L. Yu and H. Liu. Efficient feature selection via analysis of relevance and redundancy. *The Journal of Machine Learning Research*, 5:1205–1224, 2004.

[35] Z. Zhu, Y.-S. Ong, and J. M. Zurada. Identification of full and partial class relevant genes. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, 7(2):263–277, 2010.

[36] M. Zieba, J. M. Tomczak, M. Lubicz, and J. Swiatek. Boosted svm for extracting

rules from imbalanced data in application to prediction of the post-operative life expectancy in the lung cancer patients. *Applied Soft Computing*, 14:99–108, 2014.

Chapter 4

A Fuzzy-Rough Feature Selection based on Binary Shuffled Frog Leaping Algorithm

This paper is accepted in 20th International Conference on Fuzzy Information and Engineering, Vancouver, Canada, 2018, and is selected and published in International Journal of Computer and Information Engineering.

4.1 Abstract

Feature selection and attribute reduction are crucial problems, and widely used techniques in the field of machine learning, data mining and pattern recognition to overcome the well-known phenomenon of the Curse of Dimensionality. This paper presents a feature selection method that efficiently carries out attribute reduction, thereby se-

lecting the most informative features of a dataset. It consists of two components: 1) a measure for feature subset evaluation, and 2) a search strategy. For the evaluation measure, we have employed the fuzzy-rough dependency degree (FRFDD) of the lower approximation-based fuzzy-rough feature selection (L-FRFS) due to its effectiveness in feature selection. As for the search strategy, a modified version of a binary shuffled frog leaping algorithm is proposed (B-SFLA). The proposed feature selection method is obtained by hybridizing the B-SFLA with the FRDD. Nine classifiers have been employed to compare the proposed approach with several existing methods over twenty two datasets, including nine high dimensional and large ones, from the UCI repository. The experimental results demonstrate that the B-SFLA approach significantly outperforms other metaheuristic methods in terms of the number of selected features and the classification accuracy.

Index terms— Binary shuffled frog leaping algorithm, Feature selection, Fuzzy-rough set, Minimal reduct

4.2 Introduction

Feature selection (FS) is the process of selecting the most informative features of a dataset while removing the others, and many studies have been done on diverse FS methods in recent years [47, 13, 33, 31, 12, 23, 3, 24]. The feature selection process results in a reduction in the size of datasets and a retention of their critical information. Finding and removing irrelevant features (which have little/no effect on the classification results) and redundant features (which have high correlation with other features) would reduce the size of datasets, thereby improving the classification

accuracy as well as the visualization and comprehensibility of the induced concepts. The third group is the set of features that should remain at the end of the FS process.

Selecting M out of N features by means of a comprehensive search is an NP-hard problem [26]. Furthermore, it has been proven that approximating the minimal relevant subset is hard up to very large factors [26]. Therefore, greedy search methods and metaheuristic search strategies are suitable for solving this problem [46]. However, all of the greedy search methods suffer from the deficiency of becoming trapped in local optima [46]. Forward and backward search mechanisms are instances of greedy search algorithms that are widely used for FS, because of their ideal time complexity; therefore, they are not capable of avoiding local optima [46, 28]. Due to this deficiency and the inherent ability of metaheuristic search methods to find the global optimum while avoiding local optima, these search methods have been widely utilized to solve FS problems [46, 7, 25, 40].

Genetic algorithm (GA), particle swarm optimization (PSO), Tabu search and memetic algorithms are representative metaheuristic instances that, in recent years, have been very successful at solving various NP-hard engineering problems such as feature selection [46, 7, 25, 40]. Moreover, all of the above search mechanisms require an evaluation criterion for measuring the suitability of feature subsets. Based on determining the evaluation measures, a twofold taxonomy of feature selection methods has been presented in the literature [32]. In this taxonomy, feature selection strategies are categorized into 1) filter-based methods, and 2) wrapper-based methods. The former generally evaluate a feature subset by performing statistical tests on the data [32]. Thus, the filter-based methods "filter out" irrelevant features before the induction process (i.e. classification). In the wrapper-based approach, an induction

algorithm itself (i.e. classifier) is utilized for evaluating feature subsets [32]. In other words, it is used for optimizing the accuracy rate estimated by an induction algorithm. Compared to filter-based methods, wrapper-based methods are computationally prohibitive since they employ an induction model as an embedded algorithm. On the other hand, the wrapper-based methods are more accurate at finding a proper subset of informative features than filter-based methods. In the filter-based technique, a non-statistical criterion can also be used as the evaluation measure. Examples of such criteria include the dependency degree (DD) based on rough set theory [35], and the fuzzy feature saliency measure [38] based on fuzzy set theory. Recently, much research has been performed on the development of methodologies for dealing with imprecision and uncertainty [35, 38, 5]. Fuzzy and rough set theories are analogous in the sense that they can model uncertainty and inconsistency. Recent studies have shown that they are complementary in nature.

Fuzzy-rough feature selection (FRFS) is one of the most successful hybrid tools for dimensional reduction, which is capable of handling both discrete and real-valued (or a mixture of both) variables [5]. However, there are some problems regarding the use of FRFS, thoroughly addressed in [16]. For instance, pre-data discretization by using fuzzy partitions is an FRFS approach that is not very successful in terms of computation. One of the newly developed FRFS methods is the lower approximation-based fuzzy-rough feature selection (L-FRFS) [16] method. L-FRFS, introduced in [16] is a fast FRFS, and it exhibits better performance compared to previously developed FRFSs. Moreover, as stated earlier, generating all subsets of features is an NP-hard problem and computationally prohibitive. Therefore, some hill-climbing search algorithms have been proposed in the literature in order to compensate for

this computational deficiency [16].

The smallest subset of features with the highest DD is called the "minimal reduct"; it might not be found by the fuzzy-rough QuickReduct algorithm, which is an example of a hill-climbing method, both in terms of the resulting dependency measure and the subset size. Due to the deficiencies of hill-climbing approaches, metaheuristic algorithms such as GA and PSO are required in order to find such minimal reducts, especially when available data are high-dimensional. In [4, 34, 41, 43, 1] metaheuristic algorithms and rough set theory have been combined to find minimal reducts. In recent years, a few studies have also been presented in literature regarding the hybridization of fuzzy-rough and metaheuristic approaches [5, 16]. Very significant work is the combination of ACO and fuzzy-rough set for dimension reduction [14]. In this work, Jensen and Shen utilized a computationally demanding FRFS method in which continuous data have been discretized in advance by fuzzy partitions, and an ACO has been employed to find the minimal reduct [14]. As mentioned earlier, the authors have recently confirmed the time deficiencies of the fuzzy-rough method used in [16], and as an alternative have introduced the L-FRFS as a fast method.

In [44], Xiang et al. have proposed a hybrid method for feature selection by improving the diversity of species through piecewise linear chaotic maps (PWL), and increasing the speed of local search by applying sequential quadratic programming (SQP) to the binary gravitational search algorithm (GSA). The improved version of GSA has been hybridized with a 1-nearest neighbour method to from a feature selection system. A modified version of the binary PSO with the ability to avoid premature convergence utilizing both velocity and similarity of best solutions has been introduced by Vieira et al. [39]. The search method has been used to perform

simultaneous feature selection and prediction of mortality of septic patients using concurrently optimized kernel parameters of a support vector machine (SVM). On of the most recent and successful feature selection methods is gradient boosted feature selection (GBFS) proposed by Xu et al. [45]. It works based on gradient boosted trees [9]. It starts by building regression trees using CART algorithm [2], and features are selected simultaneously based on deviation in impurity function. Selecting new feature is penalized and reusing already selected features has no cost.

In the present paper, a new FRFS technique is proposed on the basis of the B-SFLA and L-FRFS. Our contributions are twofold: 1) we devise a new binary version of an SFLA that employs a new dissimilarity measure, new coefficients for self-parameter selection, and a modified ranking rule, and 2) we develop an FS method by combining the strengths of this B-SFLA and the L-FRFS. The rest of this paper is organized as follows. In Section 4.3, the background of the rough set and the shuffled frog leaping algorithm are presented. Section 4.4 illustrates the proposed feature selection method. Section 4.5 reports experimental results and finally we conclude this paper in Section 4.6.

4.3 Background

4.3.1 Rough Set

Rough set theory was proposed by Pawlak as a tool to deal, in an efficient way, with uncertainty [27], in data organized in a decision table. Let U be the universe of discourse and A be a nonempty finite set of attributes in U; information system

is shown by $I = (U, A)$. Let X be a subset of U, and P and Q be two subsets of A; approximating a subset using rough set theory is done by means of upper and lower approximations. The upper approximation of X with regard to $(\overline{P}X)$ contains objects, which are possibly classified in X regarding the attributes in P. Objects in the lower approximation $(\underline{P}X)$ are those, which are definitely classified in X regarding the attributes in P. A rough set is shown by an ordered pair, $(\underline{P}X, \overline{P}X)$. The positive region as shown in Equation 4.1 of partition $\mathbb{U}/Q$ is a set of all objects, which can be uniquely classified into blocks of the partition by means of P.

$$POS_P(Q) = \bigcup_{X \in \mathbb{U}/Q} \underline{P}X \tag{4.1}$$

Finding the dependency between attributes is one of the most important areas in data analysis. The dependency of Q on P is denoted by $P \Rightarrow_k Q$ and $k = \gamma_p(Q)$, in which γ is the dependency degree [21]. If $k = 1$ then Q completely depends on P and if $0 < k < 1$ then Q partially depends on P. The value of k is a measure of the dependency between the features P and Q. In feature selection, features which have lower dependency on each other and are highly correlated to the decision feature(s), are desired. If Q completely depends on P, then the partition which is made by P is finer than Q. The positive region of the partition $\mathbb{U}/Q$, with respect to P, which is denoted by $POS_P(Q)$, is the set of all elements which can be classified into the partition $\mathbb{U}/Q$ using P [21]. The following equation allows to calculate the dependency.

$$k = \gamma_P(Q) = \frac{|POS_P(Q)|}{|\mathbb{U}|}, \tag{4.2}$$

where notation $|.|$ is used for cardinality. The reduct is a subset of features which have the same dependency degree as employing all the features for classification. The

features that belong to the reduct set are the most informative ones while the others are either irrelevant or redundant.

One way to handle real-valued data using rough set theory is to discretize continuous data in advance and make a new crisp valued dataset. Discretization is not enough as long as the similarity between two values remains unspecified [16]. Therefore, dependency degree between the features is calculated by means of the FRDD. The fuzzy-rough set basis will be addressed thoroughly in Section 4.4.

4.3.2 Shuffled Frog Leaping Algorithm

The Shuffled frog leaping algorithm (SFLA) is a memetic metaheuristic search algorithm proposed by Eusuff et al. [8]; it is basically a combination of a shuffled complex evolution (SCE) algorithm [6] that ensures global exploration, and PSO [20] that is responsible for local search. Randomness and determinism are the results of this combination. The SFLA is based on memetics of frog-like beings. A meme is an idea or information pattern which is replicated or repeated to someone else. Memes and genes are analogous but are different in the way they propagate. A *meme* is propagated by leaping from one brain to another and can be transmitted between any individual, but a gene is propagated from parent to offspring by (sexual) reproduction.

The algorithm is inspired by real frog populations searching for food. In this algorithm, the behaviour of the population is determined by memes, and thus the population is more important than individuals. In the SFLA, frogs are partitioned into memeplexes that are evaluated individually. In each memeplex, frogs are influenced by each other and they experience meme evolution. Memetic evolution increases the

frogs' performance in terms of reaching the goal by using information from the memeplex and the best performing individual in the population. This process continues for a predefined number of iterations. Then, all memeplexes are mixed with each other to form a new set of memeplexes through shuffling. Frogs with better performance contribute more to distribute new individuals in the population. A modified version of the SFLA has been proposed by Reddy et al. [30] for solving the environmentally-constrained economic dispatch problem. The modified algorithm uses a local search as well as a new parameter to accelerate convergence.

4.4 Proposed Feature Selection Approach

In this section, the proposed approach is defined based on the two main concepts of feature selection: 1-evaluation measure, and 2- search method. The evaluation measure is fuzzy-rough dependency degree (FRDD) and the search method is a binary modification of SFLA.

4.4.1 Evaluation Measure

The QuickReduct algorithm finds a reduct set without finding all the subsets[16]. It begins with an empty set and each time selects the feature that causes the greatest increase in dependency degree (DD). The algorithm stops when adding more features does not increase the DD. Since it employs a greedy algorithm, it does not guarantee that the minimal reduct set will be found. For this reason, a new FRFS algorithm is presented in this paper. Prior to providing the details of our approach, it is necessary to introduce the definition of the FRDD. To begin with, the definition of the X-lower

and the degree of fuzzy similarity [16] are given by Equations 4.3 and 4.4, respectively.

$$\mu_{\underline{R_P}X}(x) = \inf_{y \in \mathbb{U}} I\{\eta_{R_P}(x,y), \mu_X(y)\}, \tag{4.3}$$

$$\eta_{R_P}(x,y) = \bigcap_{a \in P} \{\eta_{R_a}(x,y)\}, \tag{4.4}$$

where I is a Łukasiewicz fuzzy *implicator*, which is defined by $min(1 - x + y, 1)$. In [29], three classes of fuzzy-rough sets based on three different classes of implicators, namely S-, R-, and QL-implicators, and their properties have been investigated. Here, R_P is the fuzzy similarity relation considering the set of features in P, and $\eta_{R_P}(x,y)$ is the degree of similarity between objects x and y over all features in P. Also, $\mu_X(y)$ is the membership degree of y to X. One of the best fuzzy similarity relations as suggested in[16] is given by Equation 4.5.

$$\eta_{R_a}(x,y) \quad = \quad max\left\{min\left\{\frac{(a(y) - (a(x) - \sigma_a))}{(a(x) - (a(x) - \sigma_a))}, \frac{((a(x) + \sigma_a) - a(y))}{((a(x) + \sigma_a) - a(x))}\right\}, 0\right\}, \tag{4.5}$$

where σ_a is variance of feature a. The L-FRFS does not use the fuzzy partitioning used in FRFS, and thereby it is more computationally effective.

The FRFS can be conducted on the real-valued datasets using the lower approximation. The positive region in rough set theory is defined as a union of lower approximations. Referring to the extension principle [16], the membership of object x to a fuzzy positive region is given by Equation 4.6.

$$\mu_{POS_P(Q)}(x) = \sup_{X \in \mathbb{U}/Q} \mu_{\underline{P}X}(x). \tag{4.6}$$

If the equivalence class that includes x does not belong to a positive region, clearly x will not be part of a positive region. Using the definition of positive region, the FRDD function [16] is defined as:

$$\gamma'_P(Q) = \frac{|\mu_{POS_P(Q)}(x)|}{|\mathbb{U}|} = \frac{\sum_{x \in \mathbb{U}} \mu_{POS_P(Q)}(x)}{|\mathbb{U}|}. \tag{4.7}$$

Based on the concept of the FRDD, we have developed a new metaheuristic search mechanism in order to effectively discover the minimal reducts. Among various search algorithms, such as GA and PSO, the SFLA can be used as a promising search method for feature selection (which is an NP-hard problem), due to its performance toward global optimal solution, both from a likelihood and a speed perspective [8]. Based on the published results in [8], the GA has failed to find best values in 20% of the cases, and it also needs a higher number of function evaluations to find the optimal value, compared to the SFLA. The SFLA is capable of finding a subset of solutions along with the optimal answer as the final result. Since the feature selection problem is fundamentally binary, the need for a binary search algorithm is inevitable.

4.4.2 Search Method

The search process starts by randomly initializing each binary individual with the size of the number of features, and continues by participating in ranking, partitioning and evolutionary processes. Generally, the SFLA consists of seven steps as follows:

Step 1 ***Initialize the population:*** Choose m and n. Here, m is the number of memeplexes, and n is the number of frogs in each memeplex. The total number of frogs is then $F = m \times n$.

Step 2 ***Generate a population:*** The total number of frogs in the feasible space is $\Omega \subset \Re^d$ where d is the number of decision variables (features); the ith frog is encoded as $U(i) = (U_i^1, U_i^2, ..., U_i^d)$. Compute the fitness value for all individuals using Equation 4.7.

Step 3 ***Rank frogs:*** Sort frogs in descending order of their fitnesses, and record them in $X = \{U(i), f(i), i = 1, ..., F\}$. The position of the first (i.e., best) frog is recorded in P_X, where $P_X = U(1)$.

Step 4 ***Partition frogs into memeplexes:*** Partition the array X of frogs into m memeplexes, each containing n frogs.

$$\begin{aligned} Y^k =& [U(j)^k, f(j)^k | U(j)^k = U(k + m(j-1)), \\ f(j)^k =& f(k + m(j-1)), j = 1, ..., n], k = 1, ..., m \end{aligned} \tag{4.8}$$

Step 5 ***Memetic evolution in each memeplex:*** Each memeplex is involved in the evolution which is described later in the Step 5's subsection.

Step 6 ***Shuffle memeplexes:*** After a predefined number of evolution rounds, all memeplexes are mixed into X, and sorted in descending order.

Step 7 ***Check convergence:*** If the convergence criteria are satisfied, stop. Otherwise, go to Step 4.

Note that in the Step 5, the evolution process is repeated N times. This process is comprised of further steps, as follows:

Step 1 ***Initialization:*** Set $i_m = 0$ and $i_N = 0$ as two counters for memeplexes and evolutions, respectively.

Step 2 $i_m = i_m + 1$

Step 3 $i_N = i_N + 1$

Step 4 ***Construct a submemeplex:*** In order to avoid being trapped in local optima, a subset of memeplexes is selected for moving toward. The submemeplex selection strategy is based on a triangular probability distribution (see Equation 4.9) that assigns the highest value to a frog with the maximum fitness and the lowest value to a frog with the minimum fitness. This assignment increases the chances of a high performing frog being selected.

$$p_j = \frac{2 \times (n + 1 - j)}{n \times (n + 1)}, j = 1, ..., n \tag{4.9}$$

For example, for $j = 1$ and $j = n$, the probabilities are given by:

$$p_1 = \frac{2}{n + 1}, p_n = \frac{2}{n \times (n + 1)}$$

After the submemeplex formation, it is sorted in descending order in an array, Z, and the best and the worst positions are recorded in P_B and P_W, respectively.

Step 5 ***Improve the worst frog:*** The worst frog's position is improved using Equations 4.10 and 4.11 for positive and negative steps, respectively.

$$\text{step size } S = min\{int\{rand \times (P_B - P_W)\}, S_{max}\} \tag{4.10}$$

$$\text{step size } S = max\{int\{rand \times (P_B - P_W)\}, -S_{max}\}, \tag{4.11}$$

where *rand* is a random number, *int* is the integer part of a number, and S_{max} is the maximum step size allowed to be adopted after infection. Since the P_B and P_W are in binary form, the distance between two parameters is calculated using the *HD*; therefore, Equation 4.10 and 4.11 are modified to Equation 4.12 and 4.13 to deal with binary parameters.

$$\text{step size } S = min\{int\{rand \times HD(P_B, P_W)\}, S_{max}\} \quad (4.12)$$

$$\text{step size } S = max\{int\{rand \times HD(P_B, P_W)\}, -S_{max}\}. \quad (4.13)$$

Then, the new position is calculated by:

$$U_{(q)} = P_W + S, \quad (4.14)$$

where q is the number of randomly selected frogs from n frogs to form a memeplex and it is initialized manually. If $U_{(q)}$ is in feasible space Ω, then compute the fitness value, $f_{(q)}$; otherwise, go to the Step 6. If the newly computed $f_{(q)}$ is better than the old $f_{(q)}$, then go to the Step 8; otherwise, go to the Step 6.

Step 6 ***Compute new position:*** For real-valued frogs new position can be calculated using Equation 4.15 and 4.16, whereas for the binary-valued frogs Equation 4.17 and 4.18 can be used.

$$\text{step size } S = min\{int\{rand \times (P_X - P_W)\}, S_{max}\} \quad (4.15)$$

$$\text{step size } S = max\{int\{rand \times (P_X - P_W)\}, -S_{max}\} \tag{4.16}$$

$$\text{step size } S = min\{int\{rand \times HD(P_X, P_W)\}, S_{max}\} \tag{4.17}$$

$$\text{step size } S = max\{int\{rand \times HD(P_X, P_W)\}, -S_{max}\}. \tag{4.18}$$

If $U_{(q)}$ is in feasible space Ω, then compute the fitness value, $f_{(q)}$; otherwise, go to Step 7. If the newly computed $f_{(q)}$ is better than the old $f_{(q)}$, then go to Step 8; otherwise, go to Step 7.

Step 7 ***Censorship:*** Replace this frog with a randomly generated frog, r.

Step 8 ***Update the memeplex:*** After changing the worst frog's position in the sub-memeplex, replace Z in their original locations in Y^{i_m} . Sort Y^{i_m} in descending order.

Step 9 If $i_N < N$, go to Step 3.

Step 10 If $i_m < m$, go to Step 2.

Meanwhile, a modification for calculating the distance of the frogs is further applied to the proposed binary SFLA. The distance of the frogs that was calculated using the *HD* is replaced with a dissimilarity measure based on the fuzzy-rough set. The positive region i.e., $POS(.)$ [19] as presented in Equation 4.6 is used instead of the *HD*. The positive region sees the frogs as features and calculates the similarity between each frog and the best frog. The value of $POS(.)$ varies from zero to the

number of the variables. Since this distance must be dissimilarity, this measure is subtracted from the length of the binary frog. This measure can be employed in the Step 5, and the modified equations are given by Equation 4.19 and 4.20 are used in the Step 6.

$$\text{step size } S = min\{int\{rand \times (L - POS(P_B, P_W))\}, S_{max}\} \tag{4.19}$$

$$\text{step size } S = max\{int\{rand \times (L - POS(P_B, P_W))\}, -S_{max}\}, \tag{4.20}$$

where L is the length of a binary frog, and S_{max} is the maximum step size allowed to be adopted after evolution.

The hybridization of the B-SFLA with FRDD is suggested to discover more than one reduct with the highest dependency degree. The L-FRFS can be considered as a multi-modal problem, in which the smallest subset of features with the highest FRDD is desired. Thus, conventional evolutionary algorithms might find many global optima with the highest FRDD; however, a question arising here is "which one is the best?"; Referring to the fitness, all of these solutions are acceptable, whereas referring to the cardinality of the subsets they varies. By ranking the subsets with the same FRDD, based on the number of selected features, a new wide range of reducted subsets is provided. This range can be analyzed using the frequency of a feature's appearance in all of the reducted subsets. The most frequent features might play an important role in specifying the outcome.

The aforementioned strategy is placed in the Step 4 of meme evolution and the Step 3, ranking frogs, of the B-SFLA; however, the ranking process is primarily based on the FRDD and in the case of having several subsets with the identical FRDD, it

ranks subsets based on their cardinality. Through this process, the B-SFLA returns more than one reduct in a single run; conventional search methods do not always return more than one reduct. These minimal sets satisfy both criteria: the highest FRDD and the lowest number of selected features.

