University of Cagliari

A FRAMEWORK FOR FEATURE SELECTION IN HIGH-DIMENSIONAL DOMAINS

by

Laura Maria Cannas

A thesis submitted for the degree of

Philosophiæ Doctor

PhD School in Mathematics and Computer Science

Supervised by

Prof. Nicoletta Dessì

INF/01

2011 – 2012

To Malala Yousafzai
and to those who work to make
education accessible to everyone.

Abstract

The introduction of DNA microarray technology has lead to enormous impact in cancer research, allowing researchers to analyze expression of thousands of genes in concert and relate gene expression patterns to clinical phenotypes. At the same time, machine learning methods have become one of the dominant approaches in an effort to identify cancer gene signatures, which could increase the accuracy of cancer diagnosis and prognosis. The central challenges is to identify the group of features (i.e. the biomarker) which take part in the same biological process or are regulated by the same mechanism, while minimizing the biomarker size, as it is known that few gene expression signatures are most accurate for phenotype discrimination.

To account for these competing concerns, previous studies have proposed different methods for selecting a single subset of features that can be used as an accurate biomarker, capable of differentiating cancer from normal tissues, predicting outcome, detecting recurrence, and monitoring response to cancer treatment. The aim of this thesis is to propose a novel approach that pursues the concept of finding many potential predictive biomarkers. It is motivated from the biological assumption that, given the large numbers of different relationships which are possible between genes, it is highly possible to combine genes in many ways to produce signatures with similar predictive power. An intriguing advantage of our approach is that it increases the statistical power to capture more reliable and consistent biomarkers while a single predictor may not necessarily provide important clues as to biological differences of interest.

Specifically, this thesis presents a framework for feature selection that is based upon a genetic algorithm, a well known approach recently proposed for feature selection. To mitigate the high computationally cost usually required by this algorithm, the framework structures the feature selection process into a multi-step approach which combines different categories of data mining methods. Starting from a ranking process performed at the first step, the following steps detail a wrapper approach where a genetic algorithm is coupled with a classifier to explore different feature subspaces looking for optimal biomarkers. The thesis presents in detail the framework and its validation on popular datasets which are usually considered as benchmark by the research community. The competitive classification power of the framework has been carefully evaluated and empirically confirms the benefits of its adoption. As well, experimental results obtained by the proposed framework are comparable to those obtained by analogous literature proposals. Finally, the thesis contributes with additional experiments which confirm the framework applicability to the categorization of the subject matter of documents.

Contents

List of Figures

List of Tables

Acronyms

CHI	Chi Squared
CHI+GA/NBM	Framework instance that employs CHI as filter in layer 1 and NBM as classifier in layer 3
IG	Information Gain
IG+GA/NBM	Framework instance that employs IG as filter in layer 1 and NBM as classifier in layer 3
GA	Genetic Algorithm
GA/K-NN	Framework instance that employs K-NN as classifier in layer 3
GA/SVM	Framework instance that employs SVM as classifier in layer 3
K-NN	Nearest Neighbor Classifier
NBM	Naïve Bayes Multinomial Classifier
SVM	Support Vector Machine Classifier
TC	Text Categorization

1 Introduction

In the last decades we have witnessed numerous new technologies emerge and establish, as such as the Internet, database, GPS and mobile devices, and DNA microarray. These new products and methods have caused a rapid and sensible increment in the volume of data used and collected in a vast range of applications including search engines, geomapping, genomic and proteomics analysis, image retrieval, information retrieval, and text categorization. In this scenario, it appears evident that there is a growing need not only for storing, organizing and delivering the high amount of data but also for extracting valuable information from them via the automatic analysis of their content.

From the 50's, the extraction of knowledge from data has been the goal of data mining [1] in order to discover interesting and previously unknown patterns in datasets [2][3]. However, the recent proliferation of large data, with hundreds to thousands of features, within many domains poses to data mining unprecedented challenges [4].

Among the above domains is bioinformatics and, specifically, the microarray technology which allows to quantify the expression for thousands of genes simultaneously by measuring the hybridization from a tissue of interest to probes on a small glass or plastic slide. A typical DNA microarray consists of thousands of ordered sets of DNA fragments on a glass, filter, or silicon wafer. As one application of this technology, gene expression profiles can be generated from a collection of cancerous and non-cancerous tumor tissue samples and then stored in a database. The characteristics of these data include a very high number of attributes or features (i.e. gene expression levels), in general much larger than the number of examples.

Data mining methods have become one of the dominant approaches in microarray analysis. As a topic under the field of supervised learning, classifiers are developed and trained to label new cases according to a set of features derived from the data.

However, using too many features in the classification algorithm can be problematic, particularly if there are irrelevant features. This can lead to overfitting, in which noise or irrelevant features may exert undue influence on the classification decisions because of the modest size of the training data. Additionally, there may be redundancies in the extracted features.

Despite the early success of data mining methods, the presence of a significant number of irrelevant features – here genes in the profile that are unrelated to the disease status of the tissue – makes such analysis somewhat prone to the curse of dimensionality. Intuitively, overcoming the curse of dimensionality requires that we build classifiers relying on information exclusively from the genes in the profile that are truly relevant to the disease status of the tissue.

This problem of identifying the features most relevant to the classification task is known as feature selection: it provides a fundamental step in the analysis of such type of data. By selecting only a subset of attributes, the prediction accuracy can possibly improve and more insight in the nature of the prediction problem can be gained by identifying only the genes that are relevant to the prediction of the disease diagnosis. Moreover, the identification of a small set of genes that is indeed capable of providing complete discriminatory information, results in inexpensive diagnostic assays for only a few genes which might be developed and be widely deployed in clinical settings

As such, it is highly desirable to discard irrelevant features prior to learning, especially when the number of available features significantly outnumbers the number of examples, as is the usual case in microarray data. On the one hand, feature selection methods offer great potential to identify a small set of diagnostically relevant genes which may provide important insights into the mechanisms responsible for the disease itself. On the other hand, the curse of dimensionality makes feature selection a major bottleneck of microarrays data analysis.

It is important to notice that a gene expression has two important roles : it is both the biological mechanism underlying the disease (this means that a gene expression has biological relevance) and a relevant element in building good diagnostic prediction algorithms (i.e. a gene expression has diagnostic relevance).

Unfortunately, not all biologically relevant genes may need to be used when building accurate diagnostic classifiers while high correlation between disease status and gene expression level does not necessarily imply that the expression of that particular gene has somehow caused the disease. In other words, being diagnostic relevance neither a necessary nor a sufficient condition for biological relevance, genes selected for their diagnostic relevance should be validated for biological relevance by follow-up studies of the literature or experimental analysis. Moreover, from the biological assumption there is evidence that:

(i) there is not a unique set of genes responsible for a disease and

(ii) these sets are composed by a limited number of genes (usually no more than 4 or 5).

As a consequence, indentifying multiple biomarkers is useful to discover correlations among genes and to permit different test possibilities in the diagnostic phase.

In the current literature, studies have proposed several methods for selecting a subset of features that can be used as an accurate biomarker for diagnostic relevance. Related algorithms are commonly deterministic and attempt to find a unique biomarker, the one that leads to obtain maximum accuracy when used to predict a disease. Beside their ability to select many biomarkers, stochastic approaches, as genetic algorithms, are rarely employed because of the high computational cost deriving from applying these procedures within a large search space.

Taking a paradigm shift from current literature, this thesis proposes a novel approach that pursues the concept of finding many potential predictive biomarkers to account for both the competing concerns of diagnostic and biologic relevance. As previous mentioned, it is motivated from the biological assumption that , given the large numbers of different relationships which are possible between genes, it is highly possible to combine genes in many ways to produce signatures with similar predictive power. An intriguing advantage of our approach is that it increases the statistical power to capture more reliable and consistent biomarkers (in that supporting biological relevance) while a single predictor may not necessarily provide important clues as to biological differences of interest.

Instead of proposing a deterministic approach for selecting the best predictor, i.e. the group genes which results in the best classification accuracy, we propose and experiment the performances of a genetic algorithm and turn the feature selection problem into a problem of finding optimal subspaces for finding several best predictors.

To mitigate the high computationally cost usually required by genetic algorithms, the thesis proposes a framework which structures the feature selection process into a multi-step approach. Starting from a ranking process performed at the first step, the following steps detail a wrapper approach where the genetic algorithm is coupled with a classifier to explore different feature subspaces looking for optimal biomarkers. Each step performs a well defined task towards the dimensionality reduction of the search space.

The framework can be considered a quite general approach since it defines at each step the class of data mining methods (instead of a specific method) which can be applied. Accordingly, different categories of data mining methods can be combined. This means that for each class of methods (i.e. filter methods, classification algorithms, etc.) there is not any constraint on the subsequent implementation. Hence, the proposed approach can be classified among the hybrid feature selection techniques, as it combines multiple classes of methods within a single framework.

For the framework, two important design criteria were established. First, it should have high classification performance on high dimensional data or small sample size problems. Second, the classifier should use as few dimensions as possible in achieving its performance.

The thesis presents in detail the different phases that compose the framework and its validation on multiple public microarray datasets considered as benchmarks by the scientific community. Corresponding results are compared with those obtained on the same datasets by various feature selection techniques proposed in literature and show the validity of the framework.

Our work aims not only to establish a new model for the identification of robust gene expression signatures from accumulated microarray data, but also to demonstrate how the great wealth of microarray data can be exploited to present some innovative ideas about measuring the influence of single genes for a biological interpretation of the prediction models. These models will be increasingly useful as more and more microarray data is generated and becomes publicly available in near future. With the inclusion of more samples, cancer gene signatures will be continuously refined and consensus signatures will finally be reached.

This thesis contributes with additional experiments that pursue the idea of validating the proposed framework in others high dimensional application domains. Specifically, we present experiments which aim to confirm the benefits of the proposed framework for the categorization of the subject matter of documents. The challenge here is the development and adaptation of class prediction algorithms that reliably work in determining what the document is about. In detail, automatic text categorization is the task of assigning one or more pre-specified categories to an electronic document, based on its content. Standard categorization approaches utilize statistical or data mining approach to perform the task. The application of such approaches requires a transformation of the document text into a vector of terms (i.e. nouns, adjectives etc.). As a tissue is described by microarray datasets, in the same vein a text, is represented by a vector of elements (features), each of which is the frequency of a term. Our aims is to experiment the framework ability to analyze a large amount of documents and extract clusters of words which best categorize a class of documents. With this purpose, the framework analyzes a corpus of text documents, with the aim of finding topics within each document (term selection) and hence classifying documents in one or more topic categories (text categorization). Conducted on a large collection of news articles, the experiments show that our framework has achieved a satisfactory overall accuracy and results compare well with both traditional and recent text categorization methods proposed in literature.

The remainder of this dissertation is organized as follows. Chapter 2 first introduces the problem of microarray data analysis, gives an overview of the typical feature selection techniques used in this domain and finally describes genetic algorithms and explain their usefulness in feature selection procedures. In chapter 3 we present our framework and discuss in detail its multi-layer structure. Then, we remark the distinctive characteristics of the framework and highlight the related advantages. Chapter 4 describes the problem of gene selection and reports the wide experimental analysis made applying the framework on four DNA-microarray datasets. Chapter 5 introduces the problem of text analysis and considered the preliminary study we made in this domain. In chapter 6 related work cited in literature are presented. Chapter 7 concludes our work and gives future directions.

Analysis and results described in this thesis have been presented in:

- Laura Maria Cannas, Nicoletta Dessì, Barbara Pes: Tuning Evolutionary Algorithms in High Dimensional Classification Problems (Extended Abstract). SEBD 2010: 142-149;

- Laura Maria Cannas, Nicoletta Dessì, Barbara Pes: A Filter-Based Evolutionary Approach for Selecting Features in High-Dimensional Micro-array Data. Intelligent Information Processing 2010: 297-307;

- Laura Maria Cannas, Nicoletta Dessì, Barbara Pes: Knowledge Discovery in Gene Expression Data via Evolutionary Algorithms. DEXA Workshops 2011 (BIOKDD'11): 402-406;

- Laura Maria Cannas, Nicoletta Dessì, Barbara Pes: A Hybrid Model to Favor the Selection of High Quality Features in High Dimensional Domains. IDEAL 2011: 228-235;

- Laura Maria Cannas, Nicoletta Dessì, Stefania Dessì: A Model for Term Selection in Text Categorization Problems. DEXA Workshops 2012 (TIR'12): 169-173;

- Laura Maria Cannas, Nicoletta Dessì, Barbara Pes: Balancing effectiveness and representation level in feature selection. Submitted to Knowledge and Information Systems on 12/02/2012 – under revision;

- Laura Maria Cannas, Nicoletta Dessì, Barbara Pes: Assessing Similarity of Feature Selection Techniques in High-Dimensional Domains. Submitted to Pattern Recognition Letters on 23/06/2012 – under revision.

2 Feature Selection in Data Mining

In recent years, the development of new technologies capable of monitoring genome function has resulted in an explosion in the rate of acquisition of biomedical data and in an increasingly stable representations of genome produced from individual sample. Being capable of providing scientists with global and functional profiles of gene expression of thousands of genes simultaneously, microarrays have great potential and have demonstrated their value in many important applications in bioscience, such as discovering novel genes and hidden patterns in expression profiles, decoding pathways involved in tumor genesis, identifying potential diagnostic markers or therapeutic targets, and thus opening possibility for accurate cancer classification.

Given the presence of large quantities of high dimensional data (which may come in a variety of noisy forms) and the lack of a comprehensive understanding at the molecular level, mining microarray data presents significant challenges to data mining communities. This section gives an overview of these challenges and describes the feature selection techniques commonly used in this domain. Finally, it presents genetic algorithms and explain their usefulness in feature selection procedures.

2.1 Challenges Faced in Microarray Data Analysis

Advances in DNA microarrays allow us for the first time to obtain a "global" view of the cell and routinely investigate the biological molecular state of a cell measuring the simultaneous expression of tens of thousands of genes [5]. Gene expression data generated using microarrays is generally employed to identify genes that are differentially expressed under various experimental conditions, to identify groups of genes with similar expression profiles across various experimental conditions (co-expressed genes) and, also, to classify the biologic sample based on the pattern of expression of all or a subset of genes on the microarray. Differentially expressed genes and groups of co-expressed genes can be used to hypothesize which pathways are involved in a particular biologic process. Additionally, clusters of co-expressed genes can be used to guess the functional relationship of a clustered gene and as a starting point to analyze regulatory mechanisms underlying the co-expression. In the context of tumor classification, gene expression profiles have been used as biomarkers to define tumors as well as different sub-classes of tumors [6-10].

