Toward Integrating Feature Selection Algorithms for Classification and Clustering

Huan Liu and Lei Yu

Department of Computer Science and Engineering

Arizona State University

Tempe, AZ 85287-8809

{hliu,leiyu}@asu.edu

Abstract

This paper introduces concepts and algorithms of feature selection, surveys existing feature selection algorithms for classification and clustering, groups and compares different algorithms with a categorizing framework based on search strategies, evaluation criteria, and data mining tasks, reveals unattempted combinations, and provides guidelines in selection of feature selection algorithms. With the categorizing framework, we continue our efforts toward building an integrated system for intelligent feature selection. A unifying platform is proposed as an intermediate step. An illustrative example is presented to show how existing feature selection algorithms can be integrated into a meta algorithm that can take advantage of individual algorithms. An added advantage of doing so is to help a user employ a suitable algorithm without knowing details of each algorithm. Some real-world applications are included to demonstrate the use of feature selection in data mining. We conclude this work by identifying trends and challenges of feature selection research and development.

Keywords: Feature Selection, Classification, Clustering, Categorizing Framework, Unifying Platform, Real-World Applications

1 Introduction

As computer and database technologies advance rapidly, data accumulates in a speed unmatchable by human's capacity of data processing. Data mining [1, 29, 35, 36], as a multidisciplinary joint effort from databases, machine learning, and statistics, is championing in turning mountains of data into nuggets. Researchers and practitioners realize that in order to use data mining tools effectively, data preprocessing is essential to successful data mining [53, 74]. Feature selection is one of the important and frequently used techniques in data preprocessing for data mining [6, 52]. It reduces the number of features, removes irrelevant, redundant, or noisy data, and brings the immediate effects for applications: speeding up a data mining algorithm, improving mining performance such as predictive accuracy and result comprehensibility. Feature selection has been a fertile field of research and development since 1970's in statistical pattern recognition [5, 40, 63, 81, 90], machine learning [6, 41, 43, 44], and data mining [17, 18, 42], and widely applied to many fields such as text categorization [50, 70, 94], image retrieval [77, 86], customer relationship management [69], intrusion detection [49], and genomic analysis [91].

Feature selection is a process that selects a subset of original features. The optimality of a feature subset is measured by an evaluation criterion. As the dimensionality of a domain expands, the number of features N increases. Finding an optimal feature subset is usually intractable [44] and many problems related to feature selection have been shown to be **NP**-hard [7]. A typical feature selection process consists of four basic steps (shown in Figure 1), namely, subset generation, subset evaluation, stopping criterion, and result validation [18]. Subset generation is a search procedure [48, 53] that produces candidate feature subsets for evaluation based on a certain *search strategy*. Each candidate subset is evaluated and compared with the previous best one according to a certain *evaluation criterion*. If the new subset turns out to be better, it replaces the previous best subset. The process of subset generation and evaluation is repeated until a given stopping criterion is satisfied. Then the selected best subset usually needs to be validated by prior knowledge or different tests via synthetic and/or real-world data sets. Feature selection can be found in many areas of data mining such as classification, clustering, association rules, regression. For example,

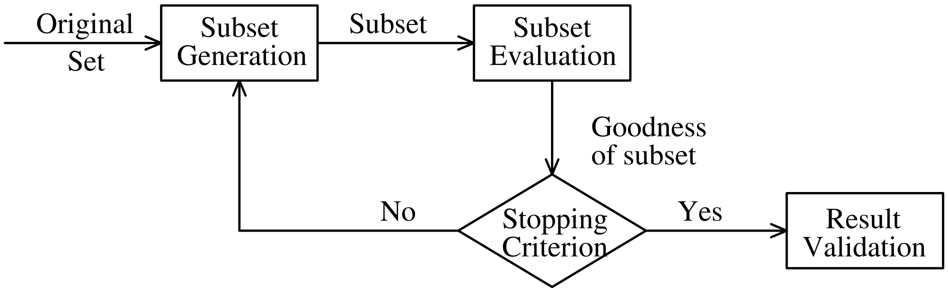

Figure 1: Four key steps of feature selection

feature selection is called subset or variable selection in Statistics [62]. A number of approaches to variable selection and coefficient shrinkage for regression are summarized in [37]. In this survey, we focus on feature selection algorithms for classification and clustering. Early research efforts mainly focus on feature selection for *classification* with labeled data [18, 25, 81] (supervised feature selection) where class information is available. Latest developments, however, show that the above general procedure can be well adopted to feature selection for *clustering* with unlabeled data [19, 22, 27, 87] (or unsupervised feature selection) where data is unlabeled.

Feature selection algorithms designed with different evaluation criteria broadly fall into three categories: the *filter* model [17, 34, 59, 95], the *wrapper* model [13, 27, 42, 44], and the *hybrid* model [15, 68, 91]. The filter model relies on general characteristics of the data to evaluate and select feature subsets without involving any mining algorithm. The wrapper model requires one predetermined mining algorithm and uses its performance as the evaluation criterion. It searches for features better suited to the mining algorithm aiming to improve mining performance, but it also tends to be more computationally expensive than the filter model [44, 48]. The hybrid model attempts to take advantage of the two models by exploiting their different evaluation criteria in different search stages.

This survey attempts to review the field of feature selection based on earlier works by Doak [25], Dash and Liu [18], and Blum and Langley [6]. The fast development of the field has produced many new feature selection methods. Novel research problems and applications emerge, and new demands for feature selection appear. In order to review the field and attempt for the next generation of feature selection methods, we aim to achieve the following objectives in this survey:

- introduce the basic notions, concepts, and procedures of feature selection,
- describe the state-of-the-art feature selection techniques,
- identify existing problems of feature selection and propose ways of solving them,
- demonstrate feature selection in real-world applications, and
- point out current trends and future directions.

This survey presents a collection of existing feature selection algorithms, and proposes a categorizing framework that systematically groups algorithms into categories and compares the commonalities and differences between the categories. It further addresses a problem springing from the very core of the success of this field - a dilemma faced by most data mining practitioners: the more feature selection algorithms available, the more difficult it is to select a suitable one for a data mining task. This survey, therefore, proposes a unifying platform that covers major factors in the selection of a suitable algorithm for an application, and paves the way for building an integrated system for intelligent feature selection.

The remainder of this paper is organized into five sections. Section 2 describes each step of the general feature selection process. Section 3 groups and compares different feature selection algorithms based on a three-dimensional categorizing framework. Section 4 introduces the development of a unifying platform and illustrates the idea of developing an integrated system for intelligent feature selection through a preliminary system. Section 5 demonstrates some real-world applications of feature selection in data mining. Section 6 concludes the survey with discussions on current trends and future directions.

2 General Procedure of Feature Selection

In this section, we explain in detail the four key steps as shown in Figure 1 of Section 1.

2.1 Subset generation

Subset generation is essentially a process of heuristic search, with each state in the search space specifying a candidate subset for evaluation. The nature of this process is determined by two basic issues. First, one must decide the search starting point (or points) which in turn influences the search direction. Search may start with an empty set and successively add features (i.e., *forward*), or start with a full set and successively remove features (i.e., *backward*), or start with both ends and add and remove features simultaneously (i.e., *bi-directional*). Search may also start with a randomly selected subset in order to avoid being trapped into local optima [25]. Second, one must decide a search strategy. For a data set with N features, there exist 2^N candidate subsets. This search space is exponentially prohibitive for exhaustive search with even a moderate N. Therefore, different strategies have been explored: complete, sequential, and random search.