Using this method, the frogs leap toward two goals simultaneously. In the very first leaps, frogs jump toward the subsets with the highest FRDD; therefore, they try to increase their fitness as much as possible. In the following leaps, when the number of frogs with the maximum fitness is increased, the population selects the individuals with both the highest FRDD and the lowest number of features. Algorithm 4.1 shows pseudo code of the proposed method. The C++ implementation of the proposed method is publicly available at GitHub. [1]

In the preparation section, parameters of the B-SFLA are initialized based on the properties of the current dataset. Then, $m \times n$ diverse subsets of features are evaluated and evolved based on FRDD and B-SFLA, respectively. Then, the outcome of the algorithm is fed to nine different classifiers to avoid any tendency toward specific classification method. Finally, the mean of the resulting classification accuracies is calculated.

Since the complexity of meta-heuristic search algorithms are very depended on their parameters, it is worth mentioning that the complexity of the FRDD is $O(n^2)$ in the worst case [15], where n is number of features.

[1] https://github.com/jracp/FuzzyRoughShuffledFrog

Algorithm 4.1: FRFS based on B-SFLA

search–evaluate;

initialize m, n, q, N, S_{max};

generate a population of $(m \times n)$ frogs;

rank frogs in X based on # of features and FRDD;

partition X into m memeplexes $Y^1, Y^2, ..., Y^m$;

while $i_m < m$ **do**

    while $i_N < N$ **do**

        construct submemeplex Z containing q frogs;

        improve the worst frog and update FRDD;

        replace infeasible and halting frogs;

        partition Z into $Y^1, Y^2, ..., Y^m$;

combine $Y^1, Y^2, ..., Y^m$ into X, update the best frog;

check the convergence criteria;

4.5 Experimental Results and Discussion

Twenty two datasets from the UCI repository of machine learning [22] including nine large datasets – namely, LSVT Voice Rehabilitation [36], Urban Land Cover [18, 17], Arrhythmia, Molecular Biology, COIL 2000 [37], CNAE-9, Madelon [10], MicroMass, and Arcene [10] – have been selected and used to perform a comparative study. These datasets and their characteristics are shown in Table 4.1. The table is sorted based on the number of samples × features.

The *fitness function* for all of the search algorithms is the FRDD depicted in

Table 4.1: Dataset characteristics

Datasets	**Samples**	**Features**
Breast Tissue	106	10
Lung Cancer	32	56
Glass	214	10
Wine	178	13
Olitos	120	25
Heart	270	13
Cleveland	303	13
Parkinson	197	23
Pima Indian Diabetes	768	8
Breast Cancer Wisconsin	699	10
Ionosphere	351	33
Sonar	208	60
Libras Movement	360	90
LSVT Voice Rehab.	126	310
Urban Land Cover	675	148
Arrhythmia	452	279
Molecular Biology	3190	60
COIL 2000	5822	85
CNAE-9	1080	857
Madelon	2000	500
MicroMass	931	1300
Arcene	200	10000

Table 4.2: GA parameters

Population	**Generation**	P_c	P_m
900	5	0.600	0.033

Table 4.3: PSO parameters

Particles	**Iteration**	C_1	C_2
900	5	2	2

Equation 4.7. The GA and PSO parameters are presented in Tables 4.2 and 4.3, respectively. For both algorithms, the population size and the number of generations are identical to B-SFLA's to enable further comparisons. As presented in [8], the SFLA parameter selection should be performed based on the properties of the problem. Parameter selection is one of the most important aspects of using search algorithms; however, it is still untouched for feature selection. Referring to the authors' recommendation in [8], for problems with 15-20 variables, the ranges in Table 4.4 are suggested. However, the parameter selection for feature selection has been formulated based on the total number of all features (all_F) using a trial and error method. The results are shown in Table 4.5. Further investigations show that the proposed parameters in Table 4.5 work remarkably well for small datasets with less than 15,000 data cells; however, parameters in Table 4.6 [8] can be used not only for small and medium datasets, but also for large ones.

The number of selected features obtained by each search algorithm is shown in Table 4.7. In terms of the number of selected features, the GBFS has selected the least

Table 4.4: SFLA parameters

m	n	N	q	S_{max}
$100 \leq m \leq 150$	$30 \leq n \leq 100$	$20 \leq N \leq 30$	20	$1.00 \times all\_F$

Table 4.5: Proposed B-SFLA parameters for datasets with size of data cells $\leq 15,000$

m	n	N	q	S_{max}
$2.20 \times all\_F$	$0.70 \times all\_F$	$0.50 \times all\_F$	$0.45 \times all\_F$	$0.50 \times all\_F$

number of features compared to the other methods; however, selecting one feature as a final result for Breast Tissue, Lung Cancer, Glass, Wine, and Sonar is not desirable both from an in-field and a data processing point of view. Selecting a very small number of features reduces the utility of feature selection methods for pre-processing and model complexity improvement.

Nine classifiers – namely PART, JRip, Naive Bayes, Bayes Net, J48, BFTree, FT, NBTree and RBFNetwork – have been chosen from different classifiers categories to classify instances of each dataset after the feature selection process. These classifiers have been implemented in Weka, a machine learning package that is ready to use [11]. For all classifiers and the feature selection methods, 10-fold cross validation (10CV) has been conducted to calculate their performance. The mean as well as

Table 4.6: Proposed B-SFLA parameters for most datasets

m	n	N	q	S_{max}
30	30	5	15	$0.45 \times all\_F$

Table 4.7: Number of selected features obtained by each search algorithm

Datasets	**L-FRFS**	**GA**	**PSO**	**GBFS**	**B-SFLA**
Breast Tissue	9	9	9	1	4
Lung Cancer	6	7	4	1	3
Glass	9	8	8	1	4
Wine	5	5	5	1	3
Olitos	5	5	5	6	5
Heart	7	8	7	4	5
Cleveland	11	10	10	4	7
Parkinson	5	6	6	3	4
Pima Indian Diabetes	8	8	8	2	6
Breast Cancer Wisconsin	7	7	7	6	7
Ionosphere	7	8	7	5	5
Sonar	5	6	6	1	5
Libras Movement	2	11	8	17	6
LSVT Voice Rehab.	5	11	7	6	7
Urban Land Cover	7	9	8	12	7
Arrhythmia	7	10	13	26	8
Molecular Biology	-	13	12	3	9
COIL 2000	29	46	33	5	8
CNAE-9	90	459	547	13	281
Madelon	-	-	-	6	7
MicroMass	33	168	142	24	141
Arcene	6	-	-	6	11

standard deviation (STD), and the best value of the nine classifiers' results over each dataset are presented in Table 4.8. The best of the mean classification accuracies are boldfaced and superscripted. The last row shows the mean of the classification accuracies' mean, the STD, and the best in which the B-SFLA gains 1.22%, 2.16%, 2.33%, 7.87% higher mean classification accuracies compared to L-FRFS, GA, PSO, and GBFS, respectively. The B-SFLA outperforms other methods not only by decreasing the model size, but also by improving classification accuracy of the resulting models. Referring to the number of selected features in Table 4.7 and the classification accuracies in Table 4.8, the GBFS has selected the least number of features and obtained the smallest classification accuracy, which is worse when compared to the unreduced datasets and to the other methods.

Table 4.9 shows the number of wins in terms of the best resulting classification accuracies. The L-FRFS has achieved the best accuracies for Breast Tissue, Glass, Wine, Ionosphere, Urban Land Cover, and CNAE-9. The GA has obtained the best classification accuracies in three cases, Breast Tissue, Breast Cancer Wisconsin and MicroMass. The PSO has obtained the highest classification accuracy for Heart dataset. The GBFS has achieved the best classification accuracies for five datasets – namely, Olitos, Cleveland, Libras Movement, Arrhythmia, and Arcene. Finally, B-SFLA has reached to the maximum number of wins for eight datasets – namely, Lung Cancer, Parkinson, Pima Indian Diabetes, Sonar, LSVT Voice Rehab., Molecular Biology, COIL 2000, and Madelon.

It is concluded that the B-SFLA is the most suitable search algorithm for FS based on the fuzzy-rough sets approach in terms of the resulting classification accuracy. Note that the B-SFLA divides the population into subpopulations, and thereby

Table 4.8: Mean, standard deviation, and best of classification accuracies (%)

Datasets	**L-FRFS**	**Best**	**GA**	**Best**	**PSO**	**Best**	**GBFS**	**Best**	**B-SFLA**	**Best**
Breast Tissue	**66.46 ± 3.69**$^{+}$	70.75	**66.46 ± 3.69**$^{+}$	70.75	66.46 ± 3.69	70.75	56.92 ± 4.42	61.32	65.09 ± 5.70	75.47
Lung Cancer	58.85 ± 12.48	77.78	41.56 ± 5.48	48.15	53.24 ± 11.53	70.37	37.04 ± 0.00	37.04	**62.96 ± 12.28**$^{+}$	77.78
Glass	**67.29 ± 7.62**$^{+}$	74.77	64.75 ± 7.76	71.96	64.75 ± 7.76	71.96	50.05 ± 5.50	54.67	65.32 ± 6.50	71.03
Wine	**95.63 ± 2.92**$^{+}$	99.44	92.38 ± 2.23	95.51	92.38 ± 2.23	95.51	66.67 ± 1.61	68.54	93.57 ± 1.97	96.07
Olitos	66.39 ± 5.50	73.33	63.89 ± 3.17	68.33	65.09 ± 3.29	70.00	**70.93 ± 4.24**$^{+}$	75.83	69.17 ± 4.06	77.50
Heart	78.48 ± 1.88	80.37	78.72 ± 1.55	80.74	**79.55 ± 3.77**$^{+}$	84.07	75.93 ± 2.10	78.89	78.85 ± 1.94	81.85
Cleveland	49.76 ± 5.58	54.88	50.73 ± 4.87	54.88	50.73 ± 4.87	54.88	**52.64 ± 2.84**$^{+}$	54.88	50.88 ± 4.11	54.88
Parkinson	85.07 ± 4.18	90.77	85.19 ± 3.20	90.26	83.36 ± 3.75	89.23	85.75 ± 3.31	90.26	**86.50 ± 3.61**$^{+}$	89.74
Pima Indian Diabetes	75.00 ± 1.23	77.34	75.00 ± 1.23	77.34	75.00 ± 1.23	77.34	64.76 ± 0.95	66.15	**75.35 ± 1.28**$^{+}$	76.69
Breast Cancer Wisconsin	96.23 ± 1.04	97.51	**96.40 ± 0.54**$^{+}$	97.36	96.13 ± 0.60	96.93	95.15 ± 0.85	96.05	96.03 ± 0.92	97.36
Ionosphere	**91.39 ± 1.04**$^{+}$	93.16	89.78 ± 1.22	92.02	89.49 ± 2.54	94.02	89.21 ± 1.40	91.74	89.65 ± 1.43	91.74
Sonar	69.82 ± 2.60	72.60	69.76 ± 2.29	73.08	64.26 ± 2.54	68.75	55.29 ± 3.69	61.06	**74.09 ± 3.45**$^{+}$	78.85
Libras Movement	21.76 ± 7.45	28.61	58.14 ± 10.11	73.94	57.73 ± 7.68	67.99	**61.36 ± 9.73**$^{+}$	74.17	53.43 ± 8.00	65.56
LSVT Voice Rehab.	79.45 ± 4.39	86.51	67.99 ± 8.10	76.98	74.52 ± 4.85	84.13	74.69 ± 10.17	80.95	**79.62 ± 5.66**$^{+}$	85.71
Urban Land Cover	**80.07 ± 2.68**$^{+}$	84.89	63.18 ± 2.87	74.37	56.50 ± 1.80	71.26	51.84 ± 1.73	83.70	77.66 ± 2.29	81.04
Arrhythmia	53.74 ± 3.10	57.52	53.60 ± 3.69	57.74	52.21 ± 4.52	56.42	**69.05 ± 2.59**$^{+}$	74.34	60.50 ± 4.11	64.60
Molecular Biology	-	-	63.18 ± 1.66	65.27	56.50 ± 1.45	59.00	51.84 ± 0.17	52.19	**80.12 ± 1.20**$^{+}$	81.38
COIL 2000	92.79 ± 2.01	94.02	92.42 ± 2.56	94.02	92.51 ± 2.40	94.02	93.97 ± 0.07	94.04	**93.98 ± 0.06**$^{+}$	94.02
CNAE-9	**88.78 ± 1.94**$^{+}$	91.57	85.77 ± 2.71	90.65	88.04 ± 3.46	92.59	53.60 ± 4.37	55.74	74.47 ± 2.32	77.96
Madelon	-	-	-	-	-	-	49.58 ± 0.72	50.80	**54.66 ± 0.68**$^{+}$	55.40
MicroMass	57.40 ± 5.16	66.90	**68.42 ± 5.44**$^{+}$	80.04	65.27 ± 4.10	74.78	63.07 ± 3.27	67.08	64.93 ± 4.02	73.20
Arcene	71.56 ± 3.00	77.00	-	-	-	-	**74.94 ± 4.45**$^{+}$	81.00	70.78 ± 5.37	78.50
Mean	72.30 ± 3.97	77.49	71.36 ± 3.72	76.67	71.19 ± 3.90	77.20	65.65 ± 3.10	70.47	**73.52 ± 3.70**$^{+}$	**78.47**$^{+}$

Table 4.9: Number of wins for each method in gaining highest classification accuracy

Algorithm	**L-FRFS**	**GA**	**PSO**	**GBFS**	**B-SFLA**
Wins	6	3	1	5	**8**$^{+}$

the diversity in the population is preserved. Such a swarm algorithm is very suitable for multi-modal optimization problems that have several optima instead of just one global optimum [42]. The feature selection based on fuzzy-rough set is an example of such problems. The main intention in the L-FRFS is to obtain the minimal reducts; there exist several minimal-reducts for a given information system that are feature subsets with the minimal possible size and maximal possible FRDD. In a single run, GA and PSO generally produce one minimal reduct for a given problem as the final solution of the L-FRFS. However, the B-SFLA returns almost all of the minimal reducts in a single run in its final population. On the other hand, the B-SFLA apparently demonstrates its suitability for solving multi-modal problems since it inherently divides the population of frogs into different subpopulations. Therefore, each of these subpopulations is able to explore and exploit one of the several existing optima in the search space. This property of the B-SFLA makes it different from the other algorithms such as GA and PSO.

4.6 Conclusion and Future Work

In this paper, a new version of the B-SFLA has been combined with the FRDD. Additionally, the performances of L-FRFS, two well-known evolutionary algorithms, the GBFS and the B-SFLA have been compared. By considering the results, the B-SFLA

approach significantly outperforms the PSO, GA, and GBFS methods, and is slightly better than L-FRFS in terms of resulting classification accuracy. Feature selection via fuzzy-rough theory is a multi-modal problem, i.e. there are some feature subsets with the same size and FRDD. In this sense, the B-SFLA is a suitable search algorithm for such problems, since it divides the population into subpopulations (called memeplexes), and by preserving the diversity, it returns multiple minimal reducts rather than returning just a single one. This means that several minimal reducts (i.e. the feature subsets with the minimum cardinality and maximum FRDDs) have been produced in a single run. This characteristic is an additional advantage of the B-SFLA over the PSO and GA algorithms. We are planning to apply our proposed method on local datasets, such as existing health data from Newfoundland and Labrador Centre for Health Information (NLCHI), and global ones in Canada, such as data from Statistics Canada. Also, we are aiming to improve time and space complexity of the B-SFLA to target big data, and perform comprehensive examinations and comparisons with the newly introduced feature selection methods.

Bibliography

[1] J. R. Anaraki and M. Eftekhari. Rough set based feature selection: A review. In *Information and Knowledge Technology (IKT), 2013 5th Conference on*, pages 301–306, May 2013.

[2] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and regression trees*. CRC press, 1984.

[3] A. M. Canuto, K. M. Vale, A. Feitos, and A. Signoretti. Reinsel: A class-based mechanism for feature selection in ensemble of classifiers. *Applied Soft Computing*, 12(8):2517 – 2529, 2012.

[4] Y. Chen, D. Miao, and R. Wang. A rough set approach to feature selection based on ant colony optimization. *Pattern Recognition Letters*, 31(3):226 – 233, 2010.

[5] C. Degang and Z. Suyun. Local reduction of decision system with fuzzy rough sets. *Fuzzy Sets and Systems*, 161(13):1871 – 1883, 2010.

[6] Q. Duan, S. Sorooshian, and V. Gupta. Effective and efficient global optimization for conceptual rainfall-runoff models. *Water Resources Research*, 28(4):1015–1031, 1992.

[7] M. ElAlami. A filter model for feature subset selection based on genetic algorithm. *Knowledge-Based Systems*, 22(5):356 – 362, 2009.

[8] M. Eusuff, K. Lansey, and F. Pasha. Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization. *Engineering Optimization*, 38(2):129–154, 2006.

[9] J. H. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[10] I. Guyon, S. Gunn, A. Ben-Hur, and G. Dror. Result analysis of the nips 2003 feature selection challenge. In *Advances in neural information processing systems*, pages 545–552, 2004.

[11] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18, Nov. 2009.

[12] X. Han. Implicit feature selection for omics data phenotype discrimination. *Applied Soft Computing*, 20:70 – 82, 2014. Hybrid intelligent methods for health technologies.

[13] E. Hancer, B. Xue, D. Karaboga, and M. Zhang. A binary {ABC} algorithm based on advanced similarity scheme for feature selection. *Applied Soft Computing*, 36:334 – 348, 2015.

[14] R. Jensen and Q. Shen. Fuzzy-rough data reduction with ant colony optimization. *Fuzzy Sets and Systems*, 149(1):5 – 20, 2005.

[15] R. Jensen and Q. Shen. *Computational intelligence and feature selection: rough and fuzzy approaches*, volume 8. John Wiley & Sons, 2008.

[16] R. Jensen and Q. Shen. New approaches to fuzzy-rough feature selection. *Fuzzy Systems, IEEE Transactions on*, 17(4):824–838, Aug 2009.

[17] B. Johnson and Z. Xie. Classifying a high resolution image of an urban area using super-object information. *ISPRS Journal of Photogrammetry and Remote Sensing*, 83:40–49, 2013.

[18] B. A. Johnson. High-resolution urban land-cover classification using a competitive multi-scale object-based approach. *Remote Sensing Letters*, 4(2):131–140, 2013.

[19] S. Kamyab, M. Eftekhari, and J. R. Anaraki. A novel rough set based dissimilarity measure and its application in multimodal optimization. In *Artificial Intelligence and Signal Processing (AISP), 2012 16th CSI International Symposium on*, pages 180–185, May 2012.

[20] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, volume 4, pages 1942–1948 vol.4, Nov 1995.

[21] J. Komorowski, Z. Pawlak, L. Polkowski, and A. Skowron. Rough sets: A tutorial. In S. K. Pal and A. Skowron, editors, *Rough-Fuzzy Hybridization: A New Trend in Decision Making*, pages 3–98. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1998.

[22] M. Lichman. UCI machine learning repository, 2013.

[23] S.-W. Lin, K.-C. Ying, C.-Y. Lee, and Z.-J. Lee. An intelligent algorithm with feature selection and decision rules applied to anomaly intrusion detection. *Applied Soft Computing*, 12(10):3285 – 3290, 2012.

[24] K. Manimala, K. Selvi, and R. Ahila. Hybrid soft computing techniques for feature selection and parameter optimization in power quality data mining. *Applied Soft Computing*, 11(8):5485 – 5497, 2011.

[25] S. Nemati, M. E. Basiri, N. Ghasem-Aghaee, and M. H. Aghdam. A novel acoga hybrid algorithm for feature selection in protein function prediction. *Expert Systems with Applications*, 36(10):12086 – 12094, 2009.

[26] R. Nock and M. Sebban. Sharper bounds for the hardness of prototype and feature selection. In H. Arimura, S. Jain, and A. Sharma, editors, *Algorithmic Learning Theory*, volume 1968 of *Lecture Notes in Computer Science*, pages 224–238. Springer Berlin Heidelberg, 2000.

[27] Z. Pawlak. Rough sets. *International Journal of Computer & Information Sciences*, 11(5):341–356, 1982.

[28] P. Pudil, J. Novoviov, and P. Somol. Feature selection toolbox software package. *Pattern Recognition Letters*, 23(4):487 – 492, 2002.

[29] A. M. Radzikowska and E. E. Kerre. A comparative study of fuzzy rough sets. *Fuzzy Sets and Systems*, 126(2):137 – 155, 2002.

[30] A. S. Reddy and K. Vaisakh. Environmental constrained economic dispatch by modified shuffled frog leaping algorithm. *Journal of Bioinformatics and Intelligent Control*, 2(3):216–222, 2013.

[31] S. Saha, R. Spandana, A. Ekbal, and S. Bandyopadhyay. Simultaneous feature selection and symmetry based clustering using multiobjective framework. *Applied Soft Computing*, 29:479 – 486, 2015.

[32] M. Sebban and R. Nock. A hybrid filter/wrapper approach of feature selection using information theory. *Pattern Recognition*, 35(4):835 – 846, 2002.

[33] N. Sreeja and A. Sankar. Pattern matching based classification using ant colony optimization based feature selection. *Applied Soft Computing*, 31:91 – 102, 2015.

[34] N. Suguna and K. Thanushkodi. A novel rough set reduct algorithm for medical domain based on bee colony optimization. *Journal of Computing*, 2(6):49–54, June 2010.

[35] K. Thangavel and A. Pethalakshmi. Dimensionality reduction based on rough set theory: A review. *Applied Soft Computing*, 9(1):1 – 12, 2009.

[36] A. Tsanas, M. A. Little, C. Fox, and L. O. Ramig. Objective automatic assessment of rehabilitative speech treatment in parkinson's disease. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 22(1):181–190, 2014.

[37] P. Van Der Putten and M. van Someren. Coil challenge 2000: The insurance company case. *Published by Sentient Machine Research, Amsterdam. Also a Leiden Institute of Advanced Computer Science Technical Report*, 9:1–43, 2000.

[38] A. Verikas, M. Bacauskiene, D. Valincius, and A. Gelzinis. Predictor output sensitivity and feature similarity-based feature selection. *Fuzzy Sets and Systems*, 159(4):422 – 434, 2008.

[39] S. M. Vieira, L. F. Mendona, G. J. Farinha, and J. M. Sousa. Modified binary {PSO} for feature selection using {SVM} applied to mortality prediction of septic patients. *Applied Soft Computing*, 13(8):3494 – 3504, 2013.

[40] S. M. Vieira, J. M. Sousa, and T. A. Runkler. Two cooperative ant colonies for feature selection using fuzzy models. *Expert Systems with Applications*, 37(4):2714 – 2723, 2010.