Different types of microarray use different technologies for measuring mRNA expression levels; detailed description of these technologies is beyond the scope of this thesis. Here we will focus on the analysis of expression data from microarray devices. Many current efforts are being directed in this direction. In a few cases the results of microarray analysis have found their way into many important applications in medicine and biology.

The main types of data analysis needed to for biomedical applications include [11]:

- Gene Selection – from a data mining prospective, this is a problem of feature selection where the goal is to find the genes most strongly related to a particular disease;

- Classification – classifying diseases or predicting outcomes based on gene expression patterns, and perhaps even identifying the best treatment for given genetic signature;

- Clustering – finding new biological classes or refining existing ones, such as groups of co-expressed genes or gene networks and genes interactions.

The above three topics present a number of challenges that need to be addressed before new knowledge about gene expression can be revealed. Some challenges regard the technology used in microarray experiments, such as the occurrence of bias caused by differences in the choice of the reagents or in the platform standards, or also the presence of mislabelled data or questioned tissues just to mention a few [12]. Other challenges directly concern the data mining analysis and are observed only in this domain [11]. These unique challenges include the following aspects.

Analyzing data with high dimensionality and small sample size. Typical data mining applications in economic, financial, and scientific domains have a large number of samples (thousands and sometimes millions), while the number of features is much smaller (at most several hundred). In contrast, a typical microarray dataset may have only a small number of samples (less than a hundred), while the number of features, corresponding to the number of genes, is typically in thousands. Given the difficulty of collecting microarray records, the number of samples is likely to remain small in many interesting cases.

Curse of dimensionality. This problem is related to the scarce number of samples in relation to number of features [13]. In fact, the sample size is often too low to permit the use of separate training and test sets as usual in machine learning. Therefore, in microarray data analysis it is felt the necessity to use re-sampling methods instead (like k-fold cross validation or leave-one-out cross validation).

High likelihood of false positives due to chance. Having so many features relative to so few samples creates a high likelihood of finding “false positives” that are due to chance, both in finding differentially expressed genes and in building predictive models. We need especially robust methods to validate the models and assess their likelihood.

Lack of absolute ground truth. Attempts to find invariant or differential molecular behaviour relevant to a given biological problem are also limited by the fact that in many cases little is known about the normal biological variation expected in a given tissue or biological state.

Assessing classifier certainty. Up to now, numerous and different methods have been applied to the problem of analyse gene expression measurements from microarrays. Just to mention a few, we cite: Support Vector Machines (SVMs) [14][15], Naïve Bayesian Classifiers [16], Artificial Neural Networks (ANNs) [17], and decision trees [18]. Some of these studies indicate that classification accuracy can be improved by reducing the number of features used as input to the machine learning method [17][19]. The reason for this is most likely that the high level of correlation between the expression levels of many genes in the cell makes much of the information from one microarray redundant. The relevance of good feature selection methods in this domain has been extensively discussed by [12][20][21][22].

Abundance of biological knowledge and difficulty of integrating it. Results from micro array data analysis are likely to be useful but only if they can be put in context and followed up with more detailed studies for example by a biologist or a clinical researcher. Often this follow up and interpretation is not done carefully enough because of the additional significant research involvement, the lack of domain expertise or proper collaborators, or due to the limitations of the computational analysis itself.

In considering this last point, it is necessary to clearly distinguish between the relevance of a gene to the biological mechanism underlying the disease and its relevance to building good diagnostic prediction algorithms [23]. On the one hand, not all biologically relevant genes may need to be used when building accurate diagnostic classifiers; on the other hand, high correlation between disease status and gene expression level does not necessarily imply that the expression of that particular gene has somehow caused the disease. So diagnostic relevance is neither a necessary nor a sufficient condition for biological relevance. Consequently, genes selected for their diagnostic relevance should be validated for biological relevance by follow-up studies of the literature or experimental analysis. Nevertheless, the fact remains that good feature selection methods can often serve as excellent guides in identifying small subsets of genes for further investigation.

Complexities of gene interaction. Genes are not independent, but the structure of their correlation is hard to estimate. Most of the current gene selection methods in use evaluate each gene separately and ignore its possible correlations. From a biological perspective, however, we know that groups of genes working together as pathway components and reflecting the states of the cell are the real atomic units, by which we might be more likely to predict the character or type of a particular sample and its corresponding biological state.

Concluding, extracting knowledge from microarray data is an important and actual scientific problem and a desirable goal of the next generation of pattern recognition and data mining methods should be to provide a more integrated and unified framework that not only builds models but also promote the interpretation and understanding of them by domain experts.

2.2 Feature Selection Techniques in Microarray Data Analysis

There are three principal reasons for our interest in feature selection [24]:

1. We can avoid overfitting and improve the generalization performance: identifying only the genes that are highly informative could enhance the accuracy of classification model to the prediction of the disease diagnosis. This effect is attributable to the overcoming of the curse of dimensionality alluded to in the previous section.

2. We can provide a faster and more cost-effective models: if it is possible to identify a small set of genes that is indeed capable of providing complete discriminatory information, inexpensive diagnostic assays for only a few genes might be developed and be widely deployed in clinical settings.

3. We can gain a deeper insight into the underlying processes that generated the data: knowing a small set of diagnostically relevant genes may provide important benefits in understanding the mechanisms responsible for the disease itself.

Moreover, differently from transformation-based reduction techniques (such as Principal Component Analysis), feature selection techniques belong to the class of selection reduction techniques, that are become the main preference in many bioinformatics applications, especially microarray data analysis, since it offers the advantage of interpretability by a domain expert.

In the context of classification, feature selection techniques can be globally organized into three categories, depending on how they combine the feature selection search with the construction of the classification model: filter method, wrapper method and embedded method [12][20].

Filter methods rank each feature according to some univariate metric that tests the discriminative power of the feature with regard to the class labels of samples. Obtained the ranked list, only the highest ranking features are used while the remaining low ranking features are eliminated [25]. Filters rely only on intrinsic characteristics of the training data to select some features without involving any learning algorithm. Filter methods have been widely utilized in microarray data analysis: they provide very easy way to calculate, can simply scale to large-scale microarray datasets since it only have a short running time, and they require a single application to provide output. Moreover, their output is intuitive and easy to understand even by biologists and domain experts in general. However, gene ranking based on univariate methods has some drawbacks. Firstly, given the ranked list of feature, filters basically separate the informative gene by choosing a threshold value. However, there is little theoretical support for determining how many genes should be chosen for classification, and the threshold used is somewhat arbitrary [26]. Another disadvantage is that the selected genes are most probably redundant: this means that top-ranked genes may carry similar discriminative knowledge towards the defined class.

Differently from filter techniques that select genes independently, wrapper methods embed a gene selection method within a classification algorithm. In the wrapper methods [27] a search is conducted in the space of genes, evaluating the goodness of each found gene subset by the estimation of the accuracy percentage of the specific classifier to be used, training the classifier only with the found genes. Wrappers are able to conduct their search evaluating also interactions and correlations among genes and, thanks to the cooperation between selection procedure and classification model, it is claimed that the wrapper approaches obtain better predictive accuracy estimates than filters [28-30]. A common drawback of wrapper methods is that they have a higher risk of over-fitting than filter techniques as they are customized for a particular classifier. Moreover they are very computationally intensive [31], particularly if the original gene set is large. Because of this, wrappers are not frequently used in microarray data analysis [29][30].

In view of the drawbacks of the filter and wrapper approaches, hybrid filter-wrapper models have been proposed that take advantage of the simplicity of the filter approach for initial gene screening and then make use of the wrapper approach to optimize classification accuracy in final gene selection [20][32]. In the hybrid model, a filter is first used to screen out a majority of (irrelevant) genes from the original set to give a filtered subset of a relatively small size (no more than few hundred from an original set of several thousands). Then, the wrapper is applied to select genes from the filtered subset to optimize the training accuracy. As the filter efficiently reduces the size of the gene set by an order of magnitude or more, the computations of the subsequent wrapper become acceptable.

With regard to the third class of feature selection methods, embedded techniques differ in that the search for an optimal subset of features is built *into* the classifier construction. Analogous to wrapper approaches, embedded approaches are thus specific to a given learning algorithm. Embedded methods have the advantage that they include the interaction with the classification model, while at the same time being far less computationally intensive than wrapper methods [20].

2.3 Genetic Algorithms

A genetic algorithm (GA) is a search and optimization tool inspired by the mechanisms of evolution [33]. These bio-inspired mechanisms have shown to be useful in numerous search and optimization domains. Given a random starting set of solutions (the initial population), genetic algorithms use principles of evolution such as reproduction, selection, crossover, and mutation (collectively known as *genetic operators*) to explore the search space and discover better solutions to a problem.

Each solution (each individual in the population) is evaluated according to a specified *fitness function* that is defined to measure the goodness of a solution. According to the principle of "survival of the fittest", solutions with higher fitness are more likely to be selected to come into play in the evolutionary process. Since the algorithm is iterative, generic operators act upon the population many times, moving the algorithm from one generation to the next, until a pre-defined number of generations or until the optimal solution is found.

Genetic algorithms are widely applied in engineering and scientific disciplines. Generally, this is due to the fact that they can be readily adapted to new problems, they are efficient with respect to other search algorithms, and also they are less prone to descending into local minima/maxima. A problem with many standard search algorithms, such as hill-climbing, is that they often find solutions in the search space which are locally – but not globally – optimum when the space is not smooth (i.e. in most real-world problems).

Genetic algorithms, due to their stochastic and population-based nature, are able to avoid this behaviour for the most part. They have therefore found favour in a large number of domains where traditional techniques would require too much computation to produce an optimal solution and where a near-optimal one will suffice.

Many studies have been published to demonstrate the efficiency of genetic algorithms in many application domains. The most influential factors in the quality of the solutions found by these algorithms are:

1. a suitable definition of the search space of the potential solutions and
2. a proper configuration of the algorithm, in terms of fitness function, genetic operators and parameters settings.

Being genetic algorithms sensitive to small changes of the initial parameter set (i.e. population size, number of generations, operator probabilities, and so on), the choice of these values is a critical aspect when the algorithm is devoted to the analysis of very large datasets.

Moreover, since GA is a randomized algorithms, it can produce different solutions among different trials for same parameter set and the same initial population.

2.4 Genetic Algorithms for Feature Selection

In numerous domains, the high dimensionality of data makes impractical the use of complete searches in wrapper procedure for feature selection problems. This limit is observed also in the analysis of microarray data.

Greedy searches, such as sequential forward selection and sequential backward selection, are often used to generate candidate gene subsets. These greedy searches are simple and fast but may result in local optimal solutions. To achieve global optimal solutions, in recent years some wrapper models replaced greedy search with heuristic search, for example introducing the use of genetic algorithms.

Unlike greedy search algorithms, GAs can avoid local optima and provide multi criteria optimization functions. An advantage of a GA is that it tends to retain good features of solutions that are inherited by successive generations. Moreover, GAs have shown their effectiveness in exploring feature spaces of high dimensionality [34][35].

3 The Proposed Framework

In this chapter, the proposed framework is presented and discussed in details. First of all, we describe its structure, focusing on its multi-layer shape and analyzing each single layer at a time. Then, we remark the distinctive characteristics of the framework and highlight the related advantages.

3.1 Framework Presentation

As previously explained in chapter 1, the framework presented in this thesis aims to reduce the number of features of a high-dimensional dataset in order to improve mining performances such as predictive accuracy and result comprehensibility. This reduction process can be categorized as a feature selection technique, as only the features considered informative for the further analysis will be selected while the others will be discarded.

This process takes as input the dataset in the shape of a matrix: rows represent the recorded samples while the columns are the features describing the data. As output, the framework returns a set of informative features that can be used for predictive modelling as well as knowledge on the analyzed domain.

Our proposal divides the feature selection process in four subsequent steps. Therefore, the framework consists of four layers, ordered by increasing complexity of the learning process. Starting from the dataset initially provided, each layer organizes a class of methods to transform data provided by the previous layer into some new form of information for future use by the next layer, until finally returning the features ultimately selected.

In details, at the first layer each single feature is weighted based on its relevance to the target class. This way a ranked list is obtained, where features appear in descending order of importance. At the second layer, a filter is repeatedly applied to the ranked list to provide different feature subsets, namely feature subspaces. Features are included in these subspaces in an incremental way based on their ranking position. At the next layer, the feature subspaces are explored by a wrapper that uses a genetic algorithm as a search strategy. Finally, at the last layer, potential useful features are evaluated to extract knowledge about the application domain.

Figure 3.1 gives an overview of the proposed framework. In particular, to better show the continuity of the learning process between two subsequent layers and remark how the output of a layer becomes the input for the following one, rectangles have been used to represent data (inputs and outputs) while ovals represent the tasks executed in the framework. Each single layer of the framework is further detailed in what follows.

3.1.1 Layer 1: Scoring Features

Starting from the input data matrix, the first task is intended to score individual features according to their discriminative power, i.e. their capacity of separating the classes. To this end, a ranking criterion is applied to the data matrix obtaining, for each individual feature, a measure that describe how strong it is correlated with the target class.

A feature with a high ranking value indicates higher discrimination of this feature compared to other categories and means that the feature contains information potentially useful for classification. Based on the ranking value they obtain, features are then returned in an ordered list where they appear in descending order of relevance.

3.1.2 Layer 2: Defining Feature Subspaces

At the layer 2, the ranked list is used to define different feature subspaces. This is performed by a filter approach: starting from the first P features of the ordered list, nested subsets of increasing size are constructed by progressively adding features (less and less correlated with the target). It results in a sequence of R feature subspaces (FSs):

$$FS_1 \subset FS_2 \subset \ldots \subset FS_R$$

where the first subspace (FS_1) includes the first P top-ranked features, the second subspace (FS_2) includes the first 2*P features, etc. Denoting with N the dimension of a generic feature subspace, one obtains:

$$N = i*P, \quad i = 1, 2, \ldots, R .$$

Although containing a subset of potentially informative features, each single subspace FS_i, i = 1, 2, …, R, cannot be considered a good predictor because its features may be mutually correlated. As such, additional work is needed for refining the above subsets by removing redundant features in order to devise more accurate and small-sized predictors.

