Classification of Heterogeneous Gene Expression Data

Benny Y. M. Fung
Department of Computing,
The Hong Kong Polytechnic University,
Hung Hom, Kowloon, Hong Kong
csymfung@comp.polyu.edu.hk

Vincent T. Y. Ng
Department of Computing,
The Hong Kong Polytechnic University,
Hung Hom, Kowloon, Hong Kong
cstyng@comp.polyu.edu.hk

ABSTRACT

Recent advanced technologies in DNA microarray analysis are intensively applied in disease classification, especially for cancer classification. Most recent proposed gene expression classifiers can successfully classify testing samples obtained from the same microarray experiment as training samples with the assumption that the symmetric errors are constant among training and testing samples. However, the classification performance is degraded with heterogeneous testing samples obtained from different microarray experiments. In this paper, we propose the "impact factors" (IFs) to measure the variations between individual classes in training samples and heterogeneous testing samples, and integrate the IFs to classifiers for classification of heterogeneous samples. Two publicly available lung adenocarcinomas gene expression data sets are used in our experiments to demonstrate the effectiveness of the IFs. It shows that, with the integration of the IFs to the Golub and Slonim (GS) and k-nearest neighbors (kNN) classifiers, the classifiers can be further improved on the classification accuracy of heterogeneous samples. Even more, the classification accuracy of the integrated GS classifier is around 90%.

Keywords

Gene expression data, classification, feature selection, significance analysis of microarrays

1. INTRODUCTION

Recent advanced technologies in DNA microarray analysis allow us to examine gene expression levels for a huge amount of genes in a single experiment. Different data mining techniques are used to analyze and discover knowledge from gene expression data. Since the number of examined genes in an experiment is in term of thousands, data mining techniques have been intensively applied in the analysis of gene expression data [6].

Microarray data contain two kinds of errors, namely *symmetric* and *random* errors [12], and normalization is a common technique to minimize the symmetric errors. Symmetric errors are defined as controllable errors which induce almost-equally variations at microarray experiments [5], and random errors are defined as uncontrollable errors which induce different degrees of variations at microarray experiments by chance. Normalization is a common pre-processing step to remove or minimize the influence of the symmetric errors and achieve data centralization. Some common approaches are global normalization, log-transformation, regression normalization, intensity-dependent normalization, and etc. Data centralization is a symmetry property of any two samples whose genes have equal or similar ratio of their expression levels. The objective is to identify a symmetry line among the gene expression levels of two samples, and transforms asymmetric gene expression levels into symmetric one with a transformation function [18]. Unfortunately, there is no efficient way to eliminate the random errors [15].

The symmetric errors are constant among samples in the same microarray experiments, but samples in different experiments have different symmetric errors. The symmetric errors within a microarray experiment can be considered as *intra-experimental variations*. Suppose that there are different data sets obtained by different microarray experiments, and then the symmetric errors of these data sets are different [21]. Hence, the differences of experimental variations among different microarray experiments can be defined as *inter-experimental variations*.

When we consider a combined data set of data sets from different microarray experiments, most normalization approaches, which work fine for a single data set, are incompatible for normalizing such combined data set because there are inter-experimental variations. Assume that the annotation of accession numbers and number of genes are the same. Practically, we can merge the data sets by appending, but this merging does not consider the inter-experimental variations, and thus the variations are distributed and shared over other data sets. The final result is like the one after an averaging process. That is, data sets originally with fewer variations are become unstable since data sets with more variations distribute their variations to them.

In this paper, we propose the "impact factors" (IFs) to measure the variations between individual classes in training samples and heterogeneous testing samples obtained from different microarray experiments, and then integrate the IFs to classifiers for classification of heterogeneous samples. The reminder of this paper is organized as follows. We review classification approaches of gene expression data in section 2. Among different classifiers, we discuss the Golub and Slonim (GS) classifier in section 3 since it is one of the recent proposed classifiers, designed for classification of gene expression data with biological relevance [16]. Furthermore, we adopt the *significant analysis of microarrays (SAM)* (i.e. a feature selection method) to select features at IFs computation [19], and the details are described in section 4. In section 5, we present the algorithm of the IFs. Experimental results are presented and discussed in section 6. Finally, we give our concluding remarks and future works.

2. RELATED WORKS

Several publications have done a comprehensive review on the progress of cancer classification [6], [11]. Their works involve of reviewing a number of traditional and microarray-specified feature selection methods and classification approaches. For the classification with machine learning approaches, support-vector

machines (SVM) [1], artificial neural network (ANN) [22], k-nearest neighbors (kNN) [11], and self-organizing map (SOM) [9] have been successfully applied to cancer classification. Their classification performances with different distance metrics, like Pearson correlation, Euclidean distance, cosine coefficient and signal-to-noise distance, are compared [6]. In addition, other approaches are Fisher linear discriminant analysis (FLDA) [9], CAST [3] and boosting [2].

Since gene expression data have a huge amount of genes, feature selection and extraction methods are used to find informative genes. These methods include principal component analysis, linear discriminant analysis, projection pursuit, and etc. A comprehensive review of feature selection and extraction methods can be found in Jain et al. [10].

3. GOLUB AND SLONIM (GS) CLASSIFIER

A binary-class classifier, especially designed for classification of binary gene expression data, is called *Golub and Slonim (GS)* classifier [16]. Among different classifiers, only tree-based (i.e. boosting) and partitioning-based (i.e. GS classifier) classifiers have the interpretation of biological relevance by the means of revealing the relationships between genes in a step-wise approach [11]. The processes of splitting a whole data set into a number of smaller subsets and performing correlation analysis on these smaller subsets provide insight understanding of selection mechanism and correlation analysis among genes. Other non-biological relevance classifiers, like ANN, SVM, and CAST, consider whole gene expression data as a set of distribution and perform classification based on the distribution of gene expression levels without the consideration of individual relationships among genes, and their correlation analysis is encapsulated in a "black box", implying that it is difficult to analyze biological relevance.

Here, we describe the approach of the GS classifier. The first step is to calculate the "signal-to-noise" (SNR) distance for genes. Assume that the expression levels of gene g in n training samples be represented by an expression vector $g=(e_1, e_2, ..., e_n)$, where e_i donates the expression level of g in tuple i, $c=\{Normal, Cancer\}$ be the class vector donating the classes of tuple i, and $(\mu_{Normal}(g), \sigma_{Normal}(g))$ and $(\mu_{Cancer}(g), \sigma_{Cancer}(g))$ be the class mean expression level and standard deviation of g in normal and cancer classes. First of all, g is normalized across samples with the mean expression level $\mu(g)$ and standard deviation $\sigma(g)$ of the gene. The SNR of g is:

$$SNR(g) = \frac{\mu_{Normal}(g) - \mu_{Cancer}(g)}{\sigma_{Normal}(g) - \sigma_{Cancer}(g)} \quad (1)$$

The second step is to construct two class-vectors to represent the overall similarities of testing sample Y to normal and cancer classes, respectively. From the *SNR*-values, a positive *SNR*-value represents that the gene is in normal class, while a negative *SNR*-value represents that the gene is in cancer class [16]. Very often, certain representative genes are sufficient to represent the overall similarities of the classes. Hence, only $k/2$ genes having the highest and lowest *SNR*-values are selected and expressed as sets $G^{k\text{-}highest}$ and $G^{k\text{-}lowest}$ to represent the overall similarities for the corresponding classes, when k genes are required. Assume that $avg(g)$ be the average value of the class mean expression levels of normal and cancer classes for gene g (i.e. $avg(g)=(\mu_{Normal}(g)+\mu_{Cancer}(g))/2$). A similarity measure called *vote factor*, $v(g)$, is calculated for g in Y by $v(g)=SNR(g)*[Y_g-avg(g)]$, where Y_g is the normalized gene expression levels of g in Y with respects to the $\mu(g)$ and $\sigma(g)$ of the same gene in the training samples. The overall similarities corresponding to the classes, which are expressed as $V_{positive}$ and $V_{negative}$ for normal and cancer classes, are the sum of $k/2$ positive and negative v-values (i.e. $V_{positive}=\Sigma(v(g)|(v(g)>0)\wedge(g\in G^{k\text{-}highest}))$ and $V_{negative}=\Sigma(v(g)|(v(g)<0)\wedge(g\in G^{k\text{-}lowest}))$). The testing sample is classified as normal for $|V_{positive}|>|V_{negative}|$. If otherwise, it is classified as cancer.