[41] X. Wang, J. Yang, X. Teng, W. Xia, and R. Jensen. Feature selection based

on rough sets and particle swarm optimization. *Pattern Recognition Letters*, 28(4):459 – 471, 2007.

[42] K.-C. Wong, C.-H. Wu, R. K. Mok, C. Peng, and Z. Zhang. Evolutionary multimodal optimization using the principle of locality. *Information Sciences*, 194:138 – 170, 2012.

[43] J. Wróblewski. Finding minimal reducts using genetic algorithms. In *Proccedings of the second annual join conference on infromation science*, pages 186–189, 1995.

[44] J. Xiang, X. Han, F. Duan, Y. Qiang, X. Xiong, Y. Lan, and H. Chai. A novel hybrid system for feature selection based on an improved gravitational search algorithm and k-nn method. *Applied Soft Computing*, 31:293 – 307, 2015.

[45] Z. Xu, G. Huang, K. Q. Weinberger, and A. X. Zheng. Gradient boosted feature selection. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 522–531. ACM, 2014.

[46] S. C. Yusta. Different metaheuristic strategies to solve the feature selection problem. *Pattern Recognition Letters*, 30(5):525 – 534, 2009.

[47] X. Zhao, D. Li, B. Yang, C. Ma, Y. Zhu, and H. Chen. Feature selection based on improved ant colony optimization for online detection of foreign fiber in cotton. *Applied Soft Computing*, 24:585 – 596, 2014.

Chapter 5

Privacy-preserving Feature Selection: A Survey and Proposing a New Set of Protocols

This paper is submitted to the Journal of Data & Knowledge Engineering, 2017.

5.1 Abstract

Feature selection is the process of sieving features, in which informative features are separated from the redundant and irrelevant ones. This process plays an important role in machine learning, data mining and bioinformatics. However, traditional feature selection methods are only capable of processing centralized datasets and are not able to satisfy today's distributed data processing needs. These needs require a new category of data processing algorithms called privacy-preserving feature se-

lection, which protects users' data by not revealing any part of the data neither in the intermediate processing nor in the final results. This is vital for the datasets which contain individuals' data, such as medical datasets. Therefore, it is rational to either modify the existing algorithms or propose new ones to not only introduce the capability of being applied to distributed datasets, but also act responsibly in handling users' data by protecting their privacy. In this paper, we will review three privacy-preserving feature selection methods and provide suggestions to improve their performance when any gap is identified. We will also propose a privacy-preserving feature selection method based on the rough set feature selection. The proposed method is capable of processing both horizontally and vertically partitioned datasets in two- and multi-parties scenarios.

Index terms— Privacy-preserving, feature selection, rough set theory

5.2 Introduction

Collecting and accumulating data in a systematic way, such as in datasets and data tables, for further processing, is very important for any organization, department and, in a broader view, to any country. A dataset is composed of several data tables, in which each table contains several columns that correspond to variables (also called features or attributes) and rows which represent records (also called samples or objects). In machine learning, each dataset contains only one data table, and dataset and data table concepts are used interchangeably. As is shown in Table 5.1, for each row of a dataset, different variables are measured and inserted into the provided cells. Later, all the collected data are processed for a variety of purposes using different sta-

tistical and mathematical methods. Table 5.1 shows a portion of Haberman's Survival dataset adopted from the UCI repository of machine learning [21]. In this dataset, each row represents a single patient, and columns show the age, year of operation and the number of auxiliary nodes of each patient. The last column presents a class label for each patient to show whether they have survived for five years or longer (represented by 1) or not (represented by 2).

Table 5.1: Partial View of Haberman's Survival Dataset

Age	**Year**	**Auxiliary nodes**	**Class**
⋮	⋮	⋮	⋮
42	61	4	1
42	62	20	1
42	65	0	1
42	63	1	1
43	58	52	2
43	59	2	2
43	64	0	2
43	64	0	2
43	63	14	1
43	64	2	1
⋮	⋮	⋮	⋮

Since the collected data can be categorized either as sensitive (e.g., medical, financial, military) or non-sensitive (e.g., publicly available data, the UCI datasets)

data, the algorithms applied should be selected accordingly, so that the data can be accessed only as is appropriate. With the dramatic increase in the amount of information generated annually, privacy challenges are becoming a serious issue for governments and health related organizations. Therefore, many countries are investing heavily in designing, implementing and applying privacy-preserving methods [15].

In US law, *privacy* is the right "to be let alone" [10] and should be protected by taking proper actions [26]. In computer science, *privacy* of individuals deals with deciding how one's information will be used. For instance, someone's health information should be kept secure and be shared only with physicians who have been chosen by the patient. These concerns necessitate a category of data-mining methods called privacy-preserving data-mining. "Privacy-preserving data-mining" refers to knowledge extraction techniques specific to privacy criteria. The main goal of these processes is to introduce a trade-off between accuracy and the amount of information revealed publicly. Generally speaking, the amount of raw data produced is much greater than the information that needs to be extracted from them. Therefore, more efforts and time are needed to process, save and maintain those data for later processing (such as classification or clustering). Many problems in machine-learning, data-mining and pattern recognition involve big datasets. A high-dimensional dataset (e.g., DNA microarray data), in terms of number of features and samples, requires a huge effort to be processed. Therefore, feature selection (FS) methods are used to effectively reduce the size of datasets (in one direction) by selecting only the most relevant columns. These methods select the most informative features, which are highly correlated with the outcome and loosely dependent on other features, so as to minimize further processing. Since the size of datasets can also be reduced in terms of

number of samples, sample selection (SS) methods have emerged to reduce the size of datasets by removing irrelevant samples. By employing FS and SS methods, dataset dimensions can be lowered and further processing can become more efficient.

In this paper, existing feature selection algorithms which consider privacy concerns as well as the application in distributed datasets will be investigated. We will also perform a thorough comparison from different aspects, such as performance, applicability, security and privacy. The rest of this paper is organized as follows: Section 2 depicts background and Section 3 surveys related work. Section 4 discusses the proposed approaches and Sections 5 concludes the paper.

5.3 Background

Vast amounts of research have been conducted in different areas of data-mining and machine-learning to satisfy the need for protecting individuals' privacy [22, 2, 25]. Surprisingly however, feature selection methods have not kept up with the developing need for privacy and security. Feature selection is the process of purifying data by retaining the most informative features while omitting the others. The important role of feature selection methods in reducing model complexity for further processing is undeniable. Each dataset contains three types of features: informative, redundant and irrelevant. The most informative, non-redundant relevant features convey sufficient amounts of information for the outcome. Redundant features contain chunks of information that are indistinguishable from other similar features and can be removed. Features belonging to the last type are unnecessary (such as a feature with constant value for all examples) and can be eliminated due to not having any information for

the classification outcome.

5.3.1 Feature Selection

Before looking at privacy-preserving aspects of data-mining, we will review some of the existing feature selection methods. Feature selection methods have been divided into two main groups: feature ranking and feature subset selection [12]. The former is a set of methods that rank features based on some specific measure values and select the top n number of features. The latter evaluates subsets of features and selects the one with the highest fitness value. Either of the aforementioned groups can be addressed using filter-based or wrapper-based approaches [19]. In the filter-based approach, a merit evaluates the quality of every feature regardless of its impact on the outcome, while wrapper-based approaches measure the effectiveness of features based on the results of already chosen classifiers. Wrapper-based methods are highly computationally-intensive and powerful in predicting the outcome compared to filter-based methods, which are faster but potentially inaccurate.

One of the most well-known feature selection methods is Relief [18], which measures the relevancy of a feature compared to other features of the same and different classes by calculating their Euclidean distance. Hall [13] has proposed a merit based on the average intra-correlation of features and inter-correlation of features to the outcome. This merit selects features that are highly correlated to the outcome while lowly correlated to the other features. Jensen et al. [17] have introduced a novel feature selection method based on the lower approximation of a fuzzy-rough set, in which dependency of the features to the outcome is calculated using a merit called

dependency degree (DD). Fuzzy-rough DD selects a new feature if it improves the discernibility power of the already selected features toward distinction of different classes of the outcome. Anaraki et al. [3] have developed a simple control criterion for the conventional fuzzy-rough feature selection (FRFS) to direct the process of adding features to the reduct set by considering a lower bound for the distinguishability power of the feature being considered. Also, they have reviewed and surveyed different methods proposed in rough set feature selection (RSFS) in [4]. Anaraki et al. [5] have introduced the following two modifications of FRFS to improve the performance of the conventional method: guiding the selection process in *equal* situations, where diverse subsets with only one different feature result in identical DD, and integrating the first improvement with the criterion that stops it [3]. Figure 5.1 shows an *equal* situation for subsets $\{b, a\}$ and $\{b, c\}$, in which the two sets differ by one member ($\{a\}$ and $\{c\}$) and for both $DD = 0.34$.

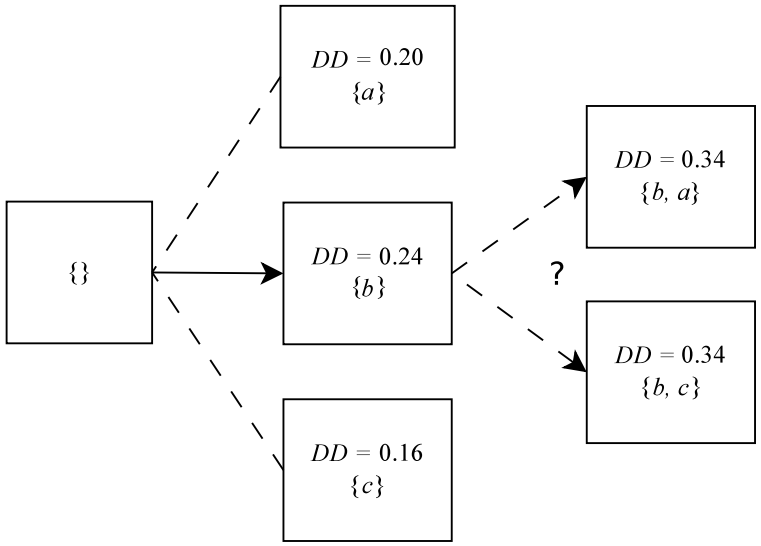


Figure 5.1: *Equal* situation

5.3.2 Privacy-Preserving Data-Mining

There are two different approaches to privacy-preserving data-mining: methods to perturb data before publishing which are called randomization, and methods to perform mathematical operations securely which are called secure multi-party computation (SMC). Figure 5.2 shows how data are represented to privacy-preserving data-mining methods.

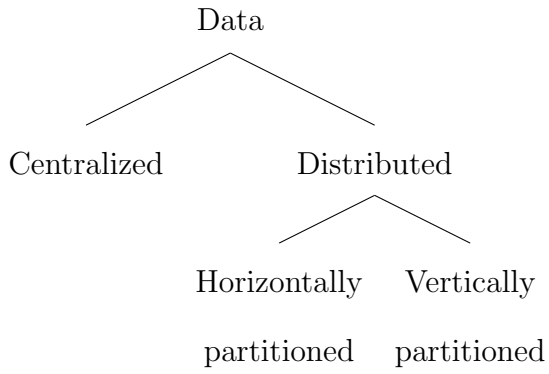


Figure 5.2: Data representation in privacy-preserving data-mining

In 2000, interestingly, two papers with the identical title of *Privacy-Preserving Data-Mining* were published [2, 22]. Agrawal and Srikant [2] proposed a secure decision tree classifier, which can be applied to the perturbed and randomized data by reconstructing distribution using a Bayesian procedure. In the other paper, Lindell and Pinkas [22] proposed a secure protocol for the ID3 classifier on two-party horizontally partitioned data. The core of their method is a secure version of $xlnx$ in which x is a two-party distributed data. Since the publication of these two seminal works, many protocols and methods have been proposed using both approaches for various data-mining, machine-learning and statistical analysis methods and algorithms.

5.3.2.1 Centralized

Randomization In this approach, data is centralized and the data owner wishes to publish their data for mining purposes. To do so, the data should be perturbed using randomization techniques before being transmitted, and are then reconstructed at the destination. The main challenge in randomization is the trade-off between privacy and accuracy.

Features in privacy-preserving data-mining are divided into three main categories: explicit identifiers (EI), quasi identifiers (QI) and sensitive identifiers (SI). Explicit identifiers are those features of a dataset which promptly reveal individuals' identification, such as name and medical care plan (MCP) number. Theses features should be removed to protect an individual's privacy. Quasi identifiers are those features which could be combined with publicly available data such as Netflix movies ranking to identify individuals. In 2006, Netflix released information on 100 million ratings to a competition called Netflix Prize to challenge researchers in order to find the best algorithm for predicting user ratings [8]. However, a few months later Netflix ratings were linked to the internet movie database (IMDB) ratings and individuals were identified [23]. Sensitive identifiers refer to that information which is private to some individuals, such as disease information in medical datasets and should be also removed from the dataset [1].

In the case of having sensitive attributes, three methods have been proposed to protect individuals' privacy as follows:

1. k-anonymity: If each record in a dataset is indistinguishable from $(k-1)$ other records (see Table 5.2 adopted from [1])

2. l-diversity: If an equivalent class of a dataset has l diverse values for the sensitive attribute

3. t-closeness: If the distance of the distribution of a sensitive attribute value in an equivalent class to the distribution of the same attribute is less than t

Table 5.2: An example of 3-anonymized dataset

Row Index	**Age**	**ZIP Code**	**Disease**
1	[20, 30]	Northeastern US	HIV
2	[30, 40]	Western US	Hepatitis C
3	[20, 30]	Northeastern US	HIV
4	[30, 40]	Western US	Hepatitis C
5	[30, 40]	Western US	Diabetes
6	[20, 30]	Northeastern US	HIV

5.3.2.2 Distributed

Secure Multi-Party Computation In SMC, *secure* mathematical and statistical computations are applied to different portions of data in the possession of different parties. This approach has the same results as non-secure algorithms; however, the main challenge in secure methods is the trade-off between security and efficiency. In an n-party environment, datasets are divided into n chunks and all parties demanding of running a specific mining algorithm (e.g., classification) or statistical analysis (e.g. correlation coefficient) on all n chunks as a single dataset without revealing any private

information to the others.

Data in SMC can be partitioned either *vertically* or *horizontally* (see Figure 5.3) and depending on how the data are partitioned, "partition-specific" methods need to be applied. It is worth mentioning that data partitioning here is different from database partitioning in the distributed database management system [7], in which the main goal is to improve performance. In *vertical* partitioning, each party (e.g., different departments of a store) might posses a subset of features (e.g., purchased items from a specific department) while accommodating all samples (e.g., customers). In *horizontal* partitioning, each party might posses a subset of samples while accommodating all features. For example, Hospital A in Newfoundland and Labrador, Hospital B in Ontario and Hospital C in British Columbia have a Haberman's Survival dataset (as shown in Table 5.1) of people in their provinces. So, they all share the same structure and features for their datasets, but different records.

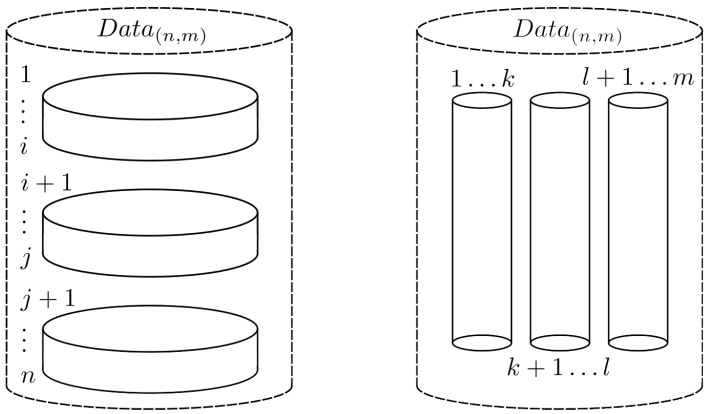


Figure 5.3: Horizontally and vertically partitioned data

5.4 Related Work

In this section we will discuss three privacy-preserving feature selection methods which have been introduced in the last seven years. Each method is presented in detail and in case of identifying any gap, some comments are provided to improve the performance of the proposed methods.

Jafer et al. [16] have proposed a privacy-aware filter-based feature selection that probes the inter-correlation of features to remove quasi-identifier (QI) features. In their paper, the authors introduce a system which contains two separate blocks: one for evaluating features, and the other one for controlling the privacy aspects of feature selection. In the former, features are ranked based on InfoGain [14] and Relief criteria [18]. In the latter, the list is traversed from bottom to top and correlation of QI features and non-QI features is calculated. By referring to the controlling values of the Correlation Block, features are selected or discarded. We have adopted Figure 5.4 from [16] to illustrate the proposed system.

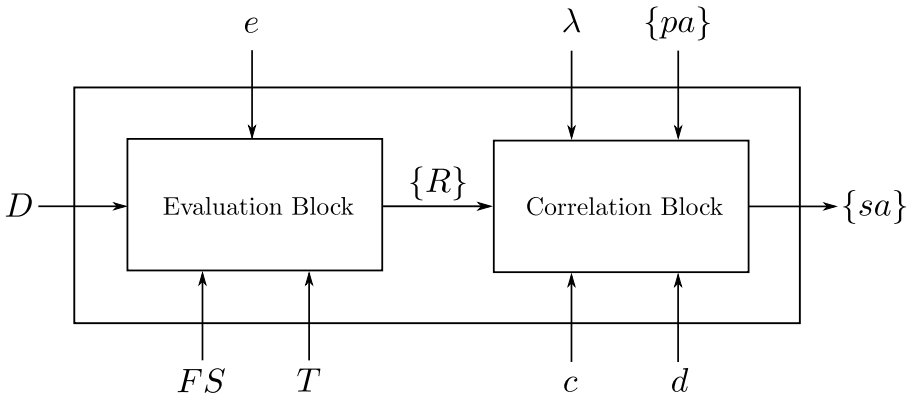


Figure 5.4: Privacy-aware filter-based feature selection

The Evaluation Block accepts dataset D, ranker threshold T, evaluation measure

e, such as information gain and chi-square, and the type of feature selection FS, such as feature subset as inputs and $\{R\}$ is the ranked list features in the intermediate output. Later, the Correlation Block takes the ranked list $\{R\}$, the privacy-breaching attributes $\{pa\}$, the correlation measure c, such as symmetric uncertainty, the discretization factor d for converting continuous values of the features to discrete ones as the requirement of correlation measure, and the balancing parameter between privacy and accuracy λ, in order to produce the set of selected attributes $\{sa\}$ as the final output. The balancing parameter λ varies from zero to one, in which moving from zero to one increases the number of the selected QI features.

After evaluating features using e, features are sorted in descending order list. Then, the correlation of the QI features in the list is calculated from bottom to up against other features using c, and those QI features which have a correlation greater than λ will be removed. This process repeats until all QI features are investigated. This method guarantees to preserve privacy and select the most important features; however, there are four concerns about the method as follows:

1. Inter-correlation of the features and the class have not been investigated
2. In case of having two perfectly correlated QI and non-QI features, only the QI feature is removed
3. The case where more than one perfectly correlated non-QI feature to a QI feature exists has not been discussed
4. Security and complexity analysis of the proposed method are missing

Banerjee and Chakravarty [6] have developed a distributed privacy-preserving

method based on the virtual dimensionality reduction method in image processing [9] to select features. This method takes advantage of correlation and covariance eigenvalues to perform feature selection on both horizontally and vertically partitioned data. It starts with calculating the correlation and the covariance matrices of a dataset, and continues with computing the eigenvalues of both of the matrices. Then, for each feature, the corresponding correlation-eigenvalue and covariance-eigenvalue is subtracted. If the resulting value is greater than a user-specified threshold δ then the feature is kept in the reduct subset, otherwise, it will be discarded. This process continues until all features of the dataset are examined.

For both horizontally and vertically partitioned data, the correlation and the covariance are calculated securely over all parties; however, the eigenvalue decomposition is done locally to reduce communication costs. In both scenarios, the threshold δ, number of rows of dataset D as $N(D)$, summation of feature j values as $FS_j(D)$, standard deviation of each feature $\sigma(j)$, and sum of product of values of features i and j as $SS_{ij}(D)$ should be calculated. Finally, the covariance and the correlation between the two features i and j are computed as shown in Equation 5.1 and 5.2, respectively.

$$COV(i,j) = \frac{SS_{ij}(D)}{N(D)} - \frac{FS_i(D) \times FS_j(D)}{N(D)^2}, \tag{5.1}$$

$$CORR(i,j) = \frac{COV(i,j)}{\sigma(i) \times \sigma(j)}. \tag{5.2}$$

For horizontally partitioned data, each party p calculates $N^p(D)$, $FS_j^p(D)$, $\sigma(j)^p$ and $SS_{ij}^p(D)$, and then they apply secure sum protocol to calculate the aggregation

results. Finally, each party performs feature selection based on the resulting eigenvalues of the calculated covariance and the correlation. With n number of records, m features and p parties, the communication cost would be $O(mp)$, since the only secure operation used in the horizontally partitioned data is secure sum.

For vertically partitioned data, each party can calculate $N(D)$, $FS_j(D)$, $\sigma(j)$ locally; however, the calculation of $SS_{ij}^{p}(D)$ depends on whether both attributes of i and j are in the same partition or not. If so, the calculation is straight forward. Otherwise, both parties should use secure dot product to calculate $SS_{ij}(D)$. The communication cost of the vertically partitioned data is $O(m^2np)$, which is mainly imposed by the secure dot product operation.

Das [11] et al. have introduced three asynchronous feature selection methods based on the misclassification gain, Gini index and entropy measures for binary-class datasets with categorical features in horizontally partitioned fashion. The main requirement for the proposed methods is a P2P network with a structured ring-based topology. The distributed setup of each measure (i.e. misclassification gain, Gini index and entropy) to evaluate every feature A_i are shown in Equations 5.3, 5.4 and 5.5, respectively.

$$\sum_{a=0}^{m_i-1} \left| \sum_{l=1}^{d} \left\{ x_{i,a0}^{(l)} - x_{i,a1}^{(l)} \right\} \right|, \tag{5.3}$$

where each feature A_i can take a value from $\{0, \ldots, m_i - 1\}$, d is the number of peers, and $x_{i,a0}^{(l)}$ and $x_{i,a1}^{(l)}$ are the number of examples with the value of $A_i = a$ and class

value of 0 and 1, respectively.

$$\sum_{a=0}^{m_i-1}\left\{\frac{\left(\sum_{l=1}^{d} x_{i,a0}^{(l)}\right)^2+\left(\sum_{l=1}^{d} x_{i,a1}^{(l)}\right)^2}{\sum_{l=1}^{d} x_{i,a}^{(l)}}\right\}, \tag{5.4}$$

where $x_{i,a}^{(l)}$ is the number of examples with $A_i = a$.

$$\sum_{a=0}^{m_i-1}\left\{\left(\sum_{l=1}^{d} x_{i,a0}^{(l)}\right)\log\left(\frac{\sum_{l=1}^{d} x_{i,a0}^{(l)}}{\sum_{l=1}^{d} x_{i,a}^{(l)}}\right)+\left(\sum_{l=1}^{d} x_{i,a1}^{(l)}\right)\log\left(\frac{\sum_{l=1}^{d} x_{i,a1}^{(l)}}{\sum_{l=1}^{d} x_{i,a}^{(l)}}\right)\right\}. \tag{5.5}$$

For computing misclassification gain, Gini index, and entropy across all peers, each peer P_i estimates Equation 5.3, 5.4, and 5.5 for feature A_i when it takes value a, respectively. This process starts from an initiator and continues by each peer with adding their value to the received data. When the initiator receives the data, it calculates the average using the asymmetric network topology version (see Equation 5.6) of the method proposed by Scherber and Papadopoulos [27].

$$z_i^{(t)} = \{1-2\rho|\Gamma_{i,1}|-\rho(n_i^*-|\Gamma_{i,1}|)\}z_i^{(t-1)}+2\rho\sum_{q\in\Gamma_{i,1}} z_q^{(t-1)}+\rho\sum_{q=1}^{n_i^*-|\Gamma_{i,1}|} z_q^{(t-1)}, \tag{5.6}$$

where $z_i^{(t)}$ is an estimate of average at the time t by ith peer, ρ is the rate of convergence, $|\Gamma_{i,1}|$ is the size of set of neighbours of peer i in hop-distance one, and n_i^* is the size of the ring formed by peer i.