3.1.3 Layer 3: Selecting Feature Subsets

The third layer involves a wrapper approach aimed at refining the feature subspaces to discover optimal predictors in each of these. As search procedure within the wrapper, we employ a genetic algorithm (GA) that explores the subspace looking for solutions, i.e. features subsets, that are optimal in terms of a given fitness function.

Detailing the operation of a GA, a population of individuals is randomly initialized from each single subspace. Each individual represents a possible solutions, i.e. a feature subset, and is encoded by a N-bit binary vector where N is the subspace size. If the ith bit has value '1' it means that the ith feature is selected in the subset, whereas if that bit has value '0' the feature is not selected. Any number of features smaller than N can be selected, meaning that larger individual can be expressed in principle from larger feature subspaces, i.e. larger predictors can be extracted from larger feature subspaces.

Individuals are first evaluated by a particular type of objective function called fitness function. While different formulations could be incorporated in our framework, we evaluate the fitness of a given individual as the predictive performance (i.e. accuracy) of a classifier learnt on it. Thus, an individual is as much strong as it provides a high classification accuracy.

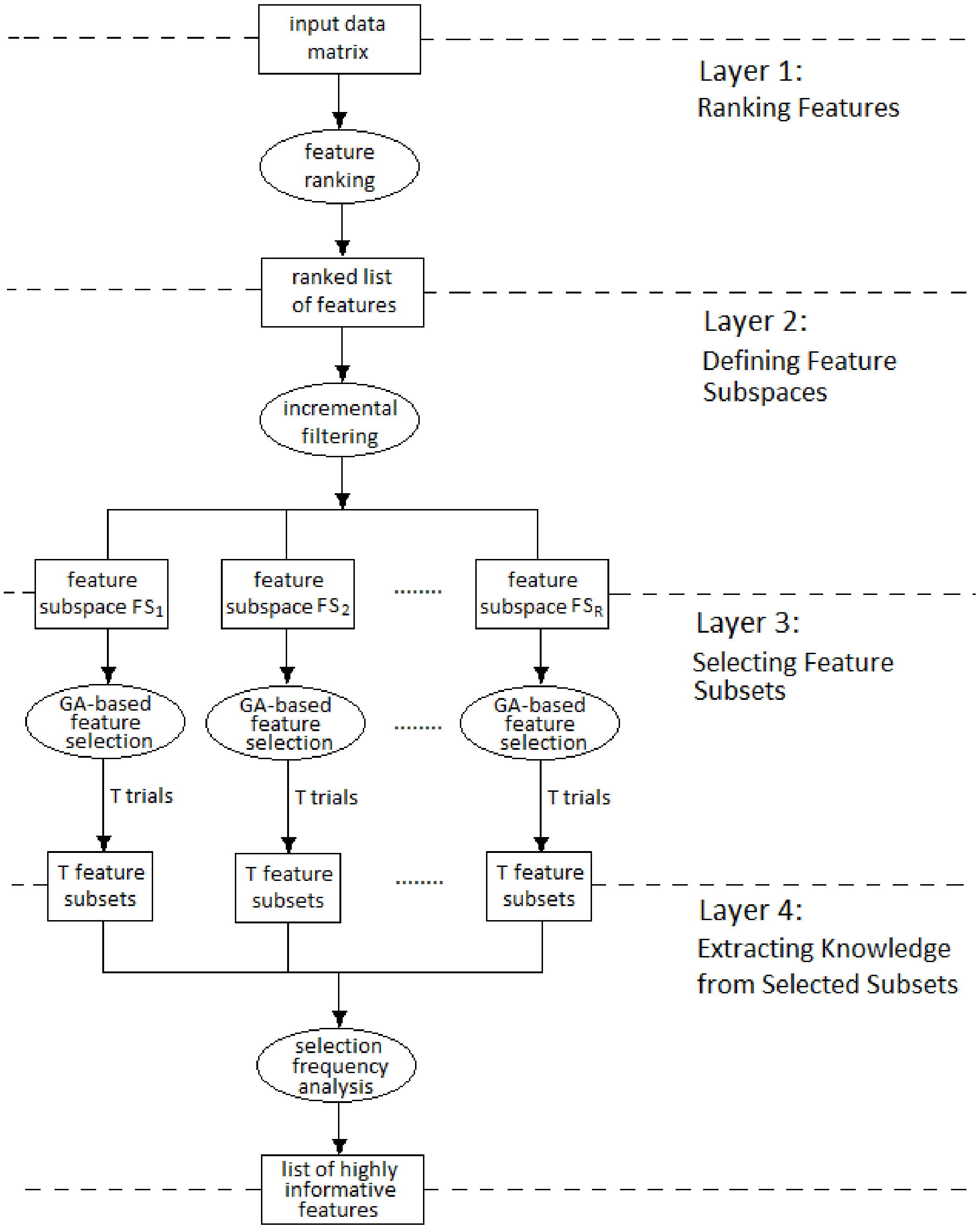
Then, the current population undergoes genetic operations (i.e. selection, mutation and crossover) and a new population is generated and evaluated. This evolution process is repeated until a pre-defined number of generations is reached. It outputs the “best individual” present in the population, i.e. the best predictor for the considered subspace.

Since GA is a randomized algorithm, it can produce different solutions among different trials for the same parameter set and the same initial population. Therefore, it is applied T times on each of the R subspaces.)

3.1.4 Layer 4: Extracting Knowledge from Selected Subsets

The fourth and final task is to extract additional domain knowledge by the predictors obtained at the previous layer. Since the GA has been applied T times on each of the R feature subspaces, we globally obtain T*R solutions (among which there may be a certain number of replicates).

This task is completed by analyzing the frequency of membership of each feature in the collected solutions. Such analysis enables evaluating the relative importance of each feature, distinguishing the features that play a primary role in discriminating the target class from those that give a complementary, yet not negligible, contribution.



3.1The proposed framework

3.2 Distinctive Aspects of the Framework

The proposed framework shows some distinctive characteristics and we highlight here the related advantages.

First. Independence of the framework from a specific implementation. As we have described in section 3.1, the framework is organized in different layers, each of one determines a type of strategy used in the feature selection process. However, as each layer refers to a general class of methods the framework is independent on the choice of the algorithms for its implementation. This means that, at each layer, the tasks are not associated to a single method, but rather to a general class of methods.

Second. Choice of a hybrid approach. In section 2.2, filter-based and wrapper-based feature selection techniques have been described, remarking their differences and discussing the pros and cons of both of them. In some specific domains (as microarray analysis) wrapper methods have been shown to generally perform better than filters but their time-consuming behaviour has made the use of the latter prominent [28-30][36].

Our framework can be classified as a hybrid feature selection approach, since it combines and takes advantage of both filter and wrapper techniques in order to overcome their limitations. In detail, the preliminary use of the filter is to guide the features research at the initial stage, permitting a rapid analysis of the dataset and ensuring that useful features are unlikely to be discarded. On the other hand, the subsequent use of the wrapper permits to refine the subspaces obtained before. In this step, the intent is to remove redundant features and obtain more accurate and small-sized predictors for the classification. Moreover, since the preliminary use of the filter permits to reduce the initial feature space, the wrapper is applied on a smaller search space and its required computational cost is reduced in consequence. This double feature selection step, therefore, makes the use of a wrapper more efficient.

Third. Parallel exploration of different feature spaces. When using a filter-based feature selection techniques, it is not clear how to determine the optimal threshold value (in our case, the optimal value for the parameter P). This value has a critical importance, as it will be used to cut the ranked list of features, distinguishing between the subset of features that will be kept for the further analysis and the subset that will be discarded instead. Since little theoretical support is presented in literature, the choice of the threshold value is often made on an “ad hoc” basis, often depending on the specific problem at hand [20].

Whereas existing hybrid models [32] usually provide a single threshold value, in the proposed framework we assign multiple values to P, i.e. we use multiple thresholds. This way, at the second layer we obtain different subspaces that will refined by the wrapper at the third layer. This parallel exploration of different feature subspaces is guided by the motivation of discovering a potentially high number of solutions, that should not necessarily seen as alternative but rather as complementary.

Fourth. GA as search strategy. Differently from filter based feature selection approaches, wrappers are able to evaluate set of feature in a time and employ a classification algorithm that will be used to build the final classifier. As an exhaustive search is impractical, greedy procedures or heuristics are usually employed to guide the combinatorial search through the space of candidate feature subsets looking for a good trade-off between performance and computational cost. In the layer 3 of our framework, the wrapper employs a GA as the search strategy. This type of algorithm belongs to the class of heuristics, it performs a random feature combination and shows its potentiality in exploring features set of high dimensionality [34][35].

Fifth. Balancing effectiveness and representation level. According to [24], two major factors seems to be particularly important in designing a suitable algorithm for feature selection in a classification task: improving the predictive accuracy and providing better understanding of the underlying concept that generated the data. Here, we denote the above factors as the *effectiveness* and the *representation level* of the feature selection process. Hence, the effectiveness deals with selecting the minimum set of features that maximize the accuracy of a classifier and the representation level concerns discovering how relevant the features are and how related to one another.

In more detail, the effectiveness attempts to capture the performance aspect of classification. From this point of view, the major challenge is finding a minimum subset of features that are useful to the prediction. Thus, this aspect is central for classification problems in which accuracy is of primary concern: the more effective the feature selection, the better the performance of the resulting classifier.

The representation level reflects the explanatory power of the selected features in representing essential knowledge about the application domain. The focus is on discovering all the variables suited to the reality that we are trying to represent, deciding how relevant they are and how related to one another. Under this paradigm, the feature selection process privileges the usefulness of the features in describing the application domain i.e. the degree of exactness with which the representation fits the reality.

Considering filters and wrappers, we can say that filter-based methods are able to give an overview on the data, thus promoting the representation level, but they do not take into account correlation between features, which reduces their usefulness for classification purposes. As they ignore the classifier to be applied, there is no support for improving the effectiveness.

On the other hand, in wrapper-based methods the whole process aims to optimize the accuracy of the particular classifier, therefore the central aspect in selecting features is the effectiveness rather than improving the representation level.

Considering feature selection from this prospective, hybrid approaches can be seen as an attempt to combine effectiveness and representation level, as they attempts to take advantage of filters and wrappers by exploiting their different evaluation criteria in different search stages [20].

In literature, to the best of our knowledge, it remains a neglected issue the formulation of feature selection methods that place the emphasis on balancing effectiveness and representation level. Usually, popular feature selection algorithms are effective in selecting a single subset of predictive features for sample class prediction. They try to achieve the best effectiveness thus discarding features which are relevant to the target concept but highly correlated to the selected ones. However, a single subset of selected features could miss important knowledge and result in a poor representation level.

The framework proposed in this thesis aims to pave the way for balancing effectiveness and representation level by performing an intelligent feature selection that aims not only to achieve good classification performance (layer 3) but also to discover different subsets of features (layer 4) relevant for the application domain and able to represent it properly.

3.3 Application contexts

The framework proposed in this thesis has been designed and developed considering the specific problem of feature selection in microarray data analysis. Therefore, particular attention has been paid to the peculiar characteristics related to the domain such as the existence of multiple biomarkers and the limited number of genes that compose each biomarker.

However, as described in section 3.2, we paid particular attention to make our approach as general as possible. The framework has been designed to be independent on the choice of the algorithms for its implementation: each layer does not refer to a single method but rather to a general class of methods. For example, in layer 2 the framework can supported a variety of popular filter techniques, in the same way as different classifiers can be employed in conjunction with the GA within the wrapper in layer 3.

Hence, the framework can be generally used to solve feature selection problems and its application can be extended to other domains. For this reason in this thesis we consider its extension to analyze a corpus of text documents, with the aim of finding within each document the words that permits to understand the document's topic (term selection) and hence classifying documents in one or more topic categories (text categorization).

In detail, Text Categorization (TC) is the study of assigning natural language documents to one or more predefined category labels. Because of the need to automatically organize the increasing number of digital documents in flexible ways, TC is receiving a crescent interest from researchers and developers. The dominant approach to this problem considers the employment of a general inductive process that automatically builds a classifier by learning, from a set of pre-classified documents, the characteristics of the categories [37].

Formally, a problem of TC can be defined as follows. Let $D = \{d_1, d_2, \ldots, d_L\}$ be a collection of L documents and $W = \{w_1, w_2, \ldots, w_M\}$ be a set of M distinct terms contained in D. Let $C = \{ c_1, c_2, \ldots, c_{|C|} \}$ be a set of predefined categories or classes. A TC process assigns a boolean value to each pair $<d_j, c_i>$ that indicates if the document d_j belongs to the category c_i.

In a multi-label TC problem each document can be assigned to any number of categories from the set C. Under the assumption that categories are stochastically independent of each other, a multi-label TC can be transformed into |C| independent (disjoint) binary TC problems, where each document is classified in one of the two disjoint categories: c and its complement $\overline{c}$. Therefore, to solve a multi-label TC problem, binary classifiers are built for each category in C and their results are then combined into a single decision.

The proposed framework is here used to address a multi-label TC problem by resolving |C| binary problems. The framework firstly selects the most representative terms for a given category c_i and then performs a binary classification process on this selection.

3.4 Experiments

In both the experimentation, we evaluate the Framework by performing two class of experiments:

1. *Baseline experiments*. To get an evaluation of the classification performance without considering the proposed framework, a classifier is trained directly on each feature subspace. The related classification performance is considered as *baseline*.

2. *Framework experiments*. We apply the proposed framework to each feature subspace and evaluate the classification performance of the selected solutions.

By comparing *baseline* and *framework* classification performance, we estimate how significant is the positive contribution brought by the framework, justifying its use.

To quantify the effectiveness of the framework, results obtained from *framework experiments* are evaluated considering, first of all, the classification performance; moreover, we measure the dimensionality of the selected subset (number of features that form the solutions) and analyzed also the computational cost.

Whereas, to evaluate the representation level, we consider the frequency of membership of each feature in the collected solutions.

4 Framework Validation

To evaluate the framework we conducted experiments on four DNA-microarray datasets. Experimental results compare well with different hybrid methods proposed in literature and show that our approach is robust and effective in finding small subsets of informative features with high classification accuracy and suitable representation level. Related results have been presented in [38][39].

4.1 Datasets

We verified the proposed framework with four popular public microarray dataset. Their main characteristics are summarized in Table 4.1. Data are represented in the shape of a matrix: rows represent the recorded samples while the columns are the genes measured in the analysis. Level of expression of genes is a continuous value.

