Learning Microarray Gene Expression Data by Hybrid Discriminant Analysis

Yijuan Lu, Qi Tian, Maribel Sanchez,
Jennifer Neary, Feng Liu, and Yufeng Wang
University of Texas at San Antonio

A proposed hybrid dimension reduction scheme—hybrid discriminant analysis—merges principal component and linear discriminant analysis in a unified framework for studying gene expression data. This flexible technique also reduces computational complexity. We conducted a set of 80 microarray experiments to test this technique as well as a boosted hybrid analysis technique.

Microarray technology offers a high-throughput means to study expression networks and gene regulatory networks in cells. The intrinsic nature of high dimensionality and small sample size in microarray data calls for effective computational methods. In this article, we propose a novel hybrid dimension reduction technique for classification that combines principal component analysis (PCA) and linear discriminant analysis (LDA)—*hybrid PCA and LDA analysis*. This technique effectively solves the singular scatter matrix problem caused by small training samples and increases the effective dimension of the projected subspace. It offers more flexibility and a richer set of alternatives to LDA and PCA in the parametric space.

In addition, we propose a boosted hybrid discriminant analysis (HDA), using the AdaBoost algorithm which provides a unified and stable solution to find close to the optimal PCA-LDA prediction result and also reduces computational complexity. Extensive experiments on the yeast cell cycle regulation data set show the superior performance of the hybrid analysis, as we explain.

Computational analysis of genes

Microarray technology provides a sizable number of high-dimensional gene expression data items having different patterns. Genes of similar function yield similar expression patterns in microarray hybridization experiments, so analyzing these data and discovering their expression patterns is fundamental in studying networks of expression and gene regulation.

Generally, we can approach the computational analysis of gene expression data in two ways: unsupervised and supervised. We consider a learning method unsupervised if no prior label or class information is given. In this case, gene expression patterns are grouped by a clustering algorithm based on a measure of distance (or similarity) between genes or samples. The commonly used clustering methods in the gene expression space are hierarchical clustering,[1] *K*-means clustering,[2] and self-organizing maps (SOMs).[3]

The unsupervised method has certain disadvantages—it can't utilize some prior information about which samples or genes are expected to group together or construct a classifier and use the classifier to predict some unknown genes. So, supervised methods designed for classification and prediction are becoming most commonly used in microarray analysis. Supervised approaches have labeled output. They can be used to construct a robust classifier, which accurately recognizes patterns from given training samples and classifies test samples into known phenotypes based on the trained classifier.

Representative supervised classification algorithms previously used in classifying gene expression data include Fisher linear discriminant analysis,[4] *k*-nearest neighbor,[5] decision tree, multilayer perceptron,[6] and support vector machines (SVMs).[7] Although these methods have achieved some useful classification results, there's still an inevitable problem plaguing efforts to analyze high-throughput microarray data: high dimensionality. The dimension of the genomic data is usually very high (typically from tens to hundreds of experiments) compared to the limited sample size. Machine learning is afflicted by what's known as the curse of dimensionality: the search space grows exponentially with the dimension. Despite the widely held view that high-throughput approaches are swamping us

1070-986X/07/$25.00 © 2007 IEEE

Published by the IEEE Computer Society

with data, in fact much of the time high dimensionality obscures the details in the data.

We can alleviate the problem of high dimensionality by dimension reduction. PCA[8] and LDA[9] are both well-known techniques for feature dimension reduction. PCA, considered one of the simplest and best-known data analysis techniques, has various applications in many fields. LDA also plays a key role in areas of science and engineering, including face recognition image retrieval, and bioinformatics.

LDA constructs the most discriminant features, by attempting to minimize the Bayes error through selection of the feature vectors ***w***, which maximizes $|w^T S_B{}^w|/|w^T S_W{}^w|$ where S_B measures the variance between the class means, and S_W measures the samples' variance in the same class. It's a simple algorithm used for both dimension reduction and classification.

Alternatively, PCA captures the most descriptive features with respect to packing the most energy—that is, possessing the most important (meaningful) information to represent the data. PCA is a useful statistical technique that has various applications in many fields[8] and is considered one of the simplest and best-known data analysis techniques. Its goal is to replace the original (numerical) variables with new numerical variables called *principal components* that have the following properties:

- they can be ranked by decreasing order of importance, and
- these new variables are uncorrelated.

By importance, we mean that the first few most important principal components account for most of the information in the data. In other words, you can then discard the original data set and replace it with a new data set with the same observations but fewer variables, without throwing away too much information.

In supervised learning, when choosing LDA and PCA, there's a tendency to prefer LDA over PCA, because, as intuition would suggest, the former deals directly with discrimination between classes, while the latter pays no particular attention to the underlying class structure. When the data for each class is represented by a single Gaussian distribution and shares a common covariance matrix, LDA will outperform PCA. However, PCA might outperform LDA when the number of samples per class is small, or when the training data samples the underlying distribution nonuniformly.[10]

Additionally, LDA can't classify small sample data effectively because a singular scatter matrix problem occurs when the number of the feature dimensions is large compared to the number of training examples. Unfortunately, the training sample sizes of microarray data are often relatively small.

Since both LDA and PCA have pros and cons, it's important to find a computational method that can effectively exploit their favorable attributes while simultaneously avoiding their unfavorable ones.