To establish a trade-off between privacy and the cost of computations, the authors have introduced an objective function for each peer i as follows:

$$f_i^{\text{obj}} = w_{ti} \times \text{threat} - w_{ci} \times \text{cost},$$

where w_{ti} and w_{ci} are the weights for *thread* and *cost*, *threat* is a measure which represents the risk that each peer might take by participating in the current computation,

and *cost* includes both computation and communication costs. The time complexity for the proposed methods based on three measures is $O(max(n_i^*, n_j^*))$, where n_i^* is the optimal value for peer P_i and n_j^* is the value for the neighbour P_j in the same ring.

In this section, we have discussed three privacy-preserving feature selection methods, each trying to address feature selection issues in distributed datasets through decentralized computation.

5.5 Discussion and Contribution

All the discussed methods have provided a variety of secure protocols for different data configurations (i.e. horizontally and vertically partitioned data) to preserve individuals' privacy. The proposed methods have been backed up with security, computational and communication complexity analyses. However, there are obvious limitations for the mentioned privacy-preserving feature selection methods from privacy-preserving aspects. On one side, they can be applied to either horizontally or vertically partitioned data which limits users by imposing specific data configuration. On the other side, they are not suited for both two- and multi-party data configurations. To address the mentioned deficiencies we are providing four privacy-preserving versions of rough set feature selection [20] for both horizontally and vertically partitioned datasets as follows:

- Two-party with horizontally partitioned data (2P-HP)
- Multi-party with horizontally partitioned data (MP-HP)
- Two-party with vertically partitioned data (2P-VP)

- Multi-party with vertically partitioned data (MP-VP)

5.6 Proposed method

Rough set theory was proposed by Pawlak as a tool for dealing with uncertainty [24]. Data in rough set theory are organized in a decision table. Table 5.3 shows a decision table adopted from [20]. Class attribute is called decision attribute and the rest are condition attributes. In Table 5.3, $\{Walk\}$ is a decision attribute and $\{Age, LEMS\}$ are condition attributes.

Table 5.3: An example of decision table

Object	**Age**	**LEMS**	**Walk**
x_1	16-30	50	Yes
x_2	16-30	0	No
x_3	31-45	1-25	No
x_4	31-45	1-25	Yes
x_5	46-60	26-49	No
x_6	16-30	26-49	Yes
x_7	46-60	26-49	No

Let $\mathbb{U} = \{x_1, x_2, \ldots, x_7\}$ be the universe of discourse and let R be the equivalence relation on $\mathbb{U}$, approximation space is shown by $(\mathbb{U}, R)$. Set of all attributes are shown by $A = \{AGE, LEMS, WALK\}$, set of all conditional attributes by $C = \{Age, LEMS\}$ and set of decision attribute(s) or class attribute(s) by

$D = \{WALK\}$. Let X be a subset of $\mathbb{U}$ and P to be a subset of A, approximating this subset using rough set theory is done by means of upper and lower approximations. Upper approximation of X with regards to ($\overline{P}X$) contains objects which are possibly classified in X regarding the attributes in P. Objects in lower approximation ($\underline{P}X$) are the ones which are surely classified in X regarding the attributes in P. Boundary region of X can be determined by subtracting upper approximation from lower approximation and where it is a non-empty set, X is called a rough set otherwise it is a crisp set. Rough set is shown by an ordered pair ($\overline{P}X$, $\underline{P}X$). Different regions are defined using this pair as below:

$$POS_P(Q) = \bigcup_{X \in \mathbb{U}/Q} \underline{P}X \tag{5.7}$$

$$NEG_P(Q) = \mathbb{U} - \bigcup_{X \in \mathbb{U}/Q} \overline{P}X \tag{5.8}$$

$$BND_P(Q) = \bigcup_{X \in \mathbb{U}/Q} \overline{P}X - \bigcup_{X \in \mathbb{U}/Q} \underline{P}X \tag{5.9}$$

Positive region of partition $\mathbb{U}/Q$ is a set of all objects which can be uniquely classified into blocks of partition by means of P. Negative region is a set of objects which cannot be classified to the partition $\mathbb{U}/Q$[20].

Finding dependency between attributes is one of the most important areas in data analysis. Let P and Q be subsets of A, dependency of Q on P are denoted by $P \Rightarrow_k Q$ and $k = \gamma_p(Q)$, in which γ is dependency degree [20]. If $k = 1$, Q depends totally on P and if $k < 1$, Q depends partially on P.

Value of k is a measure of dependency between features. In feature selection, those

features which are loosely dependent on each other and highly correlated to decision feature are desired. If Q totally depends on P, it means that the partition which is made by P is finer than Q. Calculating dependency is shown in Equation 5.10.

$$k = \gamma_P(Q) = \frac{|POS_P(Q)|}{|\mathbb{U}|} \tag{5.10}$$

The notation $|.|$ is used for cardinality. Positive region of the partition $\mathbb{U}/Q$ with respect to P which is denoted by γ, is the set of all elements which can be classified to partition $\mathbb{U}/Q$ using P [20]. Reduct is a subset of features which has the same dependency degree as employing all features for classification. Features which belong to the reduct set are information-rich and the others are irrelevant and redundant.

The QuickReduct algorithm which is given in [17] and depicted in Algorithm 5.1, calculates a reduct without finding all the subsets. It starts from an empty set and each time selects a feature which causes greatest increase in dependency degree. The algorithm stops when adding more features does not increase the dependency degree. It does not guarantee to find minimal reduct as long as it employs greedy forward search algorithm, which is vulnerable to local optimum.

The QuickReduct algorithm has been applied to the example dataset in Table 5.3. The algorithm starts by calculating dependency of the outcome $\{WALK\}$ to each conditional features $\{Age, LEMS\}$ as shown in Equation 5.11.

Algorithm 5.1: QuickReduct algorithm

C, the set of all conditional attributes;

D, the set of decision attributes;

$R \leftarrow \{\}; \gamma'_{best} = 0; \gamma'_{prev} = 0;$

do

- $T \leftarrow R;$
- $\gamma'_{prev} \leftarrow \gamma'_{best};$
- **for** $x \in (C - R)$ **do**
  - **if** $\gamma'_{R \cup \{x\}}(D) > \gamma'_T(D)$ **then**
    - $T \leftarrow R \cup \{x\};$
    - $\gamma'_{best} \leftarrow \gamma'_T(D);$
  - $R \leftarrow T;$

while $\gamma'_{best} = \gamma'_{prev};$

return $R;$

$$\begin{aligned} \gamma_{\{Age\}}(Walk) &= \frac{|POS_{\{Age\}}(Walk)|}{|\{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}|} \\ &= \frac{|\bigcup_{X \in \{x_1,x_2,x_3,x_4,x_5,x_6,x_7\}/Walk} \underline{AgeX}|}{|\{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}|} \\ &= \frac{|\{x_5, x_7\}|}{|\{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}|} = \frac{2}{7} \end{aligned} \tag{5.11}$$

$$\gamma_{\{LEMS\}}(Walk) = \frac{|POS_{\{LEMS\}}(Walk)|}{|\{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}|} = \frac{|\bigcup_{X \in \{x_1,x_2,x_3,x_4,x_5,x_6,x_7\}/Walk} \underline{LEMS}X|}{|\{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}|} = \frac{|\{x_1, x_2\}|}{|\{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}|} = \frac{2}{7}$$

Since, the dependency degree of $\{LEMS\}$ is equal to $\{Age\}$, either of them can be selected and added to the reduct set R. This process continues by selecting $\{LEMS\}$ and adding $\{Age\}$ to the reduct set; the dependency degree of the set is calculated as shown in Equation 5.12.

$$\gamma_{\{Age,LEMS\}}(Walk) = \frac{|POS_{\{Age,LEMS\}}(Walk)|}{|\{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}|} = \frac{|\bigcup_{X \in \mathbb{U}/Walk} \underline{Age, LEMS}X|}{|\{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}|} = \frac{|\{x_1, x_2, x_5, x_6, x_7\}|}{|\{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}|} = \frac{5}{7} \tag{5.12}$$

As the resulting dependency of having both features in the reduct set is greater than dependency degree of $R = \{LEMS\}$; therefore, the final result of QuickReduct algorithm is $R = \{Age, LEMS\}$.

5.6.1 Two parties with horizontally partitioned data (2P-HP)

As an illustrative example, Table 5.3 has been partitioned horizontally into two datasets $\mathbb{D}_1^H$ and $\mathbb{D}_2^H$ in possession of two parties $\mathbb{P}_1$ and $\mathbb{P}_2$, as shown in Tables 5.4 and 5.5, respectively.

Table 5.4: The first partition of horizontally partitioned $\mathbb{D}_1^H$

Object	**Age**	**LEMS**	**Walk**
x_1	16-30	50	Yes
x_2	16-30	0	No
x_3	31-45	1-25	No
x_4	31-45	1-25	Yes

Table 5.5: The second partition of data $\mathbb{D}_2^H$

Object	**Age**	**LEMS**	**Walk**
x_5	46-60	26-49	No
x_6	16-30	26-49	Yes
x_7	46-60	26-49	No

In order to uncover the required secure mathematical equations of 2P-HP for calculating dependency degree of each partitioned data $\gamma_P(Q)_{\mathbb{D}_1^H}$, the results of applying QuickReduct algorithm on each partition based on conditional feature $\{Age\}$, is calculated in Equation 5.13.

$$
\begin{aligned}
\gamma_{\{Age\}}(Walk)_{\mathbb{D}_1^H} &= \frac{|POS_{\{Age\}}(Walk)_{\mathbb{D}_1^H}|}{|\{x_1, x_2, x_3, x_4\}|} \\
&= \frac{|\bigcup_{X \in \{x_1, x_2, x_3, x_4\}/Walk} \underline{Age}X|}{|\{x_1, x_2, x_3, x_4\}|} \\
&= \frac{|\{\}|}{|\{x_1, x_2, x_3, x_4\}|} = \frac{0}{4}
\end{aligned}
\tag{5.13}
$$

$$
\begin{aligned}
\gamma_{\{Age\}}(Walk)_{\mathbb{D}_2^H} &= \frac{|POS_{\{Age\}}(Walk)_{\mathbb{D}_2^H}|}{|\{x_5, x_6, x_7\}|} \\
&= \frac{|\bigcup_{X \in \{x_5, x_6, x_7\}/Walk} \underline{Age}X|}{|\{x_5, x_6, x_7\}|} \\
&= \frac{|\{x_5, x_6, x_7\}|}{|\{x_5, x_6, x_7\}|} = \frac{3}{3}
\end{aligned}
$$

By referring to the resulting dependency degree of conditional feature $\{Age\}$ for partition $\mathbb{D}_1^H$ and $\mathbb{D}_2^H$; it can be understood that the overall dependency degree of conditional feature $\{Age\}$ cannot be calculated by simply adding corresponding numerator and denominator of $\gamma_{\{Age\}}(Walk)_{\mathbb{D}_1^H}$ to $\gamma_{\{Age\}}(Walk)_{\mathbb{D}_2^H}$ as shown in Equation 5.14.

$$
\begin{aligned}
\gamma_{\{Age\}}(Walk) &= \gamma_{\{Age\}}(Walk)_{\mathbb{D}_1^H} + \gamma_{\{Age\}}(Walk)_{\mathbb{D}_2^H} \\
&= \frac{|POS_{\{Age\}}(Walk)_{\mathbb{D}_1^H}|}{|\{x_1, x_2, x_3, x_4\}| + |\{x_5, x_6, x_7\}|} \\
&+ \frac{|POS_{\{Age\}}(Walk)_{\mathbb{D}_2^H}|}{|\{x_1, x_2, x_3, x_4\}| + |\{x_5, x_6, x_7\}|} \\
&= \frac{0}{4+3} + \frac{3}{4+3} \\
&= \frac{3}{7}
\end{aligned}
\tag{5.14}
$$

The reason is that, object x_6 is in $POS_{\{Age\}}(Walk)_{\mathbb{D}_2^H}$; whereas, if the whole feature $\{Age\}_{\mathbb{D}_1^H}$ and $\{Age\}_{\mathbb{D}_2^H}$ is considered, it is indiscernible with x_1 and x_2, which would prevent x_6 from being a member of $POS_{\{Age\}}(Walk)$ and the final dependency degree would be correct. To overcome this issue, a secure comparison is needed to compare each features' values of each partition with the other one.

All objects in positive region of each party, should be compared with the objects in the other party to decide on indiscernibilities. In case of any occurrence, numerator of dependency degree should be decreased by one.

This process starts from $\mathbb{D}_1^H$ by checking all objects in $POS_{\{Age\}}(Walk)_{\mathbb{D}_1^H}$; since there is no object in the set, the process proceeds to $\mathbb{D}_2^H$. In the second data partition, three objects have been recognized as members of $POS_{\{Age\}}(Walk)_{\mathbb{D}_2^H}$; therefore, this non-empty set leads to the commence of the secure comparison process. The secure comparison of object x_6 in $\mathbb{D}_2^H$ with the objects x_1 and x_2 in $\mathbb{D}_1^H$ recognizes three objects as indiscernible; therefore, the dependency degree in Equation 5.14 should be decreased by one as shown in Equation 5.15.

$$
\begin{aligned}
\gamma_{\{Age\}}(Walk) &= \gamma_{\{Age\}}(Walk)_{\mathbb{D}_1^H} \\
&+ \gamma_{\{Age\}}(Walk)_{\mathbb{D}_2^H} \\
&- IND_{\{Age\}}(Walk)_{\mathbb{D}} \\
&= \frac{|POS_{\{Age\}}(Walk)_{\mathbb{D}_1^H}|}{|\{x_1, x_2, x_3, x_4\}| + |\{x_5, x_6, x_7\}|} \\
&+ \frac{|POS_{\{Age\}}(Walk)_{\mathbb{D}_2^H}|}{|\{x_1, x_2, x_3, x_4\}| + |\{x_5, x_6, x_7\}|} \\
&- \frac{1}{|\{x_1, x_2, x_3, x_4\}| + |\{x_5, x_6, x_7\}|} \\
&= \frac{0}{4+3} + \frac{3}{4+3} - \frac{1}{4+3} \\
&= \frac{2}{7}
\end{aligned}
\tag{5.15}
$$

where $IND_{\{Age\}}(Walk)_{\mathbb{D}}$ is the number of indiscernible objects in both partitions.

In order to decide which feature should be selected, the feature selection process continues with calculating the dependency degree of $\{LEMS\}$. The result of applying QuickReduct algorithm on each partition, individually is shown in Equation 5.16.

$$
\begin{aligned}
\gamma_{\{LEMS\}}(Walk)_{\mathbb{D}_1^H} &= \frac{|POS_{\{LEMS\}}(Walk)_{\mathbb{D}_1^H}|}{|\{x_1, x_2, x_3, x_4\}|} \\
&= \frac{|\bigcup_{X \in \{x_1, x_2, x_3, x_4\}/Walk} \underline{LEMS}X|}{|\{x_1, x_2, x_3, x_4\}|} \\
&= \frac{|\{x_1, x_2\}|}{|\{x_1, x_2, x_3, x_4\}|} = \frac{2}{4}
\end{aligned}
\tag{5.16}
$$

$$\begin{aligned}\gamma_{\{LEMS\}}(Walk)_{\mathbb{D}_2^H} &= \frac{|POS_{\{LEMS\}}(Walk)_{\mathbb{D}_2^H}|}{|\{x_5, x_6, x_7\}|} \\ &= \frac{|\bigcup_{X \in \{x_5,x_6,x_7\}/Walk} \underline{LEMS}X|}{|\{x_5, x_6, x_7\}|} \\ &= \frac{|\{\}|}{|\{x_5, x_6, x_7\}|} = \frac{0}{3}\end{aligned}$$

After calculating the dependency degree of the two parties, number of indiscernible objects should be calculated and subtracted from the final dependency degree. The final result is shown in Equation 5.17.

$$\begin{aligned}\gamma_{\{LEMS\}}(Walk) &= \gamma_{\{LEMS\}}(Walk)_{\mathbb{D}_1^H} \\ &\quad + \gamma_{\{LEMS\}}(Walk)_{\mathbb{D}_2^H} \\ &\quad - IND_{\{LEMS\}}(Walk)_{\mathbb{D}} \\ &= \frac{|POS_{\{LEMS\}}(Walk)_{\mathbb{D}_1^H}|}{|\{x_1, x_2, x_3, x_4\}| + |\{x_5, x_6, x_7\}|} \\ &\quad + \frac{|POS_{\{LEMS\}}(Walk)_{\mathbb{D}_2^H}|}{|\{x_1, x_2, x_3, x_4\}| + |\{x_5, x_6, x_7\}|} \\ &\quad - \frac{0}{|\{x_1, x_2, x_3, x_4\}| + |\{x_5, x_6, x_7\}|} \\ &= \frac{2}{4+3} + \frac{0}{4+3} - \frac{0}{4+3} \\ &= \frac{2}{7}\end{aligned} \tag{5.17}$$

By comparing and selecting feature with the highest dependency degree, the process of feature selection proceeds to the next level by calculating the dependency degree of the new set, which contains $R = \{Age, LEMS\}$ for each parties as shown

in Equation 5.18.

$$\begin{aligned}\gamma_{\{Age,LEMS\}}(Walk)_{\mathbb{D}_1^H} &= \frac{|POS_{\{Age,LEMS\}}(Walk)_{\mathbb{D}_1^H}|}{|\{x_1,x_2,x_3,x_4\}|} \\ &= \frac{|\bigcup_{X\in\{x_1,x_2,x_3,x_4\}/Walk}\underline{Age,LEMS}X|}{|\{x_1,x_2,x_3,x_4\}|} \\ &= \frac{|\{x_1,x_2\}|}{|\{x_1,x_2,x_3,x_4\}|} = \frac{2}{4}\end{aligned} \tag{5.18}$$

$$\begin{aligned}\gamma_{\{Age,LEMS\}}(Walk)_{\mathbb{D}_2^H} &= \frac{|POS_{\{Age,LEMS\}}(Walk)_{\mathbb{D}_2^H}|}{|\{x_5,x_6,x_7\}|} \\ &= \frac{|\bigcup_{X\in\{x_5,x_6,x_7\}/Walk}\underline{Age,LEMS}X|}{|\{x_5,x_6,x_7\}|} \\ &= \frac{|\{x_5,x_6,x_7\}|}{|\{x_5,x_6,x_7\}|} = \frac{3}{3}\end{aligned}$$

The final dependency degree for $R = \{Age, LEMS\}$ is calculated and illustrated in Equation 5.19.

$$\begin{aligned}\gamma_{\{Age,LEMS\}}(Walk) &= \gamma_{\{Age,LEMS\}}(Walk)_{\mathbb{D}_1^H} \\ &\quad + \gamma_{\{Age,LEMS\}}(Walk)_{\mathbb{D}_2^H} \\ &\quad - IND_{\{Age,LEMS\}}(Walk)_{\mathbb{D}} \\ &= \frac{|POS_{\{Age,LEMS\}}(Walk)_{\mathbb{D}_1^H}|}{|\{x_1,x_2,x_3,x_4\}|+|\{x_5,x_6,x_7\}|} \\ &\quad + \frac{|POS_{\{Age,LEMS\}}(Walk)_{\mathbb{D}_2^H}|}{|\{x_1,x_2,x_3,x_4\}|+|\{x_5,x_6,x_7\}|} \\ &\quad - \frac{0}{|\{x_1,x_2,x_3,x_4\}|+|\{x_5,x_6,x_7\}|} \\ &= \frac{2}{4+3} + \frac{3}{4+3} - \frac{0}{4+3} \\ &= \frac{5}{7}\end{aligned} \tag{5.19}$$

Based on the greedy nature of QuickReduct algorithm, $R = \{Age, LEMS\}$ is selected, since it ends to the highest dependency degree.

5.6.2 Multi parties with horizontally partitioned data (MP-HP)

In multi-party environments, the most important challenge is to run secure comparison as efficient as possible. Since many parties are involved, each should calculate the dependency degree of each feature in their partitions and also find indiscernible objects. Having a record of indiscernible objects help the whole process by deciding on indiscernibility of objects from other partitions faster. When the secure comparison process is triggered, objects from the other partition are compared with the objects in the indiscernible set of the same partition, initially. If the decision on the indiscernibiliy is finalized, the corresponding dependency degree should be affected. Otherwise, a thorough comparison should be run on all non-indiscernible objects, also.

5.6.3 Two parties with vertically partitioned data (2P-VP)

In case of having vertically partitioned data, three principles should be followed as follows:

1. Each partition should have the classification results
2. Features should have the same order in the whole dataset
3. A set of indiscernible objects should be created for each partition

As an illustrative example, the dataset in Table 5.3 has been partitioned vertically into two datasets and shown in Tables 5.6 and 5.7.

Table 5.6: The first partition of vertically partitioned data $\mathbb{D}_1^V$

Object	**Age**	**Walk**
x_1	16-30	Yes
x_2	16-30	No
x_3	31-45	No
x_4	31-45	Yes
x_5	46-60	No
x_6	16-30	Yes
x_7	46-60	No

Table 5.7: The second partition of vertically partitioned data $\mathbb{D}_2^V$

Object	**LEMS**	**Walk**
x_1	50	Yes
x_2	0	No
x_3	1-25	No
x_4	1-25	Yes
x_5	26-49	No
x_6	26-49	Yes
x_7	26-49	No

By referring to the required principles for 2P-VP datasets, each partition has classification outcome and also the order of samples are preserved. The only remaining criterion is sets of indiscernible objects for both parties, which have been calculated and mentioned in Equation 5.20.

$$IND_{\{Age\}}(Walk)_{\mathbb{D}_1^V} = \{\{x_1, x_2, x_6\}, \{x_3, x_4\}\} \tag{5.20}$$
$$IND_{\{LEMS\}}(Walk)_{\mathbb{D}_2^V} = \{\{x_3, x_4\}, \{x_5, x_6, x_7\}\}$$

As calculated in Equation 5.11, the dependency degree of each feature in each partition is equal. So, one of feature should be selected to break the tie, since both of them have the same dependency degree. Regardless of which feature is selected, calculating the dependency degree of $R = \{Age, LEMS\}$ requires some efforts. Since all objects are available to each party and the exact value for dependency degree can be calculated, each party should decide on the number of indiscernible objects.

