Advancing Feature Selection Research
– ASU Feature Selection Repository

Zheng Zhao
zhaozheng@asu.edu
Computer Science & Engineering
Arizona State University

Fred Morstatter
fred.morstatter@asu.edu
Computer Science & Engineering
Arizona State University

Shashvata Sharma
shashvata.sharma@asu.edu
Computer Science & Engineering
Arizona State University

Salem Alelyani
salelyan@asu.edu
Computer Science & Engineering
Arizona State University

Aneeth Anand
aanand11@asu.edu
Computer Science & Engineering
Arizona State University

Huan Liu
huan.liu@asu.edu
Computer Science & Engineering
Arizona State University

Abstract

The rapid advance of computer based high-throughput technique have provided unparalleled opportunities for humans to expand capabilities in production, services, communications, and research. Meanwhile, immense quantities of high-dimensional data are accumulated challenging state-of-the-art data mining techniques. Feature selection is an essential step in successful data mining applications, which can effectively reduce data dimensionality by removing the irrelevant (and the redundant) features. In the past few decades, researchers have developed large amount of feature selection algorithms. These algorithms are designed to serve different purposes, are of different models, and all have their own advantages and disadvantages. Although there have been intensive efforts on surveying existing feature selection algorithms, to the best of our knowledge, there is still not a dedicated repository that collects the representative feature selection algorithms to facilitate their comparison and joint study. To fill this gap, in this work we present a feature selection repository, which is designed to collect the most popular algorithms that have been developed in the feature selection research to serve as a platform for facilitating their application, comparison and joint study. The repository also effectively assists researchers to achieve more reliable evaluation in the process of developing new feature selection algorithms.

1 Introduction

Data mining is a multidisciplinary effort to extract nuggets of knowledge from data. The proliferation of large data sets within many domains poses unprecedented challenges to data mining [20]. Not only are data sets getting larger, but new types of data have also evolved, such as data streams on the Web, microarrays in genomics and proteomics, and networks in social computing and system biology. Researchers and practitioners are realizing that in order to use data mining tools effectively, feature selection is an integral component to successful data mining [30].

Feature selection, a process of selecting a subset of original features according to certain criteria, is an important and frequently used dimensionality reduction technique for data mining [30, 18, 31]. 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, and improving mining performance such as predictive accuracy and result comprehensibility.

Feature selection has been an active field of research topic and development for decades in machine learning, and data mining, and widely applied to many fields such as genomic analysis [22], text mining [14], image retrieval [17, 53], intrusion detection [28], to name a few. As new applications emerge in recent years, many new challenges arise requiring novel theories and methods addressing high-dimensional and complex data. Stable feature selection [60], optimal redundancy removal [7] and the exploitation of auxiliary data and prior knowledge in feature selection [69, 67] are among the most fundamental and challenging problems in feature selection. Up-to-date, large volume of literature has been published along the research direction of feature selection. For example, by searching in Pubmed[1] with keywords: "gene selection", "feature selection" and "variable selection" in paper titles, one can obtain over 500 papers which are directly related to the design, the evaluation, or the application of feature selection techniques. This number can become even bigger, if we further conduct search in keywords. Given large an amount of existing works, a systematical summarization and comparison are of necessity to facilitate the research and the application of feature selection techniques. Recently, there have been quite a few surveys published to serve this purpose. For instance, two comprehensive surveys for feature (or variable) selection published in machine learning or statistic domain can be found in [18, 35]. In [47], the authors provided a good review for applying feature selection techniques in bioinformatics. In [22], the authors surveyed the filter and the wrapper model for feature selection. In [40], the authors explore the representative feature selection approaches based on sparse regularization, which is a branch of embedded model. Representative feature selection algorithms are also empirically compared and evaluated in [37, 29, 51, 27, 39, 52, 42] with different problem settings from different perspectives.

Despite the intensive efforts on surveying existing feature selection algorithms, there is still an important issues left unaddressed. When design new feature selection algorithms, or pick existing feature selection algorithms for solving a certain problem, researchers need to conduct comparison across a spectrum of different existing feature selection algorithms to obtain comprehensive views on the performance of either the new algorithms or the existing ones. Although the implementation of some popular feature selection algorithms can be found in software package, such as Weka [57], Spider [2], and MLC++ [3], there is still not a repository that dedicate on collecting representative feature selection algorithms to facilitate their comparison and joint study. To fill this gap, we developed a feature selection repository, which is designed to collect the most popular algorithms that have been developed in the feature selection research to serve as a platform to facilitate their application, comparison and joint study. The remaining parts of the paper is organized as follows, in Section 2, we provide the background on feature selection and visit its key concepts and components, and study their relationships and roles in algorithm design. In Section 3, we present the design of the feature selection repository. In Section 4, we outline the feature selection algorithms in the repository. And compare them through experiments in Section 5. Finally, we make conclusion and conduct discussion in Section 6.

2 Background on Feature Selection

The high dimensionality of data poses challenges to learning tasks due to the curse of dimensionality. In the presence of many irrelevant features, learning models tend to overfit and become less comprehensible. Feature selection is one effective means to identify relevant features for dimensionality reduction [18, 35]. Various studies show that features can be removed without performance deterioration [43, 8]. The training data can be either labeled, unlabeled or partial labeled, leading to the development of **supervised**, **unsupervised** and **semi-supervised** feature selection algorithms. Supervised feature selection [49, 56, 50, 64]

[1] http://www.ncbi.nlm.nih.gov/pubmed/

[2] http://www.kyb.tuebingen.mpg.de/bs/people/andre/spider.htm

[3] http://www.sgi.com/tech/mlc/index.html

determines feature relevance by evaluating feature's correlation with the class, and without labels, unsupervised feature selection exploits data variance and separability to evaluate feature relevance [11, 21]. Semi-supervised feature selection algorithms [68, 58] can use both labeled and unlabeled data, and its motivation is to use small amount of labeled data as additional information to improve the performance of unsupervised feature selection. Feature selection algorithms designed with different strategies broadly fall into three categories: **filter**, **wrapper** and **embedded** models. The filter model relies on the general characteristics of data and evaluates features without involving any learning algorithm. While, the wapper model requires a predetermined learning algorithm and uses its performance as evaluation criterion to select features. Algorithms with embedded model, e.g., C4.5 [46] and LARS [12], incorporate variable selection as a part of the training process, and feature relevance is obtained analytically from the objective of the learning model. Feature selection algorithms with filter and embedded models may return either a subset of selected features or the weights (measuring feature relevance) of all features. According to the type of the output, they can be divided into **feature weighting** and **subset selection** algorithms. Algorithms with wrapper model usually return feature subset. To the best of our knowledge, currently most feature selection algorithms are designed to handle learning tasks with **single data source**, although the capability of using auxiliary data sources in **multi-source feature selection** may greatly enhance the learning performance [38, 67]. Below, we visit the key concept of **relevance** & **redundancy** for feature selection, as well as the important components in a feature selection process.

Relevance and Redundancy

A popular definition for relevance is given in [24] for feature selection as the following. Let $\mathbf{F}$ be the full set of features, F_i be a feature, $\mathbf{S}_i = \mathbf{F} - \{f_i\}$. Let C denote the class label. And let P denote the conditional probability of the class label C given a feature set. The statistical relevance of a feature can be formalized as:

Definition 1 (Relevance) *A feature F_i is relevant iff*

$$\exists\ \mathbf{S}_i' \subseteq \mathbf{S}_i,\ \text{such that}\ P\left(C|G_i, \mathbf{S}_i'\right) \neq P\left(C|\mathbf{S}_i'\right). \tag{1}$$

Otherwise, the feature F_i is said to be irrelevant.

Definition 1 suggests that a feature is statistically relevant if its removal from a feature set will reduce the prediction power. The definition suggests that a feature can be statistically relevant due to two reasons: (1) it is strongly correlated with the class; or (2) it forms a feature subset with other features and the subset is strongly correlated with the class. If a feature is relevant because of the second reason, there exists feature **interaction** [63], which is also studied in machine learning as feature interaction [23, 65]. A related concept to the statistical relevance is the **redundancy**, which can be formalized as:

Definition 2 (Redundancy) *A feature F_i is redundant iff*

$$P\left(C|F_i, \mathbf{S}_i\right) = P\left(C|\mathbf{S}_i\right),\ \text{but}\ \ \exists\ \mathbf{S}_i' \subseteq \mathbf{S}_i,\ \text{such that}\ P\left(C|F_i, \mathbf{S}_i'\right) \neq P\left(C|\mathbf{S}_i'\right) \tag{2}$$

A feature, F_i, can become redundant due to the existence of other relevant features, which provide similar prediction power as F_i. Some searchers proposed to remove redundant features [7, 44, 62] from feature list, as this may improve the prediction accuracy. While, other researchers noticed that the removal of the redundant features may cause the exclusion of potential relevant features. Therefore, they propose to find surrogate features by measuring feature correlations [61], or group features with similar patterns into feature clusters [1, 60].

A unified view of feature selection process

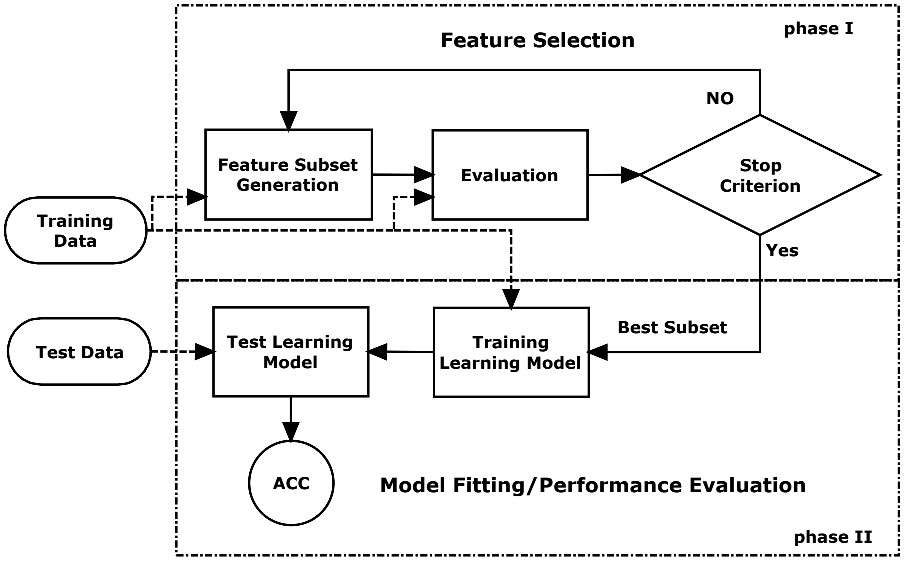


Figure 1: A unified view of feature selection process

The four components in a feature selection process includes: feature subset generation, subset evaluation, stopping criteria, and results validation. Figure 1 illustrates the four different components of a general feature selection process. In phase 1, subset generation produces candidate feature subsets based on a certain search strategy. Each candidate subset is evaluated by a certain evaluation measure and compared with the previous best one with respect to this measure. If a 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. In phase 2, The finally selected subset is subject to result validation by some given learning algorithms.