Complete search

It guarantees to find the optimal result according to the evaluation criterion used. Exhaustive search is complete (i.e., no optimal subset is missed). However, search is complete does not necessarily means that it must be exhaustive. Different heuristic functions can be used to reduce the search space without jeopardizing the chances of finding the optimal result. Hence, although the order of the search space is $O(2^N)$, a smaller number of subsets are evaluated. Some examples are *branch and bound* [67], and *beam search* [25].

Sequential search

It 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* [53]. All these approaches add or remove features one at a time. Another alternative is to add (or remove) p features in one step and remove (or add) q features in the next step $(p > q)$ [25]. 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

It starts with a randomly selected subset and proceeds in two different ways. One is to follow sequential search, which injects randomness into the above classical sequential approaches. Examples are *random-start hill-climbing* and *simulated annealing* [25]. The other is to generate the next subset in a completely random manner (i.e., a current subset does not grow or shrink from any previous subset following a deterministic rule), also known as the *Las Vegas* algorithm [10]. For all these approaches, the use of randomness helps to escape local optima in the search space, and optimality of the selected subset depends on the resources available.

2.2 Subset evaluation

As we mentioned earlier, each newly generated subset needs to be evaluated by an evaluation criterion. The goodness of a subset is always determined by a certain criterion (i.e., an optimal subset selected using one criterion may not be optimal according to another criterion). Evaluation criteria can be broadly categorized into two groups based on their dependency on mining algorithms that will finally be applied on the selected feature subset. We discuss the two groups of evaluation criteria below.

2.2.1 Independent criteria

Typically, an independent criterion 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 popular independent criteria are distance measures, information measures, dependency measures, and consistency measures [3, 5, 34, 53].

Distance measures are also known as separability, divergence, or discrimination measures. For a two-class problem, a feature X is preferred to another feature Y if X induces a greater difference between the two-class conditional probabilities than Y, because we try to find the feature that can separate the two classes as far as possible. X and Y are indistinguishable if the difference is zero.

Information measures typically determine the information gain from a feature. The information gain from a feature X is defined as the difference between the prior uncertainty and expected posterior uncertainty using X. Feature X is preferred to feature Y if the information gain from X is greater than that from Y.

Dependency measures are also known as correlation measures or similarity measures. They measure the ability to predict the value of one variable from the value of another. In feature selection for classification, we look for how strongly a feature is associated with the class. A feature X is preferred to another feature Y if the association between feature X and class C is higher than the association between Y and C. In feature selection for clustering, the association between two random features measures the similarity between the two.

Consistency measures are characteristically different from the above measures because of their heavy reliance on the class information and the use of the Min-Features bias [3] in selecting a subset of features. These measures attempt to find a minimum number of features that separate classes as consistently as the full set of features can. An inconsistency is defined as two instances having the same feature values but different class labels.

2.2.2 Dependency criteria

A dependency criterion used in the wrapper model requires a predetermined mining algorithm in feature selection and uses the performance of the mining algorithm applied on the selected subset to determine which features are selected. It usually gives superior performance as it finds features better suited to the predetermined mining algorithm, but it also tends to be more computationally expensive, and may not be suitable for other mining algorithms [6]. For example, in a task of classification, predictive accuracy is widely used as the primary measure. It can be used as a dependent criterion for feature selection. As features are selected by the classifier that later on uses these selected features in predicting the class labels of unseen instances, accuracy is normally high, but it is computationally rather costly to estimate accuracy for every feature subset [41].

In a task of clustering, the wrapper model of feature selection tries to evaluate the goodness of a feature subset by the quality of the clusters resulted from applying the clustering algorithm on the selected subset. There exist a number of heuristic criteria for estimating the quality of clustering results, such as cluster compactness, scatter separability, and maximum likelihood. Recent work on developing dependent criteria in feature selection for clustering can been found in [20, 27, 42].

2.3 Stopping criteria

A stopping criterion determines when the feature selection process should stop. Some frequently used stopping criteria are: (a) the search completes; (b) some given bound is reached, where a bound can be a specified number (minimum number of features or maximum number of iterations); (c) subsequent addition (or deletion) of any feature does not produce a better subset; and (d) a sufficiently good subset is selected (e.g., a subset may be sufficiently good if its classification error rate is less than the allowable error rate for a given task).

2.4 Result validation

A straightforward way for result validation is to directly measure the result using prior knowledge about the data. If we know the relevant features beforehand as in the case of synthetic data, we can compare this known set of features with the selected features. Knowledge on the irrelevant or redundant features can also help. We do not expect them to be selected. In real-world applications, however, we usually do not have such prior knowledge. Hence, we have to rely on some indirect methods by monitoring the change of mining performance with the change of features. For example, if we use classification error rate as a performance indicator for a mining task, for a selected feature subset, we can simply conduct the "before-and-after" experiment to compare the error rate of the classifier learned on the full set of features and that learned on the selected subset [53, 89].

3 A Categorizing Framework for Feature Selection Algorithms

Given the key steps of feature selection, we now introduce a categorizing framework that groups many existing feature selection algorithms into distinct categories, and summarize individual algorithms based on this framework.

Table 1: Categorization of feature selection algorithms in a three-dimensional framework

			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					

3.1 A categorizing framework

There exists a vast body of available feature selection algorithms. In order to better understand the inner instrument of each algorithm and the commonalities and differences among them, we develop a three-dimensional categorizing framework (shown in Table 1) based on the previous discussions. We understand that search strategies and evaluation criteria are two dominating factors in designing a feature selection algorithm, so they are chosen as two dimensions in the framework. In Table 1, under **Search Strategies**, algorithms are categorized into *Complete*, *Sequential*, and *Random*. Under **Evaluation Criteria**, algorithms are categorized into *Filter*, *Wrapper*, and *Hybrid*. We consider **Data Mining Tasks** as a third dimension because the availability of class information in *Classification* or *Clustering* tasks affects evaluation criteria used in feature selection algorithms (as discussed in Section 2.2). In addition to these three basic dimensions, algorithms within the *Filter* category are further distinguished by specific evaluation criteria including *Distance*, *Information*, *Dependency*, and *Consistency*. Within the *Wrapper* category, *Predictive Accuracy* is used for *Classification*, and *Cluster Goodness* for *Clustering*.

Many feature selection algorithms collected in Table 1 can be grouped into distinct categories according to these characteristics. The categorizing framework serves three roles. First, it reveals relationships among different algorithms: algorithms in the same block (category) are most similar to each other (i.e., designed with similar search strategies and evaluation criteria, and for the same type of data mining tasks). Second, it enables us to focus our selection of feature selection algorithms for a given task on a relatively small number of algorithms out the whole body. For example, knowing that feature selection is performed for classification, predicative accuracy of a classifier is suitable to be the evaluation criterion, and complete search is not suitable for the limited time allowed, we can conveniently limit our choices to two groups of algorithms in Table 1: one is defined by *Classification*, *Wrapper*, and *Sequential*; the other is by *Classification*, *Wrapper*, and *Random*. Both groups have more than one algorithm available [1]. Third, the framework also reveals what are missing in the current collection of feature selection algorithms. As we can see, there are many empty blocks in Table 1, indicating that no feature selection algorithm exists for these combinations which might be suitable for potential future work. In particular, for example, current feature selection algorithms for clustering are only limited to sequential search.