Hybrid feature dimension reduction

The hybrid feature dimension reduction scheme (HDA) that we propose merges LDA and PCA in a unified framework. In this technique, PCA compensates LDA for a singular scatter matrix caused by small training samples and increases the effective dimension of the projected subspace. Alternatively, LDA compensates PCA for dealing directly with discrimination between classes. The hybrid PCA and LDA analysis—which we refer to as HDA—offers more flexibility and a richer set of alternatives to LDA and PCA in the parametric space.

Hybrid discriminant analysis

It's common practice to preprocess microarray data by extracting linear and nonlinear features. In many feature extraction techniques, there's a criterion to assess the quality, which ought to be optimized, of a single feature. Often, prior information is available to formulate quality criteria, or probably even more common, the features are extracted for a certain purpose, such as to subsequently train some classifier. What we'd like to obtain is a feature that is as invariant to transformation (that is, rotation or scale) as possible while still covering as much of the information necessary for describing the data's properties of interest.

A classical and well-known technique that solves this type of problem, considering only one linear feature, is the maximization of the *Rayleigh* coefficient:[9]

$$J(W) = \frac{|W^T S_1 W|}{|W^T S_2 W|} \qquad (1)$$

Here, W denotes the weight vector of a linear feature extractor. That is, for an example **x**, the fea-

Table 1. Special cases of hybrid discriminant analysis.

Case	(λ, η)	Hybrid Discriminant Analysis
Case 1 (LDA)	(0, 0)	$W_{\text{opt}} = \underset{W}{\text{argmax}} \frac{\lvert W^T S_B W \rvert}{\lvert W^T S_W W \rvert}$
Case 2	(0, 1)	$W_{\text{opt}} = \underset{W}{\text{argmax}} \frac{\lvert W^T S_B W \rvert}{\lvert W^T \cdot I \cdot W \rvert}$
Case 3	(1, 0)	$W_{\text{opt}} = \underset{W}{\text{argmax}} \frac{\lvert W^T S_\Sigma W \rvert}{\lvert W^T S_W W \rvert}$
Case 4 (PCA)	(1, 1)	$W_{\text{opt}} = \underset{W}{\text{argmax}} \frac{\lvert W^T S_\Sigma W \rvert}{\lvert W^T \cdot I \cdot W \rvert}$
Case 5	(1/2, 1/2)	$W_{\text{opt}} = \underset{W}{\text{argmax}} \frac{\lvert W^T (S_B + S_\Sigma) W \rvert}{\lvert W^T (S_W + I) W \rvert}$

ture is given by the projections ($W^T \cdot \mathbf{x}$) and S_1 and S_2 are symmetric matrices designed so that they measure the desired information and the undesired noise along the direction W. The ratio in Equation 1 is maximized when we cover as much as possible of the desired information while avoiding the undesired.

If we look for discriminating directions for classification, we can choose S_B to measure the separation between class centers (between-class variance)—that is, S_1 in Equation 1, and S_W to measure the within-class variance (S_2 in Equation 1). In this case, we recover the Fisher discriminant, where S_B and S_W are given by

$$S_B = \sum_{j=1}^{C} N_j \cdot (m_j - m)(m_j - m)^T \tag{2}$$

$$S_W = \sum_{j=1}^{C} \sum_{i=1}^{N_j} (x_i^{(j)} - m_j)(x_i^{(j)} - m_j)^T \tag{3}$$

We use $\{x_i^{(j)}, i = 1, \ldots, N_j\}, j = 1, \ldots, C$ to denote the feature vectors of training samples. C is the number of classes where C is 2 for Fisher discriminant analysis (FDA) and C is greater than 2 for multiple discriminant analysis (MDA). N_j is the number of the samples of the jth class, $x_i^{(j)}$ is the ith sample from the jth class, m_j is the mean vector of the jth class, and m is the grand mean of all examples.

If S_1 in Equation 1 is the covariance matrix S_Σ of all the samples

$$S_\Sigma = \frac{1}{C} \sum_{j=1}^{C} \frac{1}{N_j} \sum_{i=1}^{N_j} (x_i^{(j)} - m)(x_i^{(j)} - m)^T \tag{4}$$

and S_2 is an identity matrix, we recover standard PCA.[8]

We design our optimal function as

$$W_{\text{opt}} = \arg\max_{W} \frac{\lvert W^T[(1-\lambda) \cdot S_B + \lambda \cdot S_\Sigma]W \rvert}{\lvert W^T[(1-\eta) \cdot S_W + \eta \cdot I]W \rvert} \tag{5}$$

where λ, η are two regularization parameters,[11] S_Σ is the covariance matrix of all the training samples, and I is an identity matrix. The range of the parametric pair (λ, η) is from (0, 0) to (1, 1).

With different (λ, η) values, Equation 5 provides a rich set of alternatives to PCA and LDA: ($\lambda = 0, \eta = 0$) reduces to the full LDA; ($\lambda = 1, \eta = 1$) recovers the full PCA; ($\lambda = 0, \eta = 1$) gives a subspace that is mainly defined by maximizing the scatters among all classes with minimal effort on clustering each class; ($\lambda = 1, \eta = 0$) gives a subspace that mainly preserves the most energy while minimizing the scatter matrices of within-classes; ($\lambda = 1/2, \eta = 1/2$) gives a subspace that is discriminative while preserving as much energy as possible, a tradeoff between LDA and PCA.

Table 1 summarizes the five special cases of such a hybrid analysis. All five cases fit certain gene feature distributions.

