A Feature Vector Selection Method for Cancer Classification

Zheng Yun and Kwoh Chee Keong

Bioinformatics Research Center, School of Computer Engineering, Nanyang Technological University, Singapore 639798

Email: pg04325488@ntu.edu.sg, asckkwoh@ntu.edu.sg

ABSTRACT: The high-dimensionality and insufficiency of gene expression profiles and proteomic profiles makes feature selection become a critical step in efficiently building accurate models for cancer problems based on such data sets. In this paper, we use a method, called Discrete Function Learning algorithm, to find discriminatory feature vectors based on information theory. The target feature vectors contain all or most information (in terms of entropy) of the class attribute. Two data sets are selected to validate our approach, one leukemia subtype gene expression data set and one ovarian cancer proteomic data set. The experimental results show that the our method generalizes well when applied to these insufficient and high-dimensional data sets. Furthermore, the obtained classifiers are highly understandable and accurate.

1 Introduction

The inclusion of irrelevant, redundant and noisy attributes in the model building process phase can result in poor predictive performance and increased computation. Gene expression profiles and proteomic profiles are often noisy and contain thousands of features, many of these features are not closely related to the class distinctions between tissue samples [1, 2]. Therefore, feature selection is critical for successfully classifying tissue samples based on gene expression profiles or proteomic profiles, which are often very high-dimensional and insufficient.

Feature selection methods fall into two main categories, those evaluating individual features and those evaluating feature subsets.

In the individual feature selection methods, the evaluation statistics for each feature are calculated, then a feature ranking list is provided in predefined order of the statistics. The statistics used for individual feature selection include information gain [3, 4], signal-to-noise (S2N) statistic [2], correlation coefficient (CC) [5], t-statistic [4], χ^2-statistic [6, 4]. The main shortcoming of these individual feature selection methods lies in that a larger than necessary number of redundant top features with similar gene expression patterns are selected to build the models. Hence, such choice often brings much redundancy to the models, since the selected features carry similar information about the class attribute. According to the principle of Occam's razor, these models are not optimal although accurate, since they are often complex and suffer the risk of overfitting the data sets [7]. In addition, the large number of genes in the predictors makes it difficult to know which genes are really useful for recognizing different classes.

In the feature subset selection method, a search algorithm is often employed to find the optimal feature subsets. In evaluating a feature subset, a predefined score is calculated for the feature subset. Since the number of feature subsets grows exponentially with the number of features, heuristic searching algorithms, such as forward selection, are often employed to solve the problem. Examples of feature subset selection methods are CFS (Correlation-based Feature Selection) [8], CSE (Consistency-based Subset Evaluation) [9], the WSE (Wrapper Subset Evaluation) [10]. Most feature subset selection methods use heuristic scores to evaluate feature subset under consideration, such as CFS and CSE methods. The WSE method evaluates a subset of genes by applying a target learning algorithm to the training data set with cross validation, and selects the subset of genes which produces the highest accuracy in the cross validation process. The evaluation with cross validation makes the WSE very inefficient when meeting the high-dimensional data sets like gene expression profiles.

There is another popular way of categorizing these algorithms, called "filter" and "wrapper" methods [11], based on the different nature of the metric used to evaluate features. In the filter methods, the feature selection is performed as a preprocessing step and often independent of the classification algorithms which will be applied to the processed data sets later. The WSE method mentioned above is the wrapper method.

In this paper, we use the Discrete Function Learning algorithm [12] to find discriminatory feature vectors based on information theory. The target feature vectors are supposed to contain all or most information (in terms of entropy) of the class attribute. We name the subset of the attributes (genes) in the most discriminatory gene vectors as the *essential attributes*, or the EAs for short. After the learning process, the DFL algorithm provides the classifiers as function tables which contain the EAs and the class attribute. To make use of the obtained function tables reasonably, the predictions are performed in the space defined by the EAs, called the *EA space*, with the 1-Nearest-Neighbor (1NN) algorithm [13]. Specifically, in predicting a new sample, the Hamming distances [14] (for binary and non-binary cases) of the EAs between the new sample and each rule of the classifier are calculated. Then, the classifier selects the class value of the rule which has the minimum Hamming distance to the new sample as the predicted class value.

