BIOINFORMATICS

Vol. 00 no. 00 2007
Pages 1–7

Using Rank-One Biclusters to Classify Microarray Data

Nasimeh Asgarian, Russell Greiner
Department of Computing Science, and the Alberta Ingenuity Center for Machine Learning,
University of Alberta, Edmonton, AB, Canada, T6G2E8

ABSTRACT

Motivation: A DNA-microarray measures the gene expression levels of tens of thousands of genes for a particular sample, corresponding to some specific experimental condition. Our goal is to learn a microarray classifier that can distinguish different classes — *e.g.*, to predict which patient will respond well to a treatment, based on the data from his/her microarray. Unfortunately, the large number of genes and the small number of samples make building such classifiers very challenging.

Results: This paper proposes a method for learning a microarray classifier by first reducing the dimensionality of the data matrix using *biclusters*, where each bicluster is a subset of genes and a subset of samples whose expression values have similar patterns. We propose a novel algorithm for finding biclusters from the microarray data, based on the best rank-1 matrix approximation, then show how to use these biclusters to classify novel samples. We demonstrate that our method works effectively by comparing its prediction accuracy with that of other classifiers, including one based on another bicluster algorithm, over a number of publicly available microarray datasets, both diagnostic and prognostic.

Availability: `http://www.cs.ualberta.ca/~greiner/R/RoBiC`

Contact: {nasimeh, greiner}@cs.ualberta.ca

1 INTRODUCTION

Biologists use DNA-microarrays to measure the gene expression levels of biological samples, where different samples may correspond to different experimental conditions, different states of a tissue (*e.g.*, diseased vs healthy), different individuals, etc. Users analyze such data to understand how expression levels differ in different conditions, which can help determine which genes are involved in some disease, suggest biomarkers of a specific disease, propose targets for drug intervention, and enable the use of microarrays as a screening tool (McLachlan *et al.*, 2004).

This paper focuses on classification: *e.g.*, given a patient's microarray gene expression data, predict how that patient will respond to a certain type of treatment. One challenge in learning such a classifier is the dimensions of the training data: there are typically very few samples with too many features, often less than a hundred samples versus more than 50,000 genes.

One approach around this involves reducing the dimensionality of the data, using perhaps LDA (Somorjai *et al.*, 2003) or SVD (Kluger *et al.*, 2003). We use an alternative, novel way to reduce the dimensionality, based on first finding the "*biclusters*" within the data. Notice this differs from traditional clustering methods, such as hierarchical or K-means approaches, that seek various sets of genes that are similar over *all* samples (or perhaps sets of samples that are similar over *all* genes). By contrast, a bicluster is a *specified subset of genes that are similar for a specified subset of samples*.

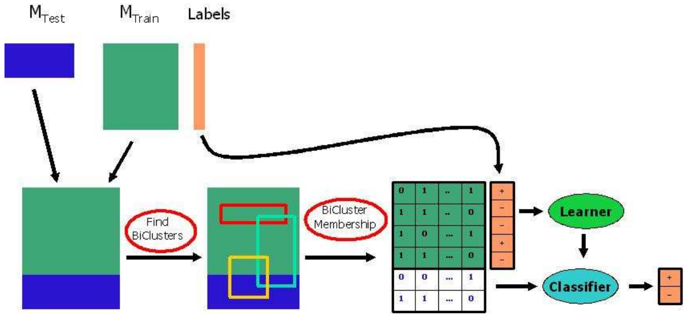

Fig. 1. The BIC "Bicluster Classifier" system first finds the biclusters within the training data (without class labels) and test data, then learns a classifier on these "bicluster membership" features, which it uses to predict labels for the test data

This has a biological motivation, in that we expect that some condition, active in some patients, will trigger some subset of genes, which means that that subset of genes will be co-expressed (that is, correlated) for the patients with that condition. But only for those patients; we expect those genes to behave essentially independently under other conditions (Madeira and Oliveira, 2004).

We describe below our particular approach, ROBIC, for finding these correlated genes and patients, based on the *component values* of the principle eigenvectors. There are many other techniques that also use eigenvectors to reduce the dimensionality of a space. These other approaches, however, work by first projecting the instances onto the space defined by the eigenvectors associated with largest eigenvalues; by contrast, our approach explicitly considers the specific *components* of some eigenvectors, and uses these real values to find the (discrete) subset of genes and subset of patients belonging to each bicluster; see Section 3.1.