With a large number of existing algorithms seen in the framework, we summarize all the algorithms into three generalized algorithms corresponding to the filter model, the wrapper model, and the hybrid model, respectively.

[1] Some other perspectives are necessary to further differentiate algorithms in each category. In-depth discussions on choosing a most suitable feature selection algorithm for a data mining problem is provided in Section 4.

3.2 Filter Algorithm

Algorithms within the filter model are illustrated through a generalized filter algorithm (shown in Table 2). For a given data set D, the algorithm starts the search from a given subset S_0 (an empty set, a full set, or any randomly selected subset) and searches through the feature space by a particular search strategy. Each generated subset S is evaluated by an independent measure M and compared with the previous best one. If it is found to be better, it is regarded as the current best subset. The search iterates until a predefined stopping criterion δ (as described in Section 2.3) is reached. The algorithm outputs the last current best subset S_{best} as the final result. By varying the search strategies and evaluation measures used in steps 5 and 6 in the algorithm, we can design different individual algorithms within the filter model. Since the filter model applies independent evaluation criteria without involving any mining algorithm, it does not inherit any bias of a mining algorithm and it is also computationally efficient.

Table 2: A generalized filter algorithm

```
Filter Algorithm
input:   $D(F_0, F_1, ..., F_{n-1})$  // a training data set with $N$ features
         $S_0$                        // a subset from which to start the search
         $\delta$                     // a stopping criterion
output:  $S_{best}$                   // an optimal subset
01 begin
02     initialize: $S_{best} = S_0$;
03     $\gamma_{best} = eval(S_0, D, M)$; // evaluate $S_0$ by an independent measure $M$
04     do begin
05         $S = generate(D)$; // generate a subset for evaluation
06         $\gamma = eval(S, D, M)$; // evaluate the current subset $S$ by $M$
07         if ($\gamma$ is better than $\gamma_{best}$)
08             $\gamma_{best} = \gamma$;
09             $S_{best} = S$;
10     end until ($\delta$ is reached);
11     return $S_{best}$;
12 end;
```

3.3 Wrapper Algorithm

A generalized wrapper algorithm (shown in Table 3) is very similar to the generalized filter algorithm except that it utilizes a predefined mining algorithm A instead of an independent measure M for subset evaluation. For each generated subset S, it evaluates its goodness by applying the mining algorithm to the data with feature subset S and evaluating the quality of mined results. Therefore, different mining algorithms will produce different feature selection results. Varying the search strategies via the function *generate(D)* and mining algorithms (A) can result in different wrapper algorithms. Since mining algorithms are used to control the selection of feature subsets, the wrapper model tends to give superior performance as feature subsets found are better suited to the predetermined mining algorithm. Consequently, it is also more computationally expensive than the filter model.

Table 3: A generalized wrapper algorithm

```
Wrapper Algorithm
input:    $D(F_0, F_1, ..., F_{n-1})$  // a training data set with $N$ features
          $S_0$                        // a subset from which to start the search
          $\delta$                     // a stopping criterion
output:   $S_{best}$                   // an optimal subset
begin
    initialize: $S_{best} = S_0$;
    $\gamma_{best} = eval(S_0, D, A)$; // evaluate $S_0$ by a mining algorithm $A$
    do begin
        $S = generate(D)$; // generate a subset for evaluation
        $\gamma = eval(S, D, A)$; // evaluate the current subset $S$ by $A$
        if ($\gamma$ is better than $\gamma_{best}$)
            $\gamma_{best} = \gamma$;
            $S_{best} = S$;
    end until ($\delta$ is reached);
    return $S_{best}$;
end;
```

3.4 Hybrid Algorithm

To take advantage of the above two models and avoid the pre-specification of a stopping criterion, the hybrid model is recently proposed to handle large data sets [15, 91]. A typical hybrid algorithm

Table 4: A generalized hybrid algorithm

```
Hybrid Algorithm
input:   $D(F_0, F_1, ..., F_{n-1})$   // a training data set with $N$ features
         $S_0$                         // a subset from which to start the search
output:  $S_{best}$                    // an optimal subset
begin
    initialize: $S_{best} = S_0$;
    $c_0 = card(S_0)$; // calculate the cardinality of $S_0$
    $\gamma_{best} = eval(S_0, D, M)$; // evaluate $S_0$ by an independent measure $M$
    $\theta_{best} = eval(S_0, D, A)$; // evaluate $S_0$ by a mining algorithm $A$
    for $c = c_0 + 1$ to $N$ begin
        for $i = 0$ to $N - c$ begin
            $S = S_{best} \cup \{F_j\}$; // generate a subset with cardinality $c$ for evaluation
            $\gamma = eval(S, D, M)$; // evaluate the current subset $S$ by $M$
            if ($\gamma$ is better than $\gamma_{best}$)
                $\gamma_{best} = \gamma$;
                $S'_{best} = S$;
        end;
        $\theta = eval(S'_{best}, D, A)$; // evaluate $S'_{best}$ by $A$
        if ($\theta$ is better than $\theta_{best}$);
            $S_{best} = S'_{best}$;
            $\theta_{best} = \theta$;
        else;
            break and return $S_{best}$;
    end;
    return $S_{best}$;
end;
```

(shown in Table 4) makes use of both an independent measure and a mining algorithm to evaluate feature subsets: it uses the independent measure to decide the best subsets for a given cardinality and uses the mining algorithm to select the final best subset among the best subsets across different cardinalities. Basically, it starts the search from a given subset S_0 (usually an empty set in sequential forward selection) and iterates to find the best subsets at each increasing cardinality. In each round for a best subset with cardinality c, it searches through all possible subsets of cardinality $c + 1$ by adding one feature from the remaining features. Each newly generated subset S with cardinality $c + 1$ is evaluated by an independent measure M and compared with the previous best one. If S is better, it becomes the current best subset S'_{best} at level $c + 1$. At the end of each iteration, a mining algorithm A is applied on S'_{best} at level $c + 1$ and the quality of the mined result θ is compared with that from the best subset at level c. If S'_{best} is better, the algorithm continues to find the best subset at the next level; otherwise, it stops and outputs the current best subset as the final best subset. The quality of results from a mining algorithm provides a natural stopping criterion in the hybrid model.

4 Toward an Integrated System for Intelligent Feature Selection

Research on feature selection has been active for decades with attempts to improve well-known algorithms or to develop new ones. The proliferation of feature selection algorithms, however, has not brought about a general methodology that allows for intelligent selection from existing algorithms. In order to make a "right" choice, a user not only needs to know the domain well (this is usually not a problem for the user), but also is expected to understand technical details of available algorithms (discussed in previous sections). Therefore, the more algorithms available, the more challenging it is to choose a suitable one for an application. Consequently, a big number of algorithms are not even attempted in practice and only a couple of algorithms are always used. Therefore, there is a pressing need for intelligent feature selection that can automatically recommend the most suitable algorithm among many for a given application. In this section, we present an integrated approach to intelligent feature selection. First, we introduce a unifying platform which serves an intermediate step toward building an integrated system for intelligent feature selection. Second, we illustrate the idea through a preliminary system based on our research.