The Leukemia dataset was produced in a study aimed at building a model to discriminate between acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL) tissues [19]. It contains 72 samples: among them, 25 samples are collected from AML patients and 47 samples are from ALL patients. Gene expression levels of 7129 genes are reported. Authors mention the difficulty of choosing the right set of informative genes, given that lots of them were highly correlated with the ALL-AML distinction.

The DLBCL dataset comes from the study [40] where the task is to discriminate between two types of lymphoma cancer. It contains 78 samples: among them, 58 are from diffuse large b-cell lymphoma ('DLBCL') samples and 19 from Follicular Lymphoma ('FL') samples. Gene expression levels of 7129 genes are reported.

The Colon Cancer dataset contains samples from 40 tumor and 22 normal colon tissues probed by an Affymetrix microarray chip measuring more than 6500 genes [41]. However, the dataset was published after a pre-filtering step and the resulting samples include only 2000 genes. The task described in the original study is to determine if groups of patients could automatically be constructed by a clustering algorithm. Since annotations of the samples are available, we address here the binary classification problem of predicting whether a sample corresponds to a tumor versus a normal colon tissue.

The Prostate dataset was first published in [42]. It is the largest dataset used in this thesis in terms of number of features (10509). Here, the task is to predict whether a sample corresponds to a tumor versus a normal tissue. The dataset contains 102 samples from 52 tumor and 50 normal prostate tissues.

4.1 Microarray datasets used in the experiments

Dataset	**No. of samples**	**Distribution among classes**	**No. of features**	**Reference**
Leukemia	72	47 ALL + 25 AML	7129	[19]
DLBCL	78	58 DLBCL + 19 FL	7129	[40]
Colon	62	40 tumor + 22 normal	2000	[41]
Prostate	102	52 tumor + 50 normal	10509	[42]

4.2 Framework Setup

We describe here the peculiar choice made for the framework setup and implementation in this specific domain.

For ranking at layer 1, we chose as metric the $\chi 2$ statistics. It is a widely used standard feature selection method that test the divergence between the observed and expected distribution of a feature. In feature selection, it evaluates features individually by measuring their $\chi 2$ statistic with respect to the classes [43].

For incremental filtering at layer 2, we set P = 10 and R = 5. Starting from the subset including the first P ranked features, namely the subset TOP10, we constructed R-1 additional nested subsets of features of increasing size by progressively adding P features (less and less correlated with the target). We denote these additional subspaces as TOP20 (i.e. the first 20 top-ranked features), TOP30 (i.e. the first 30 top-ranked features), etc. We also considered TOP80 and TOP100 in order to evaluate the proposed approach in larger feature subspaces.

At layer 3, the wrapper is based on the GA search mechanism as proposed by [44]. In our implementation, each individual is a binary vector where the values ‘1’ and ‘0’ respectively mean that the feature is included or not in the individual. The initial population is randomly initialized. Genetic operations are carried out by roulette wheel selection, single point crossover, and bit-flip mutation.

However, when using this kind of evolutionary mechanisms, it is crucial to the success a proper definition of the algorithm in terms of genetic operators and parameter settings since small changes in requirements can lead to significant differences in results. In order to find an efficient configuration of the GA we made a tuning of GA parameters. Specifically, we considered different values for the following parameters: (i) number of generations, (ii) population size, (iii) probability of crossover, and (iv) probability of mutation. We applied the GA search mechanisms T=10 times on different feature subspaces and evaluated the average accuracy, the average subset size and the computational cost.

The analysis was carried on according to two distinct phases:

A. We test the behaviour of the GA search mechanism as parameters (i) and (ii) change, while parameters (iii) and (iv) assume values consistent with the literature; Specifically, values considered for parameters are as follows: (i) number of generations: 10, 20, 30, 50, and 100; (ii) population size: 10, 20, 30, and 50; (iii) probability of crossover = 1; (iv) probability of mutation = 0.01.

B. We test the behaviour of the GA search mechanism as parameters (iii) and (iv) change, while parameters (i) and (ii) assume the best results found in the previous phase A. Values considered for parameters (iii) and (iv) are respectively: (iii) 0.6, 0.8, 1 and (iv) 0.005, 0.01, 0.02, 0.03. According to the results obtained in the phase A, we set (i) number of generations = 50 and (ii) population size = 30.

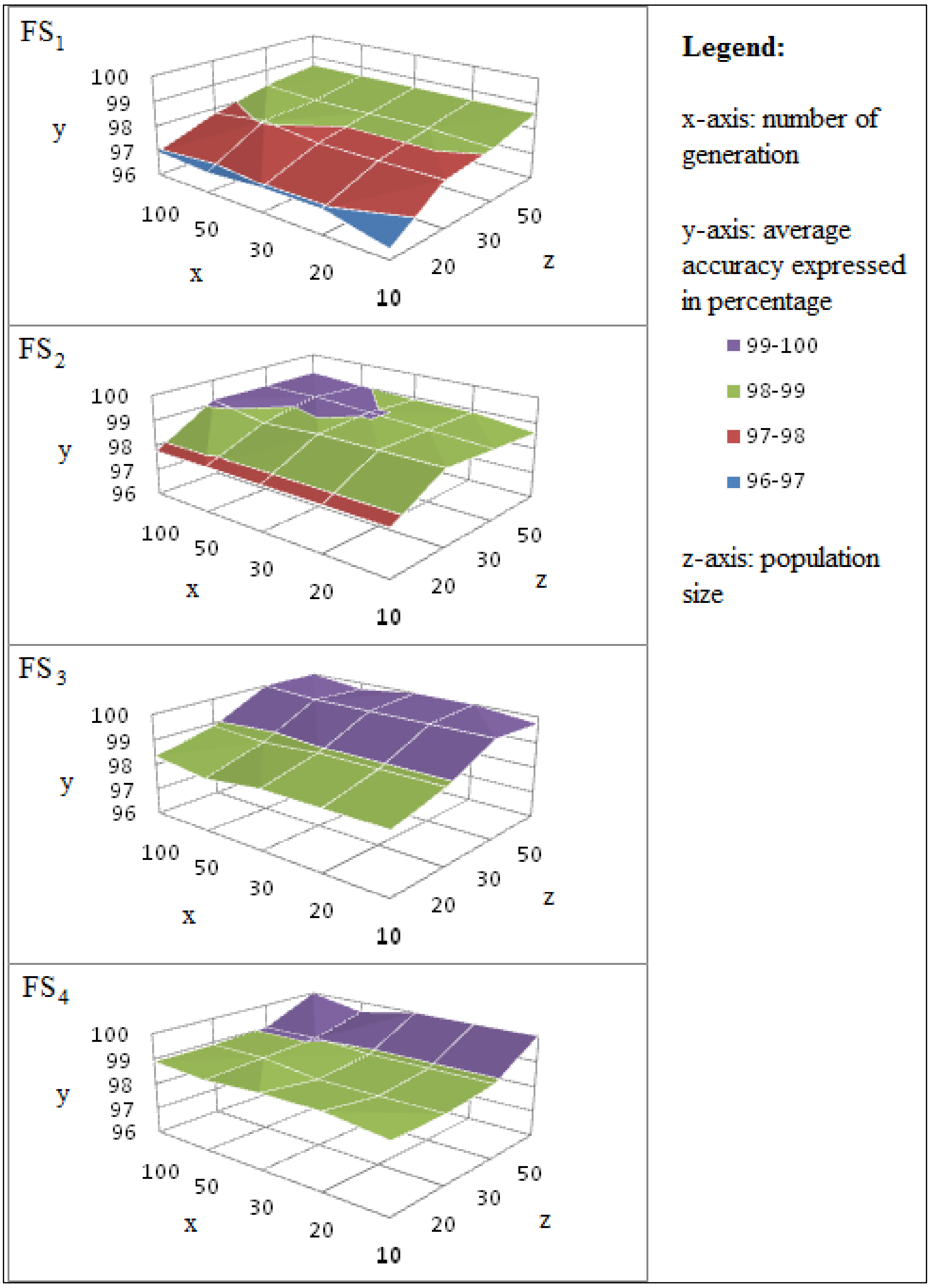
This pairing is justified because, in the literature, wide discordances can be found between the values chosen for parameters (i) and (ii). As well, parameters (iii) and (iv) typically assume values in a range that we consider in our analysis.

Figure 4.1 show some results of GA parameter tuning – phase A. Specifically, the interpolation surface expresses the global trend of the average accuracy (y-axis) vs. the number of generation (x-axis) and the population size (z-axis) on different feature subspaces. Different colours indicate different ranges of values (as shown in the enclosed legends).

Extended results have been presented in [45][46]. Therefore, being supported by this tuning analysis, we set the following values for the GA: population size = 30, number of generations = 50, probability of crossover = 1, and probability of mutation = 0.02.

To assess the population, the fitness function evaluated the predictive performance considered as the accuracy calculated on each individual. We experimented two different classifiers: a Support Vector Machine (SVM) and a Nearest Neighbor classifier (K-NN).

Support vector machines (SVM) are a set of supervised learning methods used for classification. In the most widely used two-class SVM classification method, as in this thesis, input data are viewed as two sets of vectors in the multi-dimensional input space. The SVM classifier constructs a separating hyperplane in that space, one which maximizes the margin between the two data sets. The method can also be extended to multi-class and nonlinear classification problems by using a nonlinear kernel function.



4.1 GA parameter tuning

The k-nearest neighbors algorithm (K-NN) is a method for classifying objects based on closest training examples in the feature space. It is a type of instance-based learning, or lazy learning where the function is only approximated locally and all computation is delayed until classification. A majority vote of an object's neighbors is used for classification, with the object being assigned to the class most common amongst its k nearest neighbors. If k is set to 1, then the object is simply assigned to the class of its nearest neighbor.

This choice resulted in two different implementations of the wrapper, that take the name of GA/SVM and GA/K-NN, respectively. Error estimation was performed by a 10-fold cross-validation for both SVM and K-NN classifiers. Since the GA performs a stochastic search, we considered the average results over a number T=10 of trials.

The implementations of all the algorithms are those found in the WEKA machine learning environment [47]. In particular, we used the SMO implementation [48] with linear kernel for the SVM classifier and the IBK implementation [49] with K = 1 for the K-NN classifier[1].

With regard the framework evaluation, in the *baseline experiments* each classifier (i.e. K-NN and SVM) was trained directly on each subspace TOPN in order to estimate the accuracy without the wrapper-based feature selection included in the framework. This baseline accuracy was also estimated by a 10-fold cross-validation. In the *framework experiments* we evaluated the framework applying it entirely on each subspace TOPN.

[1] Additional experiments with K = 3 resulted in non-significant differences, neither in term of classification accuracy nor subset size.

4.3 Results and Discussion

As previously said, results of our experiments will be discussed along two dimensions:

1. the effectiveness of the proposed framework in searching suitable combinations of relatively few genes that yield high classification accuracy;
2. the representation level reached by the framework in exploring how each gene may be useful in representing essential knowledge about the application domain.

4.3.1 On the Effectiveness of the Framework

We compare first the differences between the baseline accuracy and the accuracy (best and average) reached by the two implementations of the framework, GA/K-NN and GA/SVM, on each TOPN. Table 4.2 reports this comparison for each dataset.

As Table 4.2a shows, results produced by both GA/SVM and GA/K-NN on Leukemia dataset outperform baseline results from SVM and K-NN. The average accuracy of GA/SVM increases with the size of the search space until reaching 100% on TOP80 and TOP100. GA/K-NN turns out to be more effective in selecting feature subsets that perfectly discriminate the target class (namely, perfect predictors), irrespective of the size of TOPN: a search space of 10 features is sufficient to reach the maximum accuracy that is also reached in all the feature subspaces with the exception of TOP50.

Table 4.2b shows the same trend for experiments on DLBCL dataset, both with GA/SVM and GA/K-NN.

According to Table 4.2c, neither GA/SVM nor GA/K-NN are able to find perfect predictors. That is a side effect of the Colon dataset which is quite noisy and is considered one of the most difficult to classify due to a probable sample contamination problem [50]. However, the average accuracy of GA/SVM exhibits the same behavior than in previous experiments. The effectiveness of the GA/K-NN is confirmed, regardless of the size of the feature subspace: the best predictor is extracted from TOP20.

Finally, Table 4.2d reports results about Prostate dataset and shows a picture quite different from the three previous datasets. The trend of the average accuracy of GA/SVM reaches the highest value between TOP30 and TOP50 and then starts decreasing. GA/K-NN outperforms GA/SVM very slightly, since the values of the average accuracy are highly similar and, in addition, both achieve the same values of best accuracy.

Globally, results in Table 4.2 confirm that the classification can be carried out in a reduced space more accurately that in the original feature subspace as the use of an unnecessarily large gene set may decrease the effectiveness in the classification process.

By showing the trend of the average size of the selected combinations as the size of the feature subspace TOPN increases, Fig. 4.2 demonstrates the effectiveness of the framework in reducing the dimensionality of the search space.