We then identify each patient with his/her membership in these biclusters, in that bit #i is 1 for a patient if this patient belongs in bicluster #i. Hence, this process reduces the dimensionality, by mapping each patient-record of say 50,000 real values, to a small number (say 30) of bits. Our system then uses some learning algorithm (*e.g.*, SVM) to produce a classifier over these 30-tuples; see Figure 1.

Notice this method is implicitly selecting a subset of genes from the original space. For example, if biclusters #5 and #7, based respectively only on genes $\{g_1, g_6, g_{93}\}$ and $\{g_1, g_{20}, g_{21}, g_{45}\}$, are sufficient to produce an effective classifier, then we are implicitly asserting that these 7 genes are responsible for the class label. (But this might be based on the expression values for these genes in a subtle way; see Section 4.2.)

This paper's two main contributions are…

- An algorithm, RoBiC, for finding biclusters from microarray data, based on the best rank-1 matrix approximation.
- A method, BiC, for using such a biclustering algorithm to learn a sample classifier.

We demonstrate that this system works effectively across a number of standard DNA microarray classification tasks. The literature shows that many of the "diagnostic" tasks — *e.g.*, predicting whether a patient has some type of cancer — are relatively easy, as a small number of genes are sufficient to produce very high classification accuracy. We focus instead on "prognostic" predictors — *e.g.*, predicting whether a patient (who is known to have cancer) will respond to some specified treatment. These tasks are more difficult both because there tend to be no small set of genes whose expression levels distinguish the classes, and also because the sample sizes are quite small, as they include only the subset of patients with the disease.

Section 2 provides background information on biclustering methods and classification tasks, and summarizes previous work related to biclustering and classifying microarrays. Section 3 describes the biclustering problem and the structure of our algorithms: RoBiC for finding biclusters and BiC for using biclusters to classify microarray samples. Section 4 describes the results of applying our biclustering and classification method on a number of publicly available microarray datasets. We also compare our results to those of other approaches (including ones based on an alternative biclustering algorithms) on these datasets, and provide a statistical test ("permutation test") to demonstrate that our results are significant.

Web (07) provides additional information related to this work, including the datasets used in the studies reported here, as well as other studies, and also other relevant statistics about our results (*e.g.*, timing information).

2 RELATED WORK

2.1 Our Approach

The microarray data is an $n \times p$ matrix M, which is indexed by a set of rows (genes), $G = \{g_1, g_2, \ldots, g_n\}$ and a set of columns (samples), $S = \{s_1, s_2, \ldots, s_p\}$, where $|G| = n \gg p = |S|$. Each element m_{ij} is a real value, which usually represents (the logarithm of) the expression level of gene i for sample j.

A *bicluster* is a subset of genes that have a similar pattern over a subset of samples. This corresponds to a sub-matrix M_{IJ} of M, where $I \subseteq G$ is a subset of genes (rows), and $J \subseteq S$ is a subset of samples (columns). Our challenge is: Given a data matrix, M, find (and then use) a set of K biclusters, $B_k \approx M_{I_k,J_k}$ where for each $k = 1, \ldots, K$, the set of B_k's genes $I_k \subseteq G$ have a similar pattern within the set of B_k's samples $J_k \subseteq S$.

2.2 Other BiCluster Approaches

The *Plaid* system (Lazzeroni and Owen, 2002) models gene expression data $M = (m_{ij})$ using a set of biclusters (which they call "layers")

$$m_{ij} \quad = \quad \mu_0 + \textstyle\sum_{k=1}^{K} \theta_{ijk}\, [\, \mu^k + \alpha_j^k + \beta_i^k \,] \qquad (1)$$

where μ_0 is the background value for entire matrix, K is the number of biclusters, θ_{ijk} is 1 if bicluster k includes position (i, j) and 0 otherwise, and the values $\langle\, \mu^k,\ \{\alpha_j^k\}_j,\ \{\beta_i^k\}\,\rangle$ are associated with bicluster k. Hence, if a bicluster k includes (i, j), then it contributes a value there that is the sum of a background value μ^k, a value β_i^k associated with the column i and a value α_j^k associated with the row j. Note that a position can belong to 0, 1 or more biclusters.

While the Plaid model is "additive", our RoBiC approach is based on a multiplicative variant:

$$m_{ij} \quad = \quad \mu_j \ + \ \textstyle\sum_{k=1}^{K} \theta_{ijk}\, [\, \alpha_j^k \times \beta_i^k \,] \qquad (2)$$

where μ_j is just used to make each row 0-mean. Again, we allow each gene-patient pair (i, j) to belong to 0, 1 or more biclusters.