Party one (or two), needs to know if there is any intersection between the two indiscernible sets, if any, the cardinality of the subset should be subtracted from the number of objects in the dataset. The process is shown in Equation 5.21.

$$\begin{aligned} IND_{\{Age\}}(Walk)_{\mathbb{D}_1^V} &= \{\{x_1, x_2, x_6\}, \{x_3, x_4\}\} \\ IND_{\{LEMS\}}(Walk)_{\mathbb{D}_2^V} &= \{\{x_3, x_4\}, \{x_5, x_6, x_7\}\} \\ IND_{\{Age\}}(Walk)_{\mathbb{D}_1^V} \cap IND_{\{LEMS\}}(Walk)_{\mathbb{D}_2^V} & \\ = IND_{\{Age,LEMS\}}(Walk)_{\mathbb{D}} & \\ = \{\{x_3, x_4\}\} & \end{aligned} \tag{5.21}$$

Therefore, the final dependency degree is illustrated in Equation 5.22.

$$\begin{aligned} \gamma_{\{Age,LEMS\}}(Walk) &= \frac{7}{7} - \frac{2}{7} \\ &= \frac{5}{7} \end{aligned} \tag{5.22}$$

In cases which two partitions have more than one feature, each one should calculate the dependency degree of all the features, as well as, their indicernibility sets. Then, a feature with the highest dependency degree should be added to the reduct set. Therefore, there are two cases that should be addressed properly for both partitions, as follows:

1. If the selected feature is in the same partition

2. If the selected feature is in the other partition

For the partition that contains the selected feature, the only task is to build the reduct sets with two members and calculate the dependency degrees without the need of communicating with other party. However, the other partition should have indiscernibility set of the selected features to be able to find the dependency degree of the sets with two members. Hence, a secure comparison should be applied to fulfil this requirement.

5.6.4 Multi parties with vertically partitioned data (MP-VP)

In the environment with more than two parties and vertically partitioned data, the same policy for 2P-VP works fine. The only issue is the amount of communication

that is made to/from parties to compute the dependency degrees. Therefore, a computationally inexpensive secure comparison is desired to minimize the overall load.

5.7 Conclusion

Feature selection is the process of selecting important features while discarding the others. This process is usually referred to as a pre-process since it purifies data for a main process, such as classification. Almost all of the previously introduced feature selection methods are not useful for the current needs which involve distributed and decentralized datasets and parties. Therefore, researchers have tried to develop new feature selection methods which can be applied to distributed datasets. In this paper, we have reviewed three feature selection methods, and provided some suggestions to improve their performance when identified. We have also introduced a privacy-preserving feature selection method based on rough set feature selection. The proposed method, has been designed to process both horizontally and vertically partitioned datasets for either two-party or multi-party scenarios. As a future work, we are currently working on privacy-preserving protocols for other standard feature selection methods, as well as their formal security and complexity analyses, along with experimental results on real data. Also, the performance and effectiveness of the proposed method will be examined against UCI datasets. Finally, we will integrate all the proposed protocols in an online privacy-preserving feature selection tool which will be publicly available for non-commercial purposes.

Bibliography

[1] C. C. Aggarwal. *Privacy-Preserving Data Mining*, pages 663–693. Springer International Publishing, Cham, 2015.

[2] R. Agrawal and R. Srikant. Privacy-preserving data mining. In *ACM Sigmod Record*, volume 29, pages 439–450. ACM, 2000.

[3] J. R. Anaraki and M. Eftekhari. Improving fuzzy-rough quick reduct for feature selection. In *Electrical Engineering (ICEE), 2011 19th Iranian Conference on*, pages 1–6, May 2011.

[4] J. R. Anaraki and M. Eftekhari. Rough set based feature selection: a review. In *Information and Knowledge Technology (IKT), 2013 5th Conference on*, pages 301–306. IEEE, 2013.

[5] J. R. Anaraki, M. Eftekhari, and C. W. Ahn. Novel improvements on the fuzzy-rough quickreduct algorithm. *IEICE TRANSACTIONS on Information and Systems*, 98(2):453–456, 2015.

[6] M. Banerjee and S. Chakravarty. Privacy preserving feature selection for distributed data using virtual dimension. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 2281–2284. ACM, 2011.

[7] B. Beach and D. C. Platt. Distributed database management system, Apr. 27 2004. US Patent 6,728,713.

[8] J. Bennett and S. Lanning. The netflix prize. In *Proceedings of KDD cup and workshop*, volume 2007, page 35, 2007.

[9] C.-I. Chang. *Hyperspectral imaging: techniques for spectral detection and classification*, volume 1. Springer Science & Business Media, 2003.

[10] T. M. Cooley. Cooley on torts, 1888.

[11] K. Das, K. Bhaduri, and H. Kargupta. A local asynchronous distributed privacy preserving feature selection algorithm for large peer-to-peer networks. *Knowledge and Information Systems*, 24(3):341–367, 2010.

[12] M. Hall, G. Holmes, et al. Benchmarking attribute selection techniques for discrete class data mining. *Knowledge and Data Engineering, IEEE Transactions on*, 15(6):1437–1447, 2003.

[13] M. A. Hall. *Correlation-based feature selection for machine learning*. PhD thesis, The University of Waikato, 1999.

[14] M. A. Hall and L. A. Smith. Practical feature subset selection for machine learning, 1998.

[15] S. International. Sri international awarded $8.5 million darpa contract for data privacy. `https://www.sri.com/newsroom/press-releases/sri-international-awarded-85-million-darpa-contract-data-privacy`. Accessed July 26, 2016.

[16] Y. Jafer, S. Matwin, and M. Sokolova. Privacy-aware filter-based feature selec-

tion. In *Big Data (Big Data), 2014 IEEE International Conference on*, pages 1–5. IEEE, 2014.

[17] R. Jensen and Q. Shen. New approaches to fuzzy-rough feature selection. *Fuzzy Systems, IEEE Transactions on*, 17(4):824–838, 2009.

[18] K. Kira and L. A. Rendell. The feature selection problem: Traditional methods and a new algorithm. In *AAAI*, volume 2, pages 129–134, 1992.

[19] R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial intelligence*, 97(1):273–324, 1997.

[20] J. Komorowski, Z. Pawlak, L. Polkowski, and A. Skowron. Rough sets: A tutorial. In S. K. Pal and A. Skowron, editors, *Rough-Fuzzy Hybridization: A New Trend in Decision Making*, pages 3–98. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1998.

[21] M. Lichman. UCI machine learning repository, 2013.

[22] Y. Lindell and B. Pinkas. Privacy preserving data mining. In *Advances in CryptologyCRYPTO 2000*, pages 36–54. Springer, 2000.

[23] A. Narayanan and V. Shmatikov. How to break anonymity of the netflix prize dataset. *CoRR*, abs/cs/0610105, 2006.

[24] Z. Pawlak. Rough sets. *International Journal of Computer & Information Sciences*, 11(5):341–356, 1982.

[25] S. Samet. *Privacy-Preserving Data Mining*. PhD thesis, 2010.

[26] L. D. B. Samuel D. Warren. The right to privacy. *Harvard Law Review*, 4(5):193–220, 1890.

[27] D. S. Scherber and H. C. Papadopoulos. Distributed computation of averages over ad hoc networks. *IEEE journal on Selected Areas in Communications*, 23(4):776–787, 2005.

Chapter 6

A Feature Selection Based on Perturbation Theory

This paper is published in Expert Systems with Applications, 2019.

6.1 Abstract

Consider a supervised dataset $D = [A \mid \mathbf{b}]$, where $\mathbf{b}$ is the outcome column, rows of D correspond to observations, and columns of A are the features of the dataset. A central problem in machine learning and pattern recognition is to select the most important features from D to be able to predict the outcome. In this paper, we provide a new feature selection method where we use perturbation theory to detect correlations between features. We solve $AX = \mathbf{b}$ using the method of least squares and singular value decomposition of A. In practical applications, such as in bioinformatics, the number of rows of A (observations) are much less than the number of columns of A (features). So we are dealing with singular matrices with big condition numbers.

Although it is known that the solutions of least square problems in singular case are very sensitive to perturbations in A, our novel approach in this paper is to prove that the correlations between features can be detected by applying perturbations to A. The effectiveness of our method is verified by performing a series of comparisons with conventional and novel feature selection methods in the literature. It is demonstrated that in most situations, our method chooses considerably less number of features while attaining or exceeding the accuracy of the other methods.

Index terms— Feature selection, perturbation theory, least angle regression

6.2 Introduction

In machine learning and pattern recognition, feature selection is the process of selecting the most important features of a problem while removing unnecessary ones. This process plays an important role in reducing the dimension of datasets. Feature selection methods are categorized into two main groups of feature ranking and feature subset selection [10]. The former is a set of methods that ranks the features based on some measured values, and selects the top features, accordingly. The latter screens the critical features using fitness value. Both groups can be implemented using filter-based or wrapper-based approaches [14]. In the filter-based approach, a merit evaluates the quality of every feature regardless of its impact on the outcome, while the wrapper-based approaches measure the effectiveness of the features based on the results of a (a set of) classifier(s). The wrapper-based methods are highly computationally-intensive and powerful in predicting the outcome compared to the filter-based methods which are faster but less accurate.

With the emergence of high dimensional data, for example in Genomics, sophisticated feature selection methods are required to remove noisy features and detect correlation between features. It is desired that a small subset of features are selected to predict the outcome with high accuracy. The traditional feature selection methods such as principal component analysis [12] or Relief [13] have shortcomings in terms of dimensionality reduction, accuracy, as well as running time. We shall review some of the breakthrough methods that are effective in these respects.

There have been numerous methods based on the information theory, see for example [26, 20, 3]. These methods aim to minimize the feature redundancy while maximizing the features' relevancy. Most notable and widely used information theory based method is minimal-redundancy-maximal-relevance criterion (mRMR) [17]. It is shown in various studies that mRMR effectively chooses a small subset of features to predict the outcome with high accuracy. However, as it is pointed out in [25], the computational cost of mRMR on large dataset is high. In other words, it is not feasible to scale up mRMR for big datasets.

Feature selection is also referred to as variable selection in Statistics. Fundamental variable selection methods include least absolute shrinkage and selection operator (LASSO) and least angle regression (LARS). LASSO, introduced by Tibshirani [21], is a subset selection based on least squares regression. It minimizes the size of a regression model by removing those predictor variables with zero-valued coefficients by calculating Equation 6.1, the LASSO estimate, subject to $\sum_j |\beta_j| \leq t$, where β is

a vector of coefficients and $t \geq 0$ is tuning parameter

$$(\hat{\alpha}, \hat{\beta}) = \arg\min \left\{ \sum_{i=1}^{N} \left\{ b_i - \alpha - \sum_j \beta_j x_{ij} \right\}^2 \right\}, \tag{6.1}$$

and the solution for α is $\hat{\alpha} = \bar{b}$, $\hat{\beta} = (\hat{\beta}_1, \ldots, \hat{\beta}_n)^T$ are LASSO estimates where n is the total number of features, $\mathbf{b}$ represents responses, x contains predictor variables and N is the number of samples.

LARS, introduced by Efron et al. [6], is a linear regression model fitting based on the LASSO algorithm which calculates all the LASSO estimates efficiently, in combination with a forward stage-wise linear regression method within n steps, where n is number of covariates and m is number of samples. LARS starts with selecting the most relevant feature and continues by adding the next feature with the highest correlation with the current residual. Then, it continues in a direction which has equal angle from the two already selected features until the next feature is met. The complexity of LARS algorithm is $O(n^3 + mn^2)$.

In a novel work, Yamada et al. [24] proposed a non-linear feature selection method for high-dimensional datasets called Hilbert-Schmidt independence criterion least absolute shrinkage and selection operator (HSIC-Lasso), in which the most informative non-redundant features are selected using a set of kernel functions, where the solutions are found by solving a LASSO problem. The complexity of the original Hilbert-Schmidt feature selection (HSFS) is $O(n^4)$. In a recent work [25] called Least Angle Nonlinear Distributed (LAND), the authors have improved the computational power of the HSIC-Lasso. They have demonstrated via some experiments that LAND and HSIC-Lasso have attain similar classification accuracies and dimension reduction.

However, LAND has the advantage that it can be deployed on parallel distributed computing.

A method proposed by Chen et al. [5] is a feature selection called rescaled linear square regression (RLSR), where a set of coefficients for least square regression is employed to scale and rank features. The advantage of their method is that it can be applied to both supervised and semi-supervised classification problems.

In this paper, we introduce a new linear feature selection method. Linear models usually outperform nonlinear models over high-dimensional datasets. Consider a dataset D, consisting of m samples where each sample contains $n+1$ features. Let us denote by A the first n columns of D and by $\mathbf{b}$ the last column. Our objective is to remove those columns of A that do not have a significant impact on $\mathbf{b}$. So, we want to choose a subset of columns of A to express (up to an error) $\mathbf{b}$ as a linear combination of this subset. We consider the linear system $AX = \mathbf{b}$, where $X = [x_1, \ldots, x_n]^T$ is the vector of unknowns. In practical applications, the system $AX = \mathbf{b}$ may not have exact solutions. However, we want to find an X so that the distance between AX and $\mathbf{b}$ is as small as possible. That is, we want to minimize the distance $||AX - \mathbf{b}||_2$ over all X. To do so, we shall use the method of least squares and singular value decomposition (SVD) of A. The Moore-Penrose inverse A^+ of A is defined in terms of SVD of A and it is known that $X = A^+\mathbf{b}$ is the unique solution with the smallest 2-norm that satisfy the least square problem $\min_X ||AX - \mathbf{b}||_2$, see Theorem 1.

There has been extensive literature, see [9], regarding the sensitivity of solutions of least square problems when A is full-rank. It is also known and rightfully cautioned that solutions of singular systems where condition number of A is bigger than one are sensitive to perturbations in A. However, we prove in Theorem 2, that one can

use perturbations to reveal correlations between columns of A. To do so, we solve both $AX = \mathbf{b}$ and $(A + E)\tilde{X} = \mathbf{b}$ using SVD, where E is a small perturbation of A. It turns out that features $\mathbf{f}_i$ and $\mathbf{f}_j$ correlate if and only if $\mid x_i - \tilde{x}_i \mid$ and $\mid x_j - \tilde{x}_j \mid$ are close (in the magnitude of $||E||_2$.). This allows to cluster features based on the differences $\mid x_i - \tilde{x}_i \mid$.

Next, we consider the column vector $|X - \tilde{X}|$ whose values are $\mid x_i - \tilde{x_i} \mid$ and consider clustering features based on this single column. As we mentioned, features that correlate with each other fall into the same cluster. However, within a cluster there might be features that do not correlate (but have the same value for $\mid x_i - \tilde{x_i} \mid$). To break down some big clusters that contain independent features, we use a simple but efficient method based on the angle between features. In Section 6.3.2, we consider the projection of $\mathbf{b}$ into each of the hyperplanes obtained by removing one feature at a time. We construct a column that consists of the angles between each feature and the corresponding hyperplane. The third column in our clustering process consists of angles between each feature and $\mathbf{b}$.

We note that often in classification problems and real-world datasets, for example Cancer datasets, the column $\mathbf{b}$ contains nominal values (classes). One can then assign numerical values for each class. Although, this assignment is not unique our method is insensitive to the way in which the classes are numbered. The reason is, correlations between columns of A is independent of $\mathbf{b}$. Indeed, by Theorem 2, the vector $X - \tilde{X}$ consisting of the $x_i - \tilde{x}_i$ is proportional to correlations between columns of A and as such $X - \tilde{X}$ is insensitive to changes in $\mathbf{b}$. Also, if $\mathbf{b}$ changes, then all the angles between columns of A and $\mathbf{b}$ will be shifted by a fix amount (the difference of old $\mathbf{b}$ and new $\mathbf{b}$). This shows that our $n \times 3$ matrix is insensitive to the way in which we

convert classes to numerical values.

After arriving at the $n \times 3$ matrix, we use a clustering algorithm and cluster our $n \times 3$ matrix into k clusters where k is at most $\mathbf{rank}(A)$. Since we do not know the optimal k, we take the output feature subset for each k and use a classifier to get an accuracy with respect to that feature subset. Alternatively, our algorithm can take as input an integer k to represent the number of desired features and this way we can just cluster with respect to the input k and return the centroids as the selected subset of features. The final algorithm is presented in Section 6.3.3.

To the best of our knowledge, this is the first work to report on using perturbation theory in feature selection. Specifically, the fact that correlations can be detected via perturbations has not been explored before. As we can see through numerous experiments in Section 6.4, our method on average chooses smaller number of features while attaining or exceeding the classification accuracy of other methods. Also, the complexity of our algorithm is dominated by that of computing the SVD of an $m \times n$ matrix which can be done in $O(\min\{mn^2, m^2n\})$ and even faster as explained in [11]. In particular, in datasets where we have hundreds of samples and thousands of features ($m^2 \leq n$), the complexity of PFS is close to quadratic. It is also worth noting that our proposed method can be applied to both regression and classification problems. We present some further insights in Section 6.5, and conclude the paper and suggest possible future paths in Section 7.6.

6.3 Proposed Approach

Consider the system $AX = \mathbf{b}$. Since we want to know the smallest subset of columns of A that we can express $\mathbf{b}$ as a linear combination of elements of that subset, we can normalize the columns of A. So, we can assume each column of A has length 1.

In real world applications, the system $AX = \mathbf{b}$ may not have a solution. In other words, if b is not in the column space of A, there is no X such that $AX = \mathbf{b}$. Instead, we can find an X so that the distance between AX and $\mathbf{b}$ is as small as possible. That is, we want to minimize the distance $||AX - \mathbf{b}||_2$ over all X. This minimization problem is known as the method of least squares and its solutions is defined via SVD of A. Recall that the SVD of an $m \times n$ matrix A is of the form $A = USV^T$, where U is an $m \times m$ orthogonal matrix, V is an $n \times n$ orthogonal matrix, and $S = \text{diag}(\sigma_1, \ldots, \sigma_r, 0, \ldots, 0)$ is an $m \times n$ diagonal matrix. Also recall that the Moore-Penrose inverse of A is the $n \times m$ matrix $A^+ = VS^{-1}U^T$, where $S^{-1} = \text{diag}(\sigma_1^{-1}, \ldots, \sigma_r^{-1}, 0, \ldots, 0)$.

It is well-known that the least squares solutions can be given in terms of the Moore-Penrose inverse, see [9].

Theorem 1 (All Least Squares Solutions) *Let A be an $m \times n$ matrix and $\boldsymbol{b} \in \mathbb{R}^m$. Then all the solutions of $min_X||AX - \boldsymbol{b}||_2$ are of the form $y = A^+\boldsymbol{b} + q$, where $q \in \ker(A)$. Furthermore, the unique solution whose 2-norm is the smallest is given by $z = A^+\boldsymbol{b}$.*

In our method, each dataset with m samples and $n+1$ features is divided into two matrices: coefficients and constants. Coefficients matrix A involves all the feature values except for the outcome, the constant vector $\mathbf{b}$ only contains the classification

outcome. In the next section we employ perturbation theory to detect redundant features.

6.3.1 Detecting correlations via perturbation

To demonstrate how the perturbation can reveal different aspects of features, a synthetic dataset called SynthData is generated with 100 samples and six features based on the following setup:

$$\mathbf{f}_1 = rand(100), \quad \mathbf{f}_2 = rand(100),$$
$$\mathbf{f}_3 = rand(100), \quad \mathbf{f}_4 = rand(100),$$
$$\mathbf{f}_5 = 8 \times \mathbf{f}_3 + 2 \times \mathbf{f}_4, \quad \mathbf{f}_6 = 5 \times \mathbf{f}_2,$$
$$\mathbf{b} = 7 \times \mathbf{f}_1 - 3 \times \mathbf{f}_2 + 6 \times \mathbf{f}_3,$$

where $rand(100)$ generates 100 random numbers with uniform probability in the interval $(0, 1)$. So, $D = [A \mid \mathbf{b}]$, where $A = [\mathbf{f}_1 \mid \cdots \mid \mathbf{f}_6]$ is an 100×6 matrix. Now let E be a small perturbation of A and solve $AX = \mathbf{b}$ and $(A+E)\tilde{X} = \mathbf{b}$ using SVD. We have demonstrated the solutions X and $\tilde{X}$ as well as their differences in Table 6.1. As we expected, X and $\tilde{X}$ differ significantly. However, our interest is focused at the last column of Table 6.1, where we have recorded the difference between X and $\tilde{X}$.

Before we state the main theorem, we shall need to recall some facts and definitions which can be found in [9].

Let $\tilde{A} = A+E$ be a perturbation of A. Denote by $\sigma_1 \geq \sigma_2 \geq \cdots$ and $\sigma'_1 \geq \sigma'_2 \geq \cdots$ the singular values of A and $\tilde{A}$, respectively. The samllest non-zero singular value of A is denoted by $\sigma_{\min}$ and the greatest of the σ_i is denoted by $\sigma_{\max}$. It is well-known

Table 6.1: Perturbation of SynthData

	X	$\tilde{X}$	$X - \tilde{X}$
x_1	40.8401	40.8401	2.2115e-05
x_2	-8.5981	-8.5980	-1.1532e-05
x_3	17.4601	-5.9568e+03	-5.9743e+03
x_4	-3.7881	-1.4436e+03	-1.4398e+03
x_5	16.1273	6.1460e+03	6.1298e+03
x_6	-8.5981	-8.5980	-1.8675e-05

that $||A||_2 = \sigma_{\max}$. It has been of great interest to compare the σ_i and σ_i'. In this regard, we use a classical bound on the difference between σ_i and σ_i' due to Weyl:

$$|\sigma_i - \sigma_i'| \leq ||E||_2, \quad i = 1, 2, \cdots \tag{6.2}$$

We need to determine the type of perturbations we use. Indeed, we choose E to be a random matrix such that $||E||_2 \approx 10^{-s}\sigma_{\min}(A)$, for some $s \geq 0$. We set $s = 3$ where our estimates are correct up to a magnitude of 10^{-3}. We are now ready to prove the main theorem of this paper.