3 The ASU Feature Selection Repository

The ASU feature selection repository (http://featureselection.asu.edu) is designed to collect the most popular feature selection algorithms to facilitate their application, comparison and joint study. It contains two parts:

1. A MATLAB package, which contains a set of the most popular feature selection algorithms that have been developed so far and a set of auxiliary tools that facilitate the evaluation and comparison of these algorithms.

2. A web site, which allows researchers to collaborate, share their opinions on existing feature selection algorithms, and publish their own implementations.

Below we reveal the detail of the two components. We also provide a tutorial on how to set up the MATLAB program and modify it to support any user-created feature selection algorithms.

3.1 Website (http://featureselection.asu.edu)

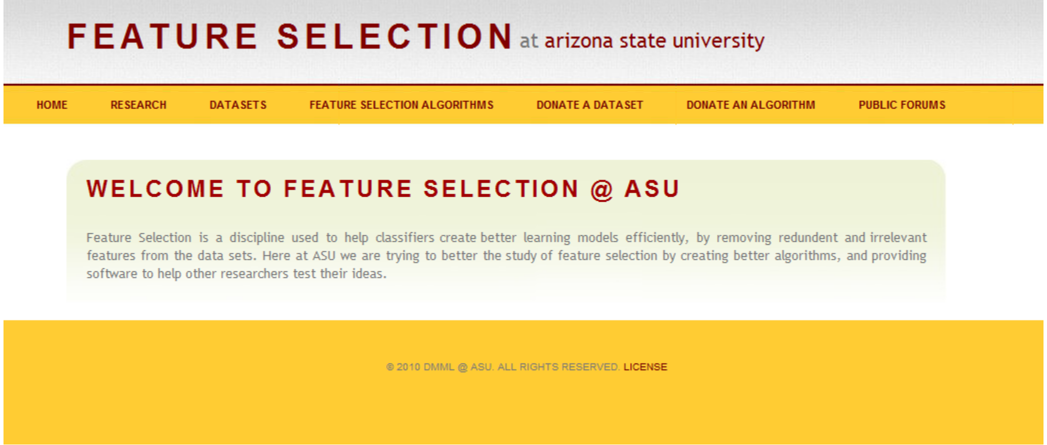
Fig. 2 shows the screen snapshot of the welcome page of the ASU feature selection repository website. It contains three portions:

1. The algorithm portion: this portion contains the feature selection algorithms, their documentation, and any associated meta information. It also allows users to rate and conduct discussions on the algorithms.

2. The dataset portion: this portion contains the benchmark datasets, which are provided together with the feature selection algorithm for serving the evaluation purpose. All benchmark datasets are currently in Matlab data format.

3. The data and algorithm donation portion: this portion allow users to donate their algorithms and benchmark datasets for sharing with other researchers.

Below we go through each of the three potions in detail.



FEATURE SELECTION at arizona state university

WELCOME TO FEATURE SELECTION @ ASU

Feature Selection is a discipline used to help classifiers create better learning models efficiently, by removing redundent and irrelevant features from the data sets. Here at ASU we are trying to better the study of feature selection by creating better algorithms, and providing software to help other researchers test their ideas.

© 2010 DMML @ ASU. ALL RIGHTS RESERVED. LICENSE

Figure 2: A screen snapshot of the welcome page of the ASU feature selection repository website.

3.1.1 The Algorithm Portion (http://featureselection.asu.edu/software.php)

The screen snapshot of the webpage for the algorithm portion can be found in Figure 3. The page lists all of the feature selection algorithms that contained in the repository. For each algorithm, the following types of information are provided: (1) a link to its documentation, (2) its corresponding reference, (3) its implementation in MATLAB coding format, (4) a link to a Microsoft Excel file containing its performance evaluation, and (5) the rating of the algorithm provided by users.

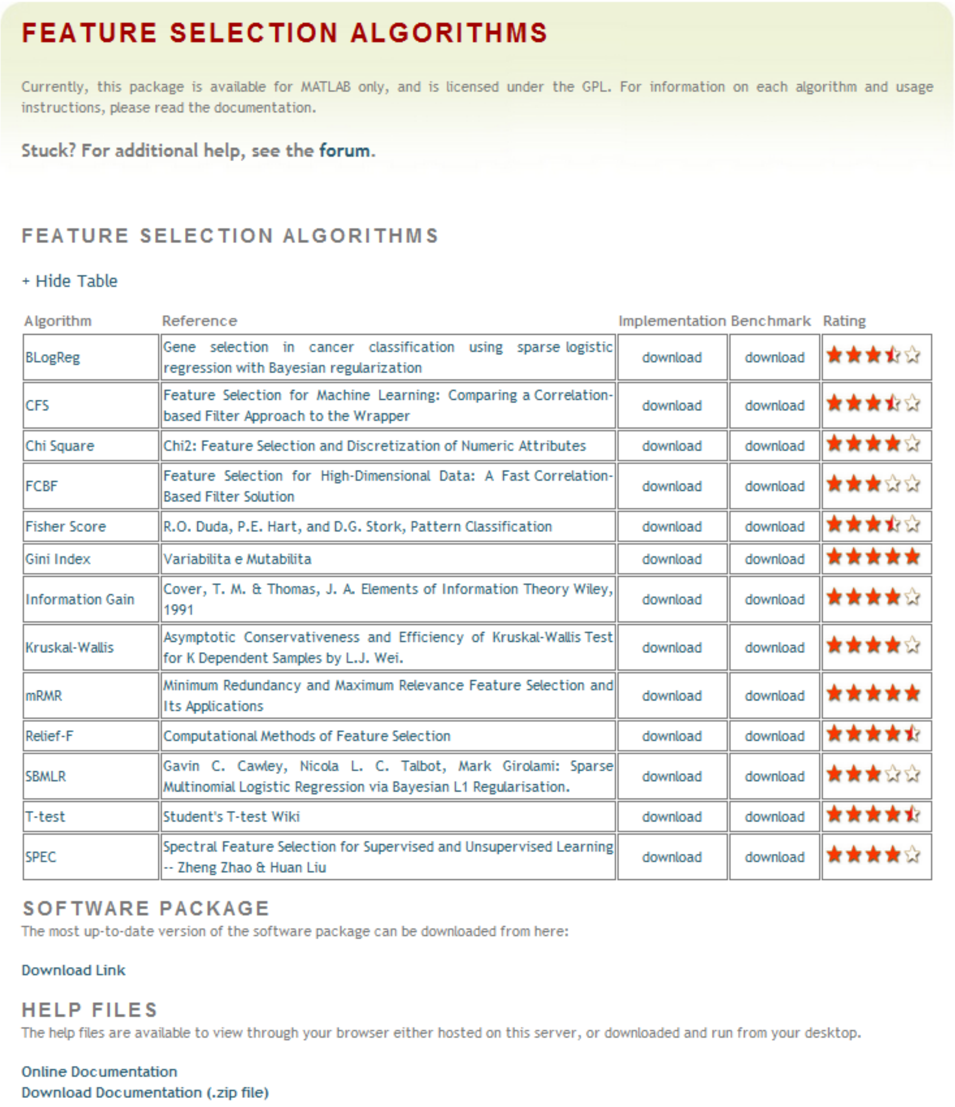
The documentation of a feature selection algorithm carries the information about the general characters of the algorithm, its usage, the example code, and a BibTeX entry of the paper, in which the algorithm is first proposed. The complete set of documentation of all the feature selection algorithm contained in the repository can be accessed online[4], or downloaded as a .zip package[5].

To evaluate the performance of each feature selection algorithm, two measurements are used: (1) accuracy rate, the accuracy rate is obtained by running three classifiers: SVM, naive Bayes, and decision tree on the top ranked features selected by different feature selection algorithms on various benchmark datasets. (2) Redundancy rates, the redundancy rates is obtained by calculating the averaged correlation among the selected features returned by different feature selection algorithms on various benchmark datasets. The details on how these two measurement are calculated can be found in the experiment part of the report.

[4] http://featureselection.asu.edu/documentation/

[5] http://featureselection.asu.edu/fs-doc.zip

Users can provide their opinions on different feature selection algorithm through two ways: (1) rating the algorithms by clicking the stars associated to each algorithm, (2) leaving comments for the algorithm in the forum under the corresponding topic.



FEATURE SELECTION ALGORITHMS

Currently, this package is available for MATLAB only, and is licensed under the GPL. For information on each algorithm and usage instructions, please read the documentation.

Stuck? For additional help, see the forum.

FEATURE SELECTION ALGORITHMS

+ Hide Table

Algorithm	Reference	Implementation	Benchmark	Rating
BLogReg	Gene selection in cancer classification using sparse logistic regression with Bayesian regularization	download	download	★★★⯪☆
CFS	Feature Selection for Machine Learning: Comparing a Correlation-based Filter Approach to the Wrapper	download	download	★★★⯪☆
Chi Square	Chi2: Feature Selection and Discretization of Numeric Attributes	download	download	★★★★☆
FCBF	Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution	download	download	★★★☆☆
Fisher Score	R.O. Duda, P.E. Hart, and D.G. Stork, Pattern Classification	download	download	★★★⯪☆
Gini Index	Variabilita e Mutabilita	download	download	★★★★★
Information Gain	Cover, T. M. & Thomas, J. A. Elements of Information Theory Wiley, 1991	download	download	★★★★☆
Kruskal-Wallis	Asymptotic Conservativeness and Efficiency of Kruskal-Wallis Test for K Dependent Samples by L.J. Wei.	download	download	★★★★☆
mRMR	Minimum Redundancy and Maximum Relevance Feature Selection and Its Applications	download	download	★★★★★
Relief-F	Computational Methods of Feature Selection	download	download	★★★★⯪
SBMLR	Gavin C. Cawley, Nicola L. C. Talbot, Mark Girolami: Sparse Multinomial Logistic Regression via Bayesian L1 Regularisation.	download	download	★★★☆☆
T-test	Student's T-test Wiki	download	download	★★★★⯪
SPEC	Spectral Feature Selection for Supervised and Unsupervised Learning -- Zheng Zhao & Huan Liu	download	download	★★★★☆

SOFTWARE PACKAGE

The most up-to-date version of the software package can be downloaded from here:

Download Link

HELP FILES

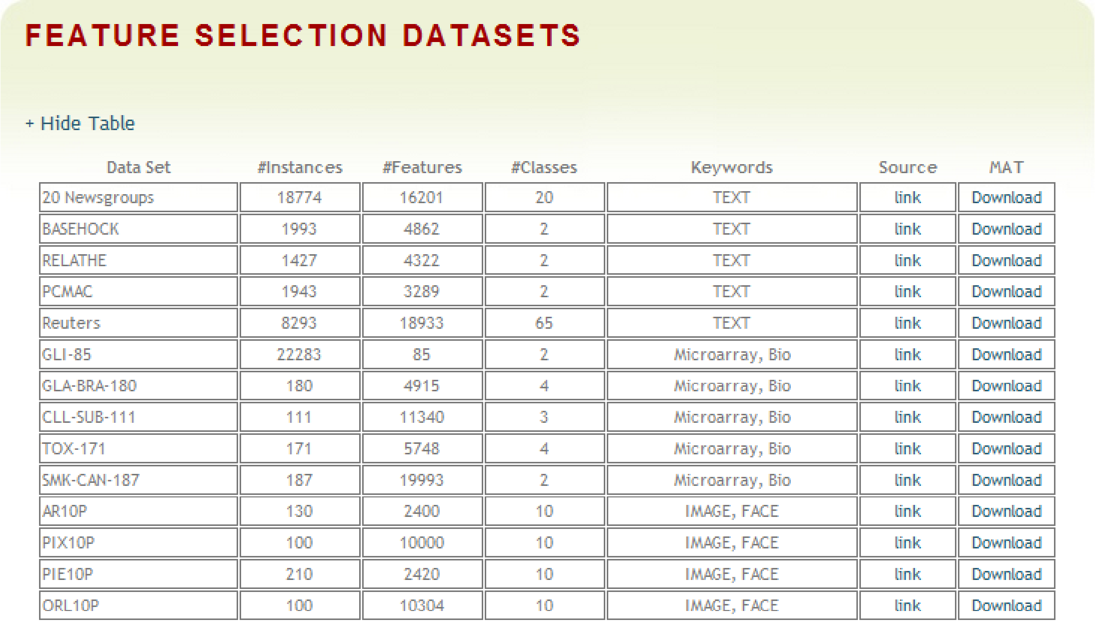
The help files are available to view through your browser either hosted on this server, or downloaded and run from your desktop.