Since $V_{positive}$ and $V_{negative}$ are in absolute distance, the relative difference, which is called *prediction strength (PS)*, between them is calculated (in equation 2). If the PS is sufficiently large, the assigned class label is confirmed. Otherwise, the new sample is classified as "uncertain".

$$PS = \frac{\max(V_{positive}, V_{negative}) - \min(V_{positive}, V_{negative})}{V_{positive} + V_{negative}} \quad (2)$$

4. SIGNIFICANCE ANALYSIS OF MICROARRAYS (SAM)

A feature selection method called *Significance Analysis of Microarrays (SAM)* has been proposed by Tusher et al. to identify significant genes in microarray experiments based on the variations of the standard deviations in repeated measurements [19]. It aims at measuring fluctuations of the expression levels for a gene across a number of microarray experiments.

It first calculates an *observed relative difference*, $d_r(g)$, for gene g in training samples X based on the ratio of change in gene expression to standard deviation for that gene. When using the same terminology as the GS classifier, $d_r(g)$ is:

$$d_r(g) = \frac{\mu_{Normal}(g) - \mu_{Cancer}(g)}{s(g) + s_0} \quad (3)$$

where $s(g)$ is a measurement of variations for gene g (in equation 4), and s_0 is a factor to adjust the $s(g)$ because of small sample size [7]. β is the average number of measurements in both classes.

$$s(g) = \sqrt{\beta \times \{\sum [X_g - \mu_{Normal}(g)]^2 + [X_g - \mu_{Cancer}(g)]^2\}}$$

$$\text{, where } \beta = \frac{1/|Normal| + 1/|Cancer|}{|Normal| + |Cancer| - 2} \quad (4)$$

SAM uses a technique of permutations of the repeated measurements to identify significant genes. The idea is to assume that a significant gene in a class should have differential expression levels over another class. However, after a certain number of permutations, such differentiations between two classes are eliminated because the permutations re-arrange the gene expression levels of both classes. After p permutations, it calculates a *permuted relative difference*, $d_p(g)$, for gene g. As a result, there are p $d_p(g)$'s for the gene, and the *mean relative difference*, $d_E(g)$, for gene g over p permutations is calculated [19].

$$d_E(g) = \frac{\sum d_p(g)}{p} \quad (5)$$

$d_E(g)$ is used as a reference point to identify those genes with differential expression levels, which are called *significant genes* in this context. For a significant gene, its $d_E(g)$ should be comparably different from its $d_r(g)$. A threshold, which is the distance in both directions away from a straight line with slope

equal to 1 across the origin, is used to exclude the insignificant genes, which are enclosed by the range.

When the threshold value is set, the smallest positive $d_r(g)$ and least negative $d_r(g)$ are used as cutoff lines to predict the number of falsely-significant genes for the corresponding threshold value. The number of predicted falsely-significant genes for $d_r(g)$ is computed by counting the number of genes at its $d_p(g)$'s that exceed the cutoff lines. Finally, the number of predicted falsely-significant genes for the threshold value is the average number of predicted falsely-significant genes for all $d_r(g)$'s.

5. IMPACT FACTORS

We propose the *impact factors (IFs)* to measure the variations between individual classes in training samples and a heterogeneous testing sample obtained from different microarray experiment. The rationale is to measure the inter-experimental variations (i.e. the differences between two microarray experiments) based on the significant genes in the training samples, and set up individual reference points corresponding to classes in the training samples from the extracted significant genes. Every reference point is then used to calculate its own relative scaling factor for the corresponding class, and this factor is used to rescale the gene expression levels in the testing sample with respects to the reference point of that class individually. With the rescaled expression levels, relative distances to the training samples are calculated. In order to enhance the discriminative powers of the IFs, only those genes with a higher relative difference are selected and integrated to classifiers for classification of heterogeneous sample. Figure 1 shows the algorithm of IFs computation.

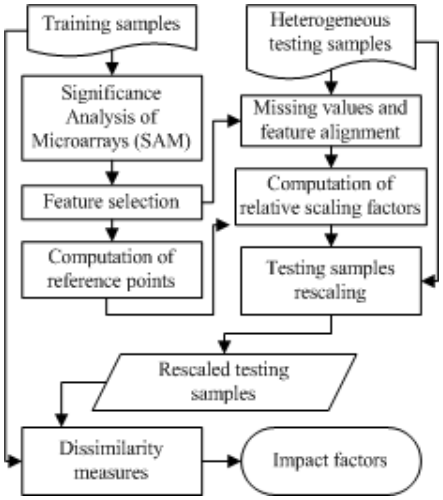

Figure 1. The algorithm of IFs computation.

In SAM, the number of predicted falsely-significant genes depends on the choice of the threshold values. In our studies, there is a convergence point for the number of predicted falsely-significant genes with an increasing threshold value. At this point, the number of predicted falsely-significant genes becomes minimal, and thus the extracted significant genes at this point have the lowest probability to be falsely-significant. Hence, the optimal threshold value, producing the optimal number of significant genes, corresponding to the smallest number of predicted falsely-significant genes at the convergence point is chosen. For example, Figure 2 shows a graph of the threshold values against the number of predicted falsely-significant genes, and the optimal threshold value is chosen at the convergence point.

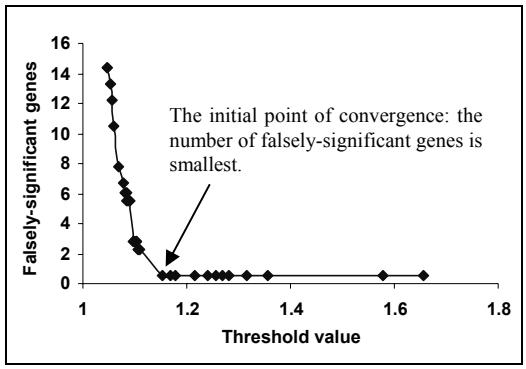

Figure 2. Selection of the optimal threshold value.

With the feature selection method, significant genes are extracted from original genes in training samples. Assume that sets G and G' be the sets of original and extracted significant genes, and hence we have $G' \subset G$. For each $g \in G'$, we calculate class trim-mean expression level. In a k% trim-mean value, data members are sorted, and $k/2$% data members at each end of the sorted list are discarded. The k% trim-mean value is then calculated from those un-discarded members only. The idea of the trim-mean value is to eliminate outliners. We choose 30% of data members to be trimmed, the 30% class trim-mean expression levels are expressed as $\mu'_{Normal}(g)$ and $\mu'_{Cancer}(g)$ for each extracted significant gene g, where $g \in G'$, in normal and cancer classes.