The rest of this paper is organized as follows. In section 2, we provide the theory foundation and related work of our method. In section 3, we briefly describe the DFL algorithm. In section 4, we show the experimental results. In section 5, we summarize the paper.

2 Background

We will first introduce some notation. We use capital letters to represent discrete random variables, such as X and Y; lower case letters to represent an instance of the random variables, such as x and y; bold capital letters, like $\mathbf{X}$, to represent a vector; and lower case bold letters, like $\mathbf{x}$, to represent an instance of $\mathbf{X}$. The cardinality of $\mathbf{X}$ is represented with $|\mathbf{X}|$. In the remainder parts of this paper, we denote the attributes except the class attribute as a set of discrete random variables $\mathbf{V} = \{X_1, \ldots, X_n\}$, the class attribute as variable Y.

In this section, we first introduce necessary background knowledge of information theory. Then, we will discuss related feature selection methods based on information theory and analyze their shortcomings.

2.1 Theoretic Background

The entropy of a discrete random variable X is defined in terms of probability of observing a particular value x of X as [15]:

$$H(X) = -\sum_x P(X=x) log P(X=x).$$

The entropy is used to describe the diversity of a variable or vector. The more diverse a variable or vector is, the larger entropy they will have. Hereafter, for the purpose of simplicity, we represent $P(X=x)$ with $p(x)$, $P(Y=y)$ with $p(y)$, and so on. The mutual information between a vector $\mathbf{X}$ and Y is defined as [15]:

$$I(\mathbf{X};Y) = H(Y) - H(Y|\mathbf{X}) = H(\mathbf{X}) - H(\mathbf{X}|Y)$$
$$= H(\mathbf{X}) + H(Y) - H(\mathbf{X},Y) \quad (1)$$

Basically, the stronger the relation between two variables, the larger mutual information they will have. Zero mutual information means the two variables are independent or have no relation.

The conditional mutual information $I(X;Y|Z)$ [16](the mutual information between X and Y given Z) is defined by

$$I(X;Y|Z) = \sum_{x,y,z} p(x,y,z) \frac{p(x,y|z)}{p(x|z)p(y|z)}.$$

The chain rule for mutual information is give by Theorem 1, for which the proof is available in [16].

Theorem 1

$$I(X_1, X_2, \ldots, X_n; Y) = \sum_{i=1}^{n} I(X_i; Y | X_{i-1}, X_{i-2}, \ldots, X_1). \quad (2)$$

2.2 Related Work

Some feature selection methods based on mutual information have been introduced. These methods also fall into two categories.

In the first category, features are ranked according to their mutual information with the class label. Then, the first k features [17] or the features with a bigger mutual information than a predefined threshold value [18] are chosen.

The second category is feature subset selection methods. In this category, the forward selection searching algorithm is often used to find the predefined k features. In the first iteration, the X_i which shares the largest mutual information with Y is selected to the target feature subset $\mathbf{U}$. Then, in the next step, the selection criterion is how much information can be added with respect to the already existing $X_{(1)}$. Therefore, the $X_{(2)}$ with maximum $I(X_i, X_{(1)}; Y) - I(X_{(1)}; Y)$ is added to $\mathbf{U}$ [19]. Formally, the features $X_{(1)}, \ldots, X_{(k)}$ are selected with the following criteria, $X_{(1)} = argmax_i I(X_i; Y)$ and

$$X_{(l)} = argmax_{X_i \in \mathbf{P}_l} min_{X_{(j)} \in \mathbf{U}_l} I(X_i, X_{(j)}; Y) - I(X_{(j)}; Y) \quad (3)$$

where $\forall l, 1 < l \leq k, i = 1, \ldots, (n-l+1), j = 1, \ldots, (l-1)$, and $\mathbf{P}_l$ is the feature pool by removing $X_{(1)}, \ldots, X_{(l)}$, $\mathbf{P}_1 = \mathbf{V} \setminus X_{(1)}$, $\mathbf{P}_{l+1} = \mathbf{P}_l \setminus X_{(l)}$, and $\mathbf{U}_l$ is the set of selected features $\mathbf{U}_1 = \{X_{(1)}\}$, $\mathbf{U}_{l+1} = \mathbf{U}_l \cup \{X_{(l)}\}$.