Online Documentation
Download Documentation (.zip file)

Figure 3: A screen snapshot of the webpage for the algorithm portion of the ASU feature selection repository. Note, the header of the webpage has been cropped for easy reading.

3.1.2 The Dataset Portion (http://featureselection.asu.edu/datasets.php)

The screen snapshot of the webpage for the dataset portion can be found in Figure 4. The page lists all of the benchmark datasets that are provided with the feature selection algorithms to facilitate their evaluation. For each benchmark dataset, the following types of information is provided, including: (1) the number of the instances in the data, (2) the number of features, (3) the number of classes, (4) its category information, (5) a link of the place, where the original data is downloaded, and (6) a link for downloading the data in Matlab format.



FEATURE SELECTION DATASETS

+ Hide Table

Data Set	#Instances	#Features	#Classes	Keywords	Source	MAT
20 Newsgroups	18774	16201	20	TEXT	link	Download
BASEHOCK	1993	4862	2	TEXT	link	Download
RELATHE	1427	4322	2	TEXT	link	Download
PCMAC	1943	3289	2	TEXT	link	Download
Reuters	8293	18933	65	TEXT	link	Download
GLI-85	22283	85	2	Microarray, Bio	link	Download
GLA-BRA-180	180	4915	4	Microarray, Bio	link	Download
CLL-SUB-111	111	11340	3	Microarray, Bio	link	Download
TOX-171	171	5748	4	Microarray, Bio	link	Download
SMK-CAN-187	187	19993	2	Microarray, Bio	link	Download
AR10P	130	2400	10	IMAGE, FACE	link	Download
PIX10P	100	10000	10	IMAGE, FACE	link	Download
PIE10P	210	2420	10	IMAGE, FACE	link	Download
ORL10P	100	10304	10	IMAGE, FACE	link	Download

Figure 4: A screen snapshot of the webpage for the dataset portion of the ASU feature selection repository. Note, the header of the webpage is cropped for easy reading.

3.1.3 The Data and Algorithm Donation Portion (http://featureselection.asu.edu/donateal.php, donateds.php)

To facilitate active communication and collaboration in the feature selection community, the web site offers a portion that allows the user to share their algorithms and datasets with the rest of the community. The screen snapshot of the algorithm donation webpage is shown in Figure 5. All of the algorithms and datasets in the repository are currently provided in the MATLAB format, but the repository allows users to donate their algorithm implementations and datasets in any format.

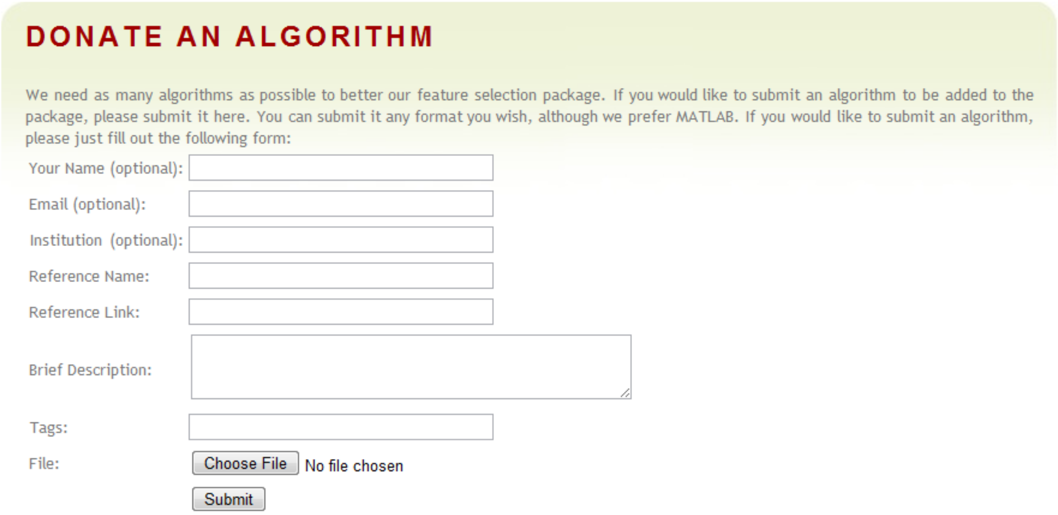
By submitting either an algorithm or a dataset, the user retains all the rights associated with those files. At the same time, the submitter is implicitly granting us the right to display the submission on the site. If at any time a submitter would like to see their submission removed, or updated, they must send an email to the administrator of the website. The contact information can be found on the contact page.

3.2 The Matlab Software Package

To facilitate the application and comparison of various feature selection algorithm, ASU feature selection repository provides software package, which contains a set of the most popular feature selection algorithms and a set of auxiliary tools for learning model fitting and algorithm evaluation. All the feature selection algorithms in the repository are either implemented in Matlab or implemented in other languages but are made accessible through an Matlab wrapper. Below we introduce the protocol used to implement the algorithms, organization of the package, and provide detailed information and examples on how to install, use, and add user implemented feature selection algorithms to the package.

3.2.1 Algorithm Implementation Protocol

The facilitate the evaluation of various feature selection algorithms in a unified framework, we defined the following protocol for algorithm implementation:



DONATE AN ALGORITHM

We need as many algorithms as possible to better our feature selection package. If you would like to submit an algorithm to be added to the package, please submit it here. You can submit it any format you wish, although we prefer MATLAB. If you would like to submit an algorithm, please just fill out the following form:

Your Name (optional):

Email (optional):

Institution (optional):

Reference Name:

Reference Link:

Brief Description:

Tags:

File:

Figure 5: A screen snapshot of the webpage for the data and algorithm donation portion of the ASU feature selection repository. Note, the header of the webpage has been cropped.

- **Input:** The input of a feature selection algorithm must contain three standard parameters:

  1. **X -** the $n \times m$ data matrix. Each of the n rows is an instance, and each of the m columns is a feature.
  2. **Y -** a vector of size n specifying the class label of the samples. The class label of samples are coded by integers in the form of 1, 2, 3, . . .
  3. **options -** a Matlab structure, with its fields specifying the model parameters of the algorithm. Different feature selection algorithms need specify different model parameters, which are described in detail in it related documentation.

- **Output:** The output of a feature selection algorithm is a MATLAB structure, named “out”. It contains three fields:

  1. **.W**, for feature weighting algorithms, this field contains the weight of each features.
  2. **.fList**, this field contains a set of feature index. For “feature weighting algorithms”, the elements in “flist” is ordered according to features relevance. That is the features in the top of the rank list are most relevant features according to the algorithm. For “subset selection algorithms”, the elements in the “fList” may or may not have order, which depends on the nature of the specific feature selection algorithms. In case the elements are ordered, all selected features in “fList” are equally relevant.
  3. **.prf**, this field specifies whether the larger the weights the more relevant the features (.prf=1), or the smaller the weights, the more relevant (.prf=-1). For “subset selection” algorithms, if (.prf=0), the features in the “fList” are not ordered.
  4. **.fImp**, this field indicates that whether the algorithm is a “subset selection” algorithm. For all subset selection algorithm, `out.fImp = true`.

Additional information about implementation and usage for each specific feature selection algorithm can be found in its related documentation.

3.2.2 Directory Organization

The feature selection package consists of six main sections:

1. **package loading**: `load_fspackage.m`. This script adds all the feature selection algorithms, benchmark data sets, and helper methods to the user's path, so that all the resources contained in the package can be easily accessed by users.

2. **feature selection algorithms**: the `fs_*` directories. The directories start with the `fs` prefix containing the implementations of feature selection algorithms. The directories containing `sup` prefix corresponds to supervised feature selection algorithms, and directories containing `uns` prefix corresponds to unsupervised feature selection algorithms.

3. **learning models**: the `classifiers/ and clusters/` directories. The two directories contain the most popular supervised and unsupervised learning models, such as SVM, Naive Bayes, J48, k-NN, and k-means. These learning models are provided as supporters to facilitate learning after feature selection. It can also be used in feature selection algorithms evaluation phase for evaluating algorithms.

4. **data preprocessors**: the `preprocessor/` directory. This directory contains supporters for data preprocessing. The functions that are supported include: data normalization, class label format conversion, kernel computation, data partition for cross-validation, etc.

5. **algorithm evaluators**: the `examples/` directory. This directory contains an evaluation framework that allows researchers to conveniently evaluate various feature selection algorithms, and automatically generate standard charts and tables to compare their performance.

6. **support libs**: the `lib` directory. This directory contains the libs that are used in the feature selection repository, which currently includes: Weka (in a jar file), lib_SVM and mutual_info. Each lib corresponding to a dedicated directory under the `lib` directory.

Figure 6 shows that how different components in a standard feature selection process are mapped to directory structure of the package. More specifically, we have the following relationship:

1. **feature selection**: the `fs_*` directories.

2. **model fitting**: the `classifiers/ and clusters/` directories.

3. **performance evaluation**: the `examples/` directory.

4. **data preprocessing**: the `preprocessor/` directory.

Table 1: Data transformation of SY2MY for a three classes problem.