The difference of the proposed hybrid analysis from the existing formulas for linear discriminant analysis is subtle but critical. Two points are worth mentioning: *regularization* and *effective dimension*. These two differences are responsible for the robust performance of hybrid PCA and LDA analysis.

Regularization. It's well known that sample-based plug-in estimates of the scatter matrices based on Equations 2–5 will be severely biased for a small number of training samples. If the number of the feature dimensions is large compared to the number of training examples, the problem becomes ill-posed, that is, $\lvert W^T S_W W \rvert = 0$ in Equation 1. We can achieve a compensation or regularization simply by adding quantities to the diagonal of the scatter matrices.[11] It's denoted as a simple regularization scheme. If we examine the denominator of Equation 5, by adding the part $\eta \cdot I$, the denominator won't become 0 even when the number of the feature dimensions is large compared to the number of training exam-

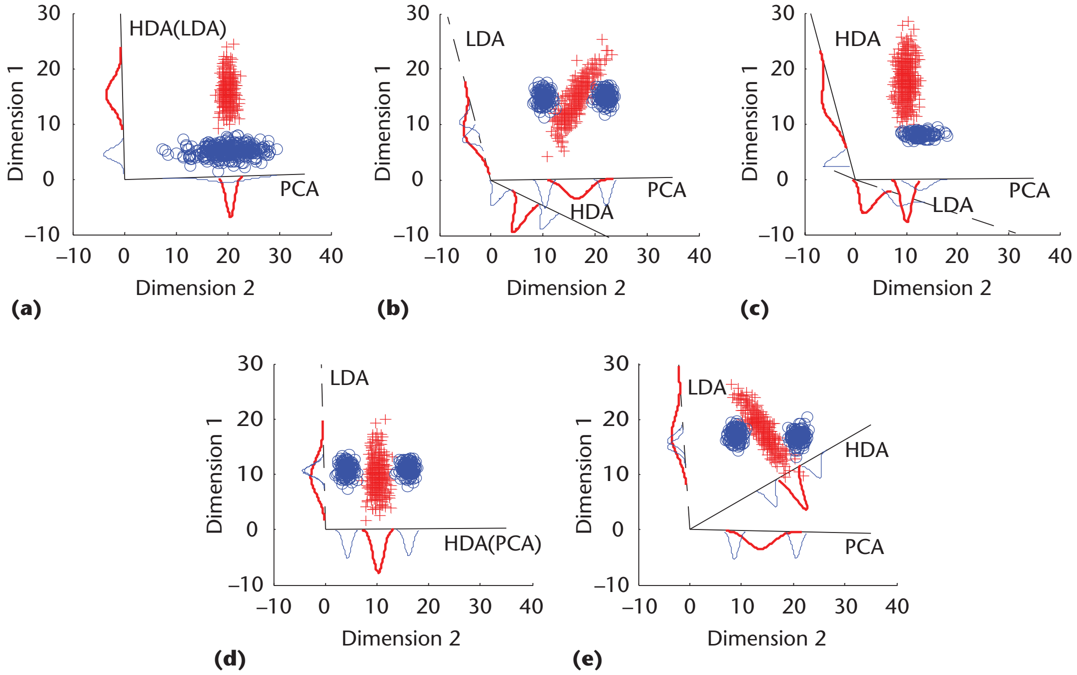


Figure 1. Comparison of principal component analysis (PCA), linear discriminant analysis (LDA), and hybrid discriminant analysis (HDA) for dimension reduction from 2D to 1D on synthetic data. (a) In Case 1, HDA and LDA yield projection with good class separation, but PCA fails. In (b), (c), and (e), HDA finds good projection with class separation while PCA and LDA produce relative bad projection in Cases 2, 3, and 5. (d) In Case 4, HDA and PCA give better class separation, but LDA fails.

ples. It's equivalent to a simple regularization scheme, which has been shown to significantly improve the classification accuracy on average by approximately 15 to 40 percent.[12] It effectively solves the singular scatter matrix problem caused by small training samples.

Effective dimension. In LDA, W maps the original d_1-dimensional data space X to a d_2-dimensional space Δ. The maximum dimension of the projected subspace is $C - 1$, where C is the number of the classes, while in PCA there's no such limitation. Due to the full rank of the $(1 - \eta) \cdot S_W + \eta \cdot I$, HDA has *effective dimension* up to d_1, while for FDA it is only 1 and for MDA it's at most $C - 1$ ($d_1 >> C$ usually). This gives the hybrid approach significantly higher capacity for informative density modeling, for which FDA has virtually none.

To show the advantages of HDA over PCA or LDA, we use synthetic data to simulate different sample distributions, as Figure 1 shows, which correspond to the five special cases in Table 1. We simulated original data in 2D feature space, and marked positive examples with a + symbol and negative examples with the letter o. In each case, we apply PCA, LDA, and HDA to find the best projection direction by each technique's criterion functions. The resulting projection lines are dotted, dash-dotted, and solid lines respectively. In addition, the distributions of the examples along these projections in the figure are bell-shaped curves along a projection line, assuming Gaussian distribution for each class. The thicker curves represent the distribution of projected positive examples; thinner curves denote the distribution of projected negative examples.