Equation 2 is clearly related to *singular value decomposition*, SVD, which is often used to reduce the dimensionality of microarray data (Alter *et al.*, 2000; Wall *et al.*, 2003; Ding, 2003; Hastie *et al.*, 2000; Kluger *et al.*, 2003). These SVD-based systems first compute a list of eigenvalue/(row)eigenvector/(column)eigenvector triples $\langle\, v^{(k)}, \alpha^{(k)}, \beta^{(k)}\,\rangle_k$ from M, where here $v^{(k)} \in \Re$ (sorted by $|v^{(1)}| \geq |v^{(2)}| \geq \ldots$), $\alpha^{(k)} \in \Re^n$ and $\beta^{(k)} \in \Re^p$, such that $M \approx \sum_k v^{(k)} \alpha^{(k)} \times {\beta^{(k)}}^T$. These systems then consider only the $k \ll p, n$ largest eigenvalues, and afterwords project each p-ary patient row vector $M(j,:)$ onto k real values $M(j,:)^T \cdot [\beta^{(1)}, \beta^{(2)}, \ldots, \beta^{(k)}]$, whose i^{th} component $M(j,:)^T \cdot \beta^{(i)}$ depends on all component of both the $M(j,:)$ and $\beta^{(i)}$ vectors. They could then use these smaller vectors, perhaps as input to a classifier.

Our RoBiC system is significantly different as it maps each patient row vector $M(j,:)$ onto *k bits*, where the i^{th} bit is 1 if this patient belongs to the i^{th} bicluster, which (we will see) is true if the *j^{th} component* of the i^{th} eigenvector $\beta^{(i)}$ is significantly larger than the other values in this eigenvector. N.b., this does not explicitly involve taking a dot-product, nor require using any other information from $M(j,:)$, as RoBiC *explicitly focuses on the* real-valued components of the leading eigenvector, *rather than on the eigen*values. Another critical difference is that RoBiC computes these eigenvalue/eigenvectors triplets *sequentially*, while standard SVD systems compute *all* of these triplets at once; see Section 4.2.

Many of these systems evaluate the quality of a bicluster by, for example, testing against a null hypothesis of no structure in the data matrix. By contrast, we use our biclusters as part of a classification task, which makes it relatively easy to evaluate their effectiveness.

Madeira and Oliveira (2004) provide a comprehensive survey that analyzes a large number of existing approaches to biclustering, organized by the type of biclusters they can find, the patterns of biclusters that are discovered, the methods used to perform the search, and the target applications. Similarly, after introducing their new algorithm (Bimax), Preli *et al.* (2006) then provide a systematic comparison and evaluation of the prominent biclustering methods, as well as clustering and hierarchical clustering, for the purpose of *gene* classification. They conclude the bicluster-based approaches are the most effective.

2.3 Classification Algorithms

Many researchers have developed supervised learning methods to produce classifiers that use a sample's gene expression profile to predict if that sample belongs to a known class or not. van't Veer *et al.* (2002) uses one such technique to try to predict the clinical outcome of breast cancer — *i.e.*, predict whether a patient will remain disease-free for a certain period of time. They first hand-selected 231 genes (of the $25,000$ genes on the microarray) that were significantly associated with the disease outcome, *based on all samples*, then ranked these genes based on the magnitude of their respective correlation coefficient. They then sequentially added the top 5 remaining genes (based on this ranking), each time evaluating the quality of the current gene set by the cross-validation classification accuracy of the classifier based on that subset. They obtained an accuracy of 83% using 70 of these genes.

Unfortunately, the authors selected the features (genes) by looking at the *entire dataset*. Instead, they should have done the feature selection within each cross-validation fold. When this correction, their prediction accuracy went down by 10%, to 73% (Molla *et al.*, 2004). We will use this BreastCancer dataset in our studies (see P_1 in Table 1 below[1]), and also use "in-fold feature selection".

Golub *et al.* (1999) developed another method for class prediction: They first find the genes whose expression values were strongly correlated with the class to be predicted, then developed a method, "neighborhood analysis", to define an "idealized expression pattern", corresponding to a gene that is uniformly high in one class and uniformly low in the other. They then select and use a subset of *informative genes*, chosen based on their correlation with the class label. Each informative gene casts a weighted vote, which are added to determine the winning class as well as a "prediction strength". They applied their method to a human leukemia database, producing a predictor for AML versus ALL class prediction. This predictor made strong prediction for 29 of the 34 samples, achieving an accuracy of 100% for those 29 samples. Notice this was "training set accuracy"; we provide generalization error for an extension of this dataset, D_3. They then use their neighborhood analysis system to predict response to chemotherapy (a prognostic task), but found no informative genes, and found no evidence of a strong multi-gene expression signature correlated with clinical outcome. We report our result on this data set (P_2) in Section 4.

