Contents

Chapter 1

Introduction and Motivation

As the advancements of technology continuously grow, the amount of generated data, as well as the amount of data to be processed, increases at an unprecedented scale [9]. The process to analyze, make sense and gain insights from such extent and variety of data is widely-known as **big data analytics** [2]. Various real-world applications of big data analytics, such as genomics, biometric recognition, text mining, text classification, etc; often involve high-dimensional data with large number of features and, sometimes, large number of instances. Consider the variety aspect in big data, different types of data also complicates the analytics process.

Machine learning are commonly used to gain insights or solve prediction task upon those data. The use of all the available features does not guarantee the best accuracy, training, and classification time of the machine learning task. In fact, having more features could incur greater computational cost and potentially cause over fitting in the machine learning model construction. In order to avoid over fitting, many algorithms employ the Occams Razor bias [6] to build a simple model that still achieves some acceptable level of performance on the training data. This bias often leads an algorithm to prefer a small number of predictive features over a large number of features that, if used in the proper combination, are fully predictive of the class label [8]. Too much irrelevant and redundant information or noisy and unreliable data could make the learning process more difficult. Having a subset of features could potentially produce better accuracy, training, and classification time. Selecting the subset of features uses **feature selection**.

Feature selection is an important technique in data preprocessing for machine learning problems. The basic task of feature selection is to reduce the number of features by removing irrelevant or redundant features and noisy data prior to model construction. It is useful to simplify models to make it more understandable, to provide cost-effective training time and to enhance generalization by reducing over fitting. The ultimate goal of feature selection is to speed up a machine learning process, and potentially improve machine learning model performance, such as predictive accuracy and result comprehensibility [14].

Decision-making process nowadays requires data to be processed in a real-time manner. Scalability becomes an important concern to choose certain machine learning model. Certainly, feature selection algorithm also needs to be scalable to produce a fully cost-effective machine learning pipeline.

Different approaches have been proposed to handle feature selection for large data. One way is to use the distributed environment to increase processing speed, as a common way for handling big data in general. Parallel feature selection algorithms are needed to run in such setting. This thesis focuses on parallel feature selection algorithms to handle large scale data. We focus on classification problem as the machine learning problem.

This chapter introduces the thesis and its purpose in relation to machine learning. Section **??** describes the motivation of this thesis and Section 1.1 describes the objectives of this thesis. The outline of this thesis is explained in Section 1.2.

Researchers conduct studies, in the area of data analytics and machine learning, concerning the feature selection. They focus on improving the efficiency of the feature selection process, feature ranking, multivariate feature selection, efficient search methods, and feature validity assessment methods [7]. Other study focus on coming up with best practices for feature selection [1, 7], or describing a feature-selection language and creating optimization methods, such as COLUMBUS [20].

Most of the existing methods are designed to run in a centralized manner. Consider current trend of unprecedented growth of data, feature selection in a centralized manner is not feasible, as it increases the running time requirements. Applying wrapper methods, for example, can make the search space extensive when we deal with high-dimensional data. Studies compared top-performing machine learning algorithms, such as logistic regression, Random Forests and SVMs [3]. However, problem of feasible and efficient computation of feature evaluation still remains as current issues, especially when dealing with large scale problems. An example of study proposed a parallel large-scale feature selection approach for logistics regression [17].

1.1 Thesis Objectives

This thesis aims to provide insights over existing feature selection algorithms in distributed machine learning context, especially to tackle large scale data. This begins by listing existing feature selection algorithms and analyze the complexity and performance of each algorithm. Then, if there is any, existing proposed parallel approach of each algorithm are discussed. An example of feasible feature selection algorithm is then selected. An experimental evaluation over selected feature selection approach aims to test its feasibility when handling large scale data in practice.