4.2 Baseline, Average and Best Accuracy

(a) Leukemia

	SVM	GA/SVM		K-NN	GA/K-NN	
	Baseline accuracy (%)	Average accuracy (%)	Best accuracy (%)	Baseline accuracy (%)	Average accuracy (%)	Best accuracy (%)
TOP10	93.1	97.0	97.5	91.7	100	100
TOP20	95.8	99.3	100	97.2	100	100
TOP30	98.6	99.6	100	94.4	100	100
TOP40	98.6	99.9	100	95.8	100	100
TOP50	97.2	99.4	100	93.1	99.9	100
TOP80	97.2	100	100	95.8	100	100
TOP100	97.2	100	100	97.2	100	100

(b) DLBCL

	SVM	GA/SVM		K-NN	GA/K-NN	
	Baseline accuracy (%)	Average accuracy (%)	Best accuracy (%)	Baseline accuracy (%)	Average accuracy (%)	Best accuracy (%)
TOP10	92.2	92.5	92.7	85.7	93.8	94.3
TOP20	94.8	96.8	97.4	93.5	99.9	100
TOP30	94.8	98.1	98.7	96.1	100	100
TOP40	96.1	98.7	100	94.8	99.7	100
TOP50	96.1	98.1	98.7	94.8	100	100
TOP80	97.4	99.1	100	96.1	100	100
TOP100	94.8	100	100	96.1	99.9	100

(c) Colon

	SVM	GA/SVM		K-NN	GA/K-NN	
	Baseline accuracy (%)	Average accuracy (%)	Best accuracy (%)	Baseline accuracy (%)	Average accuracy (%)	Best accuracy (%)
TOP10	82.3	87.1	87.1	80.6	90.8	91.3
TOP20	88.7	90.9	91.9	82.3	95.3	98.4
TOP30	87.1	90.5	91.9	83.9	94.5	95.2
TOP40	85.5	91.5	92.3	83.9	94.5	96.8
TOP50	83.9	91.5	91.9	80.6	92.7	95.2
TOP80	85.5	91.9	93.2	79.0	94.6	96.8
TOP100	87.1	93.1	94.2	79.0	93.2	95.5

(d) Prostate

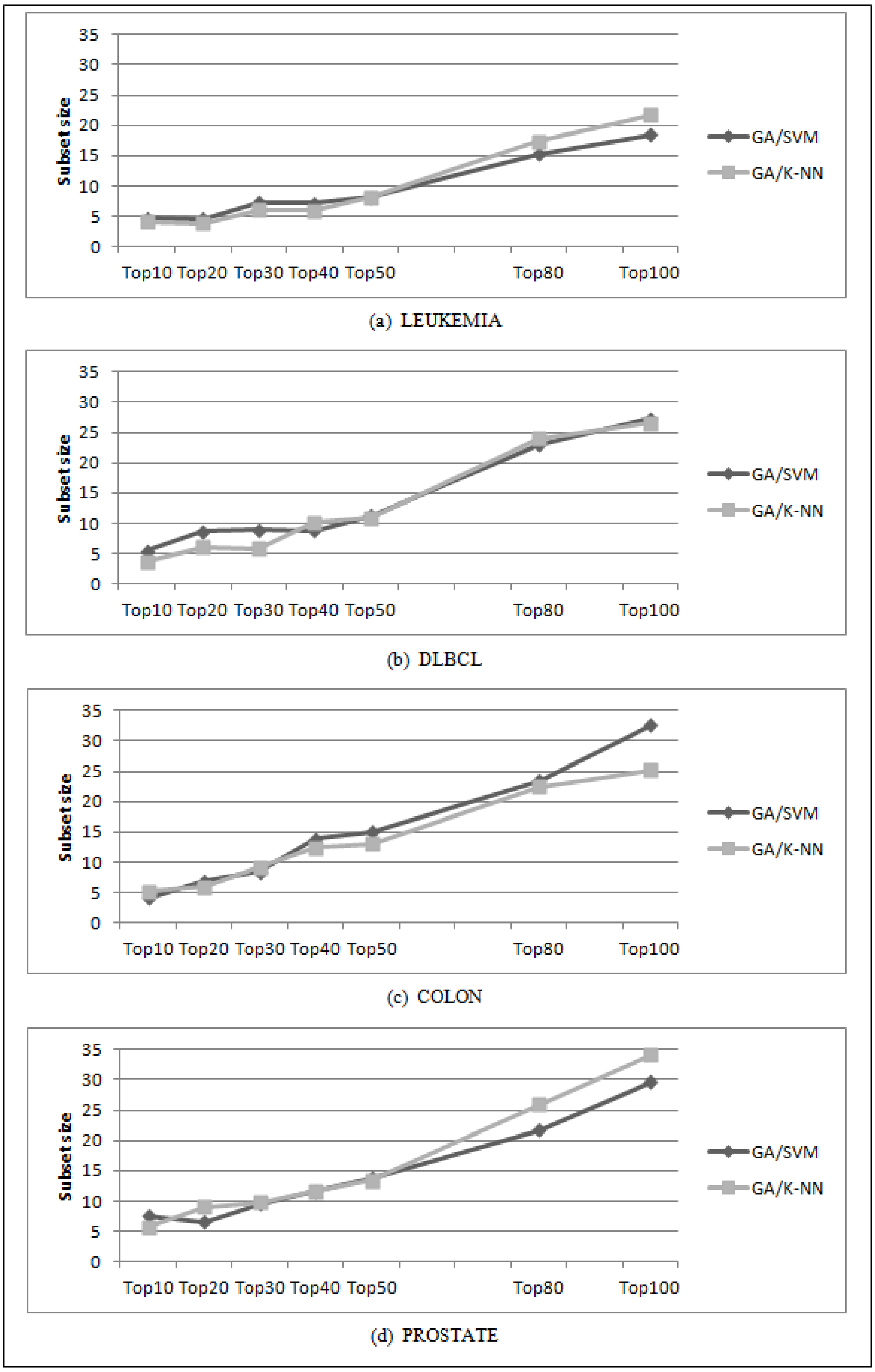
	SVM	GA/SVM		K-NN	GA/K-NN	
	Baseline accuracy (%)	Average accuracy (%)	Best accuracy (%)	Baseline accuracy (%)	Average accuracy (%)	Best accuracy (%)
TOP10	95.1	95.4	95.7	92.2	94.4	94.7
TOP20	96.1	96.6	97.1	93.1	96.2	97.1
TOP30	94.1	97.8	98.0	89.2	97.9	98.0
TOP40	97.1	97.8	98.0	93.1	98.0	98.0
TOP50	96.1	97.9	98.0	94.1	98.0	98.0
TOP80	96.1	97.3	98.0	92.2	98.0	98.0
TOP100	96.1	97.2	98.0	90.2	98.0	98.0

In particular, both GA/SVM and GA/K-NN considerably cut the size of the original TOPN whose average reduction is greater than 50% with peaks of 70-75% reached on TOP100. This trend is common to all the datasets.

As Table 5.3 shows, this reduction generates sufficient features for achieving a very high accuracy. In detail, for the best predictors selected by GA/SVM and GA/K-NN on each dataset, Table 5.3 summarizes the accuracy, the minimum size and the feature space from which the predictor was extracted. For "good" datasets such as Leukemia and DLBLC, the framework generates perfect predictors. For more difficult datasets, such as Colon and Prostate, the framework doesn't achieve the 100% accuracy albeit obtaining remarkable results.

4.3 Best predictors extracted by GA/SVM and GA/K-NN from each dataset

	Wrapper	Top	Subset size	Accuracy (%)
Leukemia	GA/SVM	TOP20	4	100
	GA/K-NN	TOP20	3	100
		TOP30	3	100
		Top40	3	100
DLBCL	GA/SVM	TOP40	8	100
	GA/K-NN	TOP20	4	100
		TOP30	4	100
Colon	GA/SVM	TOP100	39	94.2
	GA/K-NN	TOP20	4	98.4
Prostate	GA/SVM	TOP30	8	98.0
	GA/K-NN	TOP30	8	98.0
		TOP40	8	98.0



(a) LEUKEMIA

(b) DLBCL

(c) COLON

(d) PROSTATE

4.2 Average size of the selected predictors as the size of the feature space increases

Globally, the above results help to demonstrate the framework effectiveness and can be compared with those produced by different state-of-art methods in DNA-microarray literature. As reference parameters, we considered the accuracy and the number of selected features.

We present the best results achieved by GA/K-NN and omit results from GA/SVM that, except for the Colon dataset, exhibit the same trend. Table 4.4 shows this comparison.

Regarding Leukemia dataset (Table 4.4a), different methods proposed in recent literature [32][51][52][53][54] achieve 100% of accuracy, as in our approach, but the number of features they select is greater than the one obtained by GA/K-NN.

Our method shows excellent performance also in DLBCL dataset (Table 4.4b): the framework reaches 100% of accuracy with only 4 features as in [55] and outperforms the approaches proposed in [32][56][57][58].

Regarding Colon (Table 4.4c), which is recognized as one of most noisiest microarray datasets, GA/K-NN achieves better accuracy than all other methods [32][52][59][60] except for [51]; in [51], however, the number of selected features is greater than the one obtained by our framework.

Finally, in the Prostate dataset (Table 4.4d), the best performance is obtained by [32]; our approach reaches the same accuracy, with a number of features slightly superior, and outperforms all the other methods [57][58][61][62].

4.4 Framework performance vs different state-of-art methods

(a) Leukemia

	GA /K-NN	[32]	[51]	[52]	[53]	[54]
Accuracy (%)	100	100	100	100	100	100
Subset size	3	4	25	6	8	4

(b) DLBCL

	GA /K-NN	[32]	[55]	[56]	[57]	[58]
Accuracy (%)	100	100	100	93.5	92.2	96.1
Subset size	4	6	4	5	6	6

(c) Colon

	GA /K-NN	[32]	[51]	[52]	[59]	[60]
Accuracy (%)	98.4	95.2	99.4	93.6	97.0	93.6
Subset size	4	6	10	12	7	4

(d) Prostate

	GA /K-NN	[32]	[57]	[58]	[61]	[62]
Accuracy (%)	98.0	98.0	96.1	96.1	91.2	96.7
Subset size	8	6	13	11	6	19

We conclude this discussion on the effectiveness of the framework with some considerations about the pattern of agreement noticed among the four datasets. First of all, the effectiveness of both GA/SVM and GA/K-NN is always higher than the effectiveness of the baseline SVM and K-NN, as shown by the results about accuracy of the selected predictors.

However, the effectiveness of GA/SVM and GA/K-NN changes in a different way depending on the size of the initial feature subspace. The GA/SVM average accuracy tends to improve as the size of the feature subspace increases. On the contrary, GA/K-NN is much more effective in selecting predictive subsets irrespective of the size of the provided feature subspace.

As regards the average size of selected predictors, no significant difference has been found between the behaviour of GA/SVM and GA/K-NN. Finally, GA/K-NN turns out greatly superior in terms of computational cost (with execution times remarkably lower than GA/SVM), leading to an effective feature selection in a very efficient way.

4.3.2 On the Representation Level of the Framework

As previous mentioned, the effectiveness focuses on a global view about the classification accuracy obtained by a few number of selected features while the representation level aims to identify the features that provide a better understanding of the analyzed domain.

Besides predictive ability of the selected features, domain experts instinctively have high confidence in the results of a selection method that finds similar sets of features: the fact that a gene is selected by different predictors makes it more probable that this gene is an important biomarker. Hence, we assume the frequency of each gene in the selected predictors as a measure of the representation level of the framework.

Specifically, for each microarray dataset we evaluated the frequency of the features belonging to the 70 predictors obtained at layer 3. For each dataset, Table 4.5 shows the frequency of the ten most selected features and reports, in brackets, the position of each feature in the original ranked list obtained at layer 1.

Analyzing the features that are most frequently selected by GA/SVM and GA/K-NN , we note, in Table 4.5a, that the two lists have 7 features in common out of 10. Besides, features that appear only in GA/SVM list are also selected by GA/K-NN with lower frequency, and vice versa. Further, we notice that the features most frequently involved in the selected predictors are not necessarily the top-ranked ones: for example, the gene 1928 exhibits the highest frequency for GA/SVM but it is placed at ranking position 30; likewise, the most frequent gene for GA/K-NN, the gene 2354, is placed at ranking position 14. As well, genes 4951 and 5107 exhibit a very high frequency, but they are at ranking position 67 and 68, respectively. In turn, some top-ranked genes such as 3252 and 2288 do not appear at all in the two lists even if they are respectively at positions 4 and 6 of the ranked list.

Table 4.5b reports results regarding DLBCL dataset. In this case the two lists have 8 features in common. Genes 1670 and 5077 exhibit the highest frequencies for both GA/SVM and GA/K-NN even if they are placed at ranking positions 18 and 29. On the contrary, genes at ranking positions 2 and 3 do not appear in the lists.

As Table 4.5c reports, the two lists have 5 features in common. Features that exhibit the highest frequency are gene 66 for GA/SVM and gene 1772 for GA/K-NN and are placed at ranking position 15 and 10, respectively. Again, some top-ranked genes, such as those at positions 1 and 2, do not emerge.

Table 4.5d reports results regarding Prostate dataset. The two lists have 7 features in common. Features that exhibit the highest frequency, for both GA/SVM and GA/K-NN, are 4823, 10130, and 9138 and are placed at ranking positions 1, 13, and 16. Although features at ranking positions 1 and 2 are present, genes at ranking position 3 and 4 do not appear in the lists.

Again, this section on the representation level of the framework is concluded with some remarks about the pattern of agreement that we noticed among the four datasets. First of all, in each of the considered case studies, the high number of features in common between the two lists shows that GA/SVM and GA/K-NN highly agree in evaluating the relevance of features, although selecting different gene combinations (at layer 3). This suggests that the proposed framework can be useful to evaluate the relative importance of features in a context where multiple predictors may coexist, such as microarray data classification.

Another remarkable aspect to note is that the feature lists obtained at layer 4 don’t match the ranked list produced as output of the first layer; indeed the features most frequently selected are not necessarily the top-ranked ones. This outlines that, while useful in reducing the dimensionality of the initial problem, the ranking process is not by itself a suitable feature selection technique for microarray data. It can be successfully employed, instead, within hybrid filter-wrapper approaches as the one proposed here.