For heterogeneous samples, handling of missing values and feature alignment are required. Assume that set G^h be the set of genes in a heterogeneous testing sample Y^h. In fact, some extracted significant genes may not be existed in G^h (i.e. $G^h \cap G' \not\subset G'$). The objective is to construct a gene set $G^{h\prime}$ for Y^h, and thus $G^{h\prime}$ has the same set of genes as the extracted significant genes G'. (i.e. $G^{h\prime} \subseteq G' \cap G' \subseteq G^{h\prime}$). The feature alignment is performed by looking for some commonly existed genes between G^h and G' (i.e. $G^{h\prime} \subseteq G^h \cap G'$), while the handling of missing value is done by the nearest neighbor (NN) approach. The training sample with the smallest Euclidean distance to the testing sample is the nearest neighbor, and the gene expression levels in the nearest neighbor are copied to the missing expression levels of the corresponding genes in the testing sample (i.e. $G^{h\prime} \subseteq G'/G^h$).

We estimate relative scaling factors for individual classes in the training samples based on the extracted significant gene set G' and the constructed gene set for the heterogeneous sample $G^{h\prime}$, acting as reference points to rescale the gene expression levels in the heterogeneous sample. We first calculate baselines of, respectively, normal and cancer classes in the training samples. The baselines for the classes are the sum of all the corresponding class trim-mean expression levels of the significant genes g, expressing as $\sum \mu'_{Normal}(g)$ and $\sum \mu'_{Cancer}(g)$, where $g \in G'$, for normal and cancer classes, respectively. Similarly, the baseline for a heterogeneous sample Y^h is the sum of gene expression levels of the constructed genes g, expressing as $\sum Y^h{}_g$, where $g \in G^{h\prime}$. Since the relative scaling factors are used to minimize the inter-experimental variations, one possible way to perform the minimization is to amplify or reduce (i.e. rescale) the gene

expression levels of the heterogeneous sample with respects to the reference points of the corresponding individual classes. Thus, the relative scaling factors are ratio of the baselines corresponding to the classes in the training samples to the baseline of the heterogeneous testing sample. They are:

$$R_{Normal} = \sum_{g \in G'} \mu'_{Normal}(g) \Big/ \sum_{g \in G^h{}'} Y^h{}_g \quad (6)$$

$$R_{Cancer} = \sum_{g \in G'} \mu'_{Cancer}(g) \Big/ \sum_{g \in G^h{}'} Y^h{}_g \quad (7)$$

The relative scaling factors are then used to rescale the gene expression levels in the heterogeneous sample. The magnitudes of R_{Normal} and R_{Cancer} define different rescaling operations. When $R_c>0$, where $c \in \{Normal, Cancer\}$, the total expression levels of extracted significant genes of class c in training samples are higher than the total gene expression levels, corresponding to extracted significant genes, in testing sample. Hence, a process of signal enhancement is required. Similarly, signal reduction is required for the case $R_c<0$, and nothing happens for the case $R_c=0$.

The dissimilarity measures, $d_{Normal}(g)$ and $d_{Cancer}(g)$, for gene g, where $g \in G^h$, in the heterogeneous sample to both classes in the training samples are calculated by computing the ratio between them. For each $g \in G^h$, they are:

$$d_{Normal}(g) = \left| \frac{\{Y^h{}_g \times R_{Normal} - \delta_{Normal}\}}{\delta_{Normal}} \right|$$

where $\delta_{Normal} = \sum_{g \in G' \cap G^h} \mu'_{Normal}(g)$ (8)

$$d_{Cancer}(g) = \left| \frac{\{Y^h{}_g \times R_{Cancer} - \delta_{Cancer}\}}{\delta_{Cancer}} \right|$$

where $\delta_{Cancer} = \sum_{g \in G' \cap G^h} \mu'_{Cancer}(g)$ (9)

The total dissimilarities of the heterogeneous sample to both classes are calculated by the sum of all $d_{Normal}(g)$'s and $d_{Cancer}(g)$'s, respectively. By comparing these two values, we know the closeness of the heterogeneous sample to each class. If $\Sigma d_{Normal}(g) < \Sigma d_{Cancer}(g)$, the testing sample is closer to normal class than cancer class. Similar conclusion holds for $\Sigma d_{Normal}(g) > \Sigma d_{Cancer}(g)$. However, it is possible that the relative difference, $r(g)$, for gene g between the two measures is too small, and the discriminative powers are small.

$$r(g) = \frac{\max[d_{Normal}(g), d_{Cancer}(g)] - \min[d_{Normal}(g), d_{Cancer}(g)]}{\min[d_{Normal}(g), d_{Cancer}(g)]} \quad (10)$$

In order to increase the discriminative powers of the IFs, only those genes that have a large $r(g)$ in equation 10 are considered. Hence, we define a threshold value t, and only those genes with $r(g)$ higher than t are taken into the IFs computation. Finally, the IFs for both classes are:

$$IF_{Normal}(t) = \frac{\sum d_{Normal}\{g \mid (r(g) > t)\}}{n} \quad (11)$$

$$IF_{Cancer}(t) = \frac{\sum d_{Cancer}\{g \mid (r(g) > t)\}}{n} \quad (12)$$

where n is the number of the significant genes whose $r(g)$'s are higher than a required threshold value t.

Since $d_{Normal}(g)$ and $d_{Cancer}(g)$ are both in absolute values, IF_{Normal} and IF_{Cancer} must be non-negative values. The lower-bounds of IFs are equal to 0. However, the upper-bound of IFs depends on the factor $Y^h{}_g \times R_c$, where $c \in \{Normal, Cancer\}$. Since this factor is unbounded, the upper-bound is unbounded too.

5.1 Example

Consider the example of IFs computation in Figure 3. The expression levels of training samples and a heterogeneous sample are listed. First of all, SAM is performed, and significant gene set G' is extracted. For the heterogeneous sample, significant genes *1071_at* and *1319_at* are aligned corresponding to the extracted significant genes, and their expression levels are copied to $G^h{}'$. Also, significant gene *1439_s_at* is missed, and thus NN approach is performed. During the operation, assume that the first sample in cancer class is the nearest neighbor. Therefore, the gene expression level in the nearest neighbor corresponding to the missing gene expression level in the testing sample is copied to $G^h{}'$. Even more, R_{Normal}, R_{Normal}, δ_{Normal}, and δ_{Cancer} are calculated in equation 6, 7, 8, and 9. With these fours parameters, the rescaling is performed, and the corresponding $d_{Normal}(g)$ and $d_{Cancer}(g)$ are calculated for gene g. Suppose that the threshold value is set to 2 in equation 11 and 12. When calculating the $r(g)$ for gene g, where $g \in G^h$, only gene *32195_at* is included in the IFs computation since other $r(g)$'s are excluded by the required threshold value.