Y	**SY2MY(Y)**		
1	1	-1	-1
1	1	-1	-1
2	-1	1	-1
3	-1	-1	1

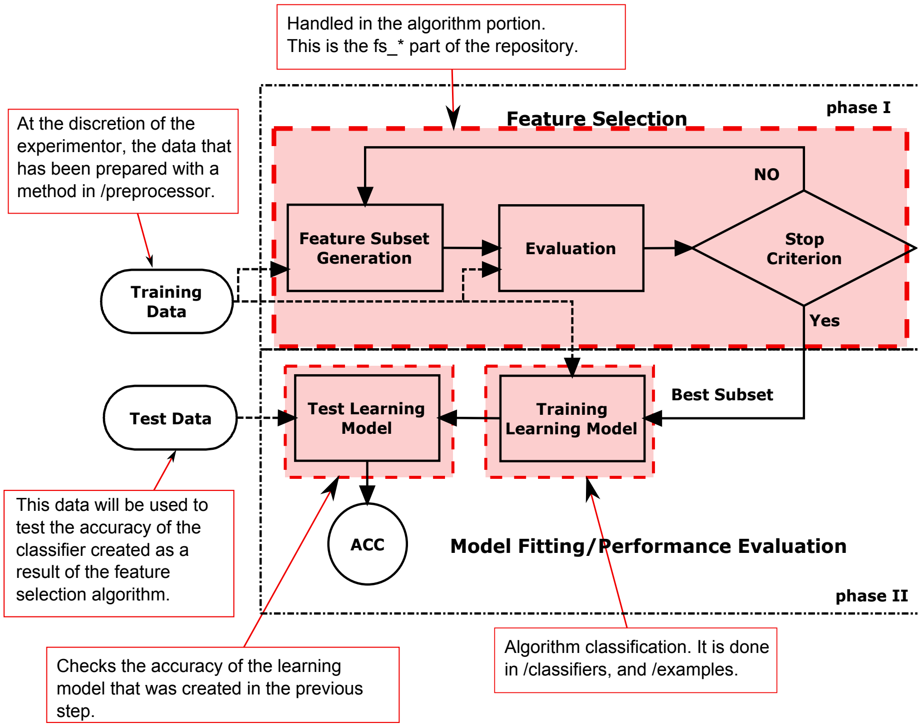

Handled in the algorithm portion.
This is the fs_* part of the repository.
At the discretion of the experimentor, the data that has been prepared with a method in /preprocessor.
phase I
Feature Selection
NO
Feature Subset Generation
Evaluation
Stop Criterion
Training Data
Yes
Best Subset
Test Data
Test Learning Model
Training Learning Model
ACC
Model Fitting/Performance Evaluation
phase II
This data will be used to test the accuracy of the classifier created as a result of the feature selection algorithm.
Checks the accuracy of the learning model that was created in the previous step.
Algorithm classification. It is done in /classifiers, and /examples.

Figure 6: How different components in a standard feature selection process are mapped to the directories of the feature selection package.

∗Hint ∗
Preprocessing Data -
Data preprocessing ensures that data are well prepared for certain types of analysis, the following data preprocessor are provided in the package:

1. Y=SY2MY(Y) - transforming Y with single column representation to its corresponding binary 1 vs. all matrix representation. An example of the conversion are shown Table 1.
2. [ID] = buildIDX(X, Y, percent, iter) - resampling per percent instances from each class for building training and test data for algorithm evaluation. X is the instance data, each row is an instance and each column a feature. Y is the list of class labels. The argument 'percent' is the percent of instances to sample. The argument 'iter' tells how many times to sample. It returns an $n \times iter$ matrix, where n is the total number of instances, and iter is times for sampling. Each column of the ID matrix corresponding to a sample and its elements are either 1 or 0, where 1 denotes the corresponding instances are sampled as a training instance.
3. [X, meanX, nor] = normData(X, opt, ctr) - normalizes data. If opt = 1, normalize features, if opt = 2, normalize the instance. Ctr is a boolean used to indicate whether to centralize the data in the data normalization process. In the output X is the normalized data, meanX is the mean of X and nor is the normalization factor computed from X. meanX and nor are usually obtained from the training data, which are used to normalize the test data in the test process.

3.2.3 Package Installation And Environment Setup

To facilitate the installation of the feature selection package, the package provides a java based installer, which is available for downloading from the web site.

- **Install -** To install the package, double click the downloaded file (in executable .jar format), and run through the wizard. It can be installed in any directory on the file system. For Microsoft Windows users, the traditional ”Program Files” structure is not necessary.

- **Uninstall -** The software package does not write to the registry, or copy any configuration files to any secret directory. Therefore, the user can just delete the folder where the package is installed.

After the package has been installed, a user can run `load_fspackage.m` to add all the feature selection algorithms, benchmark data sets, and helper methods to the user's path, so that all the resources contained in the package can be easily accessed.

∗Hint ∗

Package Shortcut -

Running `load_fspackage` every single time when the user wishes to use the feature selection package may be inconvenient. In order to alleviate this, make an application shortcut that will set up the environment. Follow these steps:

1. Open MATLAB as usual.
2. Use the `cd` function to navigate to the folder where the feature selection package is installed. If unsure as to which directory this is, it is the one which has `load_fspackage.m` in it.
3. Run the command `load_fspackage`
4. In the "Command History" window (if it is not shown, display it by going to Desktop → Command History), select all of the commands in the session by holding the control key and clicking them with the mouse. When they are all selected, right click the blue area, and click "Create Shortcut" to make a shortcut in the bar at the top of the MATLAB interface.

Clicking the shortcut to start using the feature selection package will make for a better experience with our software. Delete this manually when uninstalling the package.

3.2.4 Running An Algorithm

To run the feature selection algorithms in the package, the user needs to specify the name of the algorithm and the name of the dataset to run the algorithm upon[6]. For example, to apply the ReliefF feature selection algorithm on the `wine`[7] data can be achieved through the following steps:

1. Load the dataset (located in `/examples/data/`) with the following command:
   ```
   load 'examples/data/wine.mat'
   ```
   Two matrices 'X', and 'Y' will appear in the workspace. 'X' is the list of data points, each row being an instance. 'Y' is the class label.
2. Run the experiment with the following command:
   ```
   result = fsReliefF(X,Y)
   ```

Note, different feature selection algorithms may require different inputs and provide different outputs. Detailed information of algorithms' interface and usage examples can be found in the documentation provided with the package.

3.2.5 Running An Experiment

The package provides a routine for users to systematically evaluate the performance of feature selection algorithms. The logic flow of the routine can be found in Figure 7.

[6] Note, to facilitate the access of the feature selection algorithms in the package, one can run the `load_fspackage` function to setup the working environment.

[7] http://archive.ics.uci.edu/ml/datasets/Wine, this dataset is also provided in the package.

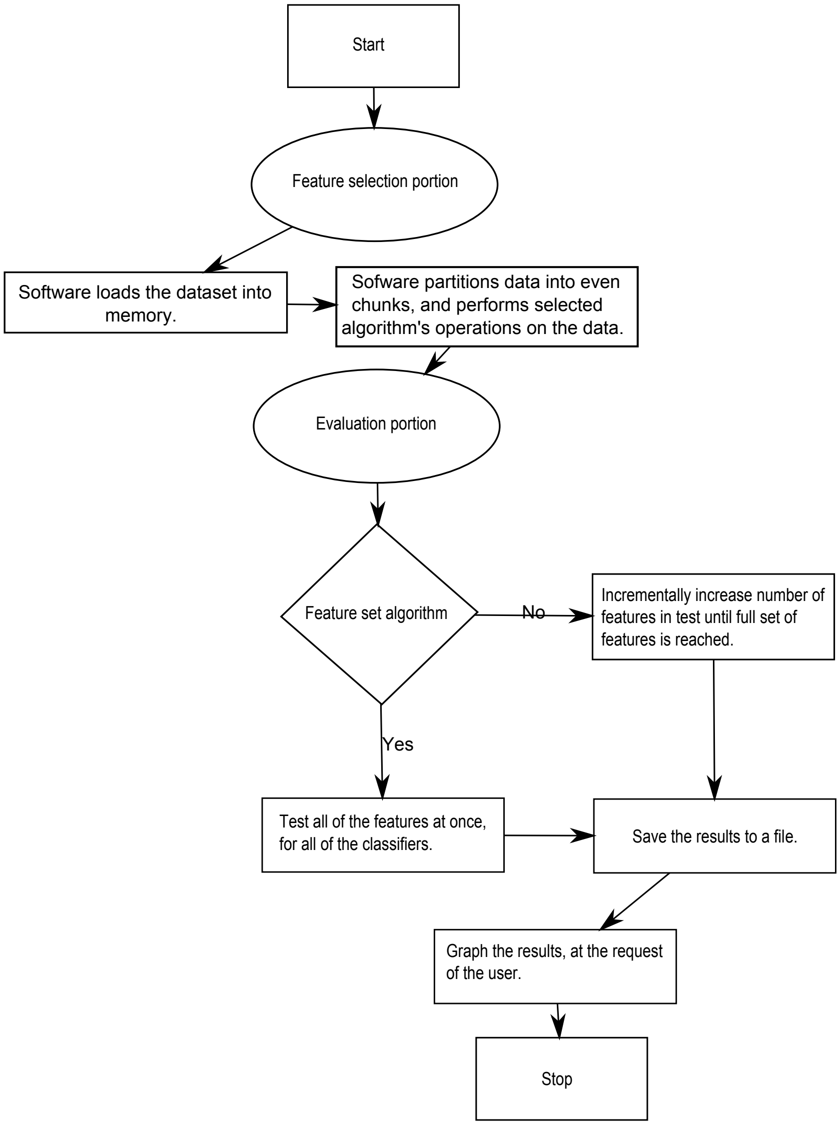

Start
Feature selection portion
Software loads the dataset into memory.
Sofware partitions data into even chunks, and performs selected algorithm's operations on the data.
Evaluation portion
Feature set algorithm
No
Incrementally increase number of features in test until full set of features is reached.
Yes
Test all of the features at once, for all of the classifiers.
Save the results to a file.
Graph the results, at the request of the user.
Stop

Figure 7: The flowchart of the performance evaluation routine (runExperiment).

Assume a user wants to evaluate the performance of ReliefF on the orlraws10P data[8]. The following steps can be used to achieve the purpose:

1. change directory: `cd '<package>/examples/result_statistic/supervised'`

2. run the experiment: `runExperiment('relieff','orlraws10P',1)`

`'<package>'` denotes the directory, where the package is installed. The third argument of `runExperiment` is boolean variable, indicating whether to generate the figures from the performance evaluation results. Figure 8 shows an example of the figures generated by the routine. Since ReliefF is a feature weighting algorithm, the routine generates an accuracy plot, which contains the accuracy achieved by the three classifier, decision tree (J48), Naive Bayes (NBC), and Support Vector Machine (SVM), using different number of features selected by ReliefF; and the redundancy rate plot, which shows the redundancy retained in the reduced data when different number of features are selected by ReliefF.

[8] In this case, orlraws10P.mat must be in the `<package>/examples/data/` directory.