The literature study aims to deliver a summary of how feature selection algorithms contribute to the performance in solving the feature selection problem. We use the classification of feature selection algorithms proposed in [14] and update the list of the algorithms with the most up-to-date improvements of each algorithm. For each of the category, the study thoroughly describes how each algorithm works and analyzes its scalability and performance from various experiments conducted by other researchers. Then, the study lists down any room of improvement or open problems left for each category that possibly leads to better performance when scaling up the data sets. The description of improvement on the scalability includes the possibility of optimization or the feasibility of parallelizing the feature selection approaches.

The thesis also conducts an experiment to verify the applicability of the selected approaches in a distributed context. This involves a complete pipeline of training, modelling and evaluating the features to examine the quality of the approximate machine learning model using the selected feature selection approach. Selected comparable approaches are implemented in Apache Spark, to allow parallelizing the computation over the training sets and potential features. The implementations are applied to two high-dimensional datasets, the internet ads data sets and the YouTube multiview video games data set, taken from UCI repository [12]. The goal of the experiment is to assess the modelling performance (accuracy) and the scalability when run in a distributed context. Additionally, the experimental evaluation is also useful to identify benefits and drawbacks for the algorithm.

1.2 Thesis Overview

The thesis is organized in five chapters. Chapter 2 provides the background study, which explains the main concepts of feature selection, common feature selection categories, and the scalability analysis for each algorithm. The reason to select a feature selection algorithm that is implemented is also explained in this chapter. Chapter 3 explains the implementation detail of the algorithm in Apache Spark. Chapter 4 explains the experiment settings, such as the chosen dataset, the classification model, and the evaluation over the model after the applying feature selection implementation. Chapter 5 concludes the thesis by listing the advantages, the drawbacks of feature selection algorithms, and the potential area of future works.

Chapter 2

Feature Selection for Classification

Feature selection can be found in many areas of data mining and machine learning, such as classification, clustering, association rules, and regression. We focus on classification problem as the machine learning model. Feature selection algorithms aims to produce a subset of features that could improve the performance of the classifier that follows the feature selection process. In case of classifier that is used for prediction problem, feature selection could provide faster and more cost-effective predictors.

A paper by Liu [14] described the feature selection process in a simple way, that it firstly selects a subset of original features, then measures the optimality of a subset by using certain evaluation criterion. The challenge of the feature selection process lies on finding an optimal feature subset, which usually is intractable (hard to control) and some problems are even NP-Hard problems. With that, it is even more challenging to apply feature selection on high-dimensional data or big data. A common approach to easily conquer big data problems is to use parallel processing. Recent study by Singh [17] proposed a parallel approach to handle large scale feature selection specific for logistic regression problem. Several other parallel feature selection approaches also exist. However, the study over the advantages and drawbacks of those algorithms are still limited.

This chapter explains the categories of feature selection method in Section 2.1 and key steps or characteristics of feature selection in Section 2.2. Then, we list down several existing feature selection algorithms. Then we describe those algorithms that represents characteristics of different approaches in Section 2.3. We discovered some feature selection algorithms that have been applied or proposed using parallel processing. In Section 2.4, those particular algorithms are explained. Finally in Section 2.5, we choose an algorithm that we implement in the next chapter and we explain the reason for choosing it.

2.1 Feature Selection Categories

Feature selection algorithms can be categorized based on its evaluation criteria into three categories:

1. **Filter methods**

   Filter methods apply statistical measures to assign a scoring to each feature and then decide to remove or not the respective features. These methods often consider each feature independently or with regard to the dependent variable. Commonly used filter methods are Chi-square test, information gain and correlation coefficient scores.

   Filter methods rely on general characteristics of the data to evaluate and select feature subsets without involving any mining algorithm. Filter methods use a suitable criterion to score the features. One approach is: feature are then ranked or ordered based on the evaluation score. This approach is known as variable ranking. A threshold is then used to remove features having evaluation score below the threshold. This can be seen as to filter out less relevant features before classification. Another approach is to firstly generate all possible subsets and evaluate the score to determine whether the it is a subset that contains only relevant features. A feature is considered relevant when it has useful information about the different classes in the data. This property can be defined as feature relevance, which provides a measurement of the features usefulness in discriminating the different classes [4].