Accession	Training samples (G)						Heterogeneous sample (G^h)
	Normal			Cancer			
100_g_at	362	178	296	273	192	272	-
1071_at	689	532	250	15	97	194	16
1319_at	1233	809	546	482	551	221	63
1439_s_at	1533	1461	586	240	499	159	-
1440_s_at	35	54	30	10	51	2	-
37659_at	362	832	178	272	576	136	-
32118_at	-	-	-	-	-	-	-21
32195_at	-	-	-	-	-	-	200
37033_s_at	3424	6519	1170	2042	309	1823	-

G'		
Accession	$\mu'_{Normal}(g)$	$\mu'_{Cancer}(g)$
1071_at	490	102
1319_at	863	418
1439_s_at	**1193**	**299**

$G^h{}'$	
Accession	Level
1071_at	16
1319_at	63
1439_s_at	**240**

R_{Normal}	R_{Cancer}
7.98	2.57

δ_{Normal}	δ_{Cancer}
1353	520

G^h		$d_{Normal}(g)$	$d_{Cancer}(g)$	$r(g)$
Accession	Level			
1071_at	16	0.91	0.92	0.01
1319_at	63	0.63	0.69	0.10
32118_at	-21	1.12	1.10	0.02
32195_at	**100**	**0.41**	**0.51**	**0.24**

IF_{Normal}	IF_{Cancer}
0.48	**0.51**

Figure 3. Example of IFs computation.

5.2 Gene Selection

Gene selection is a useful preprocessing to select informative genes for classification and hence improves classification performances. In gene expression data, samples have too many genes (i.e. features) associated with them, and many of genes are noisy or irrelevant for the differentiability between normal and cancer classes. Their inclusion in classification not only introduces confusions with informative genes, but also increases computational complexity. The calculation of the IFs only selects a set of informative genes in training samples. The general criterion for the selection is that the genes should have sufficient differentiability between two classes. From previous works, gene selection is mainly achieved by statistical measurements, which can be divided into either correlation or distance measurements. Some common correlation measurements include the Pearson correlation [2] and Spearman correlation [6], while some common distance measurements are the Euclidean distance [11], Cosine coefficient [6], signal-to-noise ratio [9] and ratio of between-groups to within-groups [8].

Most statistical-based gene selection measurements identify informative genes based on differential expression levels of a gene between two classes, assuming that all genes (i.e. features) are direct-relevant to tissues (i.e. either normal or cancer) of samples. In fact, genes, differentially expressed between two classes, do not always imply that the genes are direct-relevant to tissue of either class since some differential expressions may be caused by different compositions of cell types. In fact, normal and cancer cells have different compositions of cell types (i.e. a kind of biological mechanism), and some genes are also responsible for the mechanisms in the compositions. Some informative genes extracted by the statistical measurements may be caused by the effects of different compositions, instead of direct-relevant to the tissues of either class. This issue is called gene or sample contamination [2], [11]. SAM uses a technique of permutations to minimize biological mechanisms in gene expression data, and extracts informative genes without the inclusion of biological mechanisms. Its significance has been demonstrated by the problem: the transcriptional response of lymphoblastoid cells to ionizing radiation [19].

5.3 Integration of the IFs to Classifiers

The impact factors (IFs) are integrated to classifiers so that the integrated classifiers have the capability of classifying heterogeneous samples. The IFs are dissimilarity measures, namely IF_{Normal} and IF_{Cancer}, for normal and cancer classes, and they express the inter-experimental variations between the corresponding classes in training samples and heterogeneous testing sample.

Bias factors, ε_{Normal} and ε_{Cancer}, are introduced in the integration of the IFs to minimize the impacts of unequal proportions of extracted significant genes among classes. When the expression data are acquired from microarray experiments, it is possible that the proportions of tissue-specific genes, related to normal and cancer classes, are uneven, and hence the proportions of extracted significant genes, identified by SAM, are uneven too. Therefore, the IFs have different degrees of bias according to the classes. As a result, we introduce bias factors in the integration to adjust the importance of the IFs for the corresponding classes. In fact, the possible range of the bias factors is subjective, and may be varied from data sets to data sets.

For most classifiers using similarity/ dissimilarity measures for making classification decisions, one way to perform the integration is to multiply IFs directly to the measures since the IFs are dissimilarity measures. There are two cases for the integration. If there is a dissimilarity measure, the IF of a class is multiplied to the measure with the same class as the corresponding IF. In contrast, if there is a similarity measure, the IF of a class is multiplied to the measure with another class as the corresponding IF. Assume that $d_{dissimilarity}$ and $d_{similarity}$ are, respectively, the dissimilarity and similarity measures before the integration, while $d'_{dissimilarity}$ and $d'_{similarity}$ are, respectively, the dissimilarity and similarity measures after the integration. The general approaches of the integration are:

$$d'_{dissimilarity} = \begin{cases} d_{dissimilarity} \times IF_{Normal} \times \varepsilon_{Normal} & \text{for normal class} \\ d_{dissimilarity} \times IF_{Cancer} \times \varepsilon_{Cancer} & \text{for cancer class} \end{cases} \quad (13)$$

$$d'_{similarity} = \begin{cases} d_{similarity} \times IF_{Cancer} \times \varepsilon_{Cancer} & \text{for normal class} \\ d_{similarity} \times IF_{Normal} \times \varepsilon_{Normal} & \text{for cancer class} \end{cases} \quad (14)$$

5.3.1 Golub and Slonim (GS) classifier

For the choice of classifiers, we adopt the Golub and Slonim (GS) classifier for the first integration because (1) the GS classifier is specially designed for binary-class gene expression data, and there are two classes of samples in our data sets. It makes the classifier performs well for the classification. (2) The GS classifier calculates features with the "signal-to-noise" distance, which measures variations of a gene between two classes. Also, the SAM measures the variations of a gene between two classes. Hence, the feature selection strategy of the classifier and IFs is similar. (3) The semi-final deliveries of the GS classifier are two vote factors (i.e. $V_{positive}$ and $V_{negative}$). Also, there are two dissimilarity measures for the IFs (i.e. IF_{Normal} and IF_{Cancer}).

The integration is performed by multiplying the IF of a class to the V-value of another class since V-values are similarity measures. Supposed that $\hat{V}_{positive}$ and $\hat{V}_{negative}$ are the new vote factors to normal and cancer classes, respectively. We have:

$$\hat{V}_{positive} = V_{positive} \times IF_{Cancer} \times \varepsilon_{Cancer} \quad (15)$$

$$\hat{V}_{negative} = V_{negative} \times IF_{Normal} \times \varepsilon_{Normal}. \quad (16)$$

After the inter-experimental variations are included, the remaining steps are the same as the ordinary GS classifier. The testing sample is classified as normal for $|\hat{V}_{positive}| > |\hat{V}_{negative}|$. Similarly, it is classified as cancer for $|\hat{V}_{positive}| < |\hat{V}_{negative}|$. Even more, the class label "uncertain" is also assigned to it, whose new prediction strength is smaller than a required threshold value.

5.3.2 K-nearest neighbors (kNN) classifier

K-nearest neighbors (kNN) classifier is a multi-class classifier. It first selects features from training samples by some feature selection measurements. From the selected features, dissimilarity measures are computed between training and testing samples, and then k training samples with the lowest dissimilarity are selected as the k-nearest neighbors corresponding to the testing sample. The final step is to assign a class label to the testing sample from the k neighbors based on majority voting (i.e. the most frequent class label among the k neighbors). The choice of k is subjective. However, for most cases, k is chosen as the total number of unique class labels plus 1.