Theorem 2 *Let X and $\tilde{X}$ be solutions of $AX = \boldsymbol{b}$ and $(A + E)\tilde{X} = \boldsymbol{b}$, where E is a small enough perturbation. If a feature $\boldsymbol{f}_i$ is independent of the rest of the features then $|x_i - \tilde{x}_i| \approx 0$. Furthermore, suppose that $S' = \{\boldsymbol{f}_1, \ldots, \boldsymbol{f}_t\}$ is a subset of S such that $\sum_{i=1}^{t} c_i \boldsymbol{f}_i = 0$, for some non-zero c_i. If*

1. *any subset of S' is linearly independent,*

2. *$\boldsymbol{f}_1, \ldots, \boldsymbol{f}_t$ are linearly independent from the rest of features in S.*

Then the vectors $\begin{pmatrix} c_1 \\ \vdots \\ c_t \end{pmatrix}$ *and* $\begin{pmatrix} x_1 - \tilde{x}_1 \\ \vdots \\ x_t - \tilde{x}_t \end{pmatrix}$ *are proportional.*

Proof. From $AX = \mathbf{b}$ and $(A + E)\tilde{X} = \mathbf{b}$, we get $A(X - \tilde{X}) = E\tilde{X}$. We claim that $||E\tilde{X}|| \approx 10^{-s}$. To prove the claim, we consider the SVD of $A + E$ which is of the form $A + E = U\Sigma V^T$. So, $\tilde{X} = V\Sigma^{-1}U^T b$. Since U and V are orthogonal and for orthogonal matrices we have $||U\mathbf{v}||_2 = ||\mathbf{v}||_2$, it follows that

$$\begin{aligned} ||\tilde{X}||_2 = ||V\Sigma^{-1}U^T\mathbf{b}||_2 &= ||\Sigma^{-1}\mathbf{b}||_2 \\ &\leq ||\Sigma^{-1}||_2||\mathbf{b}||_2 = \frac{1}{\sigma_{\min}(A+E)} \\ &\leq \frac{1}{-||E||_2 + \sigma_{\min}(A)}, \end{aligned}$$

by Equation (6.2). Hence,

$$\begin{aligned} ||E\tilde{X}||_2 \leq ||E||_2||\tilde{X}||_2 &= \frac{10^{-s}\sigma_{\min}(A)}{-10^{-s}\sigma_{\min}(A) + \sigma_{\min}(A)} \\ &= \frac{10^{-s}}{1 - 10^{-s}} = \frac{1}{10^s - 1} \approx 10^{-s} \end{aligned}$$

It follows from the claim that

$$(x_1 - \tilde{x}_1)\mathbf{f}_1 + \cdots + (x_t - \tilde{x}_t)\mathbf{f}_t + \cdots + (x_n - \tilde{x}_n)\mathbf{f}_n \approx 0. \tag{6.3}$$

Now, if a feature, say $\mathbf{f}_n$, is independent of the rest of features, then it follows from Equation (6.3) that $|x_n - \tilde{x}_n| \approx 0$. Suppose now that $S' = \{\mathbf{f}_1, \ldots, \mathbf{f}_t\}$ is a linearly dependent subset of S such that $\sum_{i=1}^{t} c_i\mathbf{f}_i = 0$, for some coefficients $c_1, \ldots, c_t$. Since $\mathbf{f}_1, \ldots, \mathbf{f}_t$ are linearly independent from the rest of features in S, we get

$$(x_1 - \tilde{x}_1)\mathbf{f}_1 + \cdots + (x_t - \tilde{x}_t)\mathbf{f}_t \approx 0. \tag{6.4}$$

Now, if $\begin{pmatrix} c_1 \\ \vdots \\ c_t \end{pmatrix}$ and $\begin{pmatrix} x_1 - \tilde{x}_1 \\ \vdots \\ x_t - \tilde{x}_t \end{pmatrix}$ are not proportional, we can use Equation (6.4) and $\sum_{i=1}^{t} c_i \mathbf{f}_i = 0$ to get a dependence relation of a shorter length between the elements of S', which would contradict our assumption (1). The proof is complete. □

Consider now the correlation $\mathbf{f}_5 = 8 \times \mathbf{f}_3 + 2 \times \mathbf{f}_4$ in the SynthData dataset. As we mentioned earlier, we normalize the columns of A and replace A with $[\mathbf{f}'_1 \mid \cdots \mid \mathbf{f}'_6]$, where $\mathbf{f}'_i = \frac{\mathbf{f}_i}{||\mathbf{f}'_i||}$. Note that $||\mathbf{f}_3|| = 5.52, ||\mathbf{f}_4|| = 5.33, ||\mathbf{f}_5|| = 45.38$. We have

$$\mathbf{f}'_5 = \frac{\mathbf{f}_5}{||\mathbf{f}_5||} = \frac{8\mathbf{f}_3 + 2\mathbf{f}_4}{45.38} = 0.97\mathbf{f}'_3 + 0.23\mathbf{f}'_4$$

So, correlation vector between $\mathbf{f}'_3, \mathbf{f}'_4, \mathbf{f}'_5$ is $\begin{bmatrix} 0.97 \\ 0.23 \\ -1 \end{bmatrix}$. On the other hand, we have $\begin{bmatrix} x_3 - \tilde{x}_3 \\ x_4 - \tilde{x}_4 \\ x_5 - \tilde{x}_5 \end{bmatrix} = (-6.1298e+03) \begin{bmatrix} 0.97 \\ 0.23 \\ -1 \end{bmatrix}$. Note that in this example, weights (norms) of $8 \times \mathbf{f}_3$ and $\mathbf{f}_4$ are very close to each other compared to weight of $2 \times \mathbf{f}_4$. In general, when a dependence relation exists between a set of features, Theorem 2 along with normalization detect the two features whose weights are closest to each other compared to the others. In particular, if features $\mathbf{f}_i$ and $\mathbf{f}_j$ correlate with each other then the differences $\mid x_i - \tilde{x_i} \mid$ and $\mid x_j - \tilde{x_j} \mid$ are almost the same. The converse may not be necessarily true.

We can now consider a column vector whose values are $\mid x_i - \tilde{x_i} \mid$ and use a clustering algorithm to cluster this single column. Clearly, features that correlate

with each other fall into the same cluster. However, within a cluster there might be features that do not correlate (but have the same value for $\mid x_i - \tilde{x_i} \mid$). For this reason, we want to further refine the clustering process by computing two more characteristics of data. We shall explain this in the next section.

6.3.2 Refining the clustering process

One way to compare the similarity between vectors is by calculating the angle between them. Features that have smaller angles with the outcome $\mathbf{b}$ are informative and predictive. So we construct another column whose values are angles between the $\mathbf{f}_i$ and $\mathbf{b}$. The angle of each feature with $\mathbf{b}$ in SynthData are calculated and shown in the Table 6.2.

Table 6.2: Angle of each feature to $\mathbf{b}$ in SynthData

	$\mathbf{f}_1$	$\mathbf{f}_2$	$\mathbf{f}_3$	$\mathbf{f}_4$	$\mathbf{f}_5$	$\mathbf{f}_6$
$\mathbf{b}$	37.104	112.981	47.897	87.030	48.270	112.981

Our third column in the clustering process is obtained as follows. We remove each feature $\mathbf{f}_i$ from the matrix A along with its corresponding coefficient x_i in X. Then, the angle of resulting vector $A \setminus \{\mathbf{f}_i\} \times X \setminus \{x_i\} = \hat{\mathbf{b}}_i$ and the actual outcome $\mathbf{b}$ will be considered as a measure of the relevancy for feature $\mathbf{f}_i$. Note that the closer $\mathbf{b}$ and $\hat{\mathbf{b}}_i$ are, the less significant the vector $x_i\mathbf{f}_i$ is. Applying this process to SynthData is shown in Table 6.3.

Now we set up an $n \times 3$ matrix where the first column consists of $|x_i - \tilde{x}_i|$, the

Table 6.3: Angles of calculated $\hat{\mathbf{b}}_i$ to $\mathbf{b}$ for SynthData

	$\hat{\mathbf{b}}_1$	$\hat{\mathbf{b}}_2$	$\hat{\mathbf{b}}_3$	$\hat{\mathbf{b}}_4$	$\hat{\mathbf{b}}_5$	$\hat{\mathbf{b}}_6$
θ	40.390	7.748	14.574	3.507	13.330	7.748

second column is the angles between the $\mathbf{f}_i$'s and $\mathbf{b}$, and the third column is the angles between the $\hat{\mathbf{b}}_i$'s and $\mathbf{b}$. Next we use a clustering algorithm to cluster our $n \times 3$ into k clusters. The centroids of clusters will be chosen as our selected features. Since we do not know the optimal number of clusters, we take the output feature subset for each k and use a classifier to get an accuracy with respect to that feature subset. Alternatively, our algorithm can take as input an integer k to represent the number of desired features and this way we can just cluster with respect to the input k and return the centroids as the selected subset of features. The upper bound for the number of clusters is $\mathbf{rank}(A)$, where $\mathbf{rank}(A)$ is the numerical rank of A.

6.3.3 Algorithm

The PFS running time is $t \times (\min(m \times n^2, m^2 \times n) + k \times (3 \times n \times k))$, where $\min(m \times n^2, m^2 \times n)$ is the complexity of calculating SVD for a $m \times n$ matrix [11], and $(3 \times n \times k)$ is the time complexity of the k-means clustering algorithm to cluster a dataset of size $n \times 3$ into k clusters. Therefore, the time complexity of PFS is dominated by the complexity of SVD.

Flowchart of PFS is depicted in Figure 6.1 and is as shown in Algorithm 7.1. The MATLAB® implementation of PFS is publicly available on GitHub[1].

[1]https://github.com/jracp/PerturbationFeatureSelection

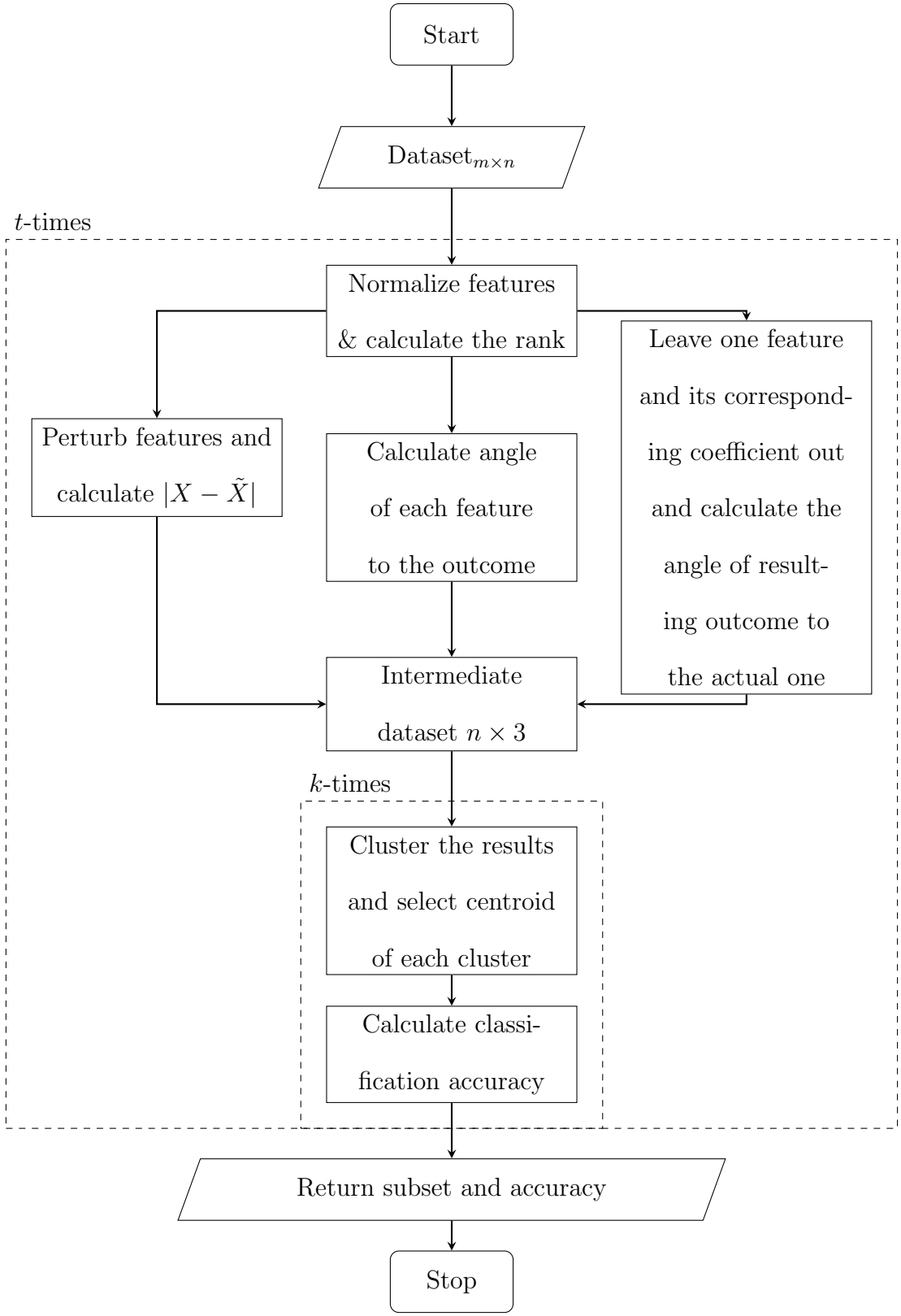


Figure 6.1: Flowchart of the proposed method

Algorithm 6.1: Perturbation-based Feature Selection

Data: $D = [A \mid \mathbf{b}]_{m \times n+1}$

Result: Subset of features and resulting accuracy

$ACC_{average}, ACC_{optimal}$: average accuracy, the optimal accuracy over t runs;

$|CLS_{average}|$: average size of subset of features over t runs;

$|CLS_{optimal}|$: size of the optimal subset of features over t runs;

Set $t = 10, c_l = 10^6, c_u = 10^5$, Normalize columns of A and $\mathbf{b}$ within $[0, 1]$;

for $i = 1$ **to** t **do**

  $E = (\frac{\max(A)}{c_u} - \frac{\min(A)}{c_l}) \cdot \mathbf{rand} + \min(A)$; $\tilde{A} = A + E$;

  $X = A^+ \times \mathbf{b}$, where A^+ is the Moore-Penrose inverse of A; $\tilde{X} = (\tilde{A})^+ \times \mathbf{b}$;

  for $j = 1$ **to** n **do**

    calculate the angle θ_j between $\mathbf{f}_j$ and $\mathbf{b}$;

    calculate the angle γ_j between $A \setminus \{\mathbf{f}_j\} \times X \setminus \{x_j\} = \hat{\mathbf{b}}_i$ and $\mathbf{b}$;

  for $k = 2$ **to rank**(A) **do**

    Form the $n \times 3$ matrix $[\text{abs}(X - \tilde{X}) \mid \theta \mid \gamma]$;

    cluster this data into k clusters;

    find and select centroid features of each cluster;

    classify D based on the selected features and return $ACC_{current}$ and $|CLS_{current}|$;

    if $ACC_{current} > ACC_{best}$ **then**

      $ACC_{best} = ACC_{current}$; $|CLS_{best}| = |CLS_{current}|$;

  $ACC(j) = ACC_{best}$;

Compute and return $ACC_{average}, |CLS_{average}|, ACC_{optimal}$, and $|CLS_{optimal}|$ based on the vector ACC;

6.4 Experimental Results

We generate the perturbation matrix E such that the entries of E are randomly chosen in the range $c_l = 10^6$ and $c_u = 10^5$.

Referring to Tran et al. [22], classification accuracy of imbalanced datasets should be calculated using Equation 6.5.

$$\frac{1}{s}\sum_{i=1}^{s}\frac{CC_i}{M_i}, \tag{6.5}$$

where s is the number of classes in dataset, CC_i is the number of correctly classified instances within class i, and M_i is the total number of samples in the class i.

When comparing two feature selection methods, there are three quantities that matter: 1) the accuracy, 2) number of selected features 3) complexity and running time.

We adopt the following formula to compare feature selection methods based on the their accuracy and selected number of features: We quantify the relative effectiveness of a feature selection methods as follows:

$$\frac{\text{classification accuracy}}{\#\,\text{selected features}}. \tag{6.6}$$

Formula (6.6) means that a feature selection method with smaller number of features and higher classification accuracy is favourable.

All the computations have been done on an ubuntu 14.04 LTS machine with Intel®Core™i5-4570, 24 GB of RAM, using MATLAB® 9.2.0.556344 (R2017a), R version 3.4.4 (2018-03-15), and Java™SE Runtime Environment (build 1.8.0_151-b12).

6.4.1 Comparisons with conventional methods

In this section, we compare PFS with Friedman's gradient boosting machine (GBM) [7]; least absolute shrinkage and selection operator (LASSO) [21]; least angle regression (LARS) [6]; rescaled linear square regression (RLSR) [5] with $k = minSelF$, where $minSelF$ is the minimum number of selected features using GBM, LASSO and LARS; and Hilbert-Schmidt independence criterion least absolute shrinkage and selection operator (HSIC-Lasso) [24]. We used gbm package in R [18] for running GBM, and MATLAB® implementations of LASSO and LARS by Sjöstrand [19], RLSR and HSIC-Lasso.

In Section 6.4.1.1, we have used k-means to cluster our $n \times 3$ matrix where the upper bound for k is the numerical rank of A. To find the best subset, we have experimented with three different classifiers, that is decision tree (DT) [4], support vector machine (SVM) [1], and k-nearest neighbour (k-NN) [2] in the inner layer. Once we find the k and corresponding subset of features that gives us the best accuracy, we output that subset as the selected features. At the outer layer of our algorithm, we always use DT for classification. To demonstrate a fair and robust result, we run the algorithm 10 times where each time a subset of features is outputted and then classified by DT. The average of accuracies as well as average size of feature subsets are reported. We have demonstrated similar experiments using fuzzy c-means in Section 6.4.1.2.

We perform a series of tests on various datasets including, one medical dataset, LSVT Voice [23], one artificial dataset Madelon and six biological datastes – namely, Colon , Lung, Lymphoma, GLIOMA, Leukemia and ALLAML – have been selected

from ASU dataset repository [15] and UCI repository of machine learning [16]. The specifications of all datasets are given in Table 7.1.

Table 6.4: Dataset Specifications

Dataset	**Samples**	**Features**
LSVT Voice	126	310
Madelon	2000	500
Colon	62	2000
Lung	203	3312
Lymphoma	96	4026
GLIOMA	50	4434
Leukemia	72	7070
ALLAML	72	7129

Note that for the experiments in this section, the decision tree classifier is applied with MATLAB®, using 70% of the data for training and 30% for testing and validating. This set up is applied to all methods including GBM, LASSO, LARS, RLSR, HSIC-Lasso, and PFS. Since PFS uses a clustering algorithm, the selected subset of features in PFS can change each run. So, we run PFS 10 times on randomly shuffled data where testing and trainings sets vary accordingly in each run.

6.4.1.1 Evaluation results using k-means

In this section, we use k-means to cluster our $n \times 3$ matrix where the upper bound for k is the numerical rank of A. To find the best subset, we have experimented with

three different classifiers, that is DT, SVM and kNN in the inner layer. Once we find the k and corresponding subset of features that gives us the best accuracy, we output that subset as the selected features. At the outer layer of our algorithm, we always use DT for classification for all the methods.

In Tables 6.5 and 6.6, we have reported the selected number of feature and classification accuracies, respectively. Note that PFS-DT, PFS-SVM, and PFS-kNN mean that we have used DT, SVM, and kNN as the inner classifier in PFS, respectively. In all the methods we have used DT to report the classification accuracy.

To demonstrate a fair and robust result, we run our algorithm 10 times where each time the dataset is randomly shuffled and a subset of features is outputted. The average of accuracies as well as average size of feature subsets are reported. Also, we use Formula 6.6 to find the optimal accuracy and subset of features amongst the 10 run. In columns corresponding to PFS-DT, PFS-SVM, and PFS-kNN, the optimal number of features and optimal classification accuracy with respect to Formula 6.6 are shown in the superscript whereas the average number of features and average of classification accuracies are shown in the subscript.

We can see from Table 6.6 that, over all, the classification accuracies of PFS-based methods are favourable to the other methods and only HSIC-Lasso is sometimes attaining similar accuracies. On the other hand, HSIC-Lasso chooses less number of features on average compared to PFS-based methods. We remark that the number of features in PFS depends on the upper bound we set for the number of clusters when we cluster our intermediate $n \times 3$ matrix. We have taken $\mathbf{rank}(A)$ as an upper bound but this bound is just a crude estimate and in the next phases of this project we shall improve this bound. Hence, it is possible to still decrease the average number

Table 6.5: Number of selected features using GBM, LASSO, LARS, RLSR, HSIC-Lasso, PFS based on decision tree classifier (PFS-DT), PFS based on support vector machine classifier (PFS-SVM) and PFS based on k-nearest neighbour classifier (PFS-kNN). For each version of PFS the mean of the number of selected features in 10 run is reported in subscript.

Dataset	**Number of selected features**							
	GBM	LASSO	LARS	RLSR	HSIC-Lasso	PFS-DT	PFS-SVM	PFS-kNN
LSVT Voice	239	126	125	125	12	$13_{45.30}$	$87_{111.90}$	$30_{94.60}$
Madelon	467	89	89	89	—	$34_{100.80}$	$6_{24.80}$	$25_{64.60}$
Colon	656	62	61	61	9	$7_{29.80}$	$22_{39.30}$	$18_{30.60}$
Lung	1503	203	202	202	134	$34_{105.00}$	$28_{100.00}$	$58_{131.20}$
Lymphoma	1491	96	95	95	181	$36_{51.80}$	$23_{44.80}$	$42_{75.50}$
GLIOMA	535	50	49	49	17	$7_{25.60}$	$17_{36.50}$	$28_{37.50}$
Leukemia	1053	72	71	71	17	$6_{46.10}$	$15_{41.00}$	$24_{49.00}$
ALLAML	1200	72	71	71	8	$15_{41.20}$	$24_{53.40}$	$8_{43.00}$

of features in PFS.

We can also observe from Table 6.6, that when kNN is used as the inner classifier, the average classification accuracies are slightly better than when DT or SVM are used. In contrast, the average number of features are slightly lower when DT is used as the inner classifier.

Table 6.6: Classification accuracies of GBM, LASSO, LARS, RLSR, HSIC-Lasso, PFS based on decision tree classifier (PFS-DT), PFS based on support vector machine classifier (PFS-SVM) and PFS based on k-nearest neighbour classifier (PFS-kNN). For each version of PFS the mean of the resulting classification accuracies in 10 run is reported in subscript.