From Theorem 1, we have $I(X_i, X_{(j)}; Y) = I(X_{(j)}; Y) + I(X_i; Y | X_{(j)})$, then

$$I(X_i; Y | X_{(j)}) = I(X_i, X_{(j)}; Y) - I(X_{(j)}; Y). \quad (4)$$

Therefore, Equation 3 is equivalent to maximizing conditional mutual information, $min_{X_{(j)} \in \mathbf{U}} I(X_i; Y | X_{(j)})$ [20] in Equation 4.

For all subset selection method mentioned above, one major shorting coming is that the candidate feature is compared to all the selected features in $\mathbf{U}$, one-by-one. The motivation underlying Equation 3 and 4 is that X_i is good only if it carries information about Y, and if this information has not been caught by any of the $X_{(j)}$ already picked [20]. However, it can not be known whether the existing features as a vector have capture the information carried by X_i or not.

3 Methods

In this section, we first discuss the motivation of our method, then briefly describe the DFL algorithm.

3.1 Motivation

$I(\mathbf{X};Y)$ is evaluated with respect to $H(Y)$ in the DFL algorithm, which is different from those in existing methods, as shown in Equation 5. Suppose that $\mathbf{X}$ is the already selected feature subset in $\mathbf{U}$, and the DFL algorithm is trying to add a new feature Z to $\mathbf{U}$, $X_{(1)} = argmax_i I(X_i; Y)$, and

$$X_{(l)} = argmax_Z I(\mathbf{X}, Z; Y), \quad (5)$$

where $\forall l$, $1 < l \leq k$, $\mathbf{U}_1 = \{X_{(1)}\}$, and $\mathbf{U}_{l+1} = \mathbf{U}_l \cup \{X_{(l)}\}$. From Theorem 1, we have

$$I(\mathbf{X}, Z; Y) = I(\mathbf{X}; Y) + I(Z; Y | \mathbf{X}). \quad (6)$$

In Equation 6, note that $I(\mathbf{X};Y)$ does not change when trying different Z. Hence, the maximization of $I(\mathbf{X}, Z; Y)$ in the DFL algorithm is actually maximizing $I(Z; Y|\mathbf{X})$, the conditional mutual information of Z and Y given the already selected features $\mathbf{X}$, i.e., the information of Y not captured by $\mathbf{X}$ but carried by Z. Equation 6 is different from Equation 4 used in [20], where the new feature is evaluated with respect to individual features in $\mathbf{U}$. By considering the selected features as vectors,

the redundancy introduced by new features to be added to $\mathbf{U}$ is automatically eliminated.

Let us further investigate the measure, $I(Z;Y|\mathbf{X})$. From Equation 1, we have

$$I(Z;Y|\mathbf{X}) = H(Y|\mathbf{X}) - H(Y|Z,\mathbf{X}). \quad (7)$$

Similar to Equation 6, $H(Y|\mathbf{X})$ does not change when trying different Z. As pointed out by Fleuret [20], the ultimate goal of feature subset selection is to find $\{Z,\mathbf{X}\}$ which minimizes $H(Y|Z,\mathbf{X})$. But $H(Y|Z,\mathbf{X})$ can not be estimated with a training set of realistic size as it requires the estimation of 2^{k+1} probabilities [20]. Hence, the authors of [20, 19] proposed the estimated increase of the information content of the feature subset using Equation 3 and 4. However, from Equation 6 and 7, it can be seen that it is not necessary to compute the $H(Y|Z,\mathbf{X})$, as the problem can be directly solved by maximizing $I(\mathbf{X},Z;Y)$ as implemented in the DFL algorithm.

Furthermore, the evaluation of the feature subsets is more efficient than penalizing the new feature with respect to every selected features, as done in [20, 19]. In the DFL algorithm $O(k \cdot n \cdot N)$ operations are necessary to choose k features. However, in method of [20, 19], it needs $O(k^2 \cdot n \cdot N)$ operations to select k features, which is less efficient.

To measure which subset of genes is optimal, we restate the following theorem, which is the theoretical foundation of our algorithm.

Theorem 2 *If the mutual information between* $\mathbf{X}$ *and* Y *is equal to the entropy of* Y, *i.e.,* $I(\mathbf{X};Y) = H(Y)$, *then* Y *is a function of* $\mathbf{X}$.