4.5 Frequency of the ten most selected features; in brackets, the position of each feature in the original ranked list

(a) Leukemia

GA/SVM		GA/K-NN	
Features	Frequency	Features	Frequency
1928 (30)	47.1%	2354 (14)	44.3%
1144 (17)	41.4%	1834 (1)	42.9%
2354 (14)	38.6%	6855 (5)	38.6%
6855 (5)	37.1%	1928 (30)	38.6%
1685 (9)	37.1%	1685 (9)	37.1%
1834 (1)	35.7%	804 (31)	32.9%
4847 (2)	34.3%	1144 (17)	31.4%
804 (31)	30.0%	1882 (3)	24.3%
2020 (22)	24.3%	5107 (68)	24.3%
2642 (28)	22.9%	4951 (67)	20.0%

(b) DLBCL

GA/SVM		GA/K-NN	
Features	Frequency	Features	Frequency
5077 (29)	70.0%	1670 (18)	65.7%
1670 (18)	64.3%	5077 (29)	37.1%
4453 (11)	45.7%	3818 (32)	37.1%
3005 (24)	42.9%	4453 (11)	34.3%
506 (1)	41.4%	373 (12)	34.3%
203 (13)	41.4%	1055 (9)	32.9%
373 (12)	37.1%	2789 (38)	31.4%
2789 (38)	37.1%	506 (1)	30.0%
3818 (32)	35.7%	3005 (24)	30.0%
4202 (5)	32.9%	6493 (43)	30.0%

(c) Colon

GA/SVM		GA/K-NN	
Features	Frequency	Features	Frequency
66 (15)	64.3%	1772 (10)	75.7%
493 (3)	61.4%	765 (4)	72.9%
1423 (5)	60.0%	1423 (5)	41.4%
1771 (6)	58.6%	267 (8)	35.7%
1772 (10)	57.1%	415 (21)	35.7%
897 (14)	52.9%	513 (7)	34.3%
1042 (19)	48.6%	1892 (20)	34.3%
765 (4)	47.1%	1771 (6)	32.9%
581 (35)	45.7%	897 (14)	32.9%
780 (12)	42.9%	822 (16)	32.9%

(d) Prostate

GA/SVM		GA/K-NN	
Features	Frequency	Features	Frequency
4823 (1)	67.1%	4823 (1)	67.1%
10130 (13)	67.1%	9138 (16)	67.1%
2718 (30)	60.0%	7346 (8)	65.7%
7652 (5)	57.1%	7652 (5)	61.4%
9138 (16)	57.1%	3997 (27)	54.3%
7515 (21)	51.4%	3124 (39)	54.3%
8765 (2)	48.6%	8765 (2)	51.4%
8009 (7)	47.1%	8009 (7)	51.4%
1943 (23)	47.1%	10130 (13)	45.7%
5648 (25)	42.9%	2718 (30)	45.7%

4.4 Concluding Remarks

In this chapter we have described the experimental analysis made on four DNA-microarray datasets. The proposed hybrid framework have shown its effectiveness in finding small subsets of informative features, in both the implementations proposed. However, the GA/K-NN version has proved to be more accurate and more efficient respect to GA/SVM, finding high predictive subsets irrespective of the size of the provided feature subspace and requiring a lower computational cost.

The framework has also shown a good ability in representing the analyzed domain, as it has been able to highlight the most relevant features that describe the data. For this reason, we believe that the framework can be useful to evaluate the relative importance of features in a context where multiple predictors may coexist, such as microarray data classification.

5 Extension to Other Domains: Preliminary Results on Text Analysis

In the second experimentation we evaluate the framework considering the problem of selecting predictive terms within documents to perform text categorization. This study can be considered as a preliminary attempt in this domain, as we consider only one corpus of text documents. Experimental results compare well both with classical learning approaches and with recent hybrid methods proposed in literature. Related results have been presented in [63].

5.1 Datasets

We tested the framework on the standard Reuters-21578 text collection [64]. This collection, as well as its earlier variants, has been a standard benchmark for TC applications for many years. It is a set of 21,578 news stories published by Reuters in 1987, which are classified according to 135 thematic categories mostly concerning business and economy. Standard splits are defined by the creators of the collection to create various subsets of the corpus and different splits have been used by researchers to test their systems. Majority of researchers used Mod-Apté split that selects 9,603 training documents and the other 3,299 test documents from 90 categories. In this thesis, we used this split too.

Moreover, we used the dataset as pre-processed in [65], which considers the 10 categories with the highest number of positive training examples. In the following we will refer to this subset as R10. For each category in R10, the framework input (i.e. the training set) is a matrix where each row represents a document d_j and columns are the related terms $\{w_1, w_2, \ldots, w_M\}$. Each document is assigned to either the category c_i or its complement $\bar{c}_i$. Table 5.1 shows the number of terms for each category in R10.

5.1 Categories in R10

R10	
Category	No. of terms
acq	7,495
corn	8,302
crude	14,466
earn	9,500
grain	12,473
interest	10,458
money-fx	7,757
ship	9,930
trade	7,600
wheat	8,626

5.2 Framework Setup

In this second experimentation, likewise we did in section 4.2, we describe how we arrange the framework setup and implementation according to the specific characteristics of the domain.

For ranking at layer 1, we chose to use two different metrics: the $\chi 2$ statistics (CHI) and the Information Gain (IG). Both of them are widely used standard feature selection methods. CHI bases on the $\chi 2$ distribution and has been previously described in section 4.2. IG evaluates the worth of an attribute by measuring the information gain with respect to the class, according to the formula:

$$\text{InfoGain}(\text{Class},\text{Attribute}) = H(\text{Class}) - H(\text{Class} \mid \text{Attribute}),$$

where H is the information entropy.

Hence, this choice resulted in two different implementations of the framework that differ in the choice of the filter technique. We named these two versions CHI+GA/NMB and IG+GA/NBM respectively.

For incremental filtering at layer 2, we set P = 10 and R = 10. That is, we start from the subset including the first 10 ranked terms, namely the subset TOP10, until reaching the subset TOP100. We also considered TOP150 and TOP200 in order to evaluate the proposed approach in larger feature subspaces.

At layer 3, the wrapper is based on the GA search mechanism and related implementation and parameter tuning are consistent to section 4.2. In this experimentation we considered the results of a number T=3 trials. To assess the population, the fitness function evaluated the predictive performance of each individual as calculated by a Naïve Bayes Multinomial (NBM) classifier [66]. A Naïve Bayes classifier is a probabilistic classifier based on applying Bayes's theorem with strong (naive) independence assumptions. The Multinomial is a version of the Naïve Bayes classifier commonly used in TC community. In this model, a document is represented by the set of word occurrences from the document. The individual word occurrences are considered to be the events and the document to be the collection of word events. When calculating the probability of a document, one multiplies the probabilities of the words that occur [67].

Despite accuracy is a traditional measures to evaluate classification effectiveness, it is not widely used in the TC domain because the two categories c_i and $\bar{c}_i$ are usually unbalanced [69]. Instead, the evaluation of classification in TC applications is usually analyzed from multiple perspectives, as using precision, recall, and their combinations. Precision measures the percentage of documents predicted to be in class c_i that in fact belong to it. Recall is the percentage of documents truly belonging to c_i that are classified into this class.

In this thesis, we chose to measure predictive performances by using the following metrics [37]:

1. F-measure, which expresses the harmonic mean between precision and recall;

2. Break Even Point (BEP), which expresses the mathematical mean between precision and recall;

3. µ-BEP, which permits a global evaluation of BEP values across the different categories present in the corpus.

The implementations of all the algorithms are those found in the WEKA machine learning environment [47].

With regard the framework evaluation, in the *baseline experiments* the NBM classifier was trained directly on each subspace TOPN in order to estimate classification performance without the wrapper-based feature selection included in the framework. This baseline accuracy was also estimated by using training and test set. In the *framework experiments* we evaluated the effectiveness of the whole framework, in its two implementations CHI+GA/NMB and IG+GA/NBM, applying them entirely on each subspace TOPN.

5.3 Results and Discussion

As previously said and similar to the discussion made in section 4.3, results of our experiments will be discussed along two dimensions:

1. the effectiveness of the proposed framework in searching suitable combinations of terms that permit to correctly categorize documents, yielding high classification performance;

2. the representation level reached by the framework in exploring how each term may be relevant to the deep understanding of the application domain.

5.3.1 On the Effectiveness of the Framework

To have an initial evaluation of the effectiveness of the framework, we first analyze results obtained in the *baseline* and the *framework experiments*. In details, we compare the performance values reached in the *baseline experiments* with those obtained by CHI+GA/NBM and IG+GA/NBM on each TOPN. For the *framework experiments*, we consider best values and average values over the 3 trials. Performance values are evaluated in terms of F-measure.

With regard to the CHI filter, we note that the F-measure obtained in the *baseline experiments* grows as the size of the TOPN becomes larger. The same increasing trend of the F-measure is noticed for CHI+GA/NBM. These considerations are valid for each category in R10. On the contrary, results achieved using the IG filter do not show a clear tendency: no analogies can be noted neither comparing *baseline* and *framework experiments* nor confronting the trends in the ten categories.

Confronting best and average values obtained both by CHI+GA/NBM and IG+GA/NBM, no significant differences have been found. Therefore, in the following analysis we will consider only the best values achieved by the two implementations of the framework.

That said, we proceed with the comparison between the baseline F-measure and the best F-measure obtained by CHI+GA/NBM and by IG+GA/NBM. Results show that the highest F-measure value achieved for each specific category (i.e. best predictor) is always reached by one of the implementations of the framework. Specifically, in 9 cases out of 10, the best result is obtained using IG+GA/NBM and only in 1 case the best result is obtained by CHI+GA/NBM.

Analyzing results, the advantage of using the IG filter in this domain turns out evident for two reasons. Firstly, comparing baseline results, the NBM classifier reaches high F-measure values even for small subspaces (i.e. small values of N) if these are extracted from the ranked term list obtained by using the IG filter. On the contrary, if the ranked list is created using CHI, the NBM needs very large subspaces to find significant results. This means that, in this domain, the IG filter is able to better evaluate the relevance of terms and scores them in a more meaningful way, respect to the CHI filter.

Secondly, comparing the experiment results, we see that the average and best values reached by IG+GA/NBM are almost always higher than the values of CHI+GA/NBM. Considering the previously discussed superiority of the IG filter, we are not surprised: as the IG filter is more sensitive in this specific domain, the TOPN extracted from its ranked list are some more profitable search spaces to apply the wrapper.

Considering now the ability of the framework to reduce the search space and select a significant subset of predictive terms (i.e. the solution), we analyze the trend of the average size of the combinations of terms selected by CHI+GA/NBM and IG+GA/NBM as the size of the feature subspace TOPN increases.

We see that also in this domain the framework proves to be effective in reducing the dimensionality of the search space as both the two implementations of the framework are able to reduce the initial dimensionality of each TOPN.

The number of selected terms is proportional to the size of the search space (i.e. the value of N) as bigger solutions are found in bigger TOPN and this trend is evident for all the categories in R10. Although this increasing trend is common for both the implementation, again IG+GA/NBM shows to be superior as it allows to select a lower number of terms in almost all the categories.

We summarize the best solutions (i.e. the best predictors) obtained by the framework in Table 5.2. For each category in R10, the best F-measure, the size of the relative predictor, and the search space from which it has been selected are reported.

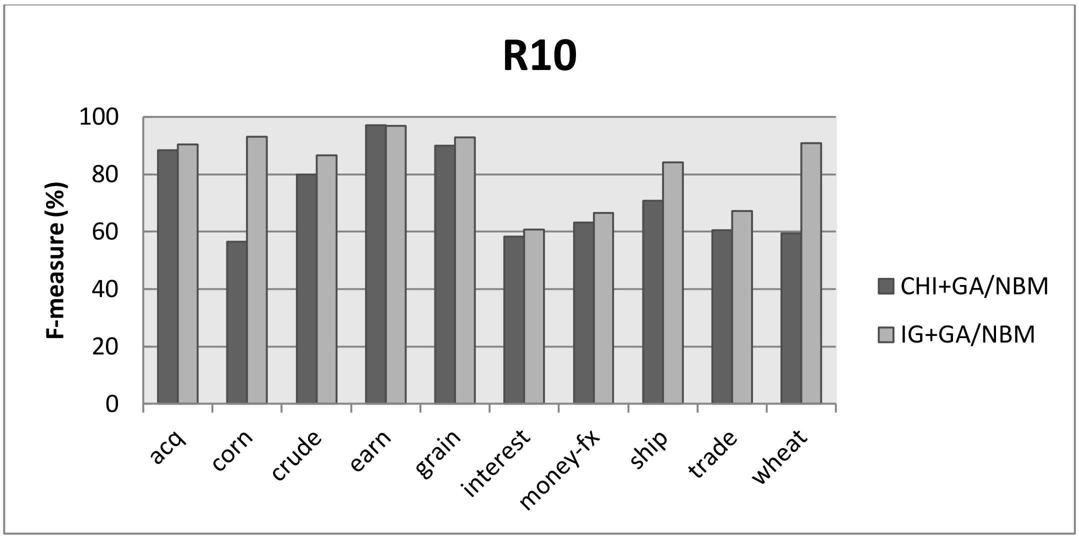
Firstly, we note that in this domain no perfect predictors (i.e. term subsets that perfectly discriminate the target class) have been found, with none of the two implementations of the framework. Considering Table 5.2, the best predictors found by CHI+GA/NBM and by IG+GA/NBM can be considered equivalent for the categories acq, earn, interest, and money. For the other 6 categories, we see that IG+GA/NBM is more effective than CHI+GA/NBM, in particular it is notably more effective for the categories corn, wheat, and ship. These considerations results more evident reported in graphical form, as depicted in Figure 5.1.

Not only does the IG+GA/NBM implementation result more effective in terms of F-measure, but it is also able to find smaller best predictor than CHI+GA/NBM. This is evident for all the categories but acq and earn, where the two framework versions are comparable both for performance and dimension of the solutions.

Moreover, CHI+GA/NBM needs very large search spaces to find its best predictor (best solutions are always found in TOP200) while IG+GA/NBM reaches higher values of F-measure starting from smaller search spaces.

5.2 Best predictors extracted by CHI+GA/NBM and IG+GA/NBM from each category in R10

Category	Filter	Top	Subset size	F-measure (%)
acq	CHI	TOP 200	107	88.46
	IG	TOP 200	105	90.36
corn	CHI	TOP 200	123	56.52
	IG	TOP 150	30	93.09
crude	CHI	TOP 200	111	79.91
	IG	TOP 50	33	86.52
earn	CHI	TOP 200	97	97.05
	IG	TOP 150	73	96.90
grain	CHI	TOP 200	73	89.82
	IG	TOP 30	13	92.79
interest	CHI	TOP 200	110	58.29
	IG	TOP 90	34	60.68
money-fx	CHI	TOP 200	111	63.21
	IG	TOP 150	69	66.51
ship	CHI	TOP 200	122	70.74
	IG	TOP 90	47	84.09
trade	CHI	TOP 200	101	60.48
	IG	TOP 60	30	67.29
wheat	CHI	TOP 150	98	59.29
	IG	TOP 40	5	90.81



5.1 Best predictors extracted by CHI+GA/NBM and IG+GA/NBM from each category in R10

Finally, we also evaluate the framework effectiveness comparing the obtained results with those produced by other learning approaches proposed in the TC literature. In particular, we consider the following classifiers: Naïve Bayes, C4.5, Ripper, and SVM (both polynomial and radial basis function – rbf) plus two hybrid approaches named Olex-GA and Olex Greedy recently proposed in [69]. As reference parameters, we considered the classification performance, expressed by BEP and µ-BEP, and the number of selected terms. We present the best results achieved by the two implementations CHI+GA/NBM and IG+GA/NBM, in the same way as the best results obtained from the other approaches are reported. Table 5.3 shows this comparison.