Dataset	**Classification Accuracy**							
	GBM	LASSO	LARS	RLSR	HSIC-Lasso	PFS-DT	PFS-SVM	PFS-kNN
LSVT Voice	73.68	73.68	72.14	63.16	78.94	$83.97_{85.26}$	$60.00_{64.46}$	$84.28_{86.86}$
Madelon	77.67	53.16	62.00	49.34	—	$76.18_{81.45}$	$62.15_{61.62}$	$83.67_{81.97}$
Colon	78.95	83.33	79.49	68.42	84.21	$100.00_{91.58}$	$89.20_{92.61}$	$84.66_{89.20}$
Lung	75.41	51.17	63.58	75.41	83.60	$96.20_{94.10}$	$100.00_{99.95}$	$100.00_{99.84}$
Lymphoma	62.07	39.21	32.19	60.71	51.72	$64.65_{55.93}$	$61.11_{62.41}$	$66.67_{69.94}$
GLIOMA	60.00	52.50	53.75	53.33	80.00	$85.42_{79.33}$	$95.00_{90.08}$	$95.00_{85.58}$
Leukemia	95.46	96.88	96.88	95.46	100.00	$96.88_{95.45}$	$97.06_{99.71}$	$97.06_{98.23}$
ALLAML	90.91	90.83	90.83	62.38	90.90	$93.33_{89.09}$	$93.33_{96.29}$	$85.71_{90.95}$

6.4.1.2 Evaluation results using fuzzy c-means

To investigate the affect of clustering method, we have also experimented with fuzzy c-means clustering algorithm for which, the results are shown in Table 6.7. We can also observe from Table 6.7 that all in all there is very little difference in average classification accuracies regardless of which classifier is used. In contrast, the average number of features are slightly lower when DT is used as the inner classifier.

Table 6.7: The number of selected features and the resulting classification accuracies using fuzzy c-means version of PFS based on decision tree classifier (PFS-DT), PFS based on support vector machine classifier (PFS-SVM) and PFS based on k-nearest neighbour classifier (PFS-kNN). For each version of PFS the mean of the number of selected features and the mean of the resulting classification accuracies is reported in subscript.

Dataset	**Number of selected features**			**Classification Accuracy**		
	PFS-DT	PFS-SVM	PFS-kNN	PFS-DT	PFS-SVM	PFS-kNN
LSVT Voice	$15_{55.70}$	$2_{87.70}$	$67_{86.40}$	$89.74_{82.43}$	$50.00_{56.00}$	$81.07_{86.11}$
Madelon	$19_{154.80}$	$15_{175.80}$	$78_{127.80}$	$75.35_{81.27}$	$62.48_{61.45}$	$79.66_{80.42}$
Colon	$11_{33.10}$	$13_{29.80}$	$13_{33.70}$	$86.67_{89.15}$	$90.91_{89.77}$	$89.20_{88.86}$
Lung	$53_{93.00}$	$66_{126.50}$	$63_{133.90}$	$95.79_{90.88}$	$99.47_{98.96}$	$100.00_{98.42}$
Lymphoma	$59_{53.20}$	$13_{37.80}$	$58_{73.40}$	$69.23_{53.18}$	$63.58_{62.28}$	$76.54_{71.05}$
GLIOMA	$5_{30.40}$	$15_{31.50}$	$17_{31.60}$	$89.58_{79.00}$	$90.00_{88.67}$	$86.67_{87.25}$
Leukemia	$7_{31.60}$	$18_{42.60}$	$17_{44.70}$	$100.00_{97.65}$	$94.12_{97.35}$	$94.12_{96.06}$
ALLAML	$27_{44.60}$	$32_{58.10}$	$8_{51.60}$	$86.09_{89.81}$	$82.86_{86.90}$	$90.00_{87.29}$

6.4.1.3 A quantified measure

In Sections 6.4.1.2 and 6.4.1.1, we have used each of k-means and fuzzy c-means as our clustering algorithm. It seems that using fuzzy c-means, our method in general chooses more features. To present and amalgamate the results of Tables 6.5, 6.6, and 6.7, we apply Formula 6.6 using average classification accuracy and average number of features to obtain a comparison in Table 6.8 between k-means and fuzzy c-means.

We can conclude that based on the measure given by Formula 6.6, our algorithm has a better performance when k-means is used for clustering.

Table 6.8: The resulting measure calculated using Equation 6.6 for k-means and c-means versions of PFS based on decision tree classifier (PFS-DT), PFS based on support vector machine classifier (PFS-SVM) and PFS based on k-nearest neighbour classifier (PFS-kNN).

Dataset	**k-means**			**c-means**		
	PFS-DT	PFS-SVM	PFS-kNN	PFS-DT	PFS-SVM	PFS-kNN
LSVT Voice	1.88	0.57	0.91	1.47	0.64	1.00
Madelon	0.81	2.54	1.26	0.52	0.34	0.62
Colon	3.95	2.35	2.96	2.69	3.06	2.66
Lung	0.89	0.99	0.75	0.96	0.77	0.73
Lymphoma	1.03	1.40	0.92	1.00	1.67	0.97
GLIOMA	3.16	2.50	2.29	2.63	2.83	2.80
Leukemia	4.52	2.41	2.00	3.12	2.30	2.18
ALLAML	2.17	1.81	2.09	2.02	1.48	1.70

6.4.2 Comparison with methods based on SVM & optimization

A recent paper by Ghaddar and Naoum-Sawaya [8] proposed a feature selection method using support vector machines (FS-SVM) for binary-class datasets, in which,

a pre-defined percentage of features is selected through adjusting l_1-norm of the classifier.

Ghaddar et al. applied their method to a set of cancer datasets (# of samples $\times$ # of features) – namely, Leukemia (72 $\times$ 7130), Lung cancer (139 $\times$ 1000), Prostate cancer (102 $\times$ 12,601) – adopted from Cancer Program at Broad Institute [2] (different form those in Table 7.1). For each dataset, a subset of positive and negative classes have been selected for training and testing purposes (see Table 7.4).

Table 6.9: Number of samples of each class for each dataset in FS-SVM

Dataset	**Train**		**Test**	
	Class 1	Class 2	Class 1	Class 2
Leukemia	24	13	23	12
Lung	9	70	8	69
Prostate	25	26	25	26

We have used PFS with DT as the inner classifier and followed the same setup to compare PFS-DT with the method proposed in [8]. To get unbiased results, we run PFS-DT 10 times where each time we shuffled and constructed test and train datasets based on the configuration in Table 7.4. The optimal and average results are reported in Table 6.10.

In order to find the highest classification accuracy, the authors in [8] have applied their method FS-SVM and limited the selected subset of features to range from 2% to 20% of total number of features. In turn, the running time of FS-SVM is very high.

[2] http://www.broad.mit.edu/cgi-bin/cancer/datasets.cgi

Table 6.10: Comparison of PFS based on decision tree classifier (PFS-DT) and FS-SVM

Dataset	**Number of selected features**		**Classification Accuracy**	
	FS-SVM	PFS-DT	FS-SVM	PFS-DT
Leukemia	142	$24_{20.4}$	80.00	$85.15_{77.34}$
Lung	20	$3_{29.90}$	97.00	$100.00_{99.28}$
Prostate	252	$29_{37.40}$	86.00	$88.23_{87.44}$

6.5 Discussions

The upper bound for the number of clusters in Algorithm 1 is the numerical rank of matrix A, which infers about the largest number of independent features. There exists various clustering algorithms and one way to improve the proposed method is to cluster the generated characteristics dataset more efficiently. Of course, the number of clusters in PFS can be set manually which adds a great flexibility in selecting a certain number of features. It is worth noting that some of the clusters that represent irrelevant features can be excluded right away before starting the clustering process. Irrelevant features can be detected by their corresponding coefficients in the solution of the least squares problem.

Since k-means and fuzzy c-means clustering method choose the initial centroids randomly, the final outcome of PFS could be different per run, which introduces a valid concern of non-reproducibility of the results. To remedy this, the proposed algorithm has iterated t-times to provide more robust and reproducible results. An alternative approach is to use a deterministic clustering algorithm which we shall

examine in the future.

The complexity of our proposed method is dominated by the complexity of calculating SVD.

6.6 Conclusions and future work

In this paper, we proposed a novel feature selection method. We divide a dataset D into a matrix A consisting of features and the vector $\mathbf{b}$ of the classification outcome, hence $D = [A \mid \mathbf{b}]$. We solve the least squares problem $\min_X ||AX - \mathbf{b}||_2$ using the singular decomposition of A. We have proved and demonstrated how perturbation theory can be used to detect correlations between features. Through this process, irrelevant features can be identified and filtered out at the very first stages of the algorithm. The main ingredient of our approach is perturbation theory and experimental results show how powerful this method is to detect and remove correlations. We have compared our method with several other methods and it is shown that PFS always chooses a fraction of the number of features selected by other methods. Furthermore, we believe PFS is robust against noise. A noisy data can be viewed as a perturbed system. So we can consider a system of the form $\tilde{A}X = \tilde{\mathbf{b}}$ and apply Theorem 2. We shall investigate the noise-robustness of PFS in future work.

We compared the results from our method with famous LASSO and LARS methods and their descendants RLSR and HSIC-Lasso, as well as, GBM against several datasets. Moreover, we compared our method with the recently proposed method based on optimizing the support vector machines (FS-SVM) [8]. The overall performance of PFS in terms of the number of selected features and resulting classification

accuracies shows its applicability and effectiveness compared to conventional and recent feature selection methods.

The advantage of the proposed method is its modularity. It can be seen as a framework for future feature selection methods, in which different characteristics of feature are extracted using a set of measures. Then, the results are grouped using a user-specified clustering method. Finally, each cluster is evaluated by an arbitrary classifier and the best subset is selected either based on the size of the selected subset or resulting classification accuracy or a combination of both, as suggested in Equation 6.6.

In a future work, we shall also investigate the effect of using different parametric and non-parametric clustering methods to compare the results and decrease the complexity of PFS. Also, we are looking at designing a version of the PFS applicable to gene datasets through a multi-stage process.

Acknowledgements

The research of the second author was supported by NSERC of Canada under grant # RGPIN 418201. The authors would like to thank the anonymous reviewers for valuable comments and feedback that helped with the exposition and clarity of results.

Bibliography

[1] E. L. Allwein, R. E. Schapire, and Y. Singer. Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of machine learning research*,

1(Dec):113–141, 2000.

[2] N. S. Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992.

[3] M. Bennasar, Y. Hicks, and R. Setchi. Feature selection using joint mutual information maximisation. *Expert Systems with Applications*, 42(22):8520–8532, 2015.

[4] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and regression trees*. CRC press, 1984.

[5] X. Chen, G. Yuan, F. Nie, and J. Z. Huang. Semi-supervised feature selection via rescaled linear regression. In *IJCAI*, pages 1525–1531, 2017.

[6] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, et al. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004.

[7] J. H. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[8] B. Ghaddar and J. Naoum-Sawaya. High dimensional data classification and feature selection using support vector machines. *European Journal of Operational Research*, 265(3):993–1004, 2018.

[9] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 2013.

[10] M. Hall, G. Holmes, et al. Benchmarking attribute selection techniques for

discrete class data mining. *Knowledge and Data Engineering, IEEE Transactions on*, 15(6):1437–1447, 2003.

[11] M. Holmes, A. Gray, and C. Isbell. Fast SVD for large-scale matrices. In *Workshop on Efficient Machine Learning at NIPS*, volume 58, pages 249–252, 2007.

[12] I. T. Jolliffe. Mathematical and statistical properties of population principal components. *Principal Component Analysis*, pages 10–28, 2002.

[13] K. Kira and L. A. Rendell. The feature selection problem: Traditional methods and a new algorithm. In *AAAI*, volume 2, pages 129–134, 1992.

[14] R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial intelligence*, 97(1):273–324, 1997.

[15] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu. Feature selection: A data perspective. *ACM Computing Surveys (CSUR)*, 50(6):94, 2017.

[16] M. Lichman. UCI machine learning repository, 2013.

[17] H. Peng, F. Long, and C. Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence*, 27(8):1226–1238, 2005.

[18] G. Ridgeway. Generalized boosted models: A guide to the gbm package. *Update*, 1(1):2007, 2007.

[19] K. Sjöstrand. Matlab implementation of LASSO, LARS, the elastic net and SPCA, jun 2005. Version 2.0.

[20] X. Sun, Y. Liu, M. Xu, H. Chen, J. Han, and K. Wang. Feature selection using dynamic weights for classification. *Know.-Based Syst.*, 37:541–549, Jan. 2013.

[21] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[22] B. Tran, B. Xue, and M. Zhang. *Using Feature Clustering for GP-Based Feature Construction on High-Dimensional Data*, pages 210–226. Springer International Publishing, Cham, 2017.

[23] A. Tsanas, M. A. Little, C. Fox, and L. O. Ramig. Objective automatic assessment of rehabilitative speech treatment in parkinson's disease. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 22(1):181–190, 2014.

[24] M. Yamada, W. Jitkrittum, L. Sigal, E. P. Xing, and M. Sugiyama. High-dimensional feature selection by feature-wise kernelized lasso. *Neural computation*, 26(1):185–207, 2014.

[25] M. Yamada, J. Tang, J. Lugo-Martinez, E. Hodzic, R. Shrestha, A. Saha, H. Ouyang, D. Yin, H. Mamitsuka, C. Sahinalp, P. Radivojac, F. Menczer, and Y. Chang. Ultra high-dimensional nonlinear feature selection for big biological data. *IEEE Transactions on Knowledge and Data Engineering*, 30(7):1352–1365, July 2018.

[26] J. Zhao, Y. Zhou, X. Zhang, and L. Chen. Part mutual information for quantifying direct associations in networks. *Proceedings of the National Academy of Sciences*, 113(18):5130–5135, 2016.

Chapter 7

A Comparative Study of Feature Selection Methods on Genomic Datasets

This paper is accepted in Artificial Intelligence for Healthcare: from black box to explainable models (AI4H:B^{2}E 2019), Cordoba, Spain.

7.1 Abstract

Feature selection plays an important role in reducing the size of datasets by choosing the most informative features and discarding the rest. The use of feature selection in microarray datasets for detecting cancer is widely investigated. In this paper we provide a series of comparisons between perturbation-based feature selection (PFS) and traditional methods, such as principal component analysis (PCA), correlation

based feature selection (CFS), and least-angle regression (LARS), and more recent methods, such as Hilbert-Schmidt independence criterion Lasso (HSIC-Lasso), minimum redundancy maximum relevance (mRMR), and a feature selection using support vector machines (FS-SVM). The performance of each method is demonstrated by conducting a series of comparisons on genomic cancer datasets, as well as, inflammatory bowel disease datasets. The experiments show that PFS and HSIC-Lasso are both scalable to large datasets.

Index terms— Perturbation, feature selection, least-angle regression, information gain, inflammatory bowel disease

7.2 Introduction

With the advancements in DNA sequencing both in terms of cost and time, it is now possible to genetically sequence a single suspect tissue for disease detection. In 2008, the first whole cancer genome was sequenced from leukaemia [24]. The fundamental task of genetic association studies is to detect genetic variations that contribute to disease status. One of the bottlenecks of working with these genomic datasets is their large-scale size that makes it difficult to render the data for meaningful analysis. Golub et al. [14] were the first to find 50 contributing genes which could accurately segregate acute myeloid leukemia from acute lymphoblastic leukemia cases.

Application of feature selection to genomic datasets is very important since the datasets are usually very large to magnitude of ten to hundred thousands features with very small number of samples. This is can be problematic for learning methods, and is referred to as *curse of dimensionality* where the number of feature are remarkably

larger than the number of samples [3].

In 2015, Hira and Gillies [17] reviewed a set of feature selection and feature extraction methods based on Pearson correlation coefficient and component analysis, respectively, as well as, wrapper methods based on support version machines. In another work by Boln-Canedo et al. [6], correlation based feature selection methods and those based on information theory were surveyed. They extend their review to wrapper and hybrid methods and investigated both binary and multi-class microarray datasets. Saeys et al. [28] also reviewed filter- and wrapper-based methods and their application in bioinformatics. They specifically focused on the application of feature selection to mass spectra and single nucleotide polymorphism analysis.

IBD is one of the most studied human disorders and it comprises Crohn's Disease (CD) and Ulcerative Colitis (UC). Due to the complexity of this disease, the recognition of these genes is a challenging task. In particular, machine learning techniques are employed in [34] to perform a risk assessment for CD and UC. They used a two-step feature selection strategy on a dataset containing 17,000 CD and 13,000 UC cases, and 22,000 controls with 178,822 features. Then they reduced the dataset by filtering out features with p-values greater than 10^{-4} and as such they obtained a reduced dataset with 10,799 features. Then they applied a penalized feature selection with L_1 penalty to select a subset of features from the reduced dataset.

This paper is organized as follows: in Section 7.3, we survey conventional and recent feature selection methods. In Section 7.4, we review perturbation based feature selection method. Next, we generate and compare the results in Section 7.5. Finally, we conclude the paper in Section 7.6.

7.3 Related works

Penalized logistic regression is a variable selection that shrinks the coefficients of less contributive variables toward zero. One of the most well-known variable selection methods based on least square regression is called least absolute shrinkage and selection operator (LASSO) [32]. Since solving the LASSO problem requires quadratic programming with linear inequality constraints, they employed Lawson and Hansen [23] method to solve the problem, However, the approach were designed to handle only one constraint as opposed to LASSO which has 2^P constraints, where p is the number of responses. Therefore, they solved the problem by introducing the constraints sequentially to satisfy Kuhn-Tucker conditions.

Later in 2004 Efron et al. [10] proposed a new variable selection called least angle regression (LARS) based on LASSO. This method starts by choosing a variable which has the highest correlation with the outcome, then it continues by finding another variable which is correlated with the current residual. Next, it continues toward the direction which is equally angled from both selected variables. The algorithm proceeds until after k steps, k regression coefficients are non-zero.

Classical feature selection methods include principal component analysis (PCA) and correlation-based feature selection (CFS). PCA is a feature extraction method in which data is transformed from its original space to a smaller one. In this method, the eigenvectors of the covariance matrix of the input dataset are calculated and the top ranked corresponding eigenvalues are selected. The complexity of the PCA for a dataset with m samples and n features is $O(\min(m^3, n^3))$ [21]. The CFS proposed by Hall in [16] is based on an equation introduced by Ghiselli [13] which calculates

feature-feature and feature-outcome correlations and chooses those features that are highly correlated to the outcome and weakly correlated to the other features.

The purpose of this paper is to compare CFS, PCA, LARS, minimal-redundancy-maximal-relevance (mRMR), Hilbert-Schmidt independence criterion least absolute shrinkage and selection operator (HSIC-Lasso), feature selection based on support vector machines (FS-SVM) with perturbation-based feature selection (PFS) against cancer datasets. Also, we will compare the performance of mRMR, HSIC-Lasso and FS-SVM on an IBD dataset containing the gene expression profiles of 59 Crohn's disease, 26 ulcerative colitis, and 42 normal samples obtained from GEO under accession number GSE3365 [7].

7.4 Feature Selection

Feature selection plays a vital role by reducing the dimension of datasets in health data and in bioinformatics. The main categorization for feature selection approaches is: feature ranking and feature subset selection [15]. The former is a subset of methods which features are ranked based on their *goodness* using a measure, see [39, 9, 19]. The latter approach has two main ingredients, search method and evaluation metric. These methods use the search method to find a subset which produces the best results regarding the evaluation metric, see [33, 12, 20]. It worth noting that feature ranking methods can be used as feature subset selection by choosing the top k features as the final subset.

A three-fold taxonomy of filter-based, wrapper-based and embedded methods [8] can be used to design a feature selection method. The former includes a set of

methods where a merit is employed to evaluate the quality of features through statistical/mathematical characteristics of the feature. These methods are fast and generally lead to less accurate models. The next subset of feature selection methods compensates the shortage of generating less accurate models by selecting features with regard to resulting accuracy produced by a classifier as a performance measure. These methods are computationally prohibitive compared to the filter-based methods but the resulting model is more accurate [22]. The last approach is embedded methods, in which features are sieved in the learning process of the induction algorithm. These methods are faster than wrapper-based methods, however, the feature selection method is tied into the learning algorithm and the resulting subset might not perform as well with the other classifiers.

7.4.1 Perturbation-based feature selection

In this section, we review our new perturbation-based feature selection method (PFS) [2]. PFS uses the method of least squares to find the most important features. The algorithm starts to calculate three properties of each feature – namely, the absolute distance of resulting $\tilde{X}$ of the perturbed features to the original X, the angle of each feature to the outcome, and the angle of the resulting outcome to the original outcome after removing each feature – to generate a $n \times 3$ table, where n is the number of features. Later, the generated table is clustered using k-means clustering into 2 to the numerical rank of the input dataset. Finally, SVM classifier is applied to all the clusters and the best subset is returned as the final outcome. The algorithm is shown in Algorithm 7.1.

Algorithm 7.1: Perturbation-based Feature Selection

Data: $D = [A \mid \mathbf{b}]_{m \times n+1}$

Result: Subset of features and resulting accuracy

Set $t = 10, s = 3$;

Normalize columns of A within $[0, 1]$;

$k = Rank(A)$;

for $i = 1$ **to** t **do**

- $E = 10^{-s} \cdot \mathbf{rand}$;
- $\tilde{A} = A + E$;
- Let $X = A^{+}\mathbf{b}$ and $\tilde{X} = (\tilde{A})^{+}\mathbf{b}$;
- **for** $j = 1$ **to** n **do**
  - Form the vector α consisting of the angles between $\mathbf{f}_j$ and $\mathbf{b}$;
  - Form the vector γ consisting of the angles between between $\hat{\mathbf{b}}_j$ and $\mathbf{b}$;
- **for** $c = 2$ **to** k **do**
  - form the $n \times 3$ matrix $[\text{abs}(X - \tilde{X}) \mid \alpha \mid \gamma]$;
  - cluster this data using k-means;
  - find and select centroid features of each cluster;
  - classify D based on the selected features;

return the classification accuracy along with the subset of features

The complexity of PFS is dominated by that of computing the SVD of an $m \times n$ matrix which can be done in $O(\min\{mn^2, m^2n\})$ and even faster as explained in [18].

7.4.2 Minimal-redundancy-maximal-relevance feature selection

Information theory based feature selection methods [38, 31, 4] use the notion of the amount of information each feature carry to the ones which are very related but redundant. One of the famous methods is minimal-redundancy-maximal-relevance criterion (mRMR) introduced by Peng et al. [27]. The computational cost of mRMR is rather high which makes it infeasible to scale up for large datasets as pointed out in [37].

7.4.3 Hilbert-Schmidt independence criterion least absolute shrinkage and selection operator feature selection

In 2014, Yamada et al. [36] introduced a method called Hilbert-Schmidt independence criterion least absolute shrinkage and selection operator (HSIC-Lasso) which is an efficient supervised feature selection for high-dimensional problems. It employes kernelized LASSO to uncover non-linear dependency between features and the the outcome. It is worth noting that HSIC-Lasso outperforms both LARS and LASSO in the sense that HSIC-Lasso attains better classification accuracy while selecting a smaller subset of features. The complexity of HSIC-Lasso is $O(n^4)$, where n is the number of features, however the current version of HSIC-Lasso is extremely tuned using C++ implementations and it can run efficiently as it can be seen in our exper-

iments.