Proof of Theorem 2 is given in our early work [21]. The entropy $H(Y)$ represents the diversity of the variable Y. The mutual information $I(\mathbf{X};Y)$ represents the relation between vector $\mathbf{X}$ and Y. From this point of view, Theorem 2 actually says that the relation between vector $\mathbf{X}$ and Y are very strong, such that there is no more diversity for Y if $\mathbf{X}$ has been known. In other words, the value of $\mathbf{X}$ can fully determine the value of Y.

3.2 Training Methods

A classification problem is trying to learn or approximate a function, which takes the values of attributes (except the class attribute) in a new sample as input and output a categorical value which indicates the class of the sample under consideration, from a given training data set. The goal of the training process is to obtain a function which makes the output value of this function be the class value of the new sample as accurately as possible. From Theorem 2, the problem is converted to finding a subset of attributes $\mathbf{U} \subseteq \mathbf{V}$ whose mutual information with Y is equal to the entropy of Y. The $\mathbf{U}$ is the EAs which we are trying to find from the data sets. For n discrete variables, there are totally 2^n subsets. Clearly, it is NP-hard to examine all possible subsets exhaustively. However, in the cancer classification problems, only a small set of genes of the human genome are responsible for the tumor cell developmental pathway [1]. Therefore, it is reasonable to reduce the searching space by considering those subsets with limited number of genes.

The detailed steps of the DFL algorithm is available in our early work [12]. Here, we will briefly describe the main steps of the DFL algorithm as shown in the following.

1. $\forall X_i \in \mathbf{V}$, compute $I(X_i;Y)$;
2. add $A = X_i$ with largest $I(X_i;Y)$ to the EA set $\mathbf{U}$;
3. $\forall X_i \in \mathbf{V} \backslash \mathbf{U}$, compute $I(\mathbf{U},X_i;Y)$;
4. add $B = X_i$ with largest $I(\mathbf{U},X_i;Y)$ to the EA set $\mathbf{U}$;
5. repeat 3-4, until find $\mathbf{U}$ so that $I(\mathbf{U};Y) = H(Y)$.

The DFL algorithm will find the most informative feature A in the first step. Then, the DFL algorithm will try every subsets with A and another remaining feature in $\mathbf{V}$, and find the most informative feature subset $\{A,B\}$ in the second step. Next, the similar calculation will be done until the target combination $\mathbf{U}$, which satisfies the criterion of Theorem 2, is found.

After $\mathbf{U}$ is found, the DFL algorithm will stop its searching process, and obtain the classifiers by deleting the non-essential attributes and duplicate rows in the training data sets.

3.3 The ε Value Method

We also introduce a method called ε value to overcome the noisy problems [12]. Theorem 2, the exact functional relation demands the strict equality between the entropy of Y, $H(Y)$ and the mutual information of $\mathbf{X}$ and Y, $I(\mathbf{X};Y)$. However, this equality is often ruined by the noisy data, like microarray gene expression data. In these cases, we have to relax the requirement to obtain a best estimated result. By defining a significant factor ε, if the difference between $I(\mathbf{X};Y)$ and $H(Y)$ is less than or equal to $\varepsilon \times H(Y)$, then the DFL algorithm will stop the searching process, and build the classifier for Y with $\mathbf{X}$ at the significant level ε.

3.4 Prediction Methods

After the DFL algorithm obtaining the classifiers as function tables of the pairs $\{\mathbf{u} \rightarrow y\}$, the most reasonable way to use such function tables is to check the input values $\mathbf{u}$, then find the corresponding output values y. Therefore, we perform predictions in the *EA space*, with the 1NN algorithm based on the Hamming distance defined as follows.

Definition 1 *Let* $1(a,b)$ *be an indicator function, which is 0 if and only if* $a = b$, *otherwise is 1. The Hamming distance between two arrays* $\mathbf{A} = [a_1,\ldots,a_n]$ *and* $\mathbf{B} = [b_1,\ldots,b_n]$ *is* $Dist(\mathbf{A},\mathbf{B}) = \sum_{i=1}^{n} 1(a_i,b_i)$.