Singh *et al.* (2002) also used a nearest neighbor (k-NN) approach, with gene selection, to predict the class label of prostate cancer samples (D_4). They then used the same approach to predict the clinical outcome, and here obtained 90% accuracy for a 5-gene model with 2 nearest neighbors.

They then used a permutation test (Mielke and Berry, 2001) to determine whether their results were statistically significant. Here, the class labels for the data points are randomly permuted — *i.e.*, randomly re-arranged, keeping only the original probability distribution of the labels. (Hence if 57% of the labels are positive in the original dataset, then 57% of the instances in the permuted set, will be labeled positive.) They then apply their learning and classification system to this new data set, with the permuted labels. Given that there is no "signal" in the labels, we expect the prediction accuracy here to be around baseline — here 57% — unless their algorithm is finding information that is not there. This is why they compare the results of their algorithm on the original unpermuted data (say 90.1% accuracy), with results of their algorithm on the permuted data, for (here) 1000 separate permuted datasets — each of which has a different permutation of the labels. In particular, they ask how often any (5-fold cross-validated) permutation score exceeded theirs, as a measure of the chance that their algorithm achieved its score on the original data by chance alone. They found accuracies that matched or exceeded their original system, 37 times. We also use their datasets (D_4 and P_4 below) and employ a permutation test to determine validity of our results.

Gordon *et al.* (2002) described yet another way to use gene expression measurements to predict clinical outcomes in cancer. They select 2 genes randomly, then use the ratios of their expression levels and choose thresholds to accurately distinguish between different classes of samples. This approach works nicely for predicting different types of tumors in mesothelioma lung cancer (D_2), but when applied on the Brain clinical outcome dataset (P_3 (Pomeroy *et al.*, 2002)), their prediction accuracy was only 68%.

The "shrunken centroid method" (Tibshirani *et al.*, 2002) is widely used for classifying microarray data. It shrinks the class centroids (by a shrinkage parameter δ, typically determined by a cross-validation process) toward the overall centroids after standardizing by the within-class standard deviation for each gene. This standardization has the effect of giving higher weight to genes whose expression is stable within samples of the same class. We compare our system with this approach below.

Pranckeviciene and Somorjai (2006) argue that one should first use feature selection to identify a good, small subset of the genes, and provides a particular method, LDA EX-FS, for this task. They compare the classification results obtained with four different classifiers, using this reduced number of genes. For microarray classification, in general, they found the best classifiers are Support Vector Machine (SVM; Alpaydin (2004)) and the shrunken centroids methods, especially when run on a small selection of good features. In addition, most findings confirm that feature selection leads to better classification performances, and that it is generally advisable to perform some dimensionality reduction, or at least exploratory analysis prior to classification. We borrow their ideas, of using feature selection (although based on a different method, using biclusters), and also using SVMs.

3 METHODS

This section describes the structure of our RoBiC algorithm for finding the biclusters, and our BiC system for using biclusters to classify microarray samples.

3.1 Finding Biclusters, RoBiC

We view each bicluster as a sub-matrix M_{IJ} whose entries are "multiplicatively correlated" — that is, we identify each sample $j \in J$ with a real value β_j, and each gene $i \in I$ with a real value α_i, and then view (this bicluster's contribution to) the matrix value $m_{ij} \in M_{IJ}$ as approximately $\alpha_i \times \beta_j$; see Equation 2. Perhaps genes $I = \{g_1, g_2, g_3\}$ are correlated, in that their expressions are in a 1 to 3 to 2 ratio. The actual expression values, however, depend on the patient: here patient p_2 is 10 times as responsive as patient p_1, and patient p_3 (resp., p_4) is 2 times (resp. 4 times) as responsive as p_1. Moreover, we assume that patients 5 through p are not involved, and similarly genes 4 through n are not involved. We would then set $\alpha' = \langle\, 1, 3, 2, 0, 0, \ldots, 0\,\rangle$ and $\beta' = \langle\, 1,\ 10,\ 2,\ 4, 0, 0, \ldots, 0\,\rangle$. Notice we set the values for the uninvolved genes and patients to 0. (Of course, this numbering is just for pedagogical purposes; we do not anticipate that the genes in a bicluster will happen to be the first three indexed in the matrix, nor should this bicluster happen to include all-and-only the first 4 patients in the study.)