2. **Wrapper methods**

Wrapper methods, like feature elimination algorithm, prepare different combinations of set of features, evaluate them using a prediction model, assign a score based on model accuracy and compare with other models. This method considers the selection of a set of features as a search problem, which can use best-first search as a methodical algorithm, random hill-climbing algorithm as a stochastic algorithm, forward/backward passes as a heuristic algorithm, or any other search methods.

Wrapper methods requires a predetermined mining algorithm or a predictor as a black box and uses its performance as the criterion to evaluate subsets. Since evaluating 2^N subsets becomes a NP-hard problem, suboptimal subsets are found by employing search algorithms, which find a subset heuristically.

3. **Embedded methods**

   This model attempts to take advantage of the filter and wrapper models by exploiting their different evaluation criteria in different search stages.

   Embedded methods learn which features best contribute to the accuracy of the model while the model is being created. Commonly used embedded methods are LASSO (which selects a few sparse features), ElasticNet, and Ridge Regression.

2.2 Key Steps of Feature Selection

Liu's work in [14] summarizes the four basic steps in feature selection process, as depicted in Figure 2-1. Subset generation step produces candidate feature subsets for evaluation. Each candidate subset is evaluated and compared with the previous best one according to a certain evaluation criterion. If the new subset is better, it replaces the previous best subset. The process of subset generation and evaluation is repeated until a given stopping criterion is satisfied. The selected best subset usually needs to be validated by prior knowledge or different tests. The detail of the four key steps are explained below.

1. **Subset generation**

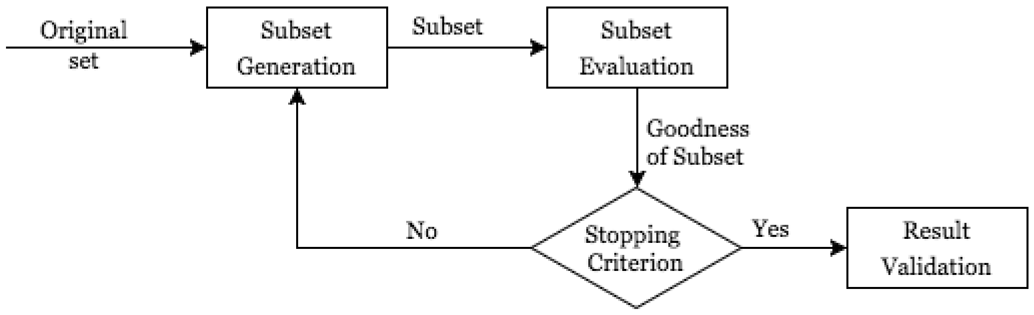


Figure 2-1: Four steps of feature selection (Source: [14])

It is a search problem that can be seen as a heuristic search, which produces candidate feature subsets for evaluation. Each state in the search space specifies a candidate subset for subset evaluation. The quality of searching depends on two basic issues.

First, search starting point determines the search direction. Forward selection starts from an empty set and incrementally add features. Backward selection starts with a full set and incrementally remove features. Bi-directional selection starts with both ends and add and remove features simultaneously. The quality of the subset produced on those search strategies heavily depends on the order of the features in the subset. In the case of getting trapped in the local minima, the subset is no longer the most optimal subset. Random selection tries to avoid that by starting with randomly selected subset [5].

Second, search strategy could affect the processing time. A data set with k features has 2^k candidate subsets. Thus, the search space is exponential and could become not scalable for large number of k. Different search strategies have been explored: complete, sequential, and random search.

- **Complete search** guarantees to find the optimal result. It does not have to be exhaustive. Different heuristic functions can reduce the search space without jeopardizing the chances of finding the optimal result. Hence, although the search space is $O(2^N)$, a smaller number of subsets are evaluated.