Note that the Hamming distance [14] is dedicated to binary arrays, however, we do not differentiate between binary or non-binary cases in this paper. We use the Hamming distance as a criterion to decide the class value of a new sample, since we believe that the rule with minimum Hamming distance to the EA values of a sample contains the maximum information of the sample. Thus, the class value of this rule is the best prediction for the sample.

In the prediction process, if a new sample has same distance to several rules, we choose the rule with the biggest count values.

Data Set	Original #	# After Discret.	k^* of DFL
T-ALL OVA	12558	1309	1
E2A. OVA	12558	718	1
TEL. OVA	12558	1309	2
BCR. OVA	12558	84	2
MLL OVA	12558	327	3
Hyper. OVA	12558	914	5
Ovarian	15154	6088	1

Table 1: The summary of the number of genes in the selected data sets.

	DFL	C4.5	NB	kNN	SVM
T-ALL	100	100/99.1	100/86.6	100/97.3	100/100
E2A.	100	100/100	100/92.0	100/99.1	100/100
TEL.	97.3	94.6/94.6	100/76.8	97.3/99.1	100/91.1
BCR.	98.2	94.6/92.9	97.3/94.6	97.3/94.6	98.2/96.4
MLL	98.2	95.5/96.4	100/94.6	100/97.3	100/100
Hyper.	93.8	85.7/90.2	99.1/79.5	99.1/96.4	99.1/83.9
sub avg.	97.9	95.1/95.5	99.4/87.4	99.0/97.3	99.6/95.2
Ovarian	98.8	97.6/92.9	84.5/76.2	91.7/92.9	98.8/100
total avg.	98.0	95.4/95.2	97.7/86.1	98.2/96.8	99.5/95.8

Table 2: The accuracies of the DFL algorithm and other well-known classification algorithms. Except the DFL algorithm, the results are for discretized/continuous data sets.

4 Results

In this section, we first introduce the data sets used in this study, then show the results.

4.1 Data Sets

To validate our method, we choose two data sets, the leukemia subtype gene expression profiles (Leukemia) [22] and the ovarian cancer proteomic mass spectromic profiles obtained with the Surface-enhanced laser desorption ionization time-of-flight (SELDI-TOF) mass spectrometry technology (Ovarian) [23].

The Leukemia data sets consist of 327 expression profiles of acute lymphoblastic leukemia (ALL), each sample with 12558 genes. The 327 samples contain all known ALL subtypes, including T-cell (T-ALL), E2A-PBX1, TEL-AML1, MLL, BCR-ABL, and hyperdiploid (Hyperdip > 50). Each class is evaluated against other classes with one of the 6 one-vs-all (OVA) classifiers. The training/testing ratio of the Leukemia data sets are 215:112.

The Ovarian data sets consists of 253 SELDI-TOF mass spectromic profiles, each sample with 15154 features. The 253 samples consist two classes, 91 the non-cancer control samples and 162 ovarian cancer samples. The training/testing ratio of the Ovarian data set is 169:84.

4.2 Results

We implement the DFL algorithm with the Java language version 1.4.1. All experiments are performed on an HP *AlphaServer* SC computer, with one EV68 1GHz CPU and 1GB memory, running the *Tru64* Unix operating system.

First, we use a widely used discretization method [24] based on entropy to discretize the selected data. This method has been implemented by the *Weka*[1] software [25]. The discretization is carried out in such a way that the training data set is first discretized. Then the testing data set is discretized according to the cutting points of genes determined in the training data set. The number of genes with more than one expression intervals, and the number of genes chosen by the DFL algorithm, i.e. the actual cardinality k^* of our classifiers, are shown in Table 1. As expected, the discretization method remove substantial amount of genes which are irrelevant to the class distinctions.

[1]The *Weka* software, available at http://www.cs.waikato.ac.nz/~ml/weka/, is written with the Java language and is an open source software issued under the GNU General Public License.

Then, we apply the DFL algorithm to the discretized data sets. To get optimal model, we change the ε value from 0 to 0.6, with a step of 0.01. For each ε value, we train a model with the DFL algorithm, then validate its performance for the testing data sets. In our implementation of the DFL algorithm, the optimal model can be automatically chosen.