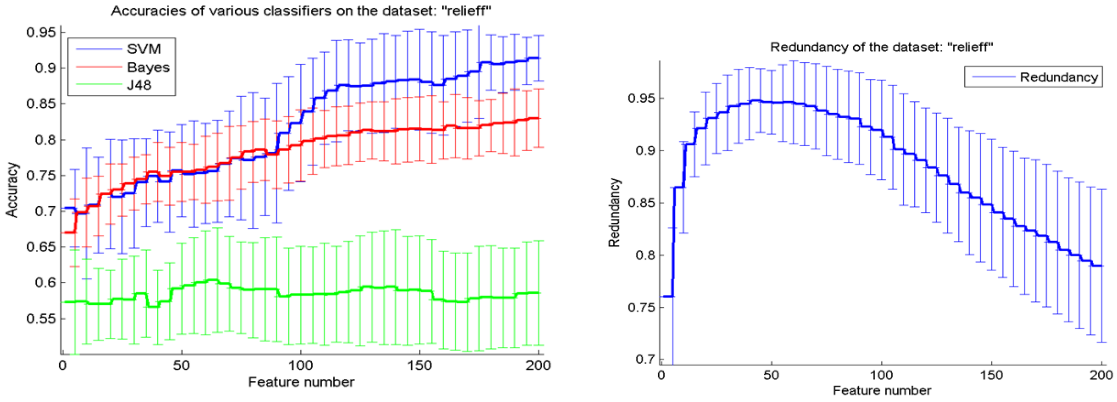


Figure 8: Feature selection algorithm performance evaluation. The left plot shows the accuracy achieved by the three classifier, decision tree (J48), Naive Bayes (NBC), and Support Vector Machine (SVM), using different number of features selected by ReliefF. The right plot shows the redundancy retained in the reduced data when different number of features are selected by ReliefF.

3.2.6 Add New Algorithms to The Package

New feature selection algorithms can be conveniently added to the package.

1. **integrating with the package**

   (a) Make sure that the algorithm conforms to the general protocol described in Section 3.2.1.

   (b) Put the algorithm in its own folder under the root directory of the package. Name the folder using the convention specified in Section 3.2.2.

   (c) Update the `load_fspackage.m` function to add the new algorithm's folder to the path. To do this, add the line: `path(path,[curPath filesep ’<New_Folder_Name>’])`.

2. **integrating with the evaluation framework**
   Editing `<package>/examples/code/result_statistic/supervised/expFun_wi_sam_feat.m`.

   (a) In the portion labeled ”Algorithm switch statement”, create a case for the algorithm.

   (b) Create a character array for the algorithm, which will be referred to as its ”keyword”.

3.2.7 Creating a Dataset

To create a dataset that conforms to the framework's protocol, it must have the following fields

1. **X** - A $n \times m$ matrix containing the data, in which, n is the number of instances and m is the number of features. Each row in the matrix corresponds to an instance, and each column corresponds to a feature.

2. **Y** - A column vector of size n contains the label of the samples. Assume the data has k different classes, the $\mathbf{Y}_i \in \{1, \ldots, k\},\ i = 1, \ldots, n$.

Save the data in Matlab format with the command: `save(’datasetname.mat’,’X’,’Y’)`. And move it to the directory `<package>/examples/data`.

4 Feature Selection Algorithms in the Repository

In this section, we briefly introduce the feature selection algorithms that has been included in the repository. Different feature selection algorithms are organized into three subsections according to the computational models they are based on.

4.1 Filter Model

4.1.1 Laplacian Score

unsupervised, filter, univariate, feature weighting

Laplacian Score is proposed in [21] to select features that retain sample locality specified by an affinity matrix $\mathbf{K}$. Given $\mathbf{K}$, its corresponding degree matrix $\mathbf{D}$ and Laplacian matrix $\mathbf{L}$, the Laplacian Score of a feature $\mathbf{f}$ is calculated in the following way:

$$\mathrm{SC}_L\left(\mathbf{f}\right) = \frac{\widetilde{\mathbf{f}}^\top \mathbf{L}\widetilde{\mathbf{f}}}{\widetilde{\mathbf{f}}^\top \mathbf{D}\widetilde{\mathbf{f}}}, \quad \text{where} \quad \widetilde{\mathbf{f}} = \mathbf{f} - \frac{\mathbf{f}^\top \mathbf{D}\mathbf{1}}{\mathbf{1}^\top \mathbf{D}\mathbf{1}}\mathbf{1}.$$

Using Laplacian Score to select k features is equivalent to optimizing the following objective:

$$\min_{i_1,\ldots,i_k} \sum_{j=1}^{k} \mathrm{SC}_L\left(\mathbf{f}_{i_j}\right) = \sum_{j=1}^{k} \frac{\widetilde{\mathbf{f}}_{i_j}^\top \mathbf{L}\widetilde{\mathbf{f}}_{i_j}}{\widetilde{\mathbf{f}}_{i_j}^\top \mathbf{D}\widetilde{\mathbf{f}}_{i_j}},$$
$$i_j \in \{1,\ldots,m\}, \quad p \neq q \to i_p \neq i_q.$$

In Laplacian Score, features are evaluated independently, therefore the optimization problem defined above can be solved by greedily picking the top k features which have the minimal SC_L values. Since features are evaluated individually, Laplacian Score cannot handle feature redundancy.

4.1.2 SPEC

unsupervised, supervised, filter, univariate, feature weighting

Proposed in [66], SPEC is an extension for Laplacian Score. In SPEC, given the affinity matrix $\mathbf{K}$, the degree matrix $\mathbf{D}$, and the normalized Laplacian matrix $\mathcal{L}$, three evaluation criteria are proposed for measuring feature relevance in the following way:

$$\mathrm{SC}_{S,1}(\mathbf{f}_i) = \widehat{\mathbf{f}}_i^\top \gamma(\mathcal{L})\ \widehat{\mathbf{f}}_i = \sum_{j=1}^{n} \alpha_j^2 \gamma(\lambda_j)$$

$$\mathrm{SC}_{S,2}(\mathbf{f}_i) = \frac{\widehat{\mathbf{f}}_i^\top \gamma(\mathcal{L})\ \widehat{\mathbf{f}}_i}{1 - \left(\widehat{\mathbf{f}}_i^\top \xi_1\right)^2} = \frac{\sum_{j=2}^{n} \alpha_j^2 \gamma(\lambda_j)}{\sum_{j=2}^{n} \alpha_j^2}$$

$$\mathrm{SC}_{S,3}(\mathbf{f}_i) = \sum_{j=1}^{k} (\gamma(2) - \gamma(\lambda_j))\alpha_j^2$$

In the above equations, $\widehat{\mathbf{f}}_i = (\mathbf{D}^{\frac{1}{2}}\mathbf{f}_i) \cdot ||(\mathbf{D}^{\frac{1}{2}}\mathbf{f}_i)||^{-1}$; (λ_j, ξ_j) is the eigensystem of $\mathcal{L}$; $\alpha_j = \cos\theta_j$, where θ_j is the angle between $\widehat{\mathbf{f}}_i$ and ξ_j; and $\gamma(\cdot)$ is an increasing function which is used to rescale the eigenvalues of $\mathcal{L}$ for denoising. The top eigenvectors of $\mathcal{L}$ are the optimal soft cluster indicators of the data [54]. By comparing with these eigenvectors, SPEC selects features that assign similar values to instances that are similar according to $\mathbf{K}$. In [66] it is shown that Laplacian Score is a special case of the second criterion, $\mathrm{SC}_{S,2}(\cdot)$, defined in SPEC. Note that SPEC also evaluates features individually, therefore it cannot handle feature redundancy.

4.1.3 Fisher Score

supervised, filter, univariate, feature weighting

Given class labels $\mathbf{y} = \{y_1, \ldots, y_n\}$, Fisher Score [9] selects features that assign similar values to the samples from the same class and different values to samples from different classes. The evaluation criterion used in Fisher Score can be formulated as:

$$\mathrm{SC}_F\left(\mathbf{f}_i\right) = \frac{\sum_{j=1}^{c} n_j \left(\mu_{i,j} - \mu_i\right)^2}{\sum_{j=1}^{c} n_j \sigma_{i,j}^2}.$$

Above μ_i is the mean of the feature $\mathbf{f}_i$, n_j is the number of samples in the jth class, and $\mu_{i,j}$ and $\sigma_{i,j}$ are the mean and the variance of $\mathbf{f}_i$ on class j, respectively. In [21], it is shown that Fisher Score is a special case of Laplacian Score, when n_l is the number of instances in l-th class and when $\mathbf{K}$ is defined as:

$$\mathbf{K}_{ij}^{FIS} = \begin{cases} \frac{1}{n_l}, & y_i = y_j = l \\ 0, & otherwise \end{cases}, \tag{3}$$

Fisher Score is an effective supervised feature selection algorithm, which has been widely applied in many real applications. However as the cases of Laplacian Score and SPEC,Fisher Score valuates features individually, therefore it cannot handle feature redundancy.

4.1.4 ReliefF

supervised, filter, univariate, feature weighting

Relief [25] and its multiclass extension ReliefF [26] are supervised feature weighting algorithms of the filter model. Assuming that p instances are randomly sampled from data, the evaluation criterion of Relief is define as:

$$\mathrm{SC}_R(\mathbf{f}_i) = \frac{1}{2}\sum_{t=1}^{p} d\left(f_{t,i} - f_{NM(\mathbf{x}_t),i}\right) - d\left(f_{t,i} - f_{NH(\mathbf{x}_t),i}\right),$$

where $f_{t,i}$ denotes the value of instance $\mathbf{x}_t$ on feature $\mathbf{f}_i$, $f_{NH(\mathbf{x}_t),i}$ and $f_{NM(\mathbf{x}_t),i}$ denote the values on the ith feature of the nearest points to $\mathbf{x}_t$ with the same and different class label respectively, and $d\left(\cdot\right)$ is a distance measurement. To handle multiclass problems, the above criterion is extended to the following form:

$$\mathrm{SC}_R(\mathbf{f}_i) = \frac{1}{p} \cdot \sum_{t=1}^{p} \left\{ -\frac{1}{m_{\mathbf{x}_t}} \sum_{\mathbf{x}_j \in NH(\mathbf{x}_t)} d\left(f_{t,i} - f_{j,i}\right) \right.$$
$$\left. + \sum_{y \neq y_{\mathbf{x}_t}} \frac{1}{m_{\mathbf{x}_t,y}} \frac{P(y)}{1 - P\left(y_{\mathbf{x}_t}\right)} \sum_{\mathbf{x}_j \in NM(\mathbf{x}_t,y)} d\left(f_{t,i} - f_{j,i}\right) \right\},$$

where $y_{\mathbf{x}_t}$ is the class label of the instance $\mathbf{x}_t$ and $P(y)$ is the probability of an instance being from the class y. $NH(\mathbf{x})$ or $NM(\mathbf{x}, y)$ denotes a set of nearest points to $\mathbf{x}$ with the same class of $\mathbf{x}$, or a different class (the class y), respectively. $m_{\mathbf{x}_t}$ and $m_{\mathbf{x}_t,y}$ are the sizes of the sets $NH(\mathbf{x}_t)$ and $NM(\mathbf{x}_t, y)$, respectively. Usually, the size of both $NH(\mathbf{x})$ and $NM(\mathbf{x}, y)$, $\forall\, y \neq y_{\mathbf{x}_t}$, is set to a pre-specified constant k. The evaluation criteria of Relief and ReliefF show that the two algorithms select features contribute to the separation of the samples from different classes. In [15], the authors related the relevance evaluation criterion of reliefF to hypothesis margin maximization, which explains that why the algorithm provide superior performance in many applications.