We can see that the results obtained using CHI+GA/NBM, with a µ-BEP of 86.06, do not significantly emerge as they surpass only Naïve Bayes (82.52), C4.5 (85.82), and the hybrid approach Olex Greedy (84.80). On the other hand, results from IG+GA/NBM compare well with the best results obtained from the other algorithms: with a µ-BEP of 88.98, this implementation of the framework outperforms all the other approaches but the SVM poly (89.91).

5.3 Framework performance vs different state-of-art methods

Category	**Naïve Bayes**	**C4.5**	**Ripper**	**SVM poly**	**SVM rbf**	**Olex greedy**	**Olex GA**	**CHI + GA/ NBM**	**IG + GA/ NBM**
acq	90.29	85.59	86.63	90.37	90.83	84.32	87.49	88.55	90.40
corn	59.41	86.73	91.79	87.16	84.74	89.38	91.07	62.85	93.20
crude	78.84	82.43	81.07	87.82	86.17	80.84	77.18	80.75	86.85
earn	96.61	95.77	95.31	97.32	96.57	93.13	95.34	97.05	96.90
grain	77.82	89.69	89.93	92.47	88.94	91.28	91.75	90.05	92.85
interest	61.71	52.93	63.15	68.16	58.71	55.96	64.59	58.35	60.70
money-fx	56.67	63.08	62.94	72.89	68.22	68.01	66.66	63.70	66.95
ship	68.68	71.72	75.91	82.66	80.40	78.49	74.81	74.05	84.10
trade	57.90	70.04	75.82	77.77	74.14	64.28	61.81	63.00	67.70
wheat	71.77	91.46	90.66	86.13	89.25	91.46	89.86	65.80	91.20
µ-BEP	**82.52**	**85.82**	**86.71**	**89.91**	**88.80**	**84.80**	**86.40**	**86.06**	**88.98**

We conclude this discussion on the effectiveness of the framework with some considerations about the pattern of agreement noticed among the ten categories. First of all, CHI+GA/NBM and IG+GA/NBM turn out to be effective as they permit to reach the highest F-measure values in each category. In particular, in 9 cases out of 10 the "best predictor" is obtained by IG+GA/NBM. Hence, this implementation of the framework has turn out to be the most effective as it reaches the highest values of F-measure, starting from relatively small search spaces, and selects on average smaller subset of terms than CHI+GA/NBM. This fact can be partially explained considering the evident supremacy, in this domain, of the IG filter respect to CHI.

Finally, we give some considerations about the computational cost required by CHI+GA/NBM and by IG+GA/NBM. From this point of view, the two versions are comparable and both require a execution time that does not exceed 200 seconds.

5.3.2 On the Representation Level of the Framework

We study now the representation level of the framework, with the aim to discover the terms that provide a better understanding of the analyzed domain. Likewise section 4.3.2, we assume the frequency of each term in the selected predictors as a measure of the representation level of the framework.

In details, for each category in R10 we evaluated the frequency of the terms belonging to the 36 predictors obtained at layer 3. For each category, we analyse the ten most selected terms, counting the frequency with which they appear in the predictors, and their position in the original ranked lists obtained at layer 1.

Comparing the two lists of terms that are most frequently selected by CHI+GA/NBM and IG+GA/NBM for each category, we can see that the lists present a number of terms in common. In particular, for 7 categories out of 10, the number of feature in common is greater than 50%. Hence, in the majority of cases, the two versions CHI+GA/NBM and IG+GA/NBM agree in evaluating the importance of features, although selecting different term subsets at layer 3. Thus, we can suppose that the proposed framework can be useful to find relevant features for the problem of text analysis, permitting to have a deeper knowledge on the domain.

Moreover, we note that the lists of terms most frequently selected do not coincide with the ranked lists produced at layer one by the filters CHI and IG. Although there is not a perfect match, it is rare to find in these lists some terms that have a low ranked position. Indeed, among the most frequently selected terms we nearly always found features that have a ranked position between 1 and 10. This shows both that the use of filter techniques is an effective way to conduct feature selection on text datasets and that the framework can be useful to find relevant features for the problem of text analysis.

5.4 Concluding Remarks

In this chapter we have considered the preliminary study we made in the domain of text analysis. Our objective is to identify, within a document, the most relevant words that describe the document’s topic, thus permitting to assign the document to one or more thematic categories. We validated the framework on the Reuters corpus.

In this domain, the framework turns out to be effective as it permits to reach significant predictive performances in each category. Moreover, related results compared well with some classical learning algorithms used in this domain and with other hybrid approaches recently proposed in literature.

With regard to the representation level, results have shown that the two implementations of the framework agree in evaluating the importance of features and helps in achieving a deeper knowledge on the domain.

6 Related Work

The problem of feature selection has received a thorough treatment in data mining and machine learning. Many surveys attempt to review the field. [70] introduces the key component of feature selection review its developments with the growth of data mining. A comprehensive survey of existing feature selection techniques and a general framework for their unification can be found in [71]. [24] reviewed feature selection algorithms from a statistical learning point of view. In [29], the authors reviewed and compared the filter with the wrapper model for feature selection. In [71] the authors systematically group algorithms into categories and compare the commonalities and differences between the categories. A comparative study of algorithms for large-scale feature selection can be found in [34] and [72]. Representative feature selection algorithms are also empirically evaluated in [26][71][73][74][75][76][77] under different problem settings and from different perspectives.

Most of the feature selection algorithms approach the task as a search problem, where each state in the search specifies a distinct subset of the possible features [78]. The search problem is combined with a criterion in order to evaluate the merit of each candidate subset of features. There are a lot of possible combinations between each search procedure and each feature evaluation measure [71]. Based on how this combination is performed, feature selection algorithms can broadly fall into the filter model and the wrapper model [27]. The filter model relies on general characteristics of the data to select predictive features (i.e., features highly correlated to the target class) without involving any mining algorithm. Conversely, the wrapper model uses the predictive accuracy of a predetermined mining algorithm to give the quality of a selected feature subset, generally producing features better suited to the classification task at hand. However, it is computationally expensive for high-dimensional data [27][78].

6.1 Related Work in Microarray Analysis

Since 2001, a significant effort has been done to develop new and adapt known feature selection techniques in the context of microarray datasets [79]. In [20] authors provided a good survey for applying feature selection techniques in bioinformatics. [12] presents a review of feature selection techniques that have been employed in microarray data based cancer classification.

In this domain, because of the high dimensionality of most microarray analyses, the filter model is often preferred in gene selection due to its computational efficiency. Examples of filter methods can be found in [80][81], which implement univariate filter methods, while examples of multivariate filter methods are in [30][82]. With regard to the wrapper model, most methods use population-based, randomized search heuristics [83][84][85][86], although also a few examples use sequential search techniques [29][87]. Hybrid and more sophisticated feature selection techniques have been explored in recent microarray research efforts [20][30][32][51][53][54][56].

As promising approaches, evolutionary algorithms have been applied to microarray analysis in order to look for the optimal or near optimal set of predictive genes [84]. The use of GAs for feature selection is firstly introduced in [88]. Example of the use of GA in gene subset selection can be found in many studies: [51][52][89] address the problem of gene selection using a standard genetic algorithm which evolves populations of possible solutions, the quality of each solution being evaluated by an SVM classifier. GAs have been employed in conjunction with different classifiers, such as Neural Networks [90] and k-Nearest Neighbor [85]. Specifically, [85] proposed a genetic algorithm/k-nearest neighbors (GA/kNN) method to identify genes that can jointly discriminate normal and tumor samples. The top genes ranked by their frequency of selection through the iterations of GA are selected as the marker genes. [86] also used a GA to search the feature space and chose the gene subset with the best fitness among all generations of GAs as the optimal subset. [91] used a estimation of distribution algorithm, a general extension of GA, to select marker genes and reported good performance.

6.2 Related Work in Text Analysis

Many information retrieval, statistical classification and machine learning techniques have been applied to TC domains. A review of machine learning algorithms for text-documents classification can be found in [92]. Examples of employed techniques are Rocchio’s algorithm [37], regression models [93], K-nearest neighbor [93], Naïve Bayes [94], SVM [36][94][95], Decision trees (e.g. C4.5 decision tree algorithm [95]), and neural networks [96] etc.

However, most algorithms may not be completely suitable when the problem of high dimensionality occurs [36][93], as even a moderately sized text collection often has tens of thousands of terms which make the classification cost prohibitive for many learning algorithms that do not scale well to large problem sizes. In addition, it is known that most terms are irrelevant for the classification task and some of them even introduce noise that may decrease the overall performance [97]. Applying dimensionality reduction techniques (i.e. feature selection or feature extraction) is beneficial for the increasing scalability, reliability, efficiency and accuracy of text classification algorithms [98].

Text categorization applications generally have massive data samples and features, which makes wrapper methods rather time-consuming and impractical for these applications [67]. For this reason, the use of faster and simpler filter approaches is prominent in the domain [36][37][93][94][99][100][101]. Examples of hybrid techniques have been recently explored and have shown promising results [63][65][69][102][103].

7 Conclusions

In this thesis we proposed a hybrid feature selection framework for dimensionality reduction in high-dimensional domains. It involves a filter and a genetic wrapper, and it is organized in a multi-layer structure that permits to exploit their different evaluation criteria in different search stages. Our goal is to perform an intelligent feature selection (combining different techniques) with the double aim to achieve good classification performance and to discover different subsets of features relevant for the application domain. In addition, our framework is guided by the motivation of discovering a potentially high number of predictive subsets and, for that, it defines different subspaces of features for searching whereas existing hybrid models usually provide a single subspace.

To evaluate the proposed framework we conducted experiments on two high-dimensional domains. An extensive analysis has been carried out on four DNA-microarray datasets (Leukemia, DLBCL, Colon Cancer, and Prostate). Our approach obtains comparable or better results than other hybrid methods proposed in literature and showed that our approach is robust and effective in finding small subsets of informative genes.

We also validate the framework with a preliminary study in the domain of text analysis. Experiments on the Reuters corpus showed that the hybrid structure of framework does not fit completely domain’s issues, however results compared well both with classical learning approaches and with recent hybrid methods proposed in literature.

The chief contributions of this thesis are:

1. to propose a multi-step using different methods to combine the GA results with a classifier;
2. to propose different predictors and compare these predictors against each other;
3. to demonstrate these techniques on real-world datasets, with a thorough investigation of the underlying parameters, including tuning GA parameters, exploring and comparing feature subset size, and evaluating possible extension of other application domains.

As future work, we will verify the proposed framework by considering a variety of high-dimensional datasets with new data types, such as mobility data and data streams, since the rapid growth of mobile devices and Web-based applications poses a challenge to deal with mining these types of data in different contexts.

8 References

[1] FAYYAD, Usama; PIATETSKY-SHAPIRO, Gregory; SMYTH, Padhraic. From data mining to knowledge discovery in databases. AI magazine, 1996, 17.3: 37.

[2] BEN-BASSAT, M. Pattern recognition and reduction of dimensionality. Handbook of Statistics-II, PR Krishnaiah and LN Kanal, eds, 1982, 773-791.

[3] SIEDLECKI, Wojciech; SKLANSKY, Jack. On automatic feature selection. International Journal of Pattern Recognition and Artificial Intelligence, 1988, 2.02: 197-220.

[4] JIAWEI, Han; KAMBER, Micheline. Data mining: concepts and techniques. San Francisco, CA, itd: Morgan Kaufmann, 2001, 5.

[5] HEGDE, Priti, et al. A concise guide to cDNA microarray analysis. Biotechniques, 2000, 29.3: 548-563.

[6] TAMAYO, Pablo; RAMASWAMY, Sridhar. Cancer genomics and molecular pattern recognition. Expression profiling of human tumors: diagnostic and research applications, 2003, 73-102.

[7] BUTTE, Atul, et al. The use and analysis of microarray data. Nature reviews drug discovery, 2002, 1.12: 951-960.

[8] XIANG, Zhaoying, et al. Microarray expression profiling: Analysis and applications. Current opinion in drug discovery & development, 2003, 6.3: 384-395.

[9] RAMASWAMY, Sridhar; GOLUB, Todd R. DNA microarrays in clinical oncology. Journal of Clinical Oncology, 2002, 20.7: 1932-1941.

[10] GOLUB, Todd R. Genome-wide views of cancer. New England Journal of Medicine, 2001, 344.8: 601-602.

[11] PIATETSKY-SHAPIRO, Gregory; TAMAYO, Pablo. Microarray data mining: facing the challenges. ACM SIGKDD Explorations Newsletter, 2003, 5.2: 1-5.

[12] GEORGE, G.; RAJ, V. Cyril. Review on Feature Selection Techniques and the Impact of SVM for Cancer Classification using Gene Expression Profile. arXiv preprint arXiv:1109.1062, 2011.

[13] DUDA, R. O.; HART, P. E.; STORK, D. G. Pattern classification. 2nd edn Wiley. New York, 2000.

[14] BEN-DOR, Amir, et al. Tissue classification with gene expression profiles.Journal of Computational Biology, 2000, 7.3-4: 559-583.

[15] FUREY, Terrence S., et al. Support vector machine classification and validation of cancer tissue samples using microarray expression data. Bioinformatics, 2000, 16.10: 906-914.

[16] KELLER, Andrew D., et al. Bayesian classification of DNA array expression data. Technical Report UW-CSE-2000-08-01, Department of Computer Science and Engineering, University of Washington, 2000.

[17] KHAN, Javed, et al. Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. Nature medicine, 2001, 7.6: 673-679.

[18] ZHANG, Heping, et al. Recursive partitioning for tumor classification with gene expression microarray data. Proceedings of the National Academy of Sciences, 2001, 98.12: 6730-6735.

[19] GOLUB, Todd R., et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. science, 1999, 286.5439: 531-537.

[20] SAEYS, Yvan; INZA, Iñaki; LARRAÑAGA, Pedro. A review of feature selection techniques in bioinformatics. Bioinformatics, 2007, 23.19: 2507-2517.

[21] GUYON, Isabelle, et al. Gene selection for cancer classification using support vector machines. Machine learning, 2002, 46.1: 389-422.

[22] YANG, Feng; MAO, K. Z. Robust feature selection for microarray data based on multicriterion fusion. Computational Biology and Bioinformatics, IEEE/ACM Transactions on, 2011, 8.4: 1080-1092.