The ratio of between-groups values to within-groups values is used to calculate the significance of features for classification. Assume that $\mu(g)$ be the mean expression level of gene g, $\mu_c(g)$ be the class mean expression level of gene g in class c, and x_{cg} be the expression level of gene g in class c. Hence, the between-groups value, *BSS(g)*, and within-groups value, *WSS(g)*, for gene g in training samples X are [8]:

$$BSS(g) = \sum_g \sum_c I(X_g = c)(\mu_c(g) - \mu(g))^2 \quad (18)$$

$$WSS(g) = \sum_g \sum_c I(X_g = c)(X_{cg} - \mu_c(g))^2 \quad (19)$$

With the *BSS(g)* and *WSS(g)*, the score, expressed as *score(g)*, of gene g is the ratio between them (i.e. *score(g)= BSS(g)/WSS(g)*). Among all genes, some genes with higher scores are selected as features for classification. From the selected genes, the Euclidean distance is used to measure the dissimilarities between training samples X and testing samples Y for gene g (in equation 20)

$$d_{Euclidean}(x,y) = \sqrt{\sum_g (X_g - Y_g)^2} \quad (20)$$

The integration of IFs into kNN classifier is different from that into GS classifier. Compared with the GS classifier, there is only one distance measure, expressing as $d_{Euclidean}(X, Y)$, instead of two measures in the GS classifier, expressing as $V_{positive}$ and $V_{negative}$. Before the k training samples are selected, the $d_{Euclidean}(X, Y)$ is multiplied by the corresponding IF whose class label is same as the class label of the training samples X since both $d_{Euclidean}(X, Y)$ and IFs are dissimilarity measurements. Assume that $\hat{d}_{Euclidean}(X,Y)$ is the new distance between X and Y. We have:

$$\hat{d}_{Euclidean}(X,Y) = \begin{cases} d_{Euclidean}(X,Y) \times IF_{Normal} \times \varepsilon_{Normal} \\ \text{where } X \text{ is in normal class} \\ d_{Euclidean}(X,Y) \times IF_{Cancer} \times \varepsilon_{Cancer} \\ \text{where } X \text{ is in cancer class} \end{cases} \quad (21)$$

From the new distance $\hat{d}_{Euclidean}(x,y)$, k training samples with the lowest dissimilarity are selected, and then the class label of the testing samples are assigned based on majority voting.

6. EVALUATIONS

We apply the IFs to the problem of classifying lung adenocarcinomas. Two publicly published data sets of lung adenocarcinomas are used. The first one is published by Bhattacharjee et al. [4], and another one is Ramaswamy et al. [14]. Since there are different accession number annotations, namely Hu680/ Hu35KsubA and U95A, we mapped the Hu680/ Hu35KsubA annotation into the U95A annotation according to the mapping table done by Ramaswamy et al. [13]. In fact, the mapping is not simply one-to-one mapping. There may be duplicated accession numbers in the mapped data set. Thus, a pre-processing is performed to merge the expression levels by averaging all expression levels of the same accession number.

6.1 Discriminative Powers of the IFs

The IFs with different values of the relative difference r for gene g are compared, and the results are shown in Table 1 and Table 2. Column 1 and 2 show different *r's* and the IFs of corresponding classes. The meta-column 3 and 4 are the mean and standard deviation (SD) of the IFs for normal and cancer classes. Since the IFs are measurement of dissimilarity, the corresponding class with the smallest mean IF-value is supposed to be the predictive class of the testing samples, showed in meta-column 5.

The use of the relative difference r is necessary since it excludes those genes with a smaller r-value between the IFs of the classes, enhancing the differentiability of the IFs. In Table 1, the predictive classes, which are determined by the smallest mean IF-value among two classes, are always same as the actual classes.

The differentiability of the IFs is acceptable in Table 1 since the predictive classes of the smallest mean IF-value are same as the actual classes in all cases, while r-value is set to be 3 in Table 2 in order to enhancing the differentiability. In Table 2, the predictive classes of the smallest mean IF-value are different from the actual classes for the first two r-values. When r-values≥3, the results have the same trends as in Table 1 (i.e. the predictive classes of the smallest mean IF-value are same as the actual class). In our investigation, a number of d_{Normal}'s are closed to 2. On the other hand, a number of d_{Cancer}'s are closed to 0. Therefore, the values of IF_{Cancer}'s are much lower than those of IF_{Normal}'s for r=0 and 1. The results become reasonably for r=3 because all these d_{Normal}'s and d_{Cancer}'s, which are included when r=0 and 1, are excluded. From the results, r-values should be equal to or higher than 3 so that those genes with small r-values are excluded, and hence the differentiability of the IFs can be enhanced.

The IFs for the data set of Bhattacharjee have higher discriminative powers than that of Ramaswamy et al. Although the class of the smallest mean IF-value is the predictive class, it is unavoidable to have overlapping cases. However, if the SD's of the IFs are small, the chance of overlapping cases will be small. In Table 1, the SD's of the IFs are small, and hence the discriminative powers of the IFs for this testing data set (i.e. Bhattacharjee) are high. However, in Table 2, the SD's of the IFs are quite high for most IF_{Normal}'s, which means that overlapping cases are more frequent in this testing data set (i.e. Ramaswamy). The facts can be shown in the classification accuracy in next experiments. In those experiments, the classification accuracy for classifying the data set of Bhattacharjee et al. is higher than that of Ramaswamy et al since the discriminative powers of IFs for the data set of Bhattacharjee et al. are higher.

Table 1: The values of IFs for data set of Bhattacharjee.

r	*IFs*	*Normal*		*Cancer*		*Predictive class*	
		Mean	*SD*	*Mean*	*SD*	*Normal*	*Cancer*
0	IF_{Normal}	**1.7**	**0.4**	3.2	2.2	**IF_{Normal}**	**IF_{Cancer}**
	IF_{Cancer}	2.2	1.2	**2.0**	**1.5**		
1	IF_{Normal}	**1.3**	**0.2**	3.3	2.4	**IF_{Normal}**	**IF_{Cancer}**
	IF_{Cancer}	1.7	1.2	**1.7**	**1.3**		
2	IF_{Normal}	**1.6**	**0.6**	3.9	2.3	**IF_{Normal}**	**IF_{Cancer}**
	IF_{Cancer}	1.6	0.6	**1.3**	**0.8**		
3	IF_{Normal}	**1.6**	0.9	4.4	2.6	**IF_{Normal}**	**IF_{Cancer}**
	IF_{Cancer}	1.8	**0.4**	**1.1**	**0.6**		
4	IF_{Normal}	**1.6**	0.7	2.1	1.1	**IF_{Normal}**	**IF_{Cancer}**
	IF_{Cancer}	1.7	**0.4**	**0.6**	**0.2**		

Table 2: The values of IFs for data set of Ramaswamy.

r	*IFs*	*Normal*		*Cancer*		*Predictive class*	
		Mean	*SD*	*Mean*	*SD*	*Normal*	*Cancer*
0	IF_{Normal}	23.7	6.9	20.2	7.7	IF_{Cancer}	**IF_{Cancer}**
	IF_{Cancer}	**5.9**	**0.9**	**5.3**	**1.0**		
1	IF_{Normal}	30.4	9.1	25.1	10.1	IF_{Cancer}	**IF_{Cancer}**
	IF_{Cancer}	**6.6**	**1.2**	**5.7**	**1.1**		
2	IF_{Normal}	**5.9**	7.7	20.4	16.9	**IF_{Normal}**	**IF_{Cancer}**

	IF_{Cancer}	9.1	**2.2**	**4.9**	**2.0**		
3	IF_{Normal}	**5.7**	9.6	19.8	20.9	**IF_{Normal}**	**IF_{Cancer}**
	IF_{Cancer}	8.6	**3.1**	**3.4**	**1.8**		
4	IF_{Normal}	**6.1**	13.7	22.5	30.5	**IF_{Normal}**	**IF_{Cancer}**
	IF_{Cancer}	9.3	6.9	**3.0**	**1.4**		

6.2 Results of the Ordinary and Integrated Classifiers

Classification results on heterogeneous testing samples, which are performed by the ordinary and integrated GS and kNN classifiers, are compared. We first evaluate the classification results of the ordinary GS classifiers with heterogeneous testing samples and different values of prediction strength (PS). Then, the IFs are integrated to the classifiers, and the same evaluation is performed again. In the experiments, r is set to 3, and ε_{Normal} and ε_{Cancer} are set as 1 and 1.5, respectively, for achieving the best performance.