From Figure 1, we can see these five cases actually represent several typical data distribution scenarios. Case 1 is the scenario where the major descriptive directions of positive genes and negative genes are upright (Figure 1a). Cases 2, 4, and 5 may correspond to the scenario that one gene function contains multiple subcategories (see Figure 1b, 1d, and 1e). They best fit the distribution where all positive gene expressions are alike, while negative ones may be irrelevant (functionally dissimilar) to each other and from different distributions. Case 3 represents the imbalanced data set. In Case 3, the size of positive genes is much larger than that of negative genes, and the negative genes may be from different smaller classes (see Figure 1c).

From projection results, we see that LDA treats positive and negative samples equally. It tries to cluster the positive samples and decrease the scatter of the negative samples, although some positive (or negative) samples perhaps come from different subclasses. This makes it a bad choice in Cases 2, 4, and 5. Similarly, because PCA captures the most descriptive features with respect to packing the most energy, it fails in Cases 1, 2, and 5. In Case 3, PCA and LDA are not applicable for imbalanced data sets, especially since the number of positive samples is much larger than that of the negative samples. The reason is that LDA or PCA tends to severely bias to cluster the positive genes, rather than cluster both positive and negative genes, since they dominate the data set.

In all five cases, HDA yields good projection with positive samples and negative samples well separated, and it outperforms PCA or LDA alone. Note that in Cases 2, 3, and 5, both PCA and LDA totally fail while HDA still produces a good projection. Clearly, no matter whether a data set is imbalanced or samples are from different subclass clusters, we've demonstrated that HDA can fit into different distributions of samples and find a balance between clustering and separating, which are embedded in the criterion function. Here, we show only five special cases of HDA. Since (λ, η) values can be any number from (0, 0) to (1, 1), we can achieve more accurate data model fitting by fine parameter tuning.

Boosted HDA

Given the data distribution and classification task, the optimal projection of HDA that offers the best classification performance could lie outside the line of PCA and LDA in the parametric space of (λ, η). But it's hard to tell which parameter is best. Searching the whole parametric space will result in extra computational complexity. It's also true that the best pair we found for one particular data set often differs from that of another data set and therefore this can't lead to a generalization.

AdaBoost, developed in the computational machine learning area, is a competitive technique. It has a theoretically justified ability to improve the performance of any weak classification algorithm in terms of bounds on the generalization error.[18]

The basic idea of boosting is to iteratively reweight the training examples based on the outputs of some weak learners. The intention is to increase the weights of the incorrectly classified examples and decrease the weights of the correctly classified examples. This forces the classifier to focus more on the incorrectly classified examples in the next iteration. The final prediction is the combination of the prediction from each classifier weighted by its classification performance—that is, the smaller the training error rate the larger the weight.

AdaBoost, therefore, gives us a general way to combine and enhance a set of PCA-LDA classifiers in the parametric space. With affordable computational cost, AdaBoost provides a unified and stable solution to find close to the optimal PCA-LDA prediction result. The reweight and retraining mechanism is expected to enhance each classifier's performance. Unlike most of the existing approaches that boost individual features to form a composite classifier, our scheme boosts both the individual features and a set of weak classifiers.

Figure 2 shows our algorithm.

Experiments and results

To evaluate our HDA on gene expression data, we conducted a series of experiments in which we applied HDA, classic dimension reduction methods such as PCA, and support vector machines (SVMs) on the same data set, and compared their results.

HDA on yeast cell cycle regulation data set

We used the baker's yeast (*Saccharomyces cerevisiae)* cell cycle expression data[13] as our benchmark test data set, which contained expression vectors from a total of 80 different DNA microarray hybridization experiments on 6,221 yeast open reading frames. We chose this data set because sequencing and functional annotation of the entire *S. cerevisiae* genome has been completed, making it an ideal testbed for estimating the accuracy of our proposed methods. According to the Comprehensive Yeast Genome Database (CYGD), a repertoire of molecular structures and functional networks in the yeast genome, 4,449 out of a total of 6,221 genes have annotated functions.

The 80 microarray experiments covered a wide spectrum of conditions for cell cycle synchronization and regulation, including tests for growth under different conditions—α factor-based synchronization, Cdc15-based synchronization, elutriation synchronization, Cln3 and Clb2 experiments—and the conditions under nitrogen deficiency and glucose depletion. The microarray data also included spotted-array samples in mitotic cell division, spore morphogenesis, and diauxic shift. Researchers have shown that combining multiple microarray studies can improve functional classification.[14] This data set, which has been used in numerous microarray studies, is publicly available at http://rana.lbl.gov/EisenData.htm.