4.1.5 t-score, F-score

supervised, filter, univariate, feature weighting

t-score is used for binary problem. For unequal sample sizes and unequal variance case the t-score can be calculated as:

$$R_t = \frac{\mu_1 - \mu_2}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}} \tag{4}$$

F-score is used to test if a feature is able to well separate samples from different classes by considering between class variance and within class variance and is calculated as:

$$R_f = \frac{\sum\limits_i \frac{n_i}{c-1} (\mu_i - \mu)^2}{\frac{1}{n-c} \sum\limits_i (n_i - 1) \sigma_i^2} \tag{5}$$

4.1.6 Chi-square Score

supervised, filter, univariate, feature weighting

Chi-square [33] is used to assess two types of comparison: tests of goodness of fit and tests of independence. In feature selection it is used as a test of independence to assess whether the class label is independent of a particular feature. Chi-square score for a feature with r different values and C classes is defined as

$$\chi^2 = \sum_{i=1}^{r} \sum_{j=1}^{C} \frac{(n_{ij} - \mu_{ij})^2}{\mu_{ij}}, \tag{6}$$

where n_{ij} is the number for samples with the i^{th} feature value. And

$$\mu_{ij} = \frac{n_{*j} n_{i*}}{n}, \tag{7}$$

where n_{i*} is the number of samples with the the i^{th} value for the particular feature, n_{*j} is the number of samples in class j and n is the number for samples.

4.1.7 Kruskal Wallis

supervised, filter, univariate, feature weighting

Kruskal Wallis test is a non-parametric method [9] that is based on ranks for comparing the population medians among groups. The first step here is to rank all data points across all groups together. And the measurement can be formulated as:

$$K = (N-1) \frac{\sum_{i=1}^{g} n_i (\overline{r}_i - \overline{r})^2}{\sum_{i=1}^{g} \sum_{j=1}^{n_i} (r_{ij} - \overline{r})^2}. \tag{8}$$

In the equation, we have:
N is the total number of observations across all groups
ni is number of observations in group 'i'
r_{ij} is rank of observation 'j' in the group 'i'

[9] An analysis method, in which there is no assumption about the distribution of the data.

$\overline{r}_i = \frac{\sum_{j=1}^{n_i} r_{ij}}{n_i}$

r is the average rank of all the observations i.e. it's the sum of N natural numbers / N,

$r = \frac{N(N+1)}{2N} = \frac{N+1}{2}$

The denominator of K could be further simplified to

$\sum_{i=1}^{g} \sum_{j=1}^{n_i} (r_{ij} - \overline{r})^2 = (r_{11} - \overline{r})^2 + (r_{12} - \overline{r})^2 + ... + (r_{gn_i} - \overline{r})^2$

$= \frac{(N(N+1)(2N+1)}{6} + ((\frac{(N+1)}{2})^2 * N) - (2 * (\frac{N+1}{2}) * (N * \frac{N+1}{2}))$

$= \frac{N(N+1)(N-1)}{12}$

Therefore, we have:

$$K = \frac{12}{N(N+1)} \sum_{i=1}^{g} n_i(\overline{r}_i - \frac{N+1}{2})^2 \quad = \frac{12}{N(N+1)} \sum_{i=1}^{g} n_i(\overline{r}_i)^2 - 3(N+1) \tag{9}$$

From the equation, we can see that the final test metric contains only the squares of the average ranks. An closely related measurement to the Kruskal Wallis is the F-Score, which does not involve rank based testing, and uses the raw measure for testing.

4.1.8 Gini Index

supervised, filter, univariate, feature weighting

Gini index [16] is a measure for quantifying a feature's ability to distinguish between classes. Given C classes, Gini Index of a feature f can be calculated as

$$GiniIndex(f) = 1 - \sum_{i=1}^{C} [p(i|f)]^2, \tag{10}$$

Gini Index can take the maximum value of 0.5 for a binary classification. Smaller the Gini Index, more relevant the feature. Gini Index of each feature is calculated independently and the top k features with the smallest Gini index are selected. It does not eliminate redundant features.

4.1.9 Information Gain

supervised, filter, univariate, feature weighting

Information Gain [6] is a measure of dependence between the feature and the class label. It is one of the most popular feature selection techniques as it is easy to compute and simple to interpret. Information Gain (IG) of a feature X and the class labels Y is calculated as

$$IG(X, Y) = H(X) - H(X|Y). \tag{11}$$

Entropy(H) is a measure of the uncertainty associated with a random variable. $H(X)$ and $H(X|Y)$ is the *entropy* of X and the *entropy* of X after observing Y, respectively.

$$H(X) = -\sum_{i} P(x_i) \log_2(P(x_i)). \tag{12}$$

$$H(X|Y) = -\sum_j P(y_j) \sum_i P(x_i|y_j) \log_2(P(x_i|y_j)). \tag{13}$$

The maximum value of information gain is 1. A feature with a high information gain is relevant. Information gain is evaluated independently for each feature and the features with the top-k values are selected as the relevant features. Information Gain does not eliminate redundant features.

4.1.10 FCBF

supervised, filter, multivariate, feature set

Fast Correlation Based Filter (FCBF) [59] is a filter model feature selection algorithm that measure feature-class and feature-feature correlation. FCBF starts by selecting a set of features S' that is highly correlated to the class with $SU \geq \delta$, where δ is a threshold given by the user. In FCBF, a feature f_i with symmetrical uncertainty $SU_{i,c}$ to the class c will be called predominant *iff* $SU_{i,c} \geq \delta$ and there is no f_j such that $SU_{j,i} \geq \ SU_{i,c} \ \forall f_j \in S'$ where $(j \neq \ i)$. However, if there exists such feature f_j where $SU_{j,i} \geq SU_{i,c}$, then f_j will be called redundant feature to f_i. Then, this set of redundant features will be denoted as S_{P_i}, which will be further split into $S^+_{P_i}$ and $S^-_{P_i}$ where they contain redundant feature to f_i with $SU_{j,c} > SU_{i,c}$ and $SU_{j,c} \leq SU_{i,c}$ respectively. Finally, FCBF applies three heuristics on S_{P_i},$S^+_{P_i}$, and $S^-_{P_i}$ that remove the redundant features and keep the feature that most relevant to the class. The symmetrical uncertainty is defined as:

$$SU(X,Y) = 2[\frac{IG(X|Y)}{H(X)+H(Y)}], \tag{14}$$

where $IG(X|Y)$, $H(X)$ and $H(X|Y)$ are defined in Eq. (11), Eq. (12) and Eq. (13) respectively. This method provides an effective way to handle feature redundancy in feature selection, and its time complexity equals to $O(m \cdot n \cdot \log n)$, where m and n are the number of instances and the number of features respectively.

4.1.11 CFS

supervised, filter, multivariate, feature set

CFS uses a correlation based heuristic to evaluate the worth of features:

$$Merit_S = \frac{k\overline{r_{cf}}}{\sqrt{k+k(k-1)r_{\overline{ff}}}}. \tag{15}$$

Here $Merit_S$ is the heuristic "merit" of a feature subset S containing k features, and we define $\overline{r_{cf}} = \sum_{f_i \in S} \frac{1}{k} \sum(f_i, c)$ is the mean feature class correlation and $\overline{r_{ff}}$ is the average feature inter-correlation. The mean feature-class correlation (numerator) is an indication to how easily a class could be predicted based on the feature. And the average feature-feature inter correlation (denominator) determines correlation between the features which indicates the level of redundancy between them. Feature correlations are estimated based on the information theory that determines the degree of association between features. The amount of information by which the entropy of Y decreases reflects the additional information about Y provided by X which is measured via Information Gain. Since, information gain is usually biased in favor of features with more values, symmetrical uncertainty is used, which is defined in Eq. (14).

CFS explores the search space using the Best First search. It estimates the utility of a feature by considering its predictive ability and the degree of correlation (redundancy) it introduces to the selected feature set. More specifically, CFS calculates feature-class and feature-feature correlations using symmetrical uncertainty and then selects a subset of features using the Best First search with a stopping criterion of five consecutive fully expanded non-improving subsets. Merits of CFS are it does not need to reserve any part of the training data for evaluation purpose and works well on smaller data sets. It selects the maximum relevant feature and avoids the re-introduction of redundancy. But the drawback is that CFS cannot handle problems where the class is numeric.

4.1.12 mRmR

supervised, filter, multivariate, feature set

Minimum-Redundancy-Maximum-Relevance (mRmR) selects features that are mutually far away from each other, while they still have "high" correlation to the classification variable. mRmR is an approximation to maximizing the dependency between the joint distribution of the selected features and the classification variable.

Minimize Redundancy

$$For\ Discrete\ variables : minW_I, W_I = \frac{1}{|s|^2}\sum_{i,j\in S} I(i,j) \tag{16}$$

$$For\ Continuous\ variables : minW_c, W_c = \frac{1}{|s|^2}\sum_{i,j} |c(i,j)| \tag{17}$$

Maximize Relevance

$$For\ Discrete\ variables : maxV_I, V_I = \frac{1}{|S|}\sum_{i\in S} I(h,i) \tag{18}$$

$$For\ Continuous\ variables : maxV_F, V_F = \frac{1}{|S|}\sum_{i\in S} F(i,h) \tag{19}$$

where S is the set of features
$I(i,j)$ is mutual information between features i and j
$c(i,j)$ is the correlation between features i and j
h = target classes
$F(i,h)$ is the F-statistic

Maximum Dependency criterion is defined by $I(S,h)$, that gives the Mutual Information between the selected variables S and the target variable h.

- For two univariate variables x and y:

$$I(x;y) = \iint p(x,y)log\frac{p(x,y)}{p(x)p(y)}dxdy \tag{20}$$

- For multivariate variables S_m and the target h:

$$I(S_m;h) = \iint p(S_m,h)log\frac{p(S_m,h)}{p(S_m)p(h)}dS_mdh \tag{21}$$

4.2 Wrapper Model

Currently, the feature selection package does not contain any algorithm of wrapper model.

4.3 Embedded Model

4.3.1 BLogReg

supervised, embedded, univariate, feature set

BLogReg is an embedded feature selection algorithm that was proposed to eliminate the regularization parameter λ from Sparse Logistic Regression SLogReg, which is proposed in [48]. SLogReg aims to promote the sparsity in the model parameter using a Laplace prior. And a major drawback of SLogReg is that it requires an expensive model selection stage to tune its model parameter. BLogReg [2], on the other hand, use a bayesian approach to overcome this shortcoming, which allows the algorithm to remove λ by maximizing the marginalized likelihood of the sparse logistic regression model. More specifically, assume the original model is defined as:

$$M = E_D + \ \lambda E_\alpha \tag{22}$$

where $\alpha = (\alpha_{\mathbf{0}}, \alpha_{\mathbf{1}}, ..., \alpha_{\mathbf{d}})$ is the parameter of the logistic regression model, d is the dimensionality of the dataset, $E_\alpha = \sum_{i=1}^{d} |\alpha_i|$, and

$$E_D = \sum_{i=1}^{l} \log 1 + \exp(-y_i + f(\mathbf{x}_i)), \tag{23}$$

$f(\mathbf{x}_i)$ is the linear regression given by $\sum_{j=1}^{d} \alpha_j x_{ij} + \alpha_0$. The maximization will has a straight-forward Bayesian interpretation:

$$p(\alpha|D, \lambda) \propto p(D|\alpha)p(\alpha|\lambda) \tag{24}$$

By marginalizing over λ, we can obtain the following optimization criterion for the model:

$$Q = E_D + N \log E_\alpha \tag{25}$$

The above optimization problem can be solved via gradient descent method. The formulation used in BLogReg restricts the algorithm to data of binary class only.