Our challenge is to find these highly correlated sub-matrices. RoBiC seeks each bicluster in two steps: It first seeks the best "over-complete" vectors α and β, which include information for the

[1] "P_i" and "D_j" refer to the datasets listed in Tables 1 and 2.

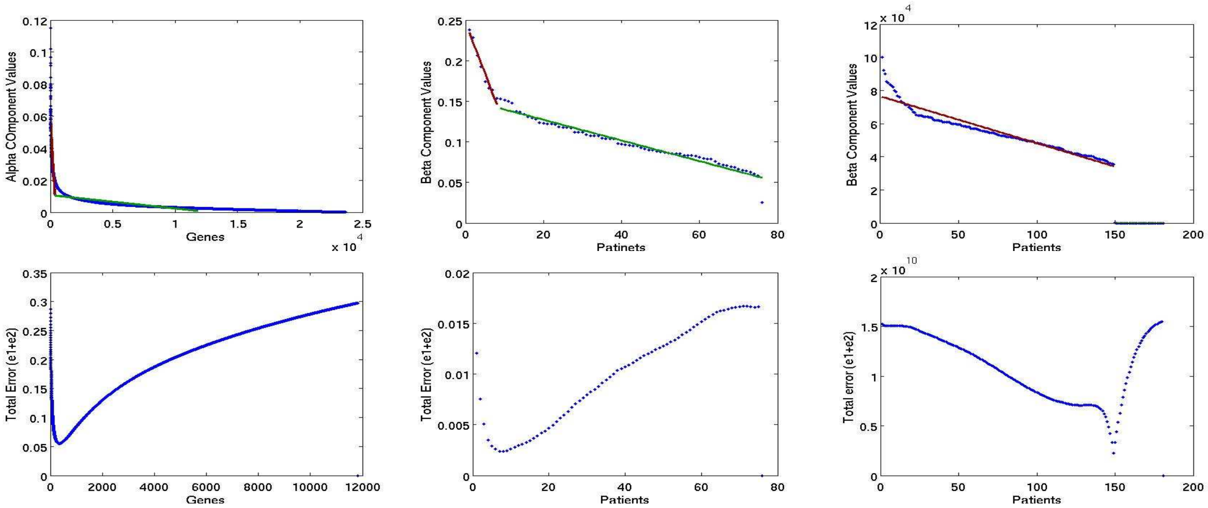

Fig. 2. [Top] Sorted eigenvector $\alpha^{(s)}$ (resp., $\beta^{(s)}$, $\beta^{(s)}$) values corresponds to genes (resp., patients, patients), with two line-segments superimposed. (Note each graphs show the *component values* of a single eigenvector; n.b., these are not the set of all eigen*values*.) [Bottom] Error values, for each position. All figures are from the first bicluster of some dataset; the left 2 columns are from the BreastCancer P_1 dataset, while the far right is from Lung Cancer D_3.

bicluster, as well as non-zero values for the other genes and samples. It then uses these vectors to find the subset of genes (samples) that belong to the bicluster — as the genes (samples) with the largest component value in the relevant eigenvector.

In more detail, given a large $n \times p$ matrix M (*e.g.*, $50,000 \times 80$), RoBiC first finds two vectors, α and β, of sizes $|\alpha| = n$ and $|\beta| = p$ that minimize

$$\|M - \alpha\,\beta^T\| \quad = \quad \sum_{ij}(m_{ij} - \alpha_i\beta_j)^2. \qquad (3)$$

that is, it seeks the best (that is, least squared error) rank-1 approximation of M. It is well known that these vectors correspond to the eigenvectors of the largest eigenvalue, which can be computed using standard singular value decomposition (SVD) (Golub and Loan, 1989)

$$[\alpha, v, \beta] = \mathrm{SVD}(M, 1)$$

(Technically the β used in Equation 3 is $v \times \beta$, where $v \in \Re$ is the largest eigenvalue.) We let $\alpha \in \Re^n$ represent the genes' vector and $\beta \in \Re^p$ represent the samples' vector.

RoBiC then sorts the absolute values in α and β in descending order producing $\alpha^{(s)}$ and $\beta^{(s)}$ — see Figure 2[Top]. RoBiC then applies a "hinge function" to $\alpha^{(s)}$ (resp., $\beta^{(s)}$) to identify the subset of genes (resp., samples) that belong to this bicluster, which are the ones whose associated eigenvector components are largest.