We use the *Weka* software (version 3.4) to evaluate the performances of other classification methods. Specifically, we compare the DFL algorithm with the C4.5 algorithm by Quinlan [26], the Naive Bayes (NB) algorithm by Langley *et al.* [27], the k-Nearest-Neighbors (kNN) algorithm by Aha *et al.* [13] and the Support Vector Machines (SVM) algorithm by Platt [28]. All these methods are implemented in the *Weka* software.

In Table 2, we first show the accuracies for the DFL algorithm and other well-known classification algorithms. As shown in Table 2, the DFL algorithm obtains better prediction performances than other algorithms when they are applied to continuous data sets. For discretized data sets, the DFL algorithm performs better than the C4.5 and NB algorithm, approximately equally to the kNN algorithm, and worse than the SVM algorithm.

The prediction performance is only one aspect of the classifiers, but not all. Next, we compare the model complexities of different methods. From Table 1, it can be seen the classifiers of our method are very simple, with only a few genes. The model from the C4.5 algorithm is comparable to our models, but the performances of the C4.5 algorithm are not better than our method. The NB, kNN and SVM algorithms build very complex models, using all genes of the data sets. The complex models from these algorithms make it difficult for the users to understand which set of genes is really important in contributing to the class distinctions between samples.

In Table 3, we also compare the performances of the DFL algorithm for the Leukemia data sets with those in the original publication [22]. As shown in Table 3, the DFL algorithm obtains comparable prediction performances to the SVM algorithm with various feature selection methods in [22]. Yeoh et al. [22] also applied other algorithms, like kNN and artificial neural networks (ANN), on the Leukemia data sets, and obtained similar accuracies as those from the SVM algorithm shown in Table 3.

Next, we also compare the model complexity of the DFL algorithm and the methods in [22]. As shown in Table 3, the CFS and top-ranking feature selection methods chose more

	DFL		Methods in Literature [22]			
Data Set	k^*	Ac.	Ac.	Al.	F.S.	k^*
T-ALL OVA	1	100	100	SVM	CFS	1
E2A. OVA	1	100	100	SVM	CFS	1
TEL. OVA	2	97	99	SVM	CFS	105
BCR. OVA	2	98	97	SVM	CFS	53
MLL OVA	3	98	98	SVM	CFS	93
Hyper. OVA	5	94	96	SVM	CFS	96
average		98	98			
T-ALL OVA	1	100	100	SVM	χ^2	20
E2A. OVA	1	100	100	SVM	χ^2	20
TEL. OVA	2	97	99	SVM	χ^2	20
BCR. OVA	2	98	95	SVM	χ^2	20
MLL OVA	3	98	100	SVM	χ^2	20
Hyper. OVA	5	94	96	SVM	χ^2	20
average		98	98			
T-ALL OVA	1	100	100	SVM	t	20
E2A. OVA	1	100	100	SVM	t	20
TEL. OVA	2	97	99	SVM	t	20
BCR. OVA	2	98	94	SVM	t	20
MLL OVA	3	98	100	SVM	t	20
Hyper. OVA	5	94	96	SVM	t	20
average		98	97			

Table 3: The comparison of the DFL algorithm and other methods in original publication [22]. The column names k^*, Ac., Al. and F.S. stand for the number of genes in the classifiers, the accuracy, the algorithm used, and the feature selection method respectively. For the F.S. column, the CFS , χ^2 and t represent the correlation-based feature selection [8], top-ranking with χ^2 test, and top-ranking with t test respectively.

Affy. NO.	Name	Description
T-ALL OVA		
38319-at	CD3D	CD3D antigen delta polypeptide TiT3 complex
E2A-PBX1 OVA		
32063-at	PBX1	pre-B-cell leukemia transcription factor 1
TEL-AML1 OVA		
41442-at	CBFA2T3	core-binding factor runt domain alpha subunit 2 translocated to 3
38652-at	FLJ20154	hypothetical protein FLJ20154
BCR-ABL OVA		
37600-at	ECM1	extracellular matrix protein 1
1636-g-at	ABL	Human proto-oncogene tyrosine-protein kinase (ABL) gene, exon 1a and exons 2-10, complete cds.
MLL OVA		
32475-at		Homo sapiens leucocyte immunoglobulin-like receptor-6b (LIR-6) mRNA, complete cds
36777-at	D12S2489E	DNA segment on chromosome 12 unique 2489 expressed sequence
34306-at	MBNL	muscleblind Drosophila like
Hyperdip>50 OVA		
31444-s-at		Human lipocortin (LIP) 2 pseudogene mRNA, complete cds
31492-at	M9	muscle specific gene
38717-at	DKFZP586 A0522	DKFZP586A0522 protein
39003-at		H.sapiens mRNA for surface glycoprotein
40570-at	FOXO1A	forkhead box O1A rhabdomyosarcoma