[23] KRISHNAPURAM, Balaji; CARIN, Lawrence; HARTEMINK, Alexander. 1 Gene expression analysis: Joint feature selection and classifier design. Kernel Methods in Computational Biology, 2004, 299-317.

[24] GUYON, Isabelle; ELISSEEFF, André. An introduction to variable and feature selection. The Journal of Machine Learning Research, 2003, 3: 1157-1182.

[25] STIGLIC, Gregor; KOKOL, Peter. Stability of ranked gene lists in large microarray analysis studies. Journal of Biomedicine and Biotechnology, 2010, 2010.

[26] LI, Tao; ZHANG, Chengliang; OGIHARA, Mitsunori. A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression. Bioinformatics, 2004, 20.15: 2429-2437.

[27] KOHAVI, Ron; JOHN, George H. Wrappers for feature subset selection. Artificial intelligence, 1997, 97.1: 273-324.

[28] HUI, Zhang; TU, Bao Ho; KAWASAKI, Saori. Wrapper Feature Extraction for Time Series Classification Using Singular Value Decomposition. 2005.

[29] INZA, Iñaki, et al. Filter versus wrapper gene selection approaches in DNA microarray domains. Artificial intelligence in medicine, 2004, 31.2: 91-103.

[30] XING, Eric P., et al. Feature selection for high-dimensional genomic microarray data. In: MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-. 2001. p. 601-608.

[31] WANG, Yu, et al. Gene selection from microarray data for cancer classification—a machine learning approach. Computational Biology and Chemistry, 2005, 29.1: 37-46.

[32] LEUNG, Yukyee; HUNG, Yeungsam. A multiple-filter-multiple-wrapper approach to gene selection and microarray data classification. IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB), 2010, 7.1: 108-117.

[33] HOLLAND, John H. Adaptation in natural and artificial systems, University of Michigan press. Ann Arbor, MI, 1975, 1.97: 5.

[34] KUDO, Mineichi; SKLANSKY, Jack. Comparison of algorithms that select features for pattern classifiers. Pattern recognition, 2000, 33.1: 25-41.

[35] HUSSEIN, Faten; KHARMA, Nawwaf; WARD, Rabab. Genetic algorithms for feature selection and weighting, a review and study. In: Document Analysis and Recognition, 2001. Proceedings. Sixth International Conference on. IEEE, 2001. p. 1240-1244.

[36] FORMAN, George. An extensive empirical study of feature selection metrics for text classification. The Journal of Machine Learning Research, 2003, 3: 1289-1305.

[37] SEBASTIANI, Fabrizio. Machine learning in automated text categorization.ACM computing surveys (CSUR), 2002, 34.1: 1-47.

[38] CANNAS, Laura M.; DESSÌ, Nicoletta; PES, Barbara. Knowledge Discovery in Gene Expression Data via Evolutionary Algorithms. DEXA Workshops 2011 (BIOKDD'11), 402-406.

[39] CANNAS, Laura M.; DESSÌ, Nicoletta; PES, Barbara. A Hybrid Model to Favor the Selection of High Quality Features in High Dimensional Domains. IDEAL 2011, 228-235.

[40] SHIPP, Margaret A., et al. Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. Nature medicine, 2002, 8.1: 68-74.

[41] ALON, Uri, et al. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays.Proceedings of the National Academy of Sciences, 1999, 96.12: 6745-6750.

[42] SINGH, Dinesh, et al. Gene expression correlates of clinical prostate cancer behavior. Cancer cell, 2002, 1.2: 203-209.

[43] LIU, Huan; SETIONO, Rudy. Chi2: Feature selection and discretization of numeric attributes. In: Tools with Artificial Intelligence, 1995. Proceedings., Seventh International Conference on. IEEE, 1995. p. 388-391.

[44] GOLDBERG, David E. Genetic algorithms in search, optimization, and machine learning. 1989.

[45] CANNAS, Laura M.; DESSÌ, Nicoletta; PES, Barbara. Tuning Evolutionary Algorithms in High Dimensional Classification Problems (Extended Abstract). SEBD 2010: 142-149.

[46] CANNAS, Laura M.; DESSÌ, Nicoletta; PES, Barbara. A Filter-Based Evolutionary Approach for Selecting Features in High-Dimensional Micro-array Data. Intelligent Information Processing 2010: 297-307.

[47] HALL, Mark, et al. The WEKA data mining software: an update. ACM SIGKDD Explorations Newsletter, 2009, 11.1: 10-18.

[48] PLATT, John, et al. Sequential minimal optimization: A fast algorithm for training support vector machines. 1998.

[49] AHA, David W.; KIBLER, Dennis; ALBERT, Marc K. Instance-based learning algorithms. Machine learning, 1991, 6.1: 37-66.

[50] YE, Jieping, et al. Using uncorrelated discriminant analysis for tissue classification with gene expression data. IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB), 2004, 1.4: 181-190.

[51] HUERTA, Edmundo; DUVAL, Béatrice; HAO, Jin-Kao. A hybrid GA/SVM approach for gene selection and classification of microarray data. Applications of Evolutionary Computing, 2006, 34-44.

[52] PENG, S., et al. Molecular classification of cancer types from microarray data using the combination of genetic algorithms and support vector machines. FEBS letters, 2003, 555.2: 358.

[53] WANG, Yuhang, et al. HykGene: a hybrid approach for selecting marker genes for phenotype classification using microarray gene expression data. Bioinformatics, 2005, 21.8: 1530-1537.

[54] NG, Manfred; CHAN, Laiwan. Informative gene discovery for cancer classification from microarray expression data. In: Machine Learning for Signal Processing, 2005 IEEE Workshop on. IEEE, 2005. p. 393-398.

[55] DEUTSCH, J. M. Evolutionary algorithms for finding optimal gene sets in microarray prediction. Bioinformatics, 2003, 19.1: 45-52.

[56] LIU, Juan; ZHOU, Huai-Bei. Tumor classification based on gene microarray data and hybrid learning method. In: Machine Learning and Cybernetics, 2003 International Conference on. IEEE, 2003. p. 2275-2280.

[57] ZHANG, Ji-Gang; DENG, Hong-Wen. Gene selection for classification of microarray data based on the Bayes error. BMC bioinformatics, 2007, 8.1: 370.

[58] DAGLIYAN, Onur, et al. Optimization based tumor classification from microarray gene expression data. PLoS One, 2011, 6.2: e14579.

[59] DEB, Kalyanmoy; REDDY, A. Raji. Reliable classification of two-class cancer data using evolutionary algorithms. BioSystems, 2003, 72.1-2: 111-129.

[60] YU, Lei; LIU, Huan. Redundancy based feature selection for microarray data. In:Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2004. p. 737-742.

[61] HEWETT, Rattikorn; KIJSANAYOTHIN, Phongphun. Tumor classification ranking from microarray data. BMC genomics, 2008, 9.Suppl 2: S21.

[62] KUCUKURAL, Alper, et al. Evolutionary selection of minimum number of features for classification of gene expression data using genetic algorithms. In:Genetic And Evolutionary Computation Conference: Proceedings of the 9 th annual conference on Genetic and evolutionary computation. 2007. p. 401-406.

[63] CANNAS, Laura M.; DESSÌ, Nicoletta; DESSÌ, Stefania. A Model for Term Selection in Text Categorization Problems. DEXA Workshops 2012 (TIR'12): 169-173.

[64] LEWIS, David D. Reuters-21578 text categorization test collection, distribution 1.0. http://www.research.att.com/~ lewis/reuters21578.html , 1997.

[65] PIETRAMALA, Adriana, et al. A genetic algorithm for text classification rule induction. Machine Learning and Knowledge Discovery in Databases, 2008, 188-203.

[66] MCCALLUM, Andrew, et al. A comparison of event models for naive bayes text classification. In: AAAI-98 workshop on learning for text categorization. 1998. p. 41-48.

[67] Tan, Feng, "Improving Feature Selection Techniques for Machine Learning" (2007). Computer Science Dissertations. Paper 27.

[68] SEBASTIANI, Fabrizio. Classification of text, automatic. The Encyclopedia of Language and Linguistics, 2006, 14: 457-462.

[69] RULLO, Pasquale, et al. Olex: effective rule learning for text categorization. Knowledge and Data Engineering, IEEE Transactions on, 2009, 21.8: 1118-1132.

[70] LIU, Huan, et al. Feature selection: An ever evolving frontier in data mining. In: Proc. The Fourth Workshop on Feature Selection in Data Mining. 2010. p. 4-13.

[71] LIU, Huan; YU, Lei. Toward integrating feature selection algorithms for classification and clustering. Knowledge and Data Engineering, IEEE Transactions on, 2005, 17.4: 491-502.

[72] HUA, Jianping; TEMBE, Waibhav D.; DOUGHERTY, Edward R. Performance of feature-selection methods in the classification of high-dimension data. Pattern Recognition, 2009, 42.3: 409-424.

[73] LIU, Huan; MOTODA, Hiroshi (ed.). Computational methods of feature selection. Chapman & Hall/CRC, 2007.

[74] SUN, Y.; BABBS, C. F.; DELP, E. J. A comparison of feature selection methods for the detection of breast cancers in mammograms: adaptive sequential floating search vs. genetic algorithm. In: Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the. IEEE, 2006. p. 6532-6535.

[75] LAI, Carmen, et al. A comparison of univariate and multivariate gene selection techniques for classification of cancer datasets. BMC bioinformatics, 2006, 7.1: 235.

[76] YU, Lei; LIU, Huan. Feature selection for high-dimensional data: A fast correlation-based filter solution. In: MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-. 2003. p. 856.

[77] YU, Lei; LIU, Huan. Efficient feature selection via analysis of relevance and redundancy. The Journal of Machine Learning Research, 2004, 5: 1205-1224.

[78] BLUM, Avrim L.; LANGLEY, Pat. Selection of relevant features and examples in machine learning. Artificial intelligence, 1997, 97.1: 245-271.

[79] JAFARI, Peyman; AZUAJE, Francisco. An assessment of recently published gene expression data analyses: reporting experimental design and statistical factors.BMC Medical Informatics and Decision Making, 2006, 6.1: 27.

[80] BALDI, Pierre; LONG, Anthony D. A Bayesian framework for the analysis of microarray expression data: regularized t-test and statistical inferences of gene changes. Bioinformatics, 2001, 17.6: 509-519.

[81] FOX, Richard J.; DIMMIC, Matthew W. A two-sample Bayesian t-test for microarray data. BMC bioinformatics, 2006, 7.1: 126.

[82] MAMITSUKA, Hiroshi. Selecting features in microarray classification using ROC curves. Pattern Recognition, 2006, 39.12: 2393-2404.

[83] BLANCO, Rosa, et al. Gene selection for cancer classification using wrapper approaches. International Journal of Pattern Recognition and Artificial Intelligence, 2004, 18.08: 1373-1390.

[84] JIRAPECH-UMPAI, Thanyaluk; AITKEN, Stuart. Feature selection and classification for microarray data analysis: Evolutionary methods for identifying predictive genes. BMC bioinformatics, 2005, 6.1: 148.

[85] LI, Leping, et al. Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the GA/KNN method. Bioinformatics, 2001, 17.12: 1131-1142.

[86] OOI, C. H.; TAN, Patrick. Genetic algorithms applied to multi-class prediction for the analysis of gene expression data. Bioinformatics, 2003, 19.1: 37-44.

[87] XIONG, Momiao; FANG, Xiangzhong; ZHAO, Jinying. Biomarker identification by feature wrappers. Genome Research, 2001, 11.11: 1878-1887.

[88] SIEDLECKI, Wojciech; SKLANSKY, Jack. A note on genetic algorithms for large-scale feature selection. Pattern Recognition Letters, 1989, 10.5: 335-347.

[89] TAN, Feng, et al. Improving feature subset selection using a genetic algorithm for microarray gene expression data. In: Evolutionary Computation, 2006. CEC 2006. IEEE Congress on. IEEE, 2006. p. 2529-2534.

[90] BEVILACQUA, Vitoantonio, et al. Genetic algorithms and artificial neural networks in microarray data analysis: a distributed approach. Engineering Letters, 2006, 13.3: 335-343.

[91] SAEYS, Yvan, et al. Fast feature selection using a simple estimation of distribution algorithm: a case study on splice site prediction. Bioinformatics, 2003, 19.suppl 2: ii179-ii188.

[92] BAHARUDIN, Baharum; LEE, Lam Hong; KHAN, Khairullah. A review of machine learning algorithms for text-documents classification. Journal of Advances in Information Technology, 2010, 1.1: 4-20.

[93] YANG, Yiming; PEDERSEN, Jan O. A comparative study on feature selection in text categorization. In: MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-. MORGAN KAUFMANN PUBLISHERS, INC., 1997. p. 412-420.

[94] WANG, Gang; LOCHOVSKY, Frederick H. Feature selection with conditional mutual information maximin in text categorization. In: Proceedings of the thirteenth ACM international conference on Information and knowledge management. ACM, 2004. p. 342-349.

[95] JOACHIMS, Thorsten. Text categorization with support vector machines: Learning with many relevant features. Machine learning: ECML-98, 1998, 137-142.

[96] YANG, Yiming; LIU, Xin. A re-examination of text categorization methods. In:Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 1999. p. 42-49.

[97] SALTON, Gerard; MCGILL, Michael J. Introduction to modern information retrieval. 1986.

[98] LEWIS, David D. Feature selection and feature extraction for text categorization. In: Proceedings of the workshop on Speech and Natural Language. Association for Computational Linguistics, 1992. p. 212-217.

[99] COMBARRO, Elias F., et al. Introducing a family of linear measures for feature selection in text categorization. Knowledge and Data Engineering, IEEE Transactions on, 2005, 17.9: 1223-1232.

[100] OLSSON, J.; OARD, Douglas W. Combining feature selectors for text classification. In: Proceedings of the 15th ACM international conference on Information and knowledge management. ACM, 2006. p. 798-799.

[101] ROGATI, Monica; YANG, Yiming. High-performing feature selection for text classification. In: Proceedings of the eleventh international conference on Information and knowledge management. ACM, 2002. p. 659-661.

[102] LIU, Jihong; WANG, Guoxiong. A hybrid feature selection method for data sets of thousands of variables. In: Advanced Computer Control (ICACC), 2010 2nd International Conference on. IEEE, 2010. p. 288-291.

[103] UĞUZ, Harun. A two-stage feature selection method for text categorization by using information gain, principal component analysis and genetic algorithm. Knowledge-Based Systems, 2011, 24.7: 1024-1032.