4.3.2 SBMLR

supervised, embedded, multivariate, feature set

To overcome the drawback of BLogReg, [3] proposed a sparse multinomial logistic regression method, SBMLR, as an extension to handle multiclass and multinomial data. SBMLR uses one-vs-all coding scheme to represent the target $\mathbf{Y}$. SBMLR minimize Eq. (22) with respect to the model parameter α as follow:

$$|\tfrac{\partial E_D}{\partial \alpha_{i,j}}| = \lambda \quad \text{if} \quad |\alpha_{i,j}| > 0 \quad \text{and} \quad |\tfrac{\partial E_D}{\partial \alpha_{i,j}}| < \lambda \quad \text{if} \quad |\alpha_{i,j}| = 0.$$

This means if the sensitivity of the log-likelihood with respect to $\alpha_{i,j} < \lambda$, then the parameter will be equal to zero and, therefore, the corresponding feature will be excluded. Notice here, $E_\alpha = \sum_{i=1}^{k} \sum_{j=1}^{d} |\alpha_{i,j}|$ since it handles multi-class data. Finally, to train the model of SBMLR, we simply need the first and the second partial derivatives of E_D *w.r.t.* $\alpha_{i,j}$, which end up to be:

$$|\frac{\partial E_D}{\partial \alpha_{i,j}}| = \sum_{n=1}^{l} y_i^n x_j^n - \sum_{n=1}^{l} t_i^n x_j^n, \tag{26}$$

where y_i^n is the probability of $x^n \in t_i$. Similar to BLogReg, SBMLR does not have model selection stage. Also, it adopts the simplified version of component-wise training algorithm form [48], which does not require Hessian matrix to train the algorithm, therefore is very efficient.

5 Empirical Study

In this section we will empirically evaluate the performance of twelve algorithms using the evaluation framework provided in the repository. Below we provide information about how each algorithm is evaluated, the evaluation criteria, and the experiment setup. The thirteen feature selection algorithms that will be evaluated in this section are listed as below:

Table 2: Feature set algorithms vs. feature weighting algorithms.

Feature Weight Algorithms	Feature Set Algorithms
Chi-Square Fisher Score Gini Index Information Gain Kruskal-Wallis mRMR ReliefF Spectrum T-test	BLogReg CFS FCBF SBMLR

Table 3: The capability of algorithms for handling feature redundancy.

Can Handle Feature Redundancy	Cannot Handle Feature Redundancy
Chi-Square Fisher Score Gini Index Information Gain Kruskal-Wallis ReliefF Spectrum T-test	BLogReg CFS FCBF mRMR SBMLR

1. BLogReg [4]: supervised, embedded, multivariate, feature set
2. CFS [19]: supervised, filter, multivariate, feature set
3. Chi-Square [33]: supervised, filter, univariate, feature weighting
4. FCBF [34], supervised, filter, multivariate, feature set
5. Fisher Score [10]: supervised, filter, univariate, feature weighting
6. Gini Index [16]: supervised, filter, univariate, feature weighting
7. Information Gain [6]: supervised, filter, univariate, feature weighting
8. Kruskal-Wallis [55]: filter, embedded, univariate, feature weighting
9. mRMR [45], supervised, filter, multivariate, feature set
10. ReliefF [32]: supervised, filter, univariate, feature weighting
11. SBMLR [5], supervised, embedded, multivariate, feature set
12. *t*-test [41]: supervised, filter, univariate, feature weighting
13. Spectrum [36]: unsupervised, filter, univariate, feature weighting

As shown in the list, among the thirteen feature selection algorithms, only one of them is unsupervised, which is Spectrum, and the others are all supervised algorithms. The categorizations of the feature selection algorithms in terms of their output types and capability on handling redundant features can be found in Table 2 and Table 3, respectively.

5.1 Datasets

To test the algorithms effectively, 10 benchmark data sets are used to test the performance of the feature selection algorithms. Detailed information of the data sets can be found in Table 4. We carefully selected data sets of different types, e.g. image data, text data and microarray data. These data sets are of different numbers of features, classes and instances. The heterogeneity of the data is important for exposing the strength and weakness of algorithms in different applications.

Table 4: Size and Dimensionality of Datasets

Data Set	Type	Num of Features	Num of Instances	Num of Classes
BASEHOCK	TEXT	4862	1993	2
CLL-SUB-111	Microarray, Bio	11340	111	3
GLA-BRA-180	Microarray, Bio	4915	180	4
GLI-85	Microarray, Bio	85	22283	2
ORL10P	Image, Face	10304	100	10
PCMAC	Text	3289	1943	2
PIX10P	Image, Face	10000	100	10
RELATHE	Text	4322	1427	2
SMK-CAN-187	Microarray, Bio	19993	187	2
TOX-171	Microarray, Bio	5748	171	4

5.2 Experiment Setup

To test performance of the algorithms, the evaluation framework introduced in section 3.2.5 is used. For algorithms of different output types, different evaluation strategies are used:

1. If it is a feature weighting algorithm, features are first ranked according to the weights of the features assigned by the algorithm. Then the quality of the first 5, 10, 15, …, 195, 200 are evaluated respectively.

2. If it is a feature set algorithm, all the selected features will be evaluated together.

To test the quality of the selected features, two metrics are used: accurate, the accuracy achieved by classifiers using selected features; redundancy rate, the redundancy rate contained in the selected features. An ideal feature selection algorithm should select features that results in high accuracy, while containing few redundant features.

5.2.1 Classifier Accuracy

For each data set, we randomly sample 50% instances as the training data and the remaining are used as test data. The process is repeated for 20 times and results in 20 different partitions of the data. The results achieved on each partition are recorded and averaged to obtain the final results. To calculate the classification accuracy, linear SVM, J48, and Naive Bayes are used. The parameters in feature selection algorithms and the SVM classifier are tuned via cross-validation on the training data. In the experiment, the paired Student t-test [41] is used to evaluate the statistical significance of the obtained results and the threshold for rejecting the null hypothesis is set to 0.05.

5.2.2 Redundancy Rate

Assume $\mathbf{F}$ is the set of selected features, and $\mathbf{X_F}$ is the data only containing features in $\mathbf{F}$. The following measurement is used for measuring redundancy rate of $\mathbf{F}$:

$$\mathrm{RED}(F) = \frac{1}{m(m-1)} \sum_{f_i, f_j \in F, i>j} c_{i,j}$$

where $c_{i,j}$ is the correlation between two features, f_i, and f_j. The measurement assesses the averaged correlation among all feature pairs, and a large value indicates that many selected features are strongly correlated and thus redundancy is expected to exist in $\mathbf{F}$.

5.3 Experiment Results

The Excel files containing the experiment results obtained from each feature selection algorithm can be downloaded from the "Algorithm" portion (http://featureselection.asu.edu/software.php) of the website as mentioned in the Section 3.1.1.

6 Conclusions and Discussions

Feature selection has been a research topic with practical significance in many areas such as statistics, pattern recognition, machine learning, and data mining (including Web mining, text mining, image processing, and microarrays analysis). The objectives of feature selection include: building simpler and more comprehensible models, improving data mining performance, and helping prepare, clean, and understand data. In this report we present a feature selection repository, which is designed to collect the most popular algorithms that have been developed in the feature selection research to serve as a platform for facilitating their application, comparison and joint study. we also briefly revisit the key concepts and the components of feature selection, and review the representative feature selection algorithms that have been implemented in the repository.

Feature selection remains and will continue to be an active field that is incessantly rejuvenating itself to answer new challenges. Below are our conjectures about some interesting research topics in feature selection of potential impact in the near future.

Feature selection for ultrahigh dimensional data: selecting features on data sets with millions of features [13]. As high-throughput techniques keep evolving, many contemporary research projects in scientific discovery generate data with ultrahigh dimensionality. For instance, the next-generation sequencing techniques in genetics analysis can generate data with several giga features on one run. Computation inherent in existing methods makes them hard to directly handle data of such high dimensionality, which raises the simultaneous challenges of computational power, statistical accuracy, and algorithmic stability. To address these challenges, researchers need to develop efficient approaches for fast relevance estimation and dimension reduction. Prior knowledge can play an important role in this study, for example, by providing effective ways to partition original feature space to subspaces, which leads to significant reduction on search space and allows the application of highly efficient parallel techniques.

Knowledge oriented sparse learning: fitting sparse learning models via utilizing multiple types of knowledge. This direction extends multi-source feature selection [70]. Sparse learning allows joint model fitting and features selection. Given multiple types of knowledge, researchers need to study how to use knowledge to guide inference for improving learning performance, such as the prediction accuracy, and model interpretability. For instance, in microarray analysis, given gene regulatory network and gene ontology annotation, it is interesting to study how to simultaneously infer with both types of knowledge, for example, via network dynamic analysis or function concordance analysis, to build accurate prediction models based on a compact set of genes. One direct benefit of utilizing existing knowledge in inference is that it can significantly increase the reliability of the relevance estimation [71]. Another benefit of using knowledge is that it may reduce cost by requiring fewer samples for model fitting.

Explanation-based feature selection (EBFS): feature selection via explaining training samples using concepts generalized from existing features and knowledge. In many real-world applications, the same phenomenon might be caused by disparate reasons. For example, in a cancer study, a certain phenotype may be related to mutations of either genes A or gene B in the same functional module M. And both gene A and gene B can cause the defect of M. Existing feature selection algorithm based on checking feature/class correlation may not work in this situation, due to the inconsistent (variable) expression pattern of gene A and gene B across the cancerous samples[10]. The generalization step in EBFS can effectively screen this variation by forming high-level concepts via using the ontology information obtained from annotation databases, such as GO. Another advantage of EBFS is that it can generate sensible explanations showing why the selected features are related. EBFS is related to explanation-based learning (EBL) and relational learning.

Funding

This work is in part sponsored by NSF-0812551

References

[1] R. Butterworth, G. Piatetsky-Shapiro, and D. A. Simovici. R. butterworth, g. piatetsky-shapiro, and d. a. simovici. In *Proceedings of the Fifth IEEE International Conference on Data Mining*, 2005.

[2] G. C. Cawley and N. L. C. Talbot. Gene selection in cancer classification using sparse logistic regression with bayesian regularization. *BIOINFORMATICS*, 22:2348–2355, 2006.