4.1 A unifying platform

In Section 3.1, we develop a categorizing framework based on three dimensions (search strategies, evaluation criteria, and data mining tasks) from an algorithm designer's perspective. However, it would be impractical to require a domain expert or a user to keep abreast of such technical details about feature selection algorithms. Moreover, in most cases, it is not sufficient to decide the most

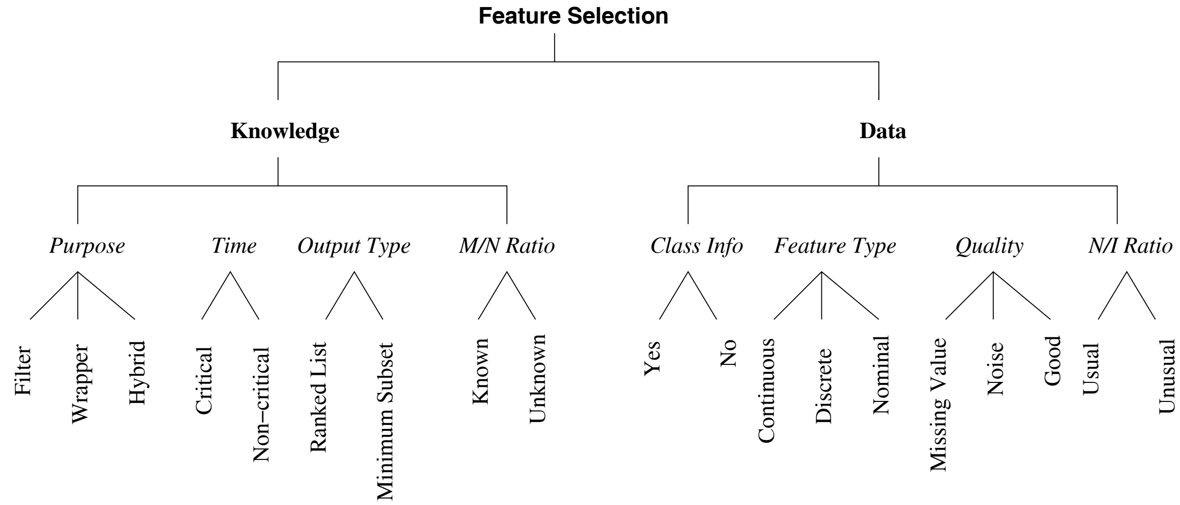

Figure 2: A unifying platform

suitable algorithm based merely on this framework. Recall the two groups of algorithms identified by the three dimensions in Section 3.1, each group still contains quite a few candidate algorithms. Assuming that we only have three wrapper algorithms: WSFG and WSBG in one group and LVW in the other group, additional information is required to decide the most suitable one for the given task. We propose a unifying platform (shown in Figure 2) that expands the categorizing framework by introducing more dimensions from a user's perspective.

At the top, knowledge and data about feature selection are two key determining factors. Currently the knowledge factor covers *Purpose* of feature selection, *Time* concern, expected *Output Type*, and M/N *Ratio* - the ratio between the expected number of selected features M and the total number of original features N. The data factor covers *Class Information*, *Feature Type*, *Quality* of data, and N/I *Ratio* - the ratio between the number of features N and the number of instances I. Each dimension is discussed below.

The *purpose* of feature selection can be broadly categorized into visualization, data understanding, data cleaning, redundancy and/or irrelevancy removal, and performance (e.g., predictive accuracy and comprehensibility) enhancement. Recall that feature selection algorithms are categorized into the filter model, the wrapper model, and the hybrid model. Accordingly, we can also summarize different purposes of feature selection into these three categories to form a generic task hierarchy, as different purposes imply different evaluation criteria and thus guide the selection of feature selection algorithms differently. For general purpose of redundancy and/or irrelevancy removal, algorithms in the filter model are good choices as they are unbiased and fast. To enhance the mining performance, algorithms in the wrapper model should be preferred than those in the filter model as they are better suited to the mining algorithms[44, 48]. Sometimes, algorithms in the hybrid model are needed to serve more complicated purposes.

The *time* concern is about whether the feature selection process is time critical or not. Different time constraints affect the selection of algorithms with different search strategies. When time is not a critical issue, algorithms with complete search are recommended to achieve higher optimality of results; otherwise, algorithms with sequential search or random search should be selected for fast results. Time constraints can also affect the choice of feature selection models as different models have different computational complexities. The filter model is preferred in applications where applying mining algorithms are too costly, or unnecessary.

The *output* type of feature selection can sometimes be known *a priori*. This aspect divides feature selection algorithms into two groups: ranked list and minimum subset. The real difference between the two is about the order among the selected features. There is no order among the features in a selected subset. One cannot easily remove any more feature from the subset, but one can do so for a ranked list by removing the least important one. Back to the previous example, among WSFG, WSBG and LVW, if we expect to get a ranked list as the result, LVW returning a minimum subset will be eliminated from the final choice.

The M/N *ratio* is also very useful in determining a proper search strategy. If the number of relevant features (M) is expected to be small, a forward complete search strategy can be afforded; if the number of irrelevant features ($N - M$) is small, a backward complete search strategy can be adopted even in time critical situations. If we have the prior knowledge that the number of irrelevant features is significantly larger than the number of relevant ones, WSFG using sequential forward search is considered a better choice than WSBG using sequential backward search.

Within the data factor, the *class* information is about whether the data contains class information or not. With class information, feature selection algorithms for classification are needed, while without class information, feature selection algorithms for clustering are needed. This dimension is essentially the same as the dimension of data mining tasks in the categorizing framework, but it reflects a user's knowledge about the data.

Different *feature types* require different data processing mechanisms. Some types of features inherit order in their values such as continuous and discrete features; some do not inherit order such as nominal features. When different feature types occur in one data set, things become more complicated: how to consider each feature's influence. Mixed data types imply that the range of values for each feature can vary significantly. It is important to recognize and allow this complexity for real-world applications in the selection of feature selection algorithms.

The *quality* of data is about whether data contains missing values or noisy data. Different feature selection algorithms require different levels of data quality to perform well. Some applications require more preprocessing such as value discretization [28, 51] and missing value treatment, while others are less stringent in this regard.

The N/I *ratio* recently becomes an interesting problem when feature selection is applied to text mining [50, 70] and genomic analysis [91]. Usually, the number of total instances I is greatly larger than the number of total features N. Sometimes, however, N could be very huge, but I small as in text mining and gene expression microarray analysis. In such cases, we should focus on algorithms that intensively work along the I dimensions (More is discussed in Section 5).

In addition to the eight dimensions in the unifying platform, domain knowledge, when available, should also be used to aid feature selection. For example, a medical doctor may know that for a certain type of patient data, some features are more indicative than others, and some may be irrelevant. The flexibility of using domain knowledge is not always required or possible, especially in data mining applications where we usually wish to let data tell us hidden patterns. When domain knowledge is available, using it will certainly speed up the feature selection process.