Table 4: The genes selected by the DFL algorithm for the Leukemia data sets.

features than the DFL algorithm does. Therefore, the SVM algorithm in [22] built more complex models than the DFL algorithm does. According to the principle of Occam's razor, the models of the DFL algorithm are more preferable to the SVM models in [22].

Furthermore, we investigate the biological role of the genes chosen by the DFL algorithm for the Leukemia data sets. As shown in Table 4, the *CDC3* and *PBX1* gene for the T-ALL and E2A-PBX1 OVA classifiers respectively are biologically relevant for the specific leukemia subtype development. They are also informative and discriminatory, as shown by the 100 prediction accuracies for these two OVA classifiers in Table 2. Actually, the CFS feature selection method used in [22] also chose these two genes for these two OVA classifiers. The *ABL* gene for the BCR-ABL OVA classifier is also closely related to the development of the specific leukemia subtype BCR-ABL.

Finally, we compare the training time of different classification methods in Table 5. Since all compared algorithms are implemented with the Java language and all experiments are performed on the same computer, the comparisons of their efficiency are meaningful. As shown in Table 5, the DFL algorithm is more efficient than other compared classification methods in most cases.

5 Conclusion

In this study, we validate the DFL algorithm with two data sets, the leukemia subtype gene expression profiles and the ovarian cancer proteomic mass spectromic profiles. The DFL algorithm achieves comparable or more competitive prediction performances than those of some other well-known classification methods with very simple and understandable rules, which suggests that the DFL algorithm generalizes well in the high-dimensional and sparse data sets, like gene expression profiles and proteomic profiles.

REFERENCES

[1] J.-P. Mira, V. Benard, J. Groffen, L. C. Sanders, and U. G. Knaus, "Endogenous, hyperactive Rac3 controls proliferation of breast cancer cells by a p21-activated kinase-dependent pathway," *PNAS*, vol. 97, no. 1, pp. 185–189, 2000.

[2] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R.

Data Set	DFL	C4.5	NB	kNN	SVM
T-ALL	0.05	0.21/6.61	0.11/5.40	0.90/4.42	0.73/15.83
E2A.	0.02	0.14/6.48	0.06/5.49	0.51/4.52	0.42/16.77
TEL.	0.17	0.41/12.24	0.11/5.49	0.92/4.43	0.96/18.90
BCR.	0.02	0.11/12.78	0.02/5.42	0.08/4.51	0.18/18.35
MLL	0.08	0.21/12.71	0.06/5.47	0.24/4.54	0.27/18.76
Hyper.	1.30	0.47/17.89	0.08/5.44	0.65/4.56	0.81/22.55
Ovarian	0.31	1.15/16.01	0.41/5.05	3.69/6.85	3.99/12.25

Table 5: The training time of the DFL algorithm and other well-known classification methods. Except the DFL algorithm, the results are for the discretized/continuous data sets. The unit is second.

Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander, "Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring," *Science*, vol. 286, no. 5439, pp. 531–537, 1999.

[3] M. A. Hall and G. Holmes, "Benchmarking attribute selection techniques for discrete class data mining," *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, pp. 1–16, 2003.

[4] H. Liu, J. Li, and L. Wong, "A comparative study on feature selection and classification methods using gene expression profiles and proteomic patterns," *Genome Informatics*, vol. 13, pp. 51–60, 2002.

[5] L. van 't Veer, H. Dai, M. van de Vijver, Y. He, A. Hart, M. Mao, H. Peterse, K. van der Kooy, M. Marton, A. Witteveen, G. Schreiber, R. Kerkhoven, C. Roberts, P. Linsley, R. Bernards, and S. Friend, "Gene expression profiling predicts clinical outcome of breast cancer," *Nature*, vol. 415, pp. 530 – 536, 2002.