To compare the performance of classification techniques, we focused on five representative functional classes: TCA cycle, respiration, cytoplasmic ribosomes, proteasomes, and histone/chromosome, previously analyzed and demon-

```
Given: Training Sample set X and corresponding label Y
       K HDA classifiers with different $(\lambda, \eta)$

Initialization: weight $w_{k,t=1}(x) = 1/|X|$

AdaBoost:
For $t = 1, \ldots, M$
  For each classifier $k = 1, \ldots, K$ do
    Train the classifier on weighted mean for all the samples,
    positive samples and negative samples and weighted scatter
    matrices in the following way. Note that
      $\sum_{x \in X} w_{k,t}(x) = 1$
    (a) Update weighted mean $\mu_{all}$, $\mu_p$, and $\mu_n$
      $\mu_{all} = \sum w_{k,t}(x) \cdot x / \sum w_{k,t}(x)$
      $\mu_p = \sum_{x \in p} w_{k,t}(x) \cdot x / \sum_{x \in p} w_{k,t}(x)$
      $\mu_n = \sum_{x \in n} w_{k,t}(x) \cdot x / \sum_{x \in n} w_{k,t}(x)$
    (b) Update within-class and between-class scatter matrices and
        covariance matrix
      $S_w = \sum_{x \in p} (x - \mu_p) w_{k,t}(x)(x - \mu_p)^T / \sum_{x \in p} w_{k,t}(x)$
           $+ \sum_{x \in n} (x - \mu_n) w_{k,t}(x)(x - \mu_n)^T / \sum_{x \in n} w_{k,t}(x)$
      $S_b = (\mu_p - \mu_{all}) \cdot (\mu_p - \mu_{all})^T \cdot \sum_{x \in p} w_{k,t}(x)$
           $+ (\mu_n - \mu_{all}) \cdot (\mu_n - \mu_{all})^T \cdot \sum_{x \in n} w_{k,t}(x)$
      $S_\Sigma = \sum_{x \in X} (x - \mu_{all}) w_{k,t}(x)(x - \mu_{all})^T / \sum_{x \in X} w_{k,t}(x)$
    Get the confidence-rated prediction on each sample
    $h_{k,t}(x) \in (-1, 1)$
    Suppose the probability of a sample $x$ belongs to a positive
    and negative class and is denoted as $P(x \in p)$ and $P(x \in n)$,
    respectively.
    If $P(x \in p) \geq P(x \in n)$,
      $h(x) = \frac{P(x \in p)}{P(x \in p) + P(x \in n)}$
    else
      $h(x) = \frac{-P(x \in n)}{P(x \in p) + P(x \in n)}$
    Compute the weight of one classifier $\alpha_{k,t}$:
      $r_{k,t} = \sum_{x \in X} w_{k,t}(x) \cdot h_{k,t}(x) \cdot y$
      $\alpha_{k,t} = \frac{1}{2} \ln(\frac{1 + r_{k,t}}{1 - r_{k,t}})$
    Update the weight of each sample
      $w_{k,t+1}(x) = w_{k,t}(x) \exp(-\alpha_{k,t} \cdot h_{k,t}(x) \cdot y) / Z_t$
    where $Z_t$ is chosen such that $\sum_{x \in X} w_{k,t}(x) = 1$
  End for each classifier
End for $t$
The final prediction $H(x) = \mathrm{sign}(\sum_{k=1..K} \sum_{t=1..M} \alpha_{k,t} \cdot h_{k,t}(x))$
```

Figure 2. Algorithm: AdaBoost with hybrid discriminant analysis.

strated to be learnable.[15,16] Biologically, the classes represent categories of genes expected to exhibit similar expression profiles.[14]

Out of the 4,449 annotated yeast genes, genes with incomplete expression data were filtered to assure accurate evaluation. The resulting data set included 2,324 annotated genes for our comprehensive evaluations. Among them, 385 genes (TCA cycle: 18, respiration: 68, cytoplasmic ribsome: 171, proteasome: 77, and histone/chromosome: 51) belong to the aforementioned five functional classes; the remaining 1,939 genes have different functions.

Experiments. A well-cited microarray classification study[15] has investigated the use of SVMs, two decision tree learners (C4.5 and MOC1), and Parzen windows in gene classification to the same data set. The congruent results are that SVMs, especially SVMs with kernel functions, significantly outperformed the other algorithms in the functional classification. Therefore, in our experiments, we focused on comparing hybrid analysis with SVMs using polynomial and radial basis kernel (RBF) functions. The polynomial kernel functions we used were $K(\boldsymbol{X}, \boldsymbol{Y}) = (\boldsymbol{X} * \boldsymbol{Y} + 1)^d$, with $d =$ 1,2,3,4; the RBF functions were $K(\boldsymbol{X}, \boldsymbol{Y}) = \exp(-\|\boldsymbol{X} - \boldsymbol{Y}\|^2 / 2\alpha^2$. In this work, α was set to be a widely used value, the median of the Euclidean distances from each positive example to the nearest negative example.[15] Besides, for the sake of showing that HDA outperforms the single method, such as single PCA, we also compared the performance of single PCA and single LDA with HDA.

To compare LDA, PCA, and HDA with SVM, we performed a two-class classification with pos-

*Table 2. Comparison of Precision, Recall, and f_measure evaluation factors for various classification methods on yeast cell cycle regulation data set (including their 95-percent-level confidence intervals).**