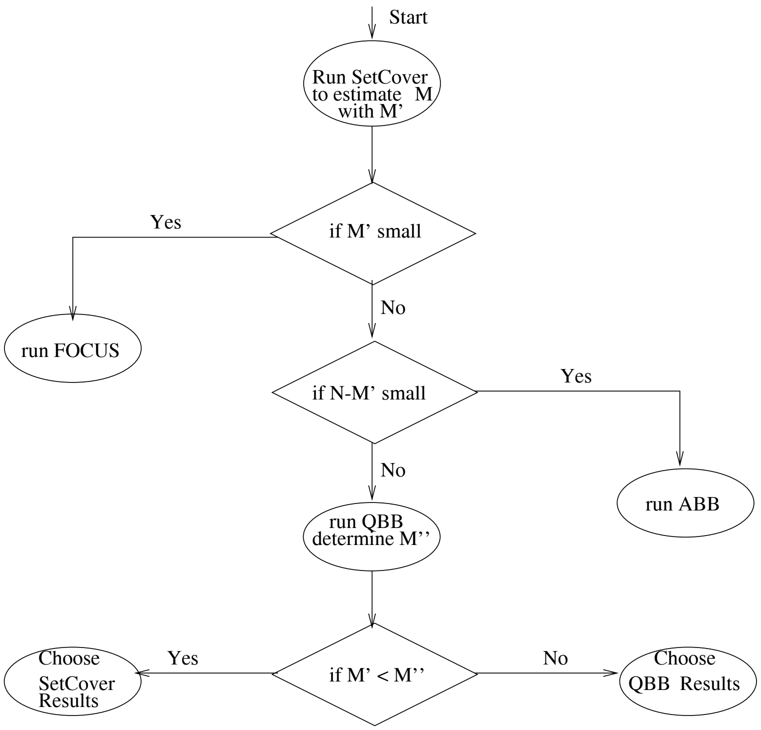

Figure 3: A preliminary integrated system

4.2 Toward an integrated system

The unifying platform serves two purposes: (1) to group existing algorithms with similar characteristics and investigate their strengths and weaknesses on the same platform, and (2) to provide a guideline in building an integrated system for intelligent feature selection. We introduce a preliminary integrated system (shown in Figure 3) by employing the information on the M/N ratio.

We focus on feature selection algorithms using the consistency evaluation criterion. The four representative algorithms employ different search strategies: Focus - forward exhaustive search, ABB - backward complete search, QBB - random search plus ABB, and SetCover - sequential search. Both theoretical analysis and experimental results suggest that each algorithm has its own strengthes and weaknesses concerning speed and optimality of results [21]. To guide the selection of a suitable algorithm among the four, the number of relevant features M is estimated as M' or M'' (shown in Figure 3), where M' is an estimate of M by SetCover, and M'' an estimate of M by QBB. With this system, Focus or ABB is recommended when either M' or $N - M'$ is small, because they guarantee the optimal selected subset. However, the two could take impractically long time to converge when the two conditions are not true. Therefore, either SetCover or QBB will be used based on the comparison of M' and M''. The two algorithms do not guarantee optimal subsets, but they are efficient in generating near optimal subsets.

The example in Figure 3 verifies the idea of automatically choosing a suitable feature selection algorithm within a limited scope based on the unifying platform. All four algorithms share the following characteristics: using an independent evaluation criterion (i.e., the filter model), searching for a minimum feature subset, time critical, and dealing with labelled data. The preliminary integrated system uses the M/N ratio to guide the selection of feature selection algorithms. How to substantially extend this preliminary work to a fully integrated system that incorporates all the factors specified in the unifying platform remains a challenging problem.

After presenting the concepts and state-of-the-art algorithms with a categorizing framework and a unifying platform, we now examine the use of feature selection in real-world data mining applications. Feature selection has found many successes in real-world applications.

5 Real-World Applications of Feature Selection

The essence of these successful applications lies at the recognition of a need for effective data preprocessing: data mining can be effectively accomplished with the aid of feature selection. Data is often collected for many reasons other than data mining (e.g., required by law, easy to collect, or simply for the purpose of book-keeping). In real-world applications, one often encounters problems such as too many features, individual features unable to independently capture significant characteristics of data, high dependency among the individual features, and emergent behaviors of combined features. Humans are ineffective at formulating and understanding hypotheses when data sets have large numbers of variables (possibly thousands in cases involving demographics and hundreds of thousands in cases involving Web browsing, microarray data analysis, or text document analysis), and people would find it easy to understand aspects of the problem in lowerdimensional subspaces [30, 72]. Feature selection can reduce the dimensionality to enable many data mining algorithms to work effectively on data with large dimensionality. Some illustrative applications of feature selection are showcased here.

Text categorization

Text categorization [50, 70] is the problem of automatically assigning predefined categories to free text documents. This problem is of great practical importance given the massive volume of online text available through the World Wide Web, Emails, and digital libraries. A major characteristic, or difficulty of text categorization problems is the high dimensionality of the feature space. The original feature space consists of many unique terms (words or phrases) that occur in documents, and the number of terms can be hundreds of thousands for even a moderate-sized text collection. This is prohibitively high for many mining algorithms. Therefore, it is highly desirable to reduce the original feature space without sacrificing categorization accuracy. In [94], different feature selection methods are evaluated and compared in the reduction of a high dimensional feature space in text categorization problems. It is reported that the methods under evaluation can effectively remove 50% - 90% of the terms while maintaining the categorization accuracy.

Image retrieval

Feature selection is applied in [86] to content-based image retrieval. Recent years have seen a rapid increase of the size and amount of image collections from both civilian and military equipments. However, we cannot access to or make use of the information unless it is organized so as to allow efficient browsing, searching and retrieving. Content-based image retrieval [77] is proposed to effectively handle the large scale of image collections. Instead of being manually annotated by text-based keywords, images would be indexed by their own visual contents (features), such as color, texture, shape, etc. One of the biggest problems to make content-based image retrieval truly scalable to large size image collections is still the "curse of dimensionality" [37]. As suggested in [77], the dimensionality of the feature space is normally of the order of 10^2. Dimensionality reduction is a promising approach to solve this problem. The image retrieval system proposed in [86] uses the theories of optimal projection to achieve optimal feature selection. Relevant features are then used to index images for efficient retrieval.

Customer relationship management

A case of feature selection is presented in [69] for customer relationship management. In the context that each customer means a big revenue and the loss of one will likely trigger a significant segment to defect, it is imperative to have a team of highly experienced experts monitor each customer's intention and movement based on massively collected data. A set of key indicators are used by the team and proven useful in predicting potential defectors. The problem is that it is difficult to find new indicators describing the dynamically changing business environment among many possible indicators (features). The machine recorded data is simply too enormous for any human expert to browse and obtain any insight from it. Feature selection is employed to search for new potential indicators in a dynamically changing environment. They are later presented to experts for scrutiny and adoption. This approach considerably improves the team's efficiency in finding new changing indicators.

Intrusion detection

As network-based computer systems play increasingly vital roles in modern society, they have become the targets of our enemies and criminals. The security of a computer system is compromised when an intrusion takes place. Intrusion detection is often used as one way to protect computer systems. In [49], Lee, Stolfo, and Mok proposed a systematic data mining framework for analyzing audit data and constructing intrusion detection models. Under this framework, a large amount of audit data is first analyzed using data mining algorithms in order to obtain the frequent activity patterns. These patterns are then used to guide the selection of system features as well as the construction of additional temporal and statistical features for another phase of automated learning. Classifiers based on these selected features are then inductively learned using the appropriately formatted audit data. These classifiers can be used as intrusion detection models since they can classify whether an observed system activity is "legitimate" or "intrusive". Feature selection plays an important role in building classification models for intrusion detection.