[6] J. Li, H. Liu, J. R. Downing, A. E.-J. Yeoh, and L. Wong, "Simple rules underlying gene expression profiles of more than six subtypes of acute lymphoblastic leukemia (ALL) patients," *Bioinformatics*, vol. 19, no. 1, pp. 71–78, 2003.

[7] E. P. Xing, M. I. Jordan, and R. M. Karp, "Feature selection for high-dimensional genomic microarray data," in *Proceedings of the 18th International Conference on Machine Learning*. Morgan Kaufmann Publishers Inc., 2001, pp. 601–608.

[8] M. Hall, "Correlation-based feature selection for machine learning," Ph.D. dissertation, Waikato University, Department of Computer Science, 1999.

[9] H. Liu and R. Setiono, "A probabilistic approach to feature selection - a filter solution," in *International Conference on Machine Learning*, 1996, pp. 319–327.

[10] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial Intelligence*, vol. 97, pp. 273–324, 1997.

[11] G. H. John, R. Kohavi, and K. Pfleger, "Irrelevant features and the subset selection problem," in *International Conference on Machine Learning*, pp. 121–129.

[12] Y. Zheng and C. K. Kwoh, "Identifying decision lists with the discrete function learning algorithm," in *Proceedings of the 2nd International Conference on Artificial Intelligence in Science And Technology, AISAT 2004*, 2004, pp. 30–35.

[13] D. W. Aha, D. Kibler, and M. K. Albert, "Instance-based learning algorithms," *Machine Learning*, vol. 6, pp. 37–66, 1991.

[14] R. Hamming, "Error detecting and error correcting codes," *Bell System Technical Jounral*, vol. 9, pp. 147–160, 1950.

[15] C. Shannon and W. Weaver, *The Mathematical Theory of Communication*. Urbana, IL.: University of Illinois Press, 1963.

[16] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. John Wiley & Sons, Inc., 1991.

[17] S. T. Dumais, J. C. Platt, D. Hecherman, and M. Sahami, "Inductive learning algorithms and representations for text categorization." in *CIKM*, 1998, pp. 148–155.

[18] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization," in *Proceedings of ICML-97, 14th International Conference on Machine Learning*, D. H. Fisher, Ed. Nashville, US: Morgan Kaufmann Publishers, San Francisco, US, 1997, pp. 412–420.

[19] M. Vidal-Naquet and S. Ullman, "Object recognition with informative features and linear classification." in *ICCV*, 2003, pp. 281–288.

[20] F. Fleuret, "Fast binary feature selection with conditional mutual information," *J. Mach. Learn. Res.*, vol. 5, pp. 1531–1555, 2004.

[21] Y. Zheng and C. K. Kwoh, "Dynamic algorithm for inferring qualitative models of gene regulatory networks," in *Proceedings of the 3rd Computational Systems Bioinformatics Conference, CSB 2004*. IEEE Computer Society Press, 2004, pp. 353–362.

[22] E.-J. Yeoh et al., "Classification and subtype discovery and and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling," *Cancer Cell*, vol. 1, pp. 133–143, 2002.

[23] I. E. F Petricoin, A. M. Ardekani, B. A. Hitt, P. J. Levine, V. A. Fusaro, S. M. Steinberg, G. B. Mills, C. Simone, D. A. Fishman, E. C. Kohn, and L. A. Liotta, "Use of proteomic patterns in serum to identify ovarian cancer," *The Lancet*, vol. 359, pp. 572–577, 2002.

[24] U. M. Fayyad and K. B. Irani, "Multi-interval discretization of continuous-valued attributes for classification learning," in *Proceedings of the 13th International Joint Conference on Artificial Intelligence, IJCAI-93*, Chambery, France, 1993, pp. 1022–1027.

[25] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, 1999.

[26] J. R. Quinlan, *C4.5: Programs for machine learning*. Morgan Kaufmann, *1993*.

[27] P. Langley, W. Iba, and K. Thompson, "An analysis of bayesian classifiers," in *National Conference on Artificial Intelligence*, 1992, pp. 223–228.

[28] J. C. Platt, *Advances in kernel methods: support vector learning*. MIT Press, 1999, ch. Fast training of support vector machines using sequential minimal optimization, pp. 185–208.