It does this by finding the "best-fitting pair of lines": That is, for each $i = 1, \ldots, n$, find the straight line that best fits the components $\langle\, \alpha_1^{(s)}, \ldots, \alpha_i^{(s)} \,\rangle$ — *i.e.*, that minimized the squared error — and then let $e_1^{(i)}$ be the residual squared error. Similarly let $e_2^{(i)}$ be the residual squared error of the best fitting line for values $\langle\, \alpha_{i+1}^{(s)}, \ldots, \alpha_n^{(s)} \,\rangle$. RoBiC then sets

$$i^* = argmin_i\{e_1^{(i)} + e_2^{(i)}\}$$

to be the index that minimizes the sum of these errors, and declares that the genes from index 1 to i^* belong to the current bicluster; see Figure 2[Bottom]. This has the desired effect of distinguishing the genes that have small values from those with significantly larges ones. RoBiC uses a similar hinge function to determine which *samples* belong to this bicluster. It also computes a bicluster-membership vector B_1, whose j^{th} element is 1 if sample j (w.r.t. the original order) belongs to the this first cluster.

```
RoBiC( M, K, $h(\cdot)$ ): Bicluster_Membership
    % M is (unlabeled) data matrix, K is # of biclusters,
    % $h(\cdot)$ is hinge function
$M^{(1)} := M$
for $k = 1..K$ do
  1. Compute the largest singular value decomposition of $M^{(k)}$,
        $[\alpha, v, \beta] \;=\; SVD(M^{(k)}, 1)$
        • $\alpha \times v \times \beta^T \;=\;$ best rank-1 approximation to $M^{(k)}$
        • $\alpha$ (resp., $\beta$) represents the genes (resp., patients)
  2. $\alpha^{(s)} \;=\; \mathrm{Sort}(\alpha);\quad \beta^{(s)} \;=\; \mathrm{Sort}(\beta)$
  3. $i \;=\; h(\alpha^{(s)});\qquad j \;=\; h(\beta^{(s)});$
        % i (resp., j) is "hinge" value for genes (resp., patients)
  4. $\alpha^{(s,i)} \;:=\; \langle\, \alpha_1^{(s)}, \alpha_2^{(s)}, ..., \alpha_i^{(s)},\; 0, 0, \ldots, 0 \,\rangle$
     $\beta^{(s,j)} \;:=\; \langle\, \beta_1^{(s)}, \beta_2^{(s)}, ..., \beta_j^{(s)},\; 0, 0, \ldots, 0 \,\rangle$
     $M^{(k+1)} := M^{(k)} \;-\; \mathrm{Unsort}(\alpha^{(s,i)}) \cdot v \cdot \mathrm{Unsort}(\beta^{(s,j)})^T$
        % Unsort($\alpha^{(s,i)}$) contains the elements of $\alpha^{(s,i)}$,
        % but in the original "pre-sorted" order
     $B_k(\ell) = \begin{cases} 1 & \text{if patient } \ell \text{ is in bicluster } k \\ 0 & \text{otherwise — } i.e., \mathrm{Unsort}(\beta^{(s,j)})[\ell] = 0 \end{cases}$
Return: $K$ bicluster_memberships $R = \{B_k\}$
```

Fig. 3. Pseudo-code for rank-1 bicluster algorithm, RoBiC

RoBiC actually needs to find a *set* of biclusters. So after finding the appropriate sparse α and β vectors for the first bicluster, RoBiC subtracts their outer product from the original matrix to produce a diminished matrix, from which it then again seeks the best bicluster, and so forth; see Equation 2. Note that each gene may belong to 0, 1, or many biclusters, as might a single patient.

Figure 3 summarizes our algorithm. *N.b.*, this process does not use the class labels for the instances.

```
BiC( $B(\cdot)$, $A(\cdot)$, $U_{Train}$, $U_{Test}$, $L_{Train}$): $L_{Test}$
    % $B(\cdot)$ is the biclustering algorithm (e.g., RoBiC(·, 30, ... ) or Plaid(· · · ) );
    % $A(\cdot)$ is the learning algorithm (e.g., SVM or NaïveBayes);
    % $U_{Test}$ is unlabeled test data; $U_{Train}$ is the features associated with the training data;
    % $L_{Train}$ is the class labels associated with the training data; the returned $L_{Test}$ are the predicted labels for the test data
  1. $R \;:=\; B(U_{Train} \cup U_{Test}, \dots)$
     % ... produces biclusters based on the unlabeled part of the training set and the test set. (This may involve feature selection.)
  2. $C \;:=\; A(R|_{Train} \;+\; L_{Train})$
     % ... is the classifier produced from the Training-set portion of $R$ along with the associated training set labels
  3. For $i = 1..u$,    $L_{Test}[i] \;:=\; C(R_{Test}[i])$
     % Use classifier $C$ to predict the class label of each sample in the test set. $R_{Test}[i]$ is the $q + i^{th}$ row of $R$.
```