		Classification Method							
	Evaluation	SVM (%)					PCA	LDA	HDA
Class	Factor	D-p 1	D-p 2	D-p 3	D-p 4	RBF	(%)	(%)	(%)
TCA cycle	Precision	0.0	**60.56**± 16.3	65± 15.6	28.89± 15.4	3.33± 6.2	0.0	35.24± 5.1	38.42± 4.4
	Recall	0.0	13.33± 4.1	16.67± 4.5	5.56± 3.1	0.56± 1.0	0.0	50.56± 5.5	**52.78**± 4.9
	*f*_measure	0.0	21.15± 6.0	25.64± 6.5	9.10± 4.9	0.95± 1.8	0.0	40.38± 4.6	43.43± 3.6
Respiration	Precision	0.0	71.21± 4.6	61.64± 4.2	47.31± 6.5	**90.17**± 6.7	0.0	40.73± 3.3	47.52± 2.6
	Recall	0.0	20.28± 1.8	22.22± 1.5	11.39± 1.9	11.94± 1.9	0.0	33.33± 2.0	**42.64**± 2.6
	*f*_measure	0.0	31.13± 2.2	32.26± 1.8	17.84± 2.7	20.68± 3.0	0.0	36.37± 2.1	**44.74**± 2.4
Cytoplasmic ribsome	Precision	88.27± 2.0	89.06± 2.0	86.12± 1.9	85.97± 2.1	**96.28**± 1.6	26.9± 2.6	68.89± 2.4	71.17± 1.8
	Recall	47.84± 2.0	46.55± 2.0	45.85± 2.0	43.27± 2.0	45.67± 2.0	4.44± 2.1	56.67± 1.8	**59.36**± 2.0
	*f*_measure	61.8± 1.9	60.89± 1.9	59.6± 1.8	57.31± 1.9	61.72± 1.9	7.56± 1.8	62.00± 1.7	**64.60**± 1.7
Proteasome	Precision	0.0	1.667± 3.1	0.0	0.83± 1.6	**72.5**± 15.2	0.0	37.74± 3.9	47.45± 4.2
	Recall	0.0	0.123± 0.2	0.0	0.12± 0.2	5.56± 1.5	0.0	15.06± 1.8	**16.05**± 1.6
	*f*_measure	0.0	0.23± 0.4	0.0	0.22± 0.4	10.17± 2.8	0.0	21.04± 2.3	**23.68**± 2.2
Histone/ Chromosome	Precision	10± 10.3	**90.28**± 8.9	65.29± 10.2	52.93± 11.1	86.67± 11.8	0.0	26.54± 5.2	32.32± 4.3
	Recall	0.59± 0.6	11.57± 1.9	11.18± 1.7	8.824± 1.8	9.02± 1.72	0.0	15.88± 3.1	**16.08**± 2.4
	*f*_measure	1.11± 1.2	20.2± 3.0	18.52± 2.7	14.66± 2.9	16.16± 3.0	0.0	19.44± 3.7	**20.99**± 2.9

* Boldface numbers, indicating the largest number in each row, identify the method that performed the best.

itive genes from one functional class and the negative genes from the remaining classes. Each gene could be classified in one of the four ways: true positive (TP), true negative (TN), false positive (FP), and false negative (FN), according to the CYGD annotation and classifier results. The yeast gene data set is imbalanced—the number of negative genes is much larger than the number of positive genes. For example, in the TCA cycle class, the number of positive instances was only 18, whereas the number of negative instances reached 2,306. In an imbalanced set such as this, accuracy and single precision aren't good evaluation metrics because the FN classifier is more important than the FP.[15] So, we chose to use *f*_measure = 2 × (Recall × Precision) / (Recall + Precision) to measure the overall performance of each classifier, taking both Precision and Recall factors into account.[17] By definition, Precision = (number of TP instances)/(number of TP + FP predictions), and Recall = (number of TP instances)/(number of TP + FN instances). Recall measures the retrieved set's completeness—that is, the percentage of retrieved objects in the correct answer set. Precision, on the other hand, measures the retrieved set's purity—that is, the percentage of relevant objects among those retrieved. Usually, a tradeoff must be made between these two measures because improving one will sacrifice the other. In imbalanced data where negative instances are dominant, Recall is the more important measure because it focuses more on FN predictions.

In our experiments, we applied each method to classify the genes in the test set to the five learnable functional classes and compared their performance. When classifying one class, we set all the genes belonging to that class positive and the remaining ones in the other classes negative. For each class, we also randomly selected 2/3 positive genes and 2/3 negative genes as a training set and the remaining gene data as a testing set to do the classification. We repeated this training and testing procedure 100 times. For HDA, we searched (λ, η) from (0, 0) to (1, 1) with step size 0.1. Therefore, each λ and η could have 11 options: 0, 0.1, 0.2, …, 0.9, and 1. Because each (λ, η) pair corresponds to one feature dimension reduction scheme between PCA and LDA—on a diagonal line from (0,0) to (1,1) or beyond PCA and LDA (on nondiagonal lines)—we could get a total of 121 different classifiers, called parameterized classifiers. Accordingly, we tested SVM, PCA, LDA, and all parameterized classifiers constructed by HDA on the same data sets. Finally, we obtained the average values of Recall, Precision, and *f*_measure for 100 rounds of each method.

Results. Table 2 lists the Precision, Recall, and *f*_measure for five different classifiers on the yeast's five functional classes and their 95-percent-level confidence interval. The first five methods are SVMs using a different polynomial kernel and an RBF kernel. Here, *D-p* 1 to *D-p* 4 represents four kinds of polynomial kernel functions with *d* from 1 to 4. The other three methods are PCA, LDA, and the best parameterized classifier constructed by HDA.

Table 2 exhibits the favorable and stable performance of HDA. From this table, we can clearly see that HDA outperformed all other methods for a total of five classes using the Recall or *f*_measure criteria, which are more important evaluation factors than Precision when working with an imbalanced data set.

The SVM method failed for most classes with a small sample size and yielded a very low *f*_measure. For example, for the Proteasome class, most SVM methods have almost zero Precision, Recall, and *f*_measure, which indicate that SVMs are nearly helpless for this class. The reason is that given a small sample size, SVMs could hardly find sufficient labeled data to train classifiers well. By contrast, hybrid analysis substantially improved the classification performance, especially on Recall and *f*_measure. In the TCA cycle class, the Recall and *f*_measure of SVMs were 16.67 percent and 25.64 percent, while the PCA-LDA method achieved 52.78 percent for Recall and 44.48 percent for *f*_measure.