- **Sequential search** gives up completeness and thus risks losing optimal subsets. There are many variations to the greedy hill-climbing approach, such as *sequential forward selection*, *sequential backward elimination*, and *bi-directional selection*. Algorithms with sequential search are simple to implement and fast in producing results as the order of the search space is usually $O(N^2)$ or less.

- **Random search** starts with a randomly selected subset and proceeds with two different ways. One is to follow search to inject randomness into the classical sequential approaches, i.e. *random-start hill-climbing* and *simulated annealing*. Another one is to generate the next subset in a completely random manner (*Las Vegas* algorithm). Both approeaches use randomness to escape local optima in the search space. The optimality of the selected subset depends on the resources available.

2. **Subset evaluation**

   After the subset generation, each candidate subset is evaluated using an evaluation criterion. The goodness of a subset is determined by a certain criterion. An optimal subset selected using one criterion may not be optimal according to another criterion. Evaluation criteria can be categorized based on their dependency on mining algorithms that will be applied onto the selected subset.

   - Independent criteria Typically this criteria is used in algorithms of the filter model. It tries to evaluate the goodness of a feature or feature subset by exploiting the intrinsic characteristics of the training data without involving any mining algorithm. Some widely-used examples are distance measures, information measures, dependency measures, and consistency measures [14].

   - Dependent criteria This criteria requires predetermined mining algorithm and uses the performance of the mining algorithm applied on the selected subset to determine which features should be selected. Therefore, this criteria is used in algorithms of the wrapper model and gives better performance on suited predetermined mining algorithm. The drawback of the this criteria is that it tends to be more computationally expensive, and may not be suitable for other mining algorithms.

3. **Stopping criterion**

   Recall that the process of subset generation and evaluation is repeated until a given stopping criterion is satisfied. For example, (a) the search completes; (b) some given bound (mininum number of features or maximum number of iterations) is reached; (c) subsequent addition (or deletion) of any feature does not produce a better subset; and (d) a sufficiently good subset is selected (e.g. its classification error rate is less than the allowable error rate).

4. **Result validation**

   One way for result validation is to directly measure the result using prior knowledge about the data. However, in most cases, we usually do not have such prior knowledge. Hence, we have to monitor the change of mining performance with the change of features.