Compared Table 3 with Table 5, the integration of the IFs can increase the relative difference between $V_{positive}$ and $V_{negative}$, and hence fewer samples are classified as "uncertain". The number of "uncertain" samples is reduced because the IFs can successfully enhance the relative difference, and thus their corresponding PS's exceed the required threshold values.

When the training and testing data sets are reversed (in Table 4 and Table 6), the number of "uncertain" samples is increased with the integration because the integrated classifiers can successfully classify the samples, but the corresponding PS's are still too small to pass the required threshold values. In fact, the number of misclassifications remains 17 for all cases with the ordinary GS classifiers because all normal samples are misclassified (in Table 4). Therefore, the number of "uncertain" samples is low. With the integration, the number of misclassifications is reduced by more than half for most cases, but the number of "uncertain" samples is increased. Actually, the number of correct classifications is still increased. Unfortunately, their PS's are insufficient to let the correctly-classified samples pass the required threshold values. It can be proved when *PS=0*. In addition, classification of "uncertain" is better than a misclassification. From the results, the integration of the IFs can still assign a correct class label to samples even though the confidence (i.e. prediction strength) is sometimes insufficient.

Table 3: Comparison of results for the data set of Bhattacharjee, obtained by the ordinary GS classifiers.

	No. of "uncertain" samples				*No. of misclassifications*			
PS / No. of features	*.0*	*.1*	*.2*	*.3*	*.0*	*.1*	*.2*	*.3*
50	0	0	1	2	2	2	2	2
100	0	0	0	4	2	2	2	2
200	0	0	0	3	2	2	2	2
300	0	2	3	4	3	2	2	2

Table 4: Comparison of results for the data set of Ramaswamy, obtained by the ordinary GS classifiers.

	No. of "uncertain" Samples				*No. of misclassifications*			
PS / No. of features	*.0*	*.1*	*.2*	*.3*	*.0*	*.1*	*.2*	*.3*
50	0	0	0	1	17	17	17	17
100	0	1	1	1	17	17	17	17
200	0	0	0	0	17	17	17	17
300	0	0	0	0	17	17	17	17

Table 5: Comparison of results for the data set of Bhattacharjee, obtained by the integrated GS classifiers.

	No. of "uncertain" samples				*No. of misclassifications*			
PS / No. of features	*.0*	*.1*	*.2*	*.3*	*.0*	*.1*	*.2*	*.3*
50	0	0	0	0	2	2	2	2
100	0	0	0	0	2	2	2	2
200	0	0	0	1	2	2	2	2
300	0	1	1	3	3	2	2	2

Table 6: Comparison of results for the data set of Ramaswamy, obtained by the integrated GS classifiers.

	No. of "uncertain" samples				*No. of misclassifications*			
PS / No. of features	*.0*	*.1*	*.2*	*.3*	*.0*	*.1*	*.2*	*.3*
50	0	0	7	17	6	6	5	4
100	0	2	6	16	6	6	5	3
200	0	2	5	13	8	6	5	4
300	0	3	8	10	9	8	6	6

For the classification results with the kNN classifiers, ε_{Normal} and ε_{Cancer} are set as 1 and 1.8, respectively, r is set to 3, and the same procedures, evaluating the GS classifiers, are performed.

In Table 7, the performance of the integration is optimal only at certain number of features like 200 features, and becomes degraded for other feature sizes, especially for larger feature sizes like 300 features. In fact, a majority of normal samples are classified as cancer for 300 features, which is similar to the one for the GS classifiers shown in Table 4. In contrast, certain amounts of cancer samples are misclassified as normal for smaller feature sizes like 50 and 100 features. The optimal results are at 200 features since a reasonably number of samples, including normal and cancer samples, can be successfully classified into the correct class label. From the results, the integrated classifiers achieve the best performance with 200 features. Although the number of correct-classified normal samples with the integrated kNN classifiers is smaller than that of the ordinary kNN classifiers, the integrated kNN classifiers have a significant improvement for classifying cancer samples, which is always better than the corresponding ordinary GS classifiers with the same number of features.

In Table 8, from 50 to 200 features, the integrated kNN classifiers perform almost the same as the ordinary kNN classifiers, but it can gradually enhance the performance with a higher number of features like 300 features.

Table 7: Comparison of results for the data set of Bhattacharjee, obtained by the GS classifiers.

No. of features	*No. of correct classifications*				*No. of misclassifications*			
	Normal		*Cancer*		*Normal*		*Cancer*	
	No IF	*IF*	*No IF*	*IF*	*No IF*	*IF*	*No IF*	*IF*
50	17	4	0	117	126	9	0	13
100	17	7	0	121	126	5	0	10
200	9	7	105	118	21	8	8	10

300	1	0	122	126	4	0	16	17

Table 8: Comparison of results for the data set of Ramaswamy, obtained by the kNN classifiers.

No. of features	*No. of correct classifications*				*No. of misclassifications*			
	Normal		*Cancer*		*Normal*		*Cancer*	
	No IF	*IF*	*No IF*	*IF*	*No IF*	*IF*	*No IF*	*IF*
50	5	5	7	7	4	4	2	2
100	6	6	7	7	4	4	1	1
200	5	6	7	8	4	3	2	1
300	5	6	7	7	4	4	2	1

6.3 Classification Performance

A contingency table of binary-class classification is used to evaluate the classification performance [17]. Table 9 shows a generic contingency table for binary-class classification in our context. From the table, the following measurements are defined.

1. *Accuracy (acc)* – it measures the proportion of correctly classified instances. (i.e. *acc=(TP+TN)/(TP+TN+FP+FN)*)
2. *Sensitivity* (S_n) – it measures the fraction of actual positive examples that are correctly classified. (i.e. S_n *=TP/TP+FN)*)
3. *Specificity* (S_p) – it measures the fraction of actual negative examples that are correctly classified. (i.e. S_p*=TN/(TN+FP)*)

Table 9: Contingency table of binary-class classification

		Predictive	
		Normal	*Cancer*
Actual	*Normal*	True Positive (TP)	False Negative (FN)
	Cancer	False Positive (FP)	True Negative (NT)

In Figure 4 and Figure 5, they show the classification accuracy, and there is a significant improvement with the integration. The accuracy of the integrated GS classifier is higher than or sometimes equals to that of the ordinary GS classifier. In Figure 4, there is no significant improvement for the integrated GS classifier with 50, 100 and 200 features. However, with 300 and 500 features, the improvements are around 10%. In addition, Figure 5 shows even better results. The accuracy of the integrated GS classifier is always higher than that of the ordinary GS classifier. The improvements are also around 10% in average. The trends of performance enhancement are kept for the integrated kNN classifier too although the rates of enhancement are not as high as for the integrated GS classifier. In Figure 4, the highest accuracy of the integrated kNN classifier is around 75%, while the integrated GS classifier is always higher than 85%. Although better results are shown in Figure 5, the highest accuracy of the integrated kNN classifier is still lower than the average accuracy of the integrated GS classifier. In fact, the integration to both classifiers most likely has higher performance than the corresponding ordinary classifiers. In terms of the classification accuracy, we can conclude that (1) the IFs improve the classification accuracy for classification of heterogeneous samples most of the times, and (2) even there is no improvement, the IFs do not deteriorate the results of the ordinary classifiers, which are not integrated with the IFs.