Not surprisingly, the SVM method showed fairly good and stable performance on all five classes, especially its relative high Precision value, but some exceptions were still observed. For example, among five classes, the *D-p* 1 SVM achieved zero Precision and zero Recall for three classes, which means that all positive instances recognized were wrong. The reason for zero precision and recall is that in the transformed space produced by *D-p* 1, it's hard to find a maximum-margin hyperplane to separate data. Despite the fact that a higher-dimensional dot product (polynomial) kernel seems to have better classification, it's difficult to tell which dimension, such as *d*, can give the best result.

Compared to HDA, any single analysis method performed much worse than the hybrid analysis on the five classes. For example, PCA failed for most classes and gave zero Precision and zero Recall, because PCA can't determine most discriminant features for these classes. The results for all of our yeast experiments show that the hybrid analysis can emphasize different aspects for the alternative schemes, and offers more flexibility than any single method does.

Table 3. Comparison of the boosted HDA classifier and best single classifier of HDA pair on Cytoplasmic ribsome class of yeast cell cycle regulation data set.

Search	*f*_measure (%) of the Best	Boosted HDA		
Space Size	Single Classifier (λ^*, η^*)	*t* = 1	*t* = 2	*t* = 3
16	61.45 (0.33, 0)	61.27	61.79	62.09
25	61.44 (0.25, 0)	61.22	61.48	61.76
36	61.46 (0, 0.6)	61.59	61.90	62.47
100	61.80 (0, 0.5)	61.61	61.74	62.04

Boosting HDA on yeast cell cycle regulation data set

In the experiments we've discussed, HDA has shown promising performance. As we can imagine, for simply searching the parametric space, the larger the searched space, the better the performance of the best single classifier. However, the exhaustive search carries a higher computational cost. Table 3 shows the boosted HDA classifier and the best single classifier of HDA analysis in different search spaces. Due to space constraints, the table shows only the performance on the Cytoplasmic ribsome class of yeast cell cycle regulation data; we obtained similar results on other classes. The range of (λ, η) is between (0, 0) and (1, 1). The search step size of λ and η is 0.25, 0.2, 0.167, and 0.1 resulting in the search space size 16, 25, 36, and 100 respectively.

Although boosted HDA didn't provide a big performance boost in this experiment, such a minor performance enhancement may be important for performance-sensitive tasks such as gene classification. What's more important, from Table 3 we find that the boosted HDA classifier isn't sensitive to the size of the search space: for example, the boosted HDA classifier from a weak set of 16 single classifiers achieves better performance (62.09 percent) than the best single classifier (61.80 percent) of search space size 100 after three iterations. So, instead of searching a large parametric space to find the best single classifier, the boosted HDA classifier provides a more efficient way to combine a small set of classifiers into a more powerful one.

Discussion and conclusions

In this article, we've proposed a novel HDA method for classification. This method addresses the high dimensionality problem by applying HDA in an optimal linear discriminant subspace. To reduce the computational complexity and combine multiple classifiers into a powerful one, we also proposed boosted HDA. In applying our proposed approach to gene classification of yeast cell cycle regulation data, HDA's demonstrated superior performance indicates that it is a promising and efficient approach to microarray data analysis.

We see three main contributions from our work. First, HDA provides a richer set of alternatives to a single method, such as LDA. As a result, it not only compensates for regularization that is afflicted by all sample-based estimation methods, but also increases the effective dimension of the projected subspace.

Second, to reduce the search time of parameter space, we propose boosted HDA. It boosts not only the individual features but also a set of weak classifiers. The weighted training scheme in AdaBoost adds indirect nonlinearity and adaptivity to the linear methods and thus enhances it by iterations. With affordable computational cost, AdaBoost can provide a unified and stable solution to find a close to optimal PCA-LDA prediction result.

Third, HDA provided insights on transcriptomic data into the dynamics of gene networks, which could shed light on as yet unrecognized network components and interactions.[19] One main limitation on the use of genomic data to better understand cellular networks in infectious agents is the inability to assign gene functionality. This study may offer an effective solution to circumvent this problem: classifying coexpressed genes in a developmental cycle helps us to identify what could conceivably be network modules.

Many interesting issues in microarray data await further investigations. The immediate improvement of the presented method is to extend the 1D vector model in HDA to a 2D image model and kernel framework, which can keep temporal information of time-series gene expression data and handle nonlinearly separated data. **MM**

Acknowledgments

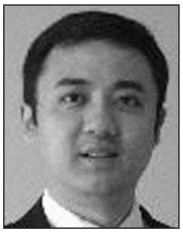
This work is supported in part by San Antonio Life Science Institute (SALSI) to Qi Tian and Feng Liu, ARO grant W911NF-05-1-0404, Dept. of Homeland Security, Office of Naval Research (DHS-ONR) N0014-07-1-0151 to Qi Tian, and NIH grant 1R21AI067543-01A1, San Antonio Area Foundation Biomedical Research Funds, and University of Texas at San Antonio Faculty Research Awards to Yufeng Wang. The project described was also supported by grant number SC1GM081068 from the National Institute of General Medical Sciences (to Yufeng Wang). The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institute of General Medical Sciences or the National Institutes of Health. Jennifer Neary is supported by NIH MBRS-RISE (Minority Biomedical Research Support Research Initiative for Scientific Enhancement, grant GM-60655).