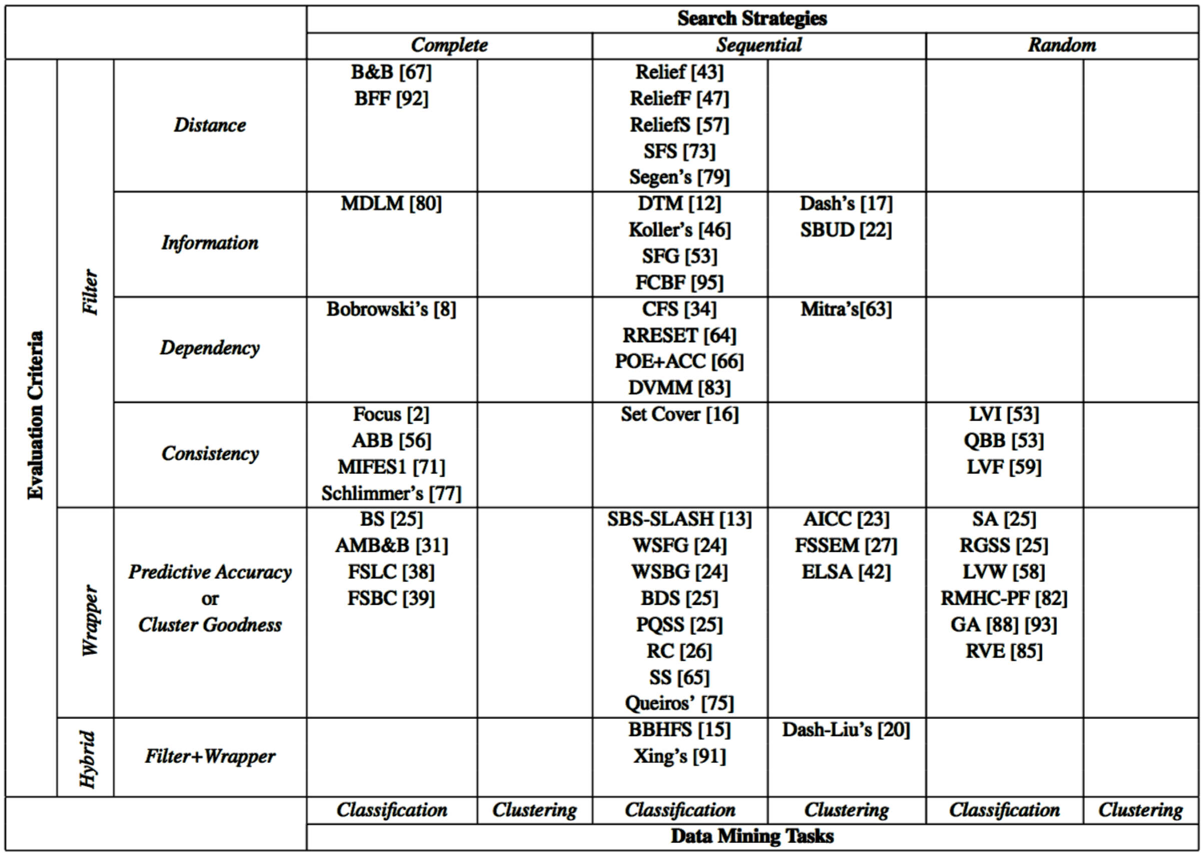
2.3 Existing Algorithms

Feature selection algorithms are categorized in three-dimensional framework in [14]. The three dimensions are evaluation criteria, search strategies and data mining tasks on which algorithms are more suitable to perform. We based our algorithms list on the categorization in Figure 2-2 and added more up-to-date algorithms to the list.

2.3.1 Filter Methods

- Branch and Bound algorithm (B&B)

  B&B [15] is a search algorithm proposed in 1977. It is a depth-first-search tree-based feature elimination strategy, by evaluating the criterion at each level and sort them, continued with pruning. The advantage of this algorithm is: it guarantees that the selected subset yields the globally best value of any criterion that satisfies monotonicity. However, as it performs the full complete search,



			Search Strategies					
			Complete		Sequential		Random	
Evaluation Criteria	Filter	*Distance*	B&B [67] BFF [92]		Relief [43] ReliefF [47] ReliefS [57] SFS [73] Segen's [79]			
		Information	MDLM [80]		DTM [12] Koller's [46] SFG [53] FCBF [95]	Dash's [17] SBUD [22]		
		Dependency	Bobrowski's [8]		CFS [34] RRESET [64] POE+ACC [66] DVMM [83]	Mitra's[63]		
		Consistency	Focus [2] ABB [56] MIFES1 [71] Schlimmer's [77]		Set Cover [16]		LVI [53] QBB [53] LVF [59]	
	Wrapper	*Predictive Accuracy* or *Cluster Goodness*	BS [25] AMB&B [31] FSLC [38] FSBC [39]		SBS-SLASH [13] WSFG [24] WSBG [24] BDS [25] PQSS [25] RC [26] SS [65] Queiros' [75]	AICC [23] FSSEM [27] ELSA [42]	SA [25] RGSS [25] LVW [58] RMHC-PF [82] GA [88] [93] RVE [85]	
	Hybrid	*Filter+Wrapper*			BBHFS [15] Xing's [91]	Dash-Liu's [20]		
			Classification	*Clustering*	*Classification*	*Clustering*	*Classification*	*Clustering*
			Data Mining Tasks					

Figure 2-2: Categorization of feature selection algorithms (Source: [14])

the number of subsets evaluated can be huge as the number of features grow. The complexity of the algorithm could grow to exponential.