Genomic analysis

Structural and functional data from analysis of the human genome has increased many folds in recent years, presenting enormous opportunities and challenges for data mining [91, 96]. In particular, gene expression microarray is a rapidly maturing technology that provides the opportunity to assay the expression levels of thousands or tens of thousands of genes in a single experiment. These assays provide the input to a wide variety of data mining tasks, including classification and clustering. However, the number of instances in these experiments is often severely limited. In [91], for example, a case involving only 38 training data points in a 7130 dimensional space is used to exemplify the above situation that are becoming increasingly common in application of data mining to molecular biology. In this extreme of very few observations on a large number of features, Xing, Jordan, and Karp investigated the possible use of feature selection on a microarray classification problem. All the classifiers tested in the experiments performed significantly better in the reduced feature space than in the full feature space.

6 Concluding Remarks and Future Directions

This survey provides a comprehensive overview of various aspects of feature selection. We introduce two architectures - a categorizing framework and a unifying platform. They categorize the large body of feature selection algorithms, reveal future directions for developing new algorithms, and guide the selection of algorithms for intelligent feature selection. The categorizing framework is developed from an algorithm designer's viewpoint that focuses on the technical details about the general procedures of feature selection process. A new feature selection algorithm can be incorporated into the framework according to the three dimensions. The unifying platform is developed from a user's viewpoint that covers the user's knowledge about the domain and data for feature selection. The unifying platform is one necessary step toward building an integrated system for intelligent feature selection. The ultimate goal for intelligent feature selection is to create an integrated system that will automatically recommend the most suitable algorithm(s) to the user while hiding all technical details irrelevant to an application.

As data mining develops and expands to new application areas, feature selection also faces new challenges. We represent here some challenges in research and development of feature selection.

Feature selection with large dimensionality

Classically, the dimensionality N is considered large if it is in the range of hundreds. However, in some recent applications of feature selection, the dimensionality can be tens or hundreds of thousands. Such high dimensionality causes two major problems for feature selection. One is the so-called "curse of dimensionality" [37]. As most existing feature selection algorithms have quadratic or higher time complexity about N, it is difficult to scale up with high dimensionality. Since algorithms in the filter model use evaluation criteria that are less computationally expensive than those of the wrapper model, the filter model is often preferred to the wrapper model in dealing with large dimensionality. Recently, algorithms of the hybrid model [15, 91] are considered to handle data sets with high dimensionality. These algorithms focus on combining filter and wrapper algorithms to achieve best possible performance with a particular mining algorithm with similar time complexity of filter algorithms. Therefore, more efficient search strategies and evaluation criteria are needed for feature selection with large dimensionality. An efficient correlation-based filter algorithm is introduced in [95] to effectively handle large-dimensional data with class information. Another difficulty faced by feature selection with data of large dimensionality is the relative shortage of instances. That is, the dimensionality N can sometimes greatly exceed the number of instances I. In such cases, we should consider algorithms that intensively work along the I dimension as seen in [91].

Feature selection with active instance selection

Traditional feature selection algorithms perform dimensionality reduction using whatever training data is given to them. When the training data set is very large, random sampling [14, 33] is commonly used to sample a subset of instances. However, random sampling is blind without exploiting any data characteristic. The concept of active feature selection is first introduced and studied in [57]. Active feature selection promotes the idea to actively select instances for feature selection. It avoids pure random sampling and is realized by selective sampling [57, 60] that takes advantage of data characteristics when selecting instances. The key idea of selective sampling is to select only those instances with high probabilities to be informative in determining feature relevance. Selective sampling aims to achieve better or equally good feature selection results with a significantly smaller number of instances than that of random sampling. Although some selective sampling methods based on data variance or class information have proven to be effective on representative algorithms [57, 60], more research efforts are needed to investigate the effectiveness of selective sampling over the vast body of feature selection algorithms.

Feature selection with new data types

The field of feature selection develops fast as data mining is an application-driven field where research questions tend to be motivated by real-world data sets. A broad spectrum of formalisms and techniques has been proposed in a large number of applications. For example, the work of feature selection mainly focused on labelled data until 1997. Since 1998, we have observed the increasing use of feature selection for unlabelled data. The best-known data type in traditional data analysis, data mining, and feature selection is N-dimensional vectors of measurements on I instances (or objects, individuals). Such data is often referred to as multivariate data and can be thought of as an $N \times I$ data matrix [84]. Since data mining emerged, a common form of data in data mining in many business contexts is records of individuals conducting transactions in applications like market basket analysis, insurance, direct-mail marketing, and health care. This type of data, if it is considered as an $N \times I$ matrix, has a very large number of possible attributes but is a very sparse matrix. For example, a typical market basket (an instance) can have tens of items purchased out of hundreds of thousands of available items. The significant and rapid growth of computer and Internet/Web techniques makes some other types of data more commonly available - text-based data (e.g., emails, on-line news, newsgroups) and semi-structure data (e.g., HTML, XML). The wide deployment of various sensors, surveillance cameras, and Internet/Web monitoring lately poses a challenge to deal with another type of data - data streams. It pertains to data arriving over time, in a nearly continuous fashion, and is often available only once or in a limited amount of time [84]. As we have witnessed a growing number of work of feature selection on unlabelled data, we are certain to anticipate more research and development on new types of data for feature selection. It does not seem reasonable to suggest that the existing algorithms can be easily modified for these new data types.

Related challenges for feature selection

Shown in Section 5 are some exemplary cases of applying feature selection as a preprocessing step in very large databases collected from Internet, business, scientific, and government applications. Novel feature selection applications will be found where creative data reduction has to be conducted when our ability to capture and store data has far outpaced our ability to process and utilize it [30]. Feature selection can help focus on relevant parts of data and improve our ability to process data. New data mining applications [4, 45] arise as techniques evolve. Scaling data mining algorithms to large databases is a pressing issue. As feature selection is one step in data preprocessing, changes need to be made for classic algorithms that require multiple database scans and/or random access to data. Research is required to overcome limitations imposed when it is costly to visit large data sets multiple times or access instances at random as in data streams [9].

It is noticed recently that in the context of clustering, many clusters may reside in different subspaces of very small dimensionality [32], either with their sets of dimensions overlapped or non-overlapped. Many subspace clustering algorithms are developed [72]. Searching for subspaces is not exactly a feature selection problem as it tries to find many subspaces while feature selection only tries to find one subspace. Feature selection can also be extended to instance selection [55] in scaling down data which is a sister issue of scaling up algorithms. In addition to sampling methods [33], a suite of methods have been developed to search for representative instances so that data mining is performed in a focused and direct way [11, 61, 76]. Feature selection is a dynamic field closely connected to data mining and other data processing techniques. This paper attempts to survey this fast developing field, show some effective applications, and point out interesting trends and challenges. It is hoped that further and speedy development of feature selection can work with other related techniques to help evolve data mining into solutions for insights.

Acknowledgments

We are very grateful to the anonymous reviewers and editor. Their many helpful and constructive comments and suggestions helped us significantly improve this work. This work is in part supported by a grant from NSF (No. 0127815).

References

[1] R. Agrawal, T. Imielinski, and A. Swami. Database mining: A performance perspective. *IEEE Transactions on Knowledge and Data Engineering*, 5(6):914–925, 1993.

[2] H. Almuallim and T.G. Dietterich. Learning with many irrelevant features. In *Proceedings of the Ninth National Conference on Artificial Intelligence*, pages 547–552, 1991.

[3] H. Almuallim and T.G. Dietterich. Learning boolean concepts in the presence of many irrelevant features. *Artificial Intelligence*, 69(1-2):279–305, 1994.

[4] C. Apte, B. Liu, P.D. Pendault, and P. Smyth. Business applications of data mining. *Communications of the Association for Computing Machinery*, 45(8):49 – 53, 2002.