Fig. 4. Pseudo-code of Bicluster Classifier algorithm BiC

3.2 Bicluster Classifier, BiC

We initially describe each sample as an n-tuple whose i^{th} component is the gene expression value for the i^{th} gene. In the typical "learning+classification task", we will start with q such training samples, each with a class label, and an additional u unlabeled test instances (typically $u = 1$). In general, $p = q + u$ is much less than n. Our goal is to predict the class labels for these u test samples, based on their gene expression values.

Instead of directly using all n features, or selecting a subset of these n genes, to predict the samples' class label, we reduce the dimension of the data matrix M from $n \times p$ real values to $K \times p$ bits corresponding to the $R = \{B_1, B_2, \dots, B_K\}$ matrix returned by RoBiC. Note $K \ll n$.

After augmenting the first q samples with their respective labels, we have data of the form suggested by the "BiCluster Membership" table in Figure 1. BiC then takes this information, and learns a classifier based on the first q labeled samples, which it applies to the remaining u unlabeled samples, to predict their respective class labels. We experimented with several learners, and here report results using SVMs. To reduce risk of over-fitting, we further reduce the dimensionality by using only a subset of the biclusters. We used Weka's built-in in-fold feature selection algorithm to find the number of biclusters that give maximum prediction accuracy on test data; see Section 2.3. Figure 4 sketches our basic bicluster classifier algorithm, BiC. Note our overall system corresponds to BiC(RoBiC, ...). We consider other variants in Section 4.2 below.

4 EXPERIMENTAL RESULTS

Our goal is to produce an effective classifier of high-dimensional microarray data. To evaluate our approach, we therefore applied our BiC(RoBiC, ...) system to several publicly available datasets, and compared our results with results obtained using other approaches, including one based on an alternative type of biclusters. In each case, our RoBiC system seeks $K = 30$ biclusters from the data set.

4.1 Prognostic Datasets

We first considered datasets for predicting the clinical outcome of a certain treatment; see top row of Table 1. All but one of the datasets were complete — *i.e.*, included a value for each gene-sample pair. For that one incomplete dataset (P_1, BreastCancer), we found that 2 of the patients were missing values for roughly half (11,776) of the genes. We therefore removed those two patients; see Section 4.3. We then removed a gene if its expression value was missing for any of the remaining patients; this meant removing 856 of the 24,481 genes.

The rest of Table 1 presents the prediction accuracies of various algorithms for the data sets, based on 5-fold cross-validation. The bottom of row 1 shows the "base-line" percentage, which is the accuracy of using the majority class. The second row presents the result of the original study. Row 3 shows the result for applying (linear kernel) SVM classifiers directly to the original data matrix, M, and row 4 corresponds to using shrunken centroids (PAM; (Tibshirani *et al.*, 2002)), which is one of the standard algorithms for learning classifiers for microarray data. (We empirically found setting $\delta = 0$ worked best for the BreastCancer dataset; we therefore continued to use this setting for the other data.) Row 5 contains the results for BiC(Plaid, SVM): *i.e.*, finding the biclusters using the Plaid (Lazzeroni and Owen, 2002) biclustering algorithm (rather than our RoBiC), then using SVM as the sample classification algorithm. Our main results appear in row 6, which shows the prediction accuracies based on the biclusters found using BiC(RoBiC, SVM), as well as the standard deviation.[2] The final row provides the results of a 1000 trial permutation test. We see, for example, that the average accuracy of BiC(RoBiC, SVM) over 1000 permutations of the BreastCancer dataset was 54.08%, and that 0 of these 1000 individual scores exceeded RoBiC's score on the original data, 90.79%. The next columns suggests that our results are significant with $p \leq 0.027$ for AML(Outcome) and with $p \leq 0.022$. for Prostate(Outcome). ((Web, 07) provides information that characterizes the biclusters found.)

4.2 Other Experiments

So far, focused on the difficult "prognostic" datasets. We also ran our algorithm on the typically easier "diagnostic" ones; see Table 2. We see that our BiC(RoBiC,SVM) system performs well on these easy tasks as well — consistently obtaining 85% accuracy, or better. In many cases, one bicluster is sufficient to produce these results, as it easily splits the data into the classes; see right-most pair of graphs in Figure 2, for Lung Cancer (D_3).