- Best First strategy for feature selection (BFF)

  BFF [18] was proposed as an improvement of B&B algorithm. It is a heuristic search strategy that guarantee the globally best subset without exhaustive enumeration for any criterion that satisfies monotonicity. The advantage is that the number of subsets evaluated by BFF is less than those needed by B&B algorithm. However, the complexity of the algorithm remains to be exponential as it utilizes graph and compute the path to select subsets.

- Relief

  Relief algorithm selects relevant features using a statistical method, does not depend on heuristics, is accurate even if features interact, and is noise-tolerant [10]. For each randomly selected instance, find nearest hit and miss to determine quality estimation. It can deal with nominal and numerical attributes, but it cannot deal with incomplete data and is limited to two-class problems. The complexity of this algorithm is $O(kN^2)$ (quadratic).

- ReliefF

  ReliefF algorithm [11] is an variant of improvements of Relief algorithm. ReliefF tries to overcome Relief algorithm's limitation on handling only binary classes. ReliefF handles multiple classes and incomplete and noisy data. For each randomly selected instance, find nearest hit from the same class. For neighbours that are not in the same class, compute the miss. Update quality estimation for each attribute based on those difference. The complexity of the algorithm is improved to $O(kN)$, however, the optimal results are not guaranteed.

- RReliefF

  ReliefF was further extended to RReliefF in order to handle continuous classes in regression. [16] shows that RReliefF is effective in selecting features in learning regression trees for regression problems.

```
Given $t$ - bucket size,

1.  set all weights $W[A_i] := 0.0$;
2.  buildKDTree($t$);
3.  $m :=$ number of buckets;
4.  for $j := 1$ to $m$ do begin
5.    randomly select an instance $X$ from $Bucket[j]$;
6.    fi nd nearest hit $H$ and nearest miss $M$;
7.    for $i := 1$ to $k$ do begin
8.      $W[A_i] := W[A_i] - diff(A_i, X, H)/m + diff(A_i, X, M)/m$;
9.    end;
10. end;
```

Figure 2-3: Relief with selective sampling algorithm

- Relief with selective sampling (ReliefS)

  ReliefS was proposed on 2004 [13] as the improvement of Relief algorithm, which creates kd-tree upon each bucket of features and adds random sampling. The algorithm in Figure 2-3 shows that ReliefS used either sample size m or bucket size t to control the kd-tree splitting process. With N number of instances, $t = N/m$. Therefore, if m is predetermined, the splitting process stops when it reaches the level where each bucket contains N/m or fewer instances. [13] also shows that ReliefS in general achieves better performance than ReliefF in terms of learning accuracy and hence selective sampling is an effective approach for active feature selection.

- Sequential Feature Selection (SFS)

  Adds one feature which gives the highest value for the objective function, then the remaining features are added individually to the current subset and the new subset is evaluated. The individual feature is permanently included in the subset if it gives the maximum classification accuracy. The process is repeated until the required number of features are added. This is a naive SFS algorithm since the dependency between the features is not accounted for. The complexity of this algorithm is quadratic.

- Sequential Floating Feature Selection (SFFS)

  This algorithm was proposed as an improvement of SFS algorithm. This algorithm works similarly to SFS but in addition, it adds another step which excludes one feature at a time from the subset obtained in the first step and evaluates the new subsets. The complexity of the algorithm remains to be quadratic.

- Sequential Backward Selection (SBS)

  This algorithm works the same way as SFS algorithm, but it starts from a full set and removes a feature on each step. The process is repeated until the required number of features needed is achieved. The complexity of this algorithm is quadratic.

2.3.2 Wrapper Methods

Wrapper methods are broadly classified into sequential selection algorithms and heuristic search algorithms [4].

- Particle Swarm Optimization (PSO)

  This algorithm yields local optimum results are employed which can produce good results and are computationally feasible.

- Genetic Algorithm (GA)

  This algorithm [19] also yields local optimum results are employed which can produce good results and are computationally feasible.