[5] M. Ben-Bassat. Pattern recognition and reduction of dimensionality. In P. R. Krishnaiah and L. N. Kanal, editors, *Handbook of statistics-II*, pages 773–791. North Holland, 1982.

[6] A.L. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97:245–271, 1997.

[7] A.L. Blum and R.L. Rivest. Training a 3-node neural networks is NP-complete. *Neural Networks*, 5:117 – 127, 1992.

[8] L. Bobrowski. Feature selection based on some homogeneity coefficient. In *Proceedings of the Ninth International Conference on Pattern Recognition*, pages 544–546, 1988.

[9] P Bradley, J. Gehrke, R. Ramakrishna, and R. Ssrikant. Scaling mining algorithms to large datbases. *Communications of the Association for Computing Machinery*, 45(8):38 – 43, 2002.

[10] G. Brassard and P. Bratley. *Fundamentals of Algorithms*. Prentice Hall, New Jersey, 1996.

[11] H. Brighton and C. Mellish. Advances in instance selection for instance-based learning algorithms. *Data Mining and Knowledge Discovery*, 6(2):153 – 172, 2002.

[12] C. Cardie. Using decision trees to improve case-based learning. In P. Utgoff, editor, *Proceedings of the Tenth International Conference on Machine Learning*, pages 25–32, 1993.

[13] R. Caruana and D. Freitag. Greedy attribute selection. In *Proceedings of the Eleventh International Conference on Machine Learning*, pages 28–36, 1994.

[14] W.G. Cochran. *Sampling Techniques*. John Wiley & Sons, 1977.

[15] S. Das. Filters, wrappers and a boosting-based hybrid for feature selection. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 74–81, 2001.

[16] M. Dash. Feature selection via set cover. In *Proceedings of IEEE Knowledge and Data Engineering Exchange Workshop*, pages 165–171, 1997.

[17] M. Dash, K. Choi, P. Scheuermann, and H. Liu. Feature selection for clustering – a filter solution. In *Proceedings of the Second International Conference on Data Mining*, pages 115–122, 2002.

[18] M. Dash and H. Liu. Feature selection for classification. *Intelligent Data Analysis: An International Journal*, 1(3):131–156, 1997.

[19] M. Dash and H. Liu. Handling large unsupervised data via dimensionality reduction. In *Proceedings of 1999 SIGMOD Research Issues in Data Mining and Knowledge Discovery (DMKD-99) Workshop*, 1999.

[20] M. Dash and H. Liu. Feature selection for clustering. In *Proceedings of the Fourth Pacific Asia Conference on Knowledge Discovery and Data Mining, (PAKDD-2000)*, pages 110–121, 2000.

[21] M. Dash, H. Liu, and H. Motoda. Consistency based feature selection. In *Proceedings of the Fourth Pacific Asia Conference on Knowledge Discovery and Data Mining, (PAKDD-2000)*, pages 98–109, 2000.

[22] M. Dash, H. Liu, and J. Yao. Dimensionality reduction of unsupervised data. In *Proceedings of the Ninth IEEE International Conference on Tools with AI (ICTAI'97)*, pages 532–539, 1997.

[23] M. Devaney and A. Ram. Efficient feature selection in conceptual clustering. In *Proceedings of the Fourteenth International Conference on Machine Learning*, pages 92–97, 1997.

[24] P.A. Devijver and J. Kittler. *Pattern Recognition: A Statistical Approach.* Prentice Hall International, 1982.

[25] J. Doak. An evaluation of feature selection methods and their application to computer security. Technical report, Davis CA: University of California, Department of Computer Science, 1992.

[26] P. Domingos. Why does bagging work? a bayesian account and its implications. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, pages 155 – 158, 1997.

[27] J. G. Dy and C. E. Brodley. Feature subset selection and order identification for unsupervised learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 247–254, 2000.

[28] U.M. Fayyad and K.B. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, pages 1022–1027, 1993.

[29] U.M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From data mining to knowledge discovery: An overview. In U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, pages 495–515. AAAI Press / The MIT Press, 1996.

[30] U.M. Fayyad and R. Uthurusamy. Evolving data mining into solutions for insights. *Communications of the Association for Computing Machinery*, 45(8):28 – 31, 2002.

[31] I. Foroutan and J. Sklansky. Feature selection for automatic classification of non-gaussian data. *IEEE Transactions on Systems, Man, and Cybernatics*, SMC-17(2):187–198, 1987.

[32] J. H. Friedman and J. J. Meulman. Clustering objects on subsets of attributes. http://citeseer.ist.psu.edu/friedman02clustering.html, 2002.

[33] B. Gu, F. Hu, and H. Liu. *Sampling: Knowing Whole from Its Part*, pages 21 – 38. In Liu and Motoda [55], 2001.

[34] M.A. Hall. Correlation-based feature selection for discrete and numeric class machine learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 359–366, 2000.

[35] J. Han and Y. Fu. Attribute-oriented induction in data mining. In U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, pages 399–421. AAAI Press / The MIT Press, 1996.

[36] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufman, 2001.

[37] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2001.

[38] M. Ichino and J. Sklansky. Feature selection for linear classifier. In *Proceedings of the Seventh International Confernce on Pattern Recognition*, pages 124–127, 1984.

[39] M. Ichino and J. Sklansky. Optimum feature selection by zero-one programming. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-14(5):737–746, 1984.

[40] A. Jain and D. Zongker. Feature selection: Evaluation, application, and small sample performance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2):153–158, 1997.

[41] G.H. John, R. Kohavi, and K. Pfleger. Irrelevant feature and the subset selection problem. In *Proceedings of the Eleventh International Conference on Machine Learning*, pages 121–129, 1994.

[42] Y. Kim, W. Street, and F. Menczer. Feature selection for unsupervised learning via evolutionary search. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 365–369, 2000.

[43] K. Kira and L.A. Rendell. The feature selection problem: Traditional methods and a new algorithm. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, pages 129–134, 1992.

[44] R. Kohavi and G.H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.

[45] R. Kohavi, N.J. Rothleder, and E. Simoudis. Emerging trends in business analytics. *Communications of the Association for Computing Machinery*, 45(8):45 – 48, 2002.

[46] D. Koller and M. Sahami. Toward optimal feature selection. In *Proceedings of the Thirteenth International Conference on Machine Learning*, pages 284–292, 1996.

[47] I. Kononenko. Estimating attributes : Analysis and extension of RELIEF. In *Proceedings of the Sixth European Conference on Machine Learning*, pages 171–182, 1994.

[48] P. Langley. Selection of relevant features in machine learning. In *Proceedings of the AAAI Fall Symposium on Relevance*, pages 140–144, 1994.

[49] W. Lee, S. J. Stolfo, and K. W. Mok. Adaptive intrusion detection: A data mining approach. *AI Review*, 14(6):533 – 567, 2000.

[50] E. Leopold and Kindermann J. Text categorization with support vector machines. how to represent texts in input space? *Machine Learning*, 46:423–444, 2002.

[51] H. Liu, F. Hussain, C.L. Tan, and M. Dash. Discretization: An enabling technique. *Data Mining and Knowledge Discovery*, 6(4):393–423, 2002.

[52] H. Liu and H. Motoda, editors. *Feature Extraction, Construction and Selection: A Data Mining Perspective*. Boston: Kluwer Academic Publishers, 1998. 2nd Printing, 2001.