7.4.4 Feature selection based on support vector machines

There much work on feature selection methods based on the support vector machines (SVM). SVM is a classification method based on optimization and there has been many variations of SVM to allow both feature selection and prediction performed simultaneously, see for example [1, 29]. The most recent of such methods called feature selection based on SVM (FS-SVM) is introduced in [11], in which, a pre-defined percentage of features is selected through adjusting l_1-norm of the classifier.

7.5 Experiments

Our ultimate goal in this section is to compare three breakthrough methods, namely PFS, HSIC-Lasso, and mRMR on some genomic datasets. Throughout this section we use SVM as the classifier for all the methods. To demonstrate the advantage of these methods to classical methods, in Section 7.5.3, we compare PFS with PCA, CFS, and LARS on five cancer datasets.

7.5.1 Data configuration

As it is pointed out in [5], it is important to withhold a section of data just for testing. It is noted that in some studies, the whole dataset is used for the feature selection process and then the model is tested on part of the data; this set up, in turn, leads to a significant bias in the estimates of the predictive accuracy. We used 70% of the data for feature selection and model creation and the remaining 30% just for testing.

Table 7.1: Dataset specifications

Dataset	**Samples**	**Features**
Colon	62	2000
Lung	203	3312
GLIOMA	50	4434
Leukemia	72	7070
ALLAML	72	7129

7.5.2 Hardware and software settings

All the computations have been done on an macOS Mojave 10.14.3 machine with Intel®Core™i7, 16 GB of RAM, using MATLAB® 9.4.0.813654 (R2018a), and Java™SE Runtime Environment (build 1.8.0_191-b12).

7.5.3 Comparisons with conventional methods

We compared our proposed method with correlation-based feature selection (CFS); principal component analysis (PCA) (in combination with Ranker search method [35] to select enough eigenvectors to cover 95% of variance in data); least absolute shrinkage and selection operator (LASSO) [32]; and least angle regression (LARS) [10].

The CFS and PCA are ready to use in WEKA [35] with default parameters. We used MATLAB® implementations of LARS and LASSO by Sjöstrand [30]. The performance of LASSO and LARS on these datasets are very similar and as such we included the results of LARS only.

Table 7.2: Number of selected features and resulting classification accuracies using PCA, CFS, LARS and PFS

Dataset	**# of selected features**				**Classification Accuracy**			
	PCA	**CFS**	**LARS**	**PFS**	**PCA**	**CFS**	**LARS**	**PFS**
Colon	45	13	61	39.30	71.67	73.21	79.49	92.61
Lung	147	74	202	100.00	76.78	63.39	63.58	99.95
GLIOMA	36	62	49	36.50	56.25	68.75	53.75	90.08
Leukemia	63	60	71	41.00	73.96	93.75	96.88	99.71
ALLAML	60	44	71	53.40	68.10	93.33	90.83	96.29

We perform a series of tests on the datasets in Table 7.1 selected from ASU dataset repository [25] and UCI repository of machine learning [26]. The results are presented in Table 7.2. We run PFS for 10 times on randomly shuffled datasets, and report the average of classification accuracies and the size of selected feature. We can see that PFS consistently uses the least number of features compared to the other methods while the classification accuracy of PFS is considerably better than the others.

The point of this experiment is to show that the performance of classical methods are inferior to newer novel methods such as PFS. As such, for the rest of this paper, we shall compare PFS, HSIC-Lasso, and mRMR along with a recent method based on support vector machines (SVM) [11].

Table 7.3: Dataset specifications used in [11]

Dataset	**Samples**	**Features**
Lung	139	1,000
Leukemia	72	7,130
Prostate	102	12,601

Table 7.4: Datasets configuration as proposed in [11]

Dataset	**Train**		**Test**	
	#Cases	**#Controls**	**#Cases**	**#Controls**
Lung	9	70	8	69
Leukemia	24	13	23	12
Prostate	25	26	25	26

7.5.4 Comparison with FS-SVM

In [11], the authors applied their method to a set of cancer datasets, as shown in Table 7.3, adopted from Cancer Program at Broad Institute[1]. For each dataset, a subset of cases and controls have been selected for training and testing purposes (see Table 7.4).

To comply with the settings proposed by Ghaddar and Naoum-Sawaya, we used the same setup and the resulting size of selected features, classification accuracies, and running times are presented in Figures **??**, **??** and **??**, respectively.

For Lung, Leukemia and Prostate datasets, mRMR, HSIC-Lasso and HSIC-Lasso

[1]http://www.broad.mit.edu/cgi-bin/cancer/datasets.cgi

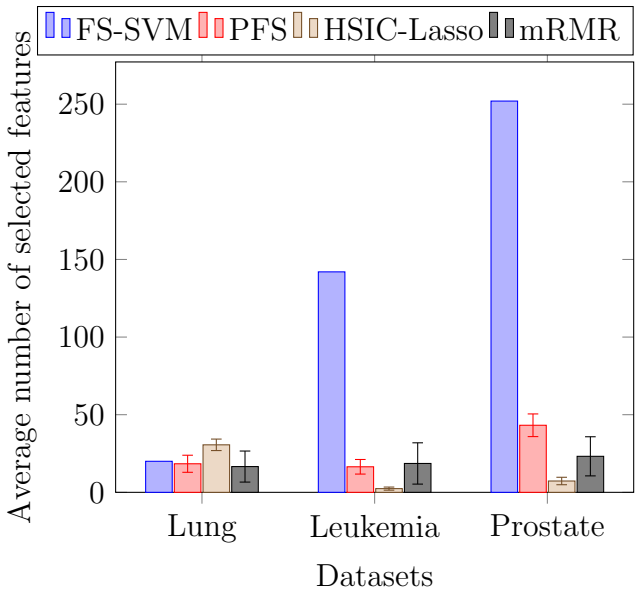


Figure 7.1: Number of selected features using FS-SVM, PFS, HSIC-Lasso and mRMR for Lung, Leukemia and Prostate datasets

has selected the smallest subsets of features, respectively. For the same order of datasets, FS-SVM, PFS, and mRMR has selected the highest classification accuracies. In the terms of running time, PFS and HSIC-Lasso performed very closely while FS-SVM run considerably slower.

7.5.5 Inflammatory bowel disease

In this section, we show experiments with a dataset containing the gene expression profiles of 59 Crohn's disease, 26 ulcerative colitis, and 42 normal samples from GEO under accession number GSE3365 [7]. The expression levels of 22,284 genes were measured using an Affymetrix Human Genome U133A Array. We applied the PFS, HSIC-Lasso, and mRMR to the GSE3365 dataset and the results are shown in Tables 7.5 and 7.6 where we run each of the PFS and HSIC-Lasso for 20 times. For each run

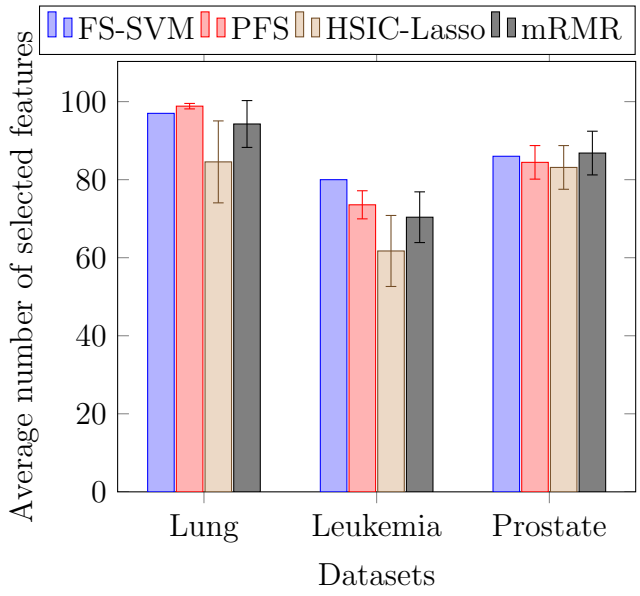


Figure 7.2: Classification accuracies (%) of the resulting subsets of FS-SVM, PFS, HSIC-Lasso and mRMR for Leukemia, Lung and Prostate datasets using SVM

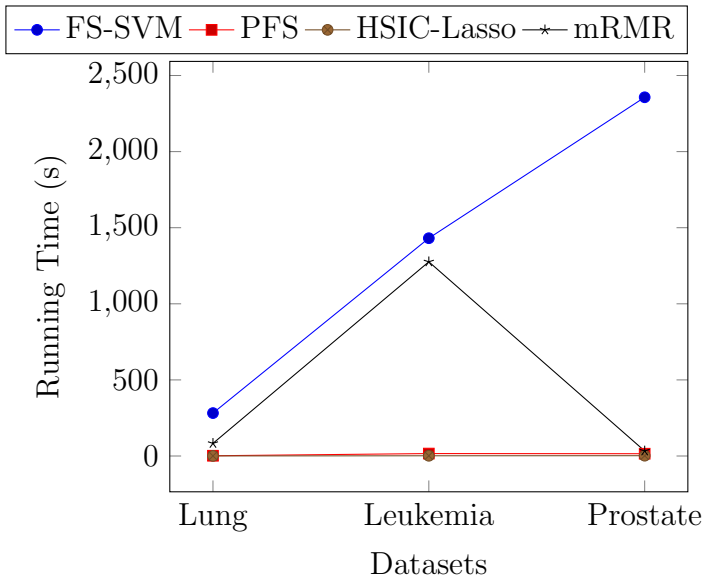


Figure 7.3: Running Time (s) of FS-SVM, PFS, HSIC-Lasso and mRMR for Leukemia, Lung and Prostate datasets using SVM

Table 7.5: Number of selected features for IBD dataset using PFS, HSIC-Lasso, and mRMR

Dataset	**# Features**		
	PFS	**HSIC-Lasso**	**mRMR**
IBD	57.90±12.50	38.20±3.40	35

Table 7.6: Resulting classification accuracies for IBD dataset for the selected features using PFS, HSIC-Lasso, and mRMR

Dataset	**Classification Accuracy**		
	PFS	**HSIC-Lasso**	**mRMR**
IBD	93.93±2.50	87.06±4.20	74.24

we have randomly shuffled the dataset and report the average and standard deviation of accuracies along with the average and standard deviation of number of selected features. We remark that the running time of mRMR on the dataset was 3262 seconds whereas the running time of PFS and HSIC-Lasso was 42 and 9 seconds, respectively. Due to high running time of mRMR, we have only reported the results of this method for a single run.

To provide further comparisons between PFS, HSIC-Lasso, and mRMR, we have created three two-class datasets from the original dataset as shown in Table 7.7. We applied PFS, HSIC-Lasso and mRMR to all three datasets and the results are shown in Table 7.8. For each of two-class datasets, we have run both PFS and HSIC-Lasso for 20 times where in each run the dataset is randomly shuffled. We have reported

Table 7.7: Two-class datasets extracted from GSE3365 dataset

Dataset	**Class**
Dataset 1	Normal and Crohn's Disease
Dataset 2	Normal and Ulcerative Colitis
Dataset 3	Ulcerative Colitis and Crohn's Disease

average classification accuracy and average number of features over the 20 run for each dataset.

7.6 Conclusions and future work

In this paper, we presented PFS, HSIC-Lasso, mRMR as feature selection method that are favorable to classical and conventional methods such as PCS, CFS, LARS, and LASSO. The performance of PFS, HSIC-Lasso, mRMR on various datasets are tested. In terms of classification accuracy and size of selected subset of features, all three methods perform well.

In particular, in datasets where we have hundreds of samples and thousands of features ($m^2 \leq n$), the complexity of PFS is close to quadratic. The current running times of PFS are based on MATLAB® which can be greatly improved by using C++ implementations.

As a future work, we will remove the clustering phase of PFS to further improve the running time and select smaller subsets while attaining similar classification accuracies.

Table 7.8: Number of selected features and the resulting classification accuracies of applying PFS, HSIC-Lasso and mRMR to the three datasets shown in Table 7.7

Method	**Dataset**	**# Features**	**CA (%)**	**Time(s)**
PFS	Dataset 1	27.40±5.10	96.61±2.60	42
	Dataset 2	27.60±7.70	97.88±1.90	51
	Dataset 3	39.90±9.90	89.89±5.60	81
HSIC-Lasso	Dataset 1	20.60±3.60	98.15±1.20	8
	Dataset 2	12.90±2.40	96.38±3.50	5
	Dataset 3	9.90±3.50	91.75±5.30	6
mRMR	Dataset 1	30	95.83	1006
	Dataset 2	25	96.15	1275
	Dataset 3	25	89.16	1868

Acknowledgement

The research of the second author was supported by NSERC of Canada under grant # RGPIN 418201.

Bibliography

[1] M. F. Akay. Support vector machines combined with feature selection for breast cancer diagnosis. *Expert Systems with Applications*, 36(2, Part 2):3240 – 3247, 2009.

[2] J. R. Anaraki and H. Usefi. A feature selection based on perturbation theory. *Expert Systems with Applications*, 127:1–8, 2019.

[3] R. Bellman. Dynamic programming. *Science*, 153(3731):34–37, 1966.

[4] M. Bennasar, Y. Hicks, and R. Setchi. Feature selection using joint mutual information maximisation. *Expert Systems with Applications*, 42(22):8520–8532, 2015.

[5] M. L. e. a. Bermingham. Application of high-dimensional feature selection: evaluation for genomic prediction in man. *Scientific reports*, 5 10312, 2015.

[6] V. Bolón-Canedo, N. Sánchez-Marono, A. Alonso-Betanzos, J. M. Benítez, and F. Herrera. A review of microarray datasets and applied feature selection methods. *Information Sciences*, 282:111–135, 2014.

[7] M. E. Burczynski, R. L. Peterson, N. C. Twine, K. A. Zuberek, B. J. Brodeur, L. Casciotti, V. Maganti, P. S. Reddy, A. Strahs, F. Immermann, et al. Molecu-

lar classification of crohn's disease and ulcerative colitis patients using transcriptional profiles in peripheral blood mononuclear cells. *The journal of molecular diagnostics*, 8(1):51–61, 2006.

[8] G. Chandrashekar and F. Sahin. A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1):16–28, 2014.

[9] Y.-W. Chang and C.-J. Lin. Feature ranking using linear svm. In *Causation and Prediction Challenge*, pages 53–64, 2008.

[10] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, et al. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004.

[11] B. Ghaddar and J. Naoum-Sawaya. High dimensional data classification and feature selection using support vector machines. *European Journal of Operational Research*, 265(3):993–1004, 2018.

[12] I. A. Gheyas and L. S. Smith. Feature subset selection in large dimensionality domains. *Pattern recognition*, 43(1):5–13, 2010.

[13] E. Ghiselli. *Theory of psychological measurement*. McGraw-Hill, 1964.

[14] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 2013.

[15] M. Hall, G. Holmes, et al. Benchmarking attribute selection techniques for discrete class data mining. *Knowledge and Data Engineering, IEEE Transactions on*, 15(6):1437–1447, 2003.

[16] M. A. Hall. *Correlation-based feature selection for machine learning*. PhD thesis, The University of Waikato, 1999.

[17] Z. M. Hira and D. F. Gillies. A review of feature selection and feature extraction methods applied on microarray data. *Advances in bioinformatics*, 2015, 2015.

[18] M. Holmes, A. Gray, and C. Isbell. Fast svd for large-scale matrices. In *Workshop on Efficient Machine Learning at NIPS*, volume 58, pages 249–252, 2007.

[19] S. J. Hong. Use of contextual information for feature ranking and discretization. *IEEE transactions on knowledge and data engineering*, 9(5):718–730, 1997.

[20] Q. Hu, D. Yu, J. Liu, and C. Wu. Neighborhood rough set based heterogeneous feature subset selection. *Information sciences*, 178(18):3577–3594, 2008.

[21] I. M. Johnstone and A. Y. Lu. Sparse principal components analysis. *arXiv preprint arXiv:0901.4392*, 2009.

[22] R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial intelligence*, 97(1):273–324, 1997.

[23] C. L. Lawson and R. J. Hanson. *Solving least squares problems*, volume 15. Siam, 1995.

[24] T. J. Ley, E. R. Mardis, L. Ding, B. Fulton, M. D. McLellan, K. Chen, D. Dooling, B. H. Dunford-Shore, S. McGrath, M. Hickenbotham, et al. DNA sequencing of a cytogenetically normal acute myeloid leukaemia genome. *Nature*, 456(7218):66, 2008.

[25] J. Li, K. Cheng, S. Wang, F. Morstatter, T. Robert, J. Tang, and H. Liu. Feature selection: A data perspective. *arXiv:1601.07996*, 2016.

[26] M. Lichman. UCI machine learning repository, 2013.

[27] H. Peng, F. Long, and C. Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence*, 27(8):1226–1238, 2005.

[28] Y. Saeys, I. Inza, and P. Larrañaga. A review of feature selection techniques in bioinformatics. *bioinformatics*, 23(19):2507–2517, 2007.

[29] Y.-H. Shao, C.-N. Li, M.-Z. Liu, Z. Wang, and N.-Y. Deng. Sparse lq-norm least squares support vector machine with feature selection. *Pattern Recognition*, 78:167 – 181, 2018.

[30] K. Sjöstrand. Matlab implementation of LASSO, LARS, the elastic net and SPCA, jun 2005. Version 2.0.

[31] X. Sun, Y. Liu, M. Xu, H. Chen, J. Han, and K. Wang. Feature selection using dynamic weights for classification. *Know.-Based Syst.*, 37:541–549, Jan. 2013.

[32] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[33] C. Wang, Q. Hu, X. Wang, D. Chen, Y. Qian, and Z. Dong. Feature selection based on neighborhood discrimination index. *IEEE transactions on neural networks and learning systems*, 29(7):2986–2999, 2018.

[34] Z. Wei, W. Wang, J. Bradfield, J. Li, C. Cardinale, E. Frackelton, C. Kim, F. Mentch, K. Van Steen, P. M. Visscher, et al. Large sample size, wide variant spectrum, and advanced machine-learning technique boost risk prediction for inflammatory bowel disease. *The American Journal of Human Genetics*, 92(6):1008–1012, 2013.

[35] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.

[36] M. Yamada, W. Jitkrittum, L. Sigal, E. P. Xing, and M. Sugiyama. High-dimensional feature selection by feature-wise kernelized lasso. *Neural computation*, 26(1):185–207, 2014.

[37] M. Yamada, J. Tang, J. Lugo-Martinez, E. Hodzic, R. Shrestha, A. Saha, H. Ouyang, D. Yin, H. Mamitsuka, C. Sahinalp, P. Radivojac, F. Menczer, and Y. Chang. Ultra high-dimensional nonlinear feature selection for big biological data. *IEEE Transactions on Knowledge and Data Engineering*, 30(7):1352–1365, July 2018.

[38] J. Zhao, Y. Zhou, X. Zhang, and L. Chen. Part mutual information for quantifying direct associations in networks. *Proceedings of the National Academy of Sciences*, 113(18):5130–5135, 2016.

[39] Q. Zou, J. Zeng, L. Cao, and R. Ji. A novel features ranking metric with application to scalable visual and bioinformatics data classification. *Neurocomputing*, 173:346–354, 2016.

Chapter 8

Summary

Feature selection is a very important pre-processing stage in which irrelevant and redundant features are removed from a dataset to attain/improve resulting classification accuracy and regression error. The other benefit of using feature selection methods before applying any induction learning algorithm is that it minimizes computational complexity and the effect of noise in the process of generating models.

Finding and removing irrelevant features is a relatively easier process when compared to redundant features, because irrelevant features usually show no evidence of incorporation in deciding the outcome. However, redundant features might show some level of correlation not only to the outcome but also to the other features. There has been a significant amount of research on finding and removing redundant features in the past with a remarkable success rate; however, we have a long way to go in designing and implementing effective algorithms, particularly for large datasets.

In this thesis, we took two major paths toward uncovering and discarding redundant features. First, we used fuzzy-rough set theory to simultaneously select the most

informative features and samples using a shuffled frog leaping algorithm. Then, we used multi-tree genetic programming to classify the results and to finally show the applicability of the proposed improvements against a brain signal dataset captured using functional near-infrared spectroscopy.

To further our investigations in utilizing fuzzy-rough set for feature selection, we designed a new measure in which we focus on removing redundant features by incorporating the fuzzy-rough dependency degree of each feature to the rest of the features. To show the potential of the newly designed measure, we compared our method to three well-known feature selection methods in a two-step fashion. At first, we applied our method to a list of 25 datasets, classified the resulting subsets using nine classifiers and reported the average classification accuracies. In the next step, we compared all methods using two newly introduced performance measures that were specially crafted to amplify the importance of the reduction ratio.

Improving and designing new measures based on fuzzy-rough set theory was one phase of our research in this area. The other exciting phase was to modify a shuffled frog leaping algorithm so that the similarity between individuals can be calculated using a fuzzy-rough positive region. In this way, we proposed a version of a shuffled frog leaping algorithm that works in complete harmony with the evaluation measure (i.e. fuzzy-rough dependency degree).

To widen the application of fuzzy-rough set feature selection to an area where researchers might not have direct access to data or data are distributed between several parties, we designed four privacy-preserving versions of the original fuzzy-rough set-based feature selection for four different configurations: two parties with horizontally partitioned data, multi-parties with horizontally partitioned data, two

parties with vertically partitioned data, and multi-parties with vertically partitioned data.

The other primary path we took in this thesis was using perturbation theory for feature selection. We proposed a new feature selection based on a brilliant architectural monument called Menar Jonban (Shaking minarets) to accurately detect and remove redundant features by perturbation theory. We clustered like-behaved features in a certain number of groups and refined each cluster using calculating the angle of each feature to the outcome, as well as, the resulting angle of the determined outcome after removing each feature to the original one. We showed the effect of choosing different clustering methods on the resulting subsets by comparing k-means and fuzzy c-means.

To investigate the applicability of the proposed method to genomic datasets, we designed a study where we compared our proposed method with three traditional methods: principal component analysis, correlation-based feature selection, and least angle regression, and three well-known and powerful feature selection methods: minimum redundancy maximum relevance, Hilbert-Schmidt independence criterion Lasso, and a feature selection based on support vector machines. We compared all methods over eight genomic datasets with the number of features ranging from 2,000 to 12,000.

In this thesis, we provided a series of methods for effectively finding and removing redundant features and assessed their performance against a variety of datasets. In the near future, we are planning to improve our recent feature selection method based on perturbation theory by using density-based clustering methods to remove the need for brute-forcing over all possible values of k, although we limited the upper bound. We are also aiming to improve the performance of choosing redundant features by

incorporating more measures to help the clustering method refine the results even further. Moreover, we will expand the applicability of the proposed method by providing a website that will accept a dataset and will apply an extensively optimized C++ implementation of the method in the back-end and email back the results.