[3] G. C. Cawley, N. L. C. Talbot, and M. Girolami. Sparse multinomial logistic regression via bayesian l1 regularisation. In *In Advances in Neural Information Processing Systems*, 2007.

[4] Gavin C. Cawley and Nicola L. C. Talbot. Gene selection in cancer classification using sparse logistic regression with bayesian regularization. *Bioinformatics*, 22(19):2348–2355, 2006.

[5] Gavin C. Cawley, Nicola L. C. Talbot, and Mark Girolami. Sparse multinomial logistic regression via bayesian l1 regularisation. In *NIPS*, pages 209–216, 2006.

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

[7] C. Ding and H. Peng. Minimum redundancy feature selection from microarray gene expression data. In *Proceedings of the Computational Systems Bioinformatics conference (CSB'03)*, pages 523–529, 2003.

[8] D. Donoho. Formost large underdetermined systems of linear equations, the minimal l1-norm solution is also the sparsest solution. *Comm. Pure Appl. Math.*, 59(7):907–934, 2006.

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

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

[11] Jennifer G. Dy and Carla E. Brodley. Feature selection for unsupervised learning. *J. Mach. Learn. Res.*, 5:845–889, 2004.

[10] For a cancerous sample, either gene A or gene B has abnormal expression, but not both.

[12] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32:407–49, 2004.

[13] Jianqing Fan, Richard Samworth, and Yichao Wu. Ultrahigh dimensional feature selection: Beyond the linear model. *Journal of Machine Learning Research*, 10:2013–2038, 2009.

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

[15] R. Gilad-Bachrach, A. Navot, and N. Tishby. Margin based feature selection - theory and algorithms. In *Twenty-first international conference on Machine learning (ICML)*, 2004.

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

[17] R. Gonzalez and R. Woods. *Digital Image Processing.* Addison-Wesley, 2nd edition, 1993.

[18] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[19] Mark A. Hall and Lloyd A. Smith. Feature selection for machine learning: Comparing a correlation-based filter approach to the wrapper, 1999.

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

[21] X. He, D. Cai, and P. Niyogi. Laplacian score for feature selection. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, Cambridge, MA, 2005. MIT Press.

[22] Inaki Inza, Pedro Larranaga, Rosa Blanco, and Antonio J. Cerrolaza. Filter versus wrapper gene selection approaches in dna microarray domains. *Artificial Intelligence in Medicine*, 31:91–103, 2004.

[23] A. Jakulin and I. Bratko. Testing the significance of attribute interactions. In *Twenty-first international conference on Machine learning (ICML).* ACM Press, 2004.

[24] G.H. John, R. Kohavi, and K. Pfleger. Irrelevant feature and the subset selection problem. In W.W. Cohen and Hirsh H., editors, *Machine Learning: Proceedings of the Eleventh International Conference*, pages 121–129, New Brunswick, N.J., 1994. Rutgers University.

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

[26] I. Kononenko. Estimating attributes : Analysis and extension of RELIEF. In F. Bergadano and L. De Raedt, editors, *Proceedings of the European Conference on Machine Learning, April 6-8*, pages 171–182, Catania, Italy, 1994. Berlin: Springer-Verlag.

[27] Carmen Lai, Marcel J T Reinders, Laura J van't Veer, and Lodewyk F A Wessels. A comparison of univariate and multivariate gene selection techniques for classification of cancer datasets. *BMC Bioinformatics*, 7:235, 2006.

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

[29] Tao Li, Chengliang Zhang, and Mitsunori Ogihara. A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression. *Bioinformatics*, 20(15):2429–2437, Oct 2004.

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

[31] H. Liu and H. Motoda, editors. *Computational Methods of Feature Selection.* Chapman and Hall/CRC Press, 2007.

[32] H. Liu and H. Motoda, editors. *Computational Methods of Feature Selection.* Chapman & Hall, 2008.

[33] H. Liu and R. Setiono. Chi2: Feature selection and discretization of numeric attributes. In J.F. Vassilopoulos, editor, *Proceedings of the Seventh IEEE International Conference on Tools with Artificial Intelligence, November 5-8, 1995*, pages 388–391, Herndon, Virginia, 1995. IEEE Computer Society.

[34] H. Liu and L. Yu. Feature selection for high-dimensional data: A fast correlation-based filter solution. In *Correlation-Based Filter Solution". In Proceedings of The Twentieth International Conference on Machine Leaning (ICML-03)*, pages 856–863, Washington, D.C., 2003. ICM.

[35] H. Liu and L. Yu. Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering*, 17:491–502, 2005.

[36] Huan Liu and Zheng Zhao. Spectral feature selection for supervised and unsupervised learning. *Proceedings of the 24th International Conference on Machine Learning*, 2007.

[37] Huiqing Liu, Jinyan Li, and Limsoon Wong. A comparative study on feature selection and classification methods using gene expression profiles and proteomic patterns. *Genome Inform*, 13:51–60, 2002.

[38] J. Lu, G. Getz, E. A. Miska, E. Alvarez-Saavedra, J. Lamb, D. Peck, A. Sweet-Cordero, B. L. Ebert, R. H. Mak, A. Ferrando, J. R. Downing, T. Jacks, H. R. Horvitz, and T. R. Golub. Microrna expression profiles classify human cancers. *Nature*, 435:834–838, 2005.

[39] Shuangge Ma. Empirical study of supervised gene screening. *BMC Bioinformatics*, 7:537, 2006.

[40] Shuangge Ma and Jian Huang. Penalized feature selection and classification in bioinformatics. *Brief Bioinform*, 9(5):392–403, Sep 2008.

[41] Runger Montgomery and Hubele. *Engineering Statistics.* John Wiley & Sons, Hoboken, NJ, 2007.

[42] Carl Murie, Owen Woody, Anna Lee, and Robert Nadon. Comparison of small n statistical tests of differential expression applied to microarrays. *BMC Bioinformatics*, 10(1):45, Feb 2009.

[43] A. Y. Ng. Feature selection, l1 vs. l2 regularization, and rotational invariance. In *ICML '04: Twenty-first international conference on Machine learning.* ACM Press, 2004.

[44] Chia Huey Ooi, Madhu Chetty, and Shyh Wei Teng. Differential prioritization between relevance and redundancy in correlation-based feature selection techniques for multiclass gene expression data. *BMC Bioinformatics*, 7:320, 2006.

[45] F. Ding C. Peng, H. Long. Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 27(8):1226–1238, 2005.

[46] J. R. Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann, 1993.

[47] Yvan Saeys, Iaki Inza, and Pedro Larraaga. A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517, Oct 2007.

[48] S. K. Shevade and S. S. Keerthi. A simple and efficient algorithm for gene selection using sparse logistic regression. *BIOINFORMATICS*, 19:2246–2253, 2003.

[49] M. R. Sikonja and I. Kononenko. Theoretical and empirical analysis of Relief and ReliefF. *Machine Learning*, 53:23–69, 2003.

[50] L. Song, A. Smola, A. Gretton, K. Borgwardt, and J. Bedo. Supervised feature selection via dependence estimation. In *International Conference on Machine Learning*, 2007.

[51] Y. Sun, C. F. Babbs, and E. J. Delp. A comparison of feature selection methods for the detection of breast cancers in mammograms: adaptive sequential floating search vs. genetic algorithm. *Conf Proc IEEE Eng Med Biol Soc*, 6:6532–6535, 2005.

[52] Michael D Swartz, Robert K Yu, and Sanjay Shete. Finding factors influencing risk: Comparing bayesian stochastic search and standard variable selection methods applied to logistic regression models of cases and controls. *Stat Med*, 27(29):6158–6174, Dec 2008.

[53] 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.

[54] U. von Luxburg. A tutorial on spectral clustering. Technical report, Max Planck Institute for Biological Cybernetics, 2007.

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

[56] J. Weston, A. Elisseff, B. Schoelkopf, and M. Tipping. Use of the zero norm with linear models and kernel methods. *Journal of Machine Learning Research*, 3:1439–1461, 2003.

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

[58] Zenglin Xu, Rong Jin, Jieping Ye, Michael R. Lyu, and Irwin King. Discriminative semi-supervised feature selection via manifold regularization. In *IJCAI' 09: Proceedings of the 21th International Joint Conference on Artificial Intelligence*, 2009.

[59] L. Yu and H. Liu. Feature selection for high-dimensional data: A fast correlation-based filter solution. In T. Fawcett and N. Mishra, editors, *Proceedings of the 20th International Conference on Machine Learning (ICML-03), August 21-24, 2003*, pages 856–863, Washington, D.C., 2003. Morgan Kaufmann.

[60] Lei Yu, Chris Ding, and Steven Loscalzo. Stable feature selection via dense feature groups. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2008.

[61] Lei Yu, Jessica L Rennert, Huan Liu, and Michael E Berens. Exploiting statistical redundancy in expression microarray data to foster biological relevancy. Technical report, Department of Computer Science and Engineering, Arizona State Univeristy, 2005. TR-05-005.

[62] Li-Juan Zhang, Zhou-Jun Li, Huo-Wang Chen, and Jian Wen. Minimum redundancy gene selection based on grey relational analysis. In *Proc. Sixth IEEE International Conference on Data Mining Workshops ICDM Workshops 2006*, pages 120–124, Dec. 2006.

[63] Xiang Zhang, Fei Zou, and Wei Wang. Fastchi: an efficient algorithm for analyzing gene-gene interactions. In *Pacific Symposium on Biocomputing (PSB)*, 2009.

[64] Yi Zhang, Chris Ding, and Tao Li. Gene selection algorithm by combining relieff and mrmr. *BMC Genomics*, 9:S27, 2008.

[65] Z. Zhao and H. Liu. Searching for interacting features. In *International Joint Conference on AI (IJCAI)*, 2007.

[66] Z. Zhao and H. Liu. Spectral feature selection for supervised and unsupervised learning. In *International Conference on Machine Learning (ICML)*, 2007.

[67] Z. Zhao, J. Wang, H. Liu, J. Ye, and Y. Chang. Identifying biologically relevant genes via multiple heterogeneous data sources. In *The Fourteenth ACM SIGKDD International Conference On Knowledge Discovery and Data Mining (SIGKDD 2008)*, 2008.

[68] Zheng Zhao and Huan Liu. Semi-supervised feature selection via spectral analysis. In *Proceedings of SIAM International Conference on Data Mining (SDM)*, 2007.

[69] Zheng Zhao and Huan Liu. Multi-source feature selection via geometry-dependent covariance analysis. *JMLR Workshop and Conference Proceedings Volume 4: New challenges for feature selection in data mining and knowledge discovery*, 4:36–47, 2008.

[70] Zheng Zhao and Huan Liu. Multi-source feature selection via geometry-dependent covariance analysis. In *Journal of Machine Learning Research, Workshop and Conference Proceedings Volume 4: New challenges for feature selection in data mining and knowledge discovery*, volume 4, pages 36–47, 2008.

[71] Zheng Zhao, Jiangxin Wang, Shashvata Sharma, Nitin Agarwal, Huan Liu, and Yung Chang. An integrative approach to identifying biologically relevant genes. In *Proceedings of SIAM International Conference on Data Mining (SDM)*, 2010.