[53] H. Liu and H. Motoda. *Feature Selection for Knowledge Discovery and Data Mining*. Boston: Kluwer Academic Publishers, 1998.

[54] H. Liu and H. Motoda. *Less Is More*, chapter 1, pages 3 – 12. In [52], 1998. 2nd Printing, 2001.

[55] H. Liu and H. Motoda, editors. *Instance Selection and Construction for Data Mining*. Boston: Kluwer Academic Publishers, 2001.

[56] H. Liu, H. Motoda, and M. Dash. A monotonic measure for optmial feature selection. In *Proceedings of the Tenth European Conference on Machine Learning*, pages 101–106, 1998.

[57] H. Liu, H. Motoda, and L. Yu. Feature selection with selective sampling. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pages 395–402, 2002.

[58] H. Liu and R. Setiono. Feature selection and classification - a probabilistic wrapper approach. In T. Tanaka, S. Ohsuga, and M. Ali, editors, *Proceedings of the Ninth International Conference on Industrial and Engineering Applications of AI and ES*, pages 419–424, 1996.

[59] H. Liu and R. Setiono. A probabilistic approach to feature selection - a filter solution. In *Proceedings of the Thirteenth International Conference on Machine Learning*, pages 319–327, 1996.

[60] H. Liu, L. Yu, M. Dash, and H. Motoda. Active feature selection using classes. In *Proceedings of the Seventh Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 474–485, 2003.

[61] D. Madigan, N. Raghavan, W. DuMouchel, C. Nason, M. Posse, and G. Ridgeway. Likelihood-based data squashing: A modeling approach to instance construction. *Data Mining and Knowledge Discovery*, 6(2):173 – 190, 2002.

[62] A. Miller. *Subset Selection in Regression*. Chapman & Hall/CRC, 2 edition, 2002.

[63] P. Mitra, C. A. Murthy, and S. K. Pal. Unsupervised feature selection using feature similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):301–312, 2002.

[64] M. Modrzejewski. Feature selection using rough sets theory. In P.B. Brazdil, editor, *Proceedings of the European Conference on Machine Learning*, pages 213–226, 1993.

[65] A. W. Moore and M. S. Lee. Efficient algorithms for minimizing cross validation error. In *Proceedings of the Eleventh International Conference on Machine Learning*, pages 190–198, 1994.

[66] A. N. Mucciardi and E. E. Gose. A comparison of seven techniques for choosing subsets of pattern recognition. *IEEE Transactions on Computers*, C-20:1023–1031, 1971.

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

[68] A. Y. Ng. On feature selection: learning with exponentially many irrelevant features as training examples. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 404–412, 1998.

[69] K.S. Ng and H. Liu. Customer retention via data mining. *AI Review*, 14(6):569 – 590, 2000.

[70] K. Nigam, A. K. Mccallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39:103–134, 2000.

[71] A. L. Oliveira and A. S. Vincentelli. Constructive induction using a non-greedy strategy for feature selection. In *Proceedings of the Ninth International Conference on Machine Learning*, pages 355–360, 1992.

[72] L. Parsons, E. Haque, and H. Liu. Subspace clustering for high dimensional data: a review. *SIGKDD Explorations*, 6(1):90–105, 2004.

[73] P. Pudil and J. Novovicova. *Novel Methods for Subset Selection with Respect to Problem Knowledge*, pages 101 – 116. In Liu and Motoda [52], 1998. 2nd Printing, 2001.

[74] D. Pyle. *Data Preparation for Data Mining*. Morgan Kaufmann Publishers, 1999.

[75] C. E. Queiros and E. S. Gelsema. On feature selection. In *Proceedings of the Seventh International Conference on Pattern Recognition*, pages 128–130, 1984.

[76] T. Reinartz. A unifying veiw on instance selection. *Data Mining and Knowledge Discovery*, 6(2):191 – 210, 2002.

[77] Y. Rui, T. S. Huang, and S. Chang. Image retrieval: Current techniques, promising directions and open issues. *Visual Communication and Image Representation*, 10(4):39–62, 1999.

[78] J. C. Schlimmer. Efficiently inducing determinations : a complete and systematic search algorithm that uses optimal pruning. In *Proceedings of the Tenth International Conference on Machine Learning*, pages 284–290, 1993.

[79] J. Segen. Feature selection and constructive inference. In *Proceedings of the Seventh International Conference on Pattern Recognition*, pages 1344–1346, 1984.

[80] J. Sheinvald, B. Dom, and W. Niblack. A modelling approach to feature selection. In *Proceedings of the Tenth International Conference on Pattern Recognition*, pages 535–539, 1990.

[81] W. Siedlecki and J. Sklansky. On automatic feature selection. *International Journal of Pattern Recognition and Artificial Intelligence*, 2:197–220, 1988.

[82] D. B. Skalak. Prototype and feature selection by sampling and random mutation hill climbing algorithms. In *Proceedings of the Eleventh International Conference on Machine Learning*, pages 293–301, 1994.

[83] N. Slonim, G. Bejerano, S. Fine, and N. Tishbym. Discriminative feature selection via multiclass variable memory markov model. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pages 578–585, 2002.

[84] P. Smyth, D. Pregibon, and C. Faloutsos. Data-driven evolution of data mining algorithms. *Communications of the Association for Computing Machinery*, 45(8):33 – 37, 2002.

[85] D. J. Stracuzzi and P. E. Utgoff. Randomized variable elimination. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pages 594–601, 2002.

[86] D. L. Swets and J. J. Weng. Efficient content-based image retrieval using automatic feature selection. In *IEEE International Symposium on Computer Vision*, pages 85–90, 1995.

[87] L. Talavera. Feature selection as a preprocessing step for hierarchical clustering. In *Proceedings of Internationl Conference on Machine Learning (ICML'99)*, pages 389–397, 1999.

[88] H. Vafaie and I. F. Imam. Feature selection methods: genetic algorithms vs. greedy-like search. In *Proceedings of the International Conference on Fuzzy and Intelligent Control Systems*, 1994.

[89] I.H. Witten and E. Frank. *Data Mining - Pracitcal Machine Learning Tools and Techniques with JAVA Implementations*. Morgan Kaufmann Publishers, 2000.

[90] N. Wyse, R. Dubes, and A.K. Jain. A critical evaluation of intrinsic dimensionality algorithms. In E.S. Gelsema and L.N. Kanal, editors, *Pattern Recognition in Practice*, pages 415–425. Morgan Kaufmann Publishers, Inc., 1980.

[91] E. Xing, M. Jordan, and R. Karp. Feature selection for high-dimensional genomic microarray data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 601–608, 2001.

[92] L. Xu, P. Yan, and T. Chang. Best first strategy for feature selection. In *Proceedings of the Ninth International Conference on Pattern Recognition*, pages 706–708, 1988.

[93] J. Yang and V. Honavar. *Feature Subset Selection Using A Genetic Algorithm*, pages 117 – 136. In Liu and Motoda [52], 1998. 2nd Printing, 2001.

[94] Y. Yang and J. O. Pederson. A comparative study on feature selection in text categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning*, pages 412–420, 1997.

[95] L. Yu and H. Liu. Feature selection for high-dimensional data: a fast correlation-based filter solution. In *Proceedings of the twentieth International Conference on Machine Learning*, pages 856–863, 2003.

[96] L. Yu and H. Liu. Redundancy based feature selection for microarray data. In *Proceedings of the Tenth ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2004.