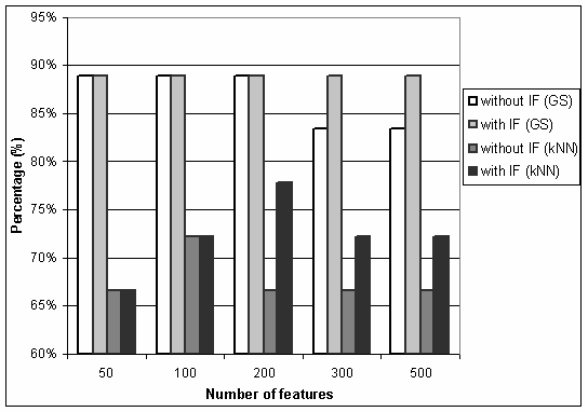

Figure 4. Classification accuracy (acc) for data set of Ramaswamy.

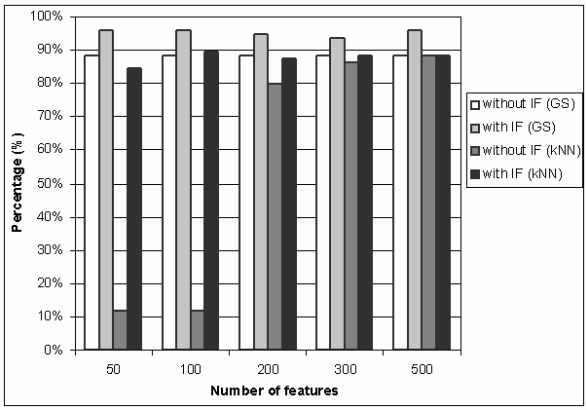

Figure 5. Classification accuracy (acc) for data set of Bhattacharjee.

For the sensitivities (in Figure 6 and Figure 7), the integrated GS classifiers outperform the ordinary GS classifiers, while the integrated kNN classifiers perform worse than the ordinary kNN classifiers for some cases because of certain amount of bias to normal samples. The integrated GS classifiers can recall higher or sometimes equal proportion of normal samples than the ordinary GS classifiers. The same sensitivities between the integrated and ordinary GS classifiers are hold with 50, 100 and 200 features in Figure 6. For the integrated kNN classifier, it sometimes behaves worse than the ordinary kNN classifier. In Figure 7, the integrated kNN classifier has serious performance degradation with 50 and 100 features. In fact, the ordinary kNN classifiers have a higher sensitivity since it has bias to normal class, showing in the corresponding specificity. Actually, almost all cancer samples are illogically misclassified as normal. Without the consideration of these two feature numbers, the integrated kNN classifier can most likely maintain the same sensitivities as the ordinary kNN classifier. Hence, it is still able to maintain the ordinary sensitivities to some extents.

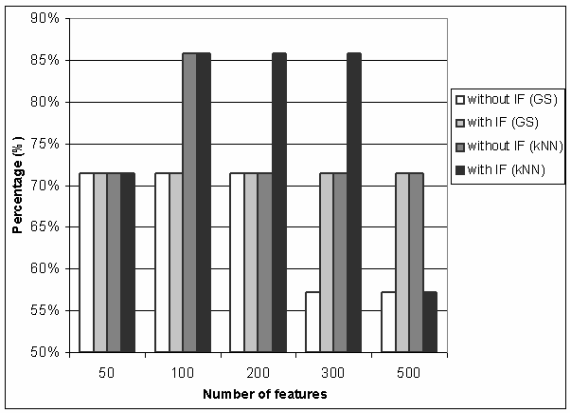

Figure 6. Sensitivity (S_n) for data set of Ramaswamy.

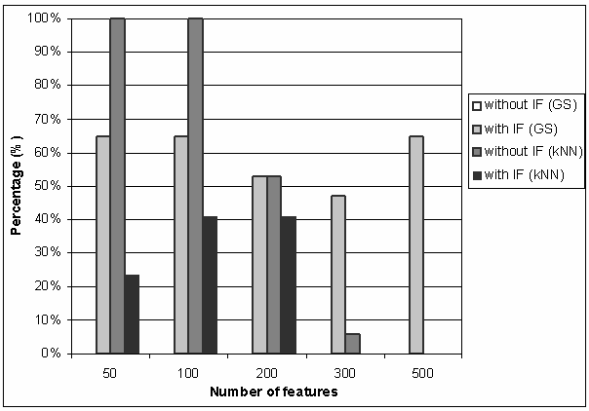

Figure 7. Sensitivity (S_n) for data set of Bhattacharjee.

For the specificities (in Figure 8 and Figure 9), the integrated GS classifiers are most likely to have the same performance as the ordinary GS classifiers, while the integrated kNN classifiers have quite significant improvements over some cases. For all cases, both the integrated and ordinary GS classifiers can achieve 100%. For the kNN classifiers, there is the most significant improvement with 50 and 100 features as shown in Figure 9. For other cases, the integrated kNN classifiers also have better performance than the corresponding ordinary classifiers, even though the improvements are not as high as the one with 50 and 100 features.

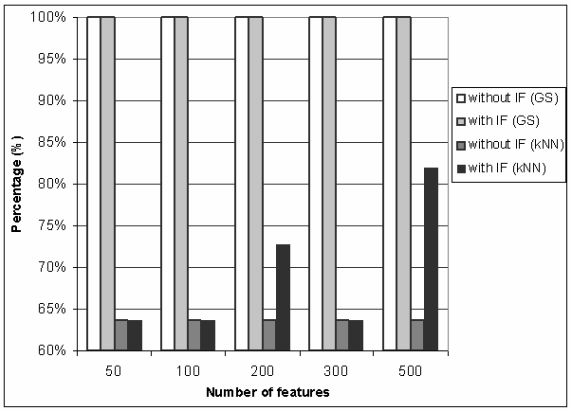

Figure 8. Specificity (S_p) for data set of Ramaswamy.

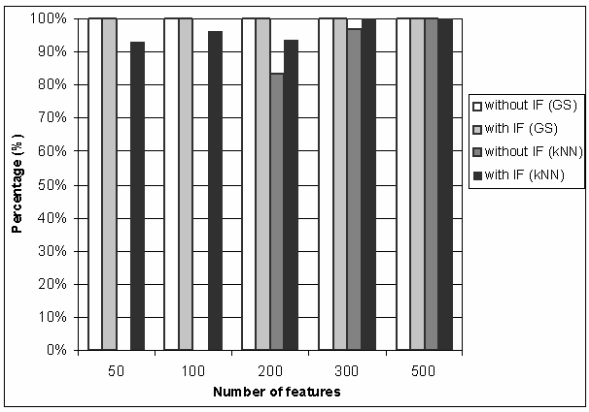

Figure 9. Specificity (S_p) for data set of Bhattacharjee et al.