References

1. M.B. Eisen et al., "Cluster Analysis and Display of Genome-Wide Expression Patterns," *Proc. Nat'l Academy of Science,* vol. 95, 1998, pp. 14863-14868.
2. S. Tavazoie et al., "Systematic Determination of Genetic Network Architecture," *Nature Genetics,* vol. 22, no. 3, 1999, pp. 281-285.
3. P. Tamayo et al., "Interpreting Patterns of Gene Expression with Self-Organizing Maps: Methods and Application to Hematopoietic Differentiation," *Proc. Nat'l Academy of Science,* vol. 96, 1999, pp. 2907-2912.
4. S. Dudoit, J. Fridlyand, and T.P. Speed, *Comparison of Discrimination Methods for the Classification of Tumors Using Gene Expression Data,* tech. report 576, 2000.
5. L. Li 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,* vol. 17, no. 12, 2001, pp. 1131-1142.
6. J. Khan et al., "Classification and Diagnostic Prediction of Cancers Using Gene Expression Profiling and Artificial Neural Networks," *Nature Medicine,* vol. 7, no. 6, 2001, pp. 673-679.
7. M.P.S. Brown et al., "Knowledge-Based Analysis of Microarray Gene Expression Data by Using Support Vector Machines," *Proc. Nat'l Academy of Science,* 2000, pp. 262-267.
8. I.T. Jolliffe, *Principal Component Analysis,* 2nd ed., Springer-Verlag, 2002.
9. R. Duda, P. Hart, and D. Stork, *Pattern Classification,* 2nd ed., John Wiley & Sons, 2001.
10. R. Beveridge et al., "A Nonparametric Statistical Comparison of Principal Component and Linear Discriminant Subspaces for Face Recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, IEEE CS Press, 2001, pp. 535-542.
11. J.H. Friedman, "Regularized Discriminant Analysis," *J. Am. Statistical Assoc.,* vol. 84, no. 405, 1989, pp. 165-175.
12. Q. Tian et al., "Parameterized Discriminant Analysis for Image Classification," *Proc. IEEE Int'l Conf. on Multimedia and Expo* (ICME), IEEE CS Press, 2004, pp. 5-8.

13. M.B. Eisen et al., "Cluster Analysis and Display of Genome-Wide Expression Patterns," *Proc. Nat'l Academy of Science,* vol. 95, no. 25, 1998, pp. 14863-14868.
14. S. Ng, S. Tan, and V.S. Sundararajan, "On Combing Multiple Microarray Studies for Improved Functional Classification by Whole-Data Set Feature Selection," *Genome Informatics,* vol. 14, 2003, pp. 44-53.
15. M.P. Brown et al., "Knowledge-Based Analysis of Microarray Gene Expression Data by Using Support Vector Machines," *Proc. Nat'l Academy of Science,* vol. 97, no. 1, 2000, pp. 262-267.
16. A. Mateos et al., "Systematic Learning of Gene Functional Classes from DNA Array Expression Data by Using Multiplayer Perceptrons," *Genome Research,* vol. 12, no. 11, 2002, pp. 1703-1715.
17. C. Van Rijsbergen, *Information Retrieval,* 2nd ed., Butterworth-Heinemann, 1979.
18. Y. Freund, "Boosting a Weak Learning Algorithm by Majority,*" Information and Computation,* vol. 121, no. 2, 1995, pp. 256-285.
19. P.M. Bowers et al., "Use of Logic Relationships to Decipher Protein Network Organization," *Science,* vol. 306, no. 5705, 2004, pp. 2246-2249.

Yijuan Lu is a PhD candidate in computer science at the University of Texas at San Antonio (UTSA). Her research interests include pattern recognition and bioinformatics. She received the 2007 HEB Dissertation Fellowship. She received a BS in computer science from Auhui University.



Qi Tian is an assistant professor in the Computer Science Department of UTSA. His research interests include multimedia information retrieval, computer vision, and bioinformatics. He received a PhD in electrical and computer engineering from the University of Illinois at Urbana-Champaign. He is a senior member of the IEEE and a member of the ACM.

Maribel Sanchez is a systems analyst at UTSA. She was a recipient of the National Institutes of Health (NIH) Minority Biomedical Research Support—Research Initiative in Science Enhancement (MBRS-RISE) and Minority Access to Research Careers—Undergraduate Student Training for Academic Research fellowships. She received dual BS degrees in biology and computer science at UTSA.

Jennifer Neary is a PhD student in bioinformatics at UTSA. Jennifer is supported by MBRS-RISE, a federally funded research training program for minorities and disadvantaged students. She earned a BS in biochemistry from Angelo State University and an MS in biology from the University of Texas at the Permian Basin.

Feng Liu is a professor in the Department of Pharmacology and Biochemistry at the University of Texas Health Science Center at San Antonio. His research interests include insulin signal transduction pathways and molecular mechanisms regulating aging. He received a PhD in biochemistry from Iowa State University.

Yufeng Wang is an assistant professor in the Department of Biology and the South Texas Center for Emerging Infectious Diseases at UTSA. Her research interests include comparative genomics, systems biology, and molecular evolution of infectious diseases. She received a PhD in bioinformatics and computational biology from Iowa State University.

Readers may contact Yijuan Lu at lyijuan@cs.utsa.edu, Qi Tian at qitian@cs.utsa.edu, and Yufeng Wang at yufeng.wang@utsa.edu.

For further information on this or any other computing topic, please visit our Digital Library at http://computer.org/publications/dlib.