2.3.3 Embedded Methods

- Boosting Based Hybrid Feature Selection (BBHFS)

2.4 Parallel Algorithms

2.5 Selected Algorithm

Chapter 3

Implementation

3.1 Feature selection implementation

3.2 Classification implementation

Chapter 4

Evaluation

4.1 Experiment Setting

4.2 Experiment Results

Chapter 5

Conclusion

5.1 Summary of Algorithms Survey

5.2 Future Works

Bibliography

[1] Varun Aggarwal and Sassoon Kosian. Feature selection and dimension reduction techniques in sas. *NorthEast SAS Users Group (NESUG)*, 2011.

[2] Mark A. Beyer and Douglas Laney. The importance of big data: A definition. Technical report, Gartner, 2012.

[3] Rich Caruana, Nikos Karampatziakis, and Ainur Yessenalina. An empirical evaluation of supervised learning in high dimensions. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 96–103, New York, NY, USA, 2008. ACM.

[4] Girish Chandrashekar and Ferat Sahin. A survey on feature selection methods. *Comput. Electr. Eng.*, 40(1):16–28, January 2014.

[5] Justin Doak. An Evaluation of Feature Selection Methods and Their Application to Computer Security. Technical report, UC Davis Department of Computer Science, 1992.

[6] Dragan Gamberger and Nada Lavrac. Conditions for occam's razor applicability and noise elimination. In *Proceedings of the 9th European Conference on Machine Learning*, ECML '97, pages 108–123, London, UK, UK, 1997. Springer-Verlag.

[7] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, March 2003.

[8] Mark A. Hall. Correlation-based feature selection for discrete and numeric class machine learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, pages 359–366, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[9] H. V. Jagadish, Johannes Gehrke, Alexandros Labrinidis, Yannis Papakonstantinou, Jignesh M. Patel, Raghu Ramakrishnan, and Cyrus Shahabi. Big data and its technical challenges. *Commun. ACM*, 57(7):86–94, July 2014.

[10] Kenji Kira and Larry A. Rendell. The feature selection problem: Traditional methods and a new algorithm. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, AAAI'92, pages 129–134. AAAI Press, 1992.

[11] Igor Kononenko. Estimating attributes: Analysis and extensions of relief. In *Proceedings of the European Conference on Machine Learning on Machine Learning*, ECML-94, pages 171–182, Secaucus, NJ, USA, 1994. Springer-Verlag New York, Inc.

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

[13] Huan Liu, Hiroshi Motoda, and Lei Yu. A selective sampling approach to active feature selection. *Artificial Intelligence*, 159(12):49 – 74, 2004.

[14] Huan Liu and Lei Yu. Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering*, 17(4):491–502, April 2005.

[15] P. M. Narendra and K. Fukunaga. A branch and bound algorithm for feature subset selection. *IEEE Transactions on Computers*, C-26(9):917–922, Sept 1977.

[16] Marko Robnik-Sikonja and Igor Kononenko. An adaptation of relief for attribute estimation in regression. In *Proceedings of the Fourteenth International Conference on Machine Learning*, ICML '97, pages 296–304, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.

[17] Sameer Singh, Jeremy Kubica, Scott Larsen, and Daria Sorokina. *Parallel Large Scale Feature Selection for Logistic Regression*, chapter 99, pages 1172–1183.

[18] Lei Xu, Pingfan Yan, and Tong Chang. Best first strategy for feature selection. In *Pattern Recognition, 1988., 9th International Conference on*, pages 706–708 vol.2, Nov 1988.

[19] J. Yang and V. Honavar. Feature subset selection using a genetic algorithm. *IEEE Intelligent Systems and their Applications*, 13(2):44–49, Mar 1998.

[20] Ce Zhang, Arun Kumar, and Christopher Ré. Materialization optimizations for feature selection workloads. *ACM Transactions on Database Systems*, 41(1):2:1–2:32, February 2016.