7. CONCLUDING REMARKS

Gene expression data are biased since there are symmetric errors in microarray experiments. Most normalization methods can minimize the errors among samples obtained from the same microarray experiments. However, normalization of samples from different microarray experiments is more challenging since there are inter-experimental variations. With the feature selection by SAM, we have proposed the "impact factors" (IFs) to measure the variations between individual classes in training samples and heterogeneous testing samples. The IFs are integrated to the Golub and Slonim (GS) and k-nearest neighbors (kNN) classifiers for classification of heterogeneous samples. Our evaluations show that the discriminative powers of the IFs between normal and cancer class are high. In addition, the integration of the IFs into the classifiers can further improve on the classification performance of heterogeneous samples. After the IFs integration, the classification performance for heterogeneous samples is either improved and no deterioration in many cases for our experiments. Even more, the classification accuracy of the integrated GS classifier is around 90%.

The future works are to integrate the impact factors into a meta-classification framework. In such framework, other classifiers, like ANN, SVM, etc, are considered, and the final classification decision is made by the integration of all or some results from these classifiers. Furthermore, other feature selection methods will also be considered to decide the robustness of the impact factors.

REFERENCES

[1] Aliferis, C.F., Hardin, D. and Massion, P.P. (2002): Machine learning models for lung cancer classification using array comparative genomic hybridization. Proc. of the American Medical Informatics Association 2002 Symposium, San Antonio, USA, pp. 7-11, AMIA.

[2] Ben-Dor, A., Bruhn, L., Friedman, N., Nachman, I., Schummer, M. and Yakhini, Z. (2000): Tissue classification with gene expression profiles. Journal of Computational Biology, 7, pp. 559-584.

[3] Ben-Dor, A., Shamir, R. and Yakhini, Z. (1999): Clustering gene expression patterns. Journal of Computational Biology, 6, pp. 281-297.

[4] Bhattacharjee, A., Richards, W., Staunton, J., Li, C., Monti, S., Vasa, P., Ladd, C., Beheshti, J., Bueno, R., Gillette, M., Loda, M., Weber, G., Mark, E., Lander, E., Wong, W., Johnson, B., Golub, T., Sugarbaker, D., and Meyerson, M. (2001): Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses. Proc. of the Natl. Acad. of Sci. USA, 98(24), pp. 13790-13795.

[5] Bilban, M., Buehler, L.K., Head, S, Desoye, G. and Quaranta, V. (2000): Normalizing DNA microarray data. Current Issues in Molecular Biology, 4(2), pp. 57-64.

[6] Cho, S.B. and Won, H.H. (2003): Machine learning in DNA microarray analysis for cancer classification. Proc. of the First Asia Pacific Bioinformatics Conference, Adelaide, Australia, 19, pp. 189-198, Australian Computer Society.

[7] Chu, G., Narasimhan, B., Tibshirani, R. and Tusher, V. (2001): Significant analysis of microarrays. Users Guide and Technical Document, Technical Report, Department of Biological Science, University of Tulsa, USA.

[8] Dudoit, S., Fridlyand, J., and Speed, T.P. (2002): Comparison of discrimination methods for the classification of tumors using gene expression data. Journal of the American Statistical Association, 97, pp. 77-87.

[9] Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., GaasenBeek, M., Mesirov, J.P., Coller, H., Loh, M.L., Downing, J.R., Caligiuri, M.A., Blomfield, C.D. and Lander, E.S. (1999): Molecular classification of cancer: Class discovery and class predication by gene-expression monitoring. Science, 286, pp. 531-537.

[10] Jain, A.K., Duin, R.P.W. and Mao, J. (2000): Statistical pattern recognition: A review. IEEE Trans. on Pattern Analysis and Machine Intelligence, 22(1), pp. 4-37.

[11] Lu, Y. and Han, J. (2003): Cancer classification using gene expression data. Information Systems, 28(4), pp. 243-268.

[12] Morrison, N. and Hoyle, D.C. (2003): Normalization: Concepts and methods for normalizing microarray data. In A Practical Approach to MicroArray Data Analysis. Berrar, D.P., Dubitzky, W. and Granzow, M. (eds.). Boston, Kluwer Academic Publishers.

[13] Ramaswamy, S., Ross, K.N., Lander, E.S. and Golub, T.R. (2003): Evidence for a molecular signature of metastasis in primary solid tumors. Nature Genetics, 33, pp. 49-54.

[14] Ramaswamy, S., Tamayo, P., Rifkin, R., Mukherjee, S., Yeang, C.H., Angelo, M., Ladd, C., Reich, M., Latulippe, E., Mesirov, J.P., Poggio, T., Gerald, W., Loda, M., Lander, E.S. and Golub, T.R. (2001): Multi-class cancer diagnosis using tumor gene expression signatures. Proc. of the Natl. Acad. of Sci. USA, 98(26), pp. 15149-15154.

[15] Schuchhardt, J., Beule, D., Malik, A., Wolski, E., Eickhoff, H., Lehrach, H. and Herzel, H. (2000): Normalization Strategies for cDNA Microarrays. Nucleic Acids Research, 28(10), E47.

[16] Slonim, D., Tamayo, P., Mesirov, J., Golub, T. and Lander, E. (2000): Class prediction and discovery using gene expression data. Proc. of the 4th Annual International Conference on Computational Molecular Biology, Tokyo, Japan, pp. 263-272, Universal Academy Press.

[17] Tan, A.C. and Gilbert, D. (2003): An Empirical Comparison of Supervised Machine Learning Techniques in Bioinformatics. Proc. of the First Asia Pacific Bioinformatics Conference, Adelaide, Australia, 19, pp. 219-222, Australian Computer Society.

[18] Tsodikov, A., Szabo, A. and Jones, D. (2002): Adjustments and measures of differential expression for microarray data. Bioinformatics, 18(2), pp. 251-260.

[19] Tusher, V.G., Tibshirani, R., and Chu, G. (2001): Significance analysis of microarrays applied to the ionizing radiation response. Proc. of the Natl. Acad. of Sci. USA, 98(9), pp. 5116-5121.

[20] Virtanen, C., Ishikawa, Y., Honjoh, D., Kimura, M., Shimane, M., Miyoshi, T., Nomura, H. and Jones, M.H (2002): Integrated classification of lung tumors and cell lines by expression profiling. Proc. of the Natl. Acad. of Sci. USA, 99(19), pp. 12357-12362.

[21] Yang, Y.H., Dudoit, S., Luu, P., Lin, D.M., Peng, V., Ngai, J. and Speed, T.P. (2002): Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation. Nucleic Acids Research, 30(4), e15.

[22] Yao, X. and Liu, Y. (1999): Neural networks for breast cancer diagnosis. Proc. of the 1999 Congress on Evolutionary Computation, New York, USA, pp. 1760-1767, IEEE Press.

ABOUT THE AUTHORS

Benny Y. M. Fung is a MPhil student at Department of Computing, the Hong Kong Polytechnic University. He received a Bachelor degree in Computing from the Hong Kong Polytechnic University in 2002. His current research interests include data mining and bioinformatics.

Dr. Vincent T. Y. Ng is the Associate Professor for the Hong Kong Polytechnic University. He received a PhD degree from the Simon Fraser University. Prior to joining the Polytechnic in 1994, he worked for many years in epidemiology and biometry. He has been involved in the research and development of the patient information system, cancer mapping, and many other clinical studies. He has been awarded the Best Teacher and the Best Consultant of the Department in 1999 and 2000, respectively. In 2000, he received "The President's Award for Outstanding Performance/Achievement 1999" in the teaching category. At present, his research interests include databases, data mining, XML, Internet computing and medical informatics.