One concern is our methodology, which finds biclusters using feature values from both the test and the training data (but not the labels). While this is clearly appropriate in the transductive framework, it does not directly match the canonical inductive framework, which typically involves classifying a single instance at a time, based on a classifier learned from the training set.

[2] The large variance values here are due mostly to the small sample size.

Table 1. Empirical Results: Prediction accuracies of various learners, using 30 biclusters and 5-fold cross validation

	Info/Dataset	P_1: BreastCancer (van't Veer *et al.*, 2002)	P_2: AML(Outcome) (Golub *et al.*, 1999)	P_3: Brain(Outcome) (Pomeroy *et al.*, 2002)	P_4: Prostate(Outcome) (Singh *et al.*, 2002)
1	#Samples / #Genes Class distr'n	76 / 23,625 32 / 49 = 57.89%	15 / 7,129 8 / 7 = 53.33%	60 / 7,129 39 / 21 = 65%	21 / 12,600 8 / 13 = 61.90%
2	Original Result	73%		78%	**90%**
3	SVM	67.11%	53.33%	65%	47.62%
4	PAM	83%	60%	62%	38%
5	BiC(Plaid, SVM)	63.16%	60%	65%	61.90%
6	BiC(RoBiC**, SVM)**	**90.79% ± 7.6**	**80% ± 18.2**	**95% ± 7.5**	**85.71% ± 12.0**
7	Permutation Test	54.08% (0/1000)	51.66% (27/1000)	60.50% (0/1000)	56.77% (22/1000)

To match the more familiar model, we considered learning a classification function for each bicluster, based only on the training data. That is, suppose we determine that genes $\{g_1, g_2, g_3\}$ belong to bicluster#1, which includes samples $\{s_1, \ldots, s_{30}\}$ but not $\{s_{31}, \ldots, s_{80}\}$. We would then learn a classifier that, given the expression values of these 3 genes, predicts whether the associated sample belongs in the cluster or not.

Notice this "bicluster membership predictor", BMP$_1$, was based only on the training data. Hence, the training phase would produce $K = 30$ such BMP$_i$s — one for each bicluster — and also a classifier that maps from these K bits to a class label, corresponding perhaps to the disease outcome. (This is reminiscent of stacking (Wolpert, 1992).) Then, to classify a novel test sample, we would run the K BMP$_i$s on that sample's data to produce K bits (or fewer if we used some feature selection), then let BiC use those values to predict the outcome. We did try this but found that this approach did not work well. For example, for the BreastCancer dataset, this method produced a (5-fold cross-validated) prediction accuracy of only 73.75%.

Towards explaining this, recall that our biclusters are actually built *sequentially*: bicluster#k is based on the matrix $M^{(k)}$ formed after subtracting off biclusters $B_1, \ldots B_{k-1}$. So a classifier trained on the values of set of genes associated with $M^{(k)}$ will not perform well for a novel patient whose expression values correspond to $M^{(1)}$, as the expression values of some of $M^{(k)}$'s gene/patient pairs m_{ij} will differ from the corresponding values in $M^{(1)}$; see Equation 2. Similarly, we cannot simply train all of the classifiers on the $M^{(1)}$ data, as each individual m_{ij} value is actually the sum of all biclusters that include this pair (Equation 2), which means these membership functions are not independent — a patient's membership in one bicluster might throw off his membership in another.

Our RoBiC system finds the top pair of eigenvectors from a matrix, then subtracts their (truncated) outer-product and recurs, finding the top eigenvectors-pair from the "reduced" matrix, and so forth, $K = 30$ times. Another obvious approach is simply to find the top 30 pairs of eigenvectors at once — *i.e.*, simply compute SVD(M,30) once, rather than (in essence) computing SVD(M,1) 30 times. We found, however, that this did not work. For the BreastCancer dataset P_1, the (5-fold cross-validated) prediction accuracy was only 60.53%. To understand why, notice that RoBiC does not deal with the complete eigenvectors, but instead "truncates" them by setting most of the values to 0; see Step 5 of Figure 3. This non-linearity makes a significant difference.

We also considered other learning algorithms in the role of SVM here, including naive bayes, decision trees, nearest neighbor, and SVM with various kernel. We found that SVM with the linear kernel works best. We also experimented with a number of alternative hinge functions, but found that none worked as well as the "best-2-line" hinge function described above; see (Web, 07).

4.3 Discussion

The data in Tables 1 and 2 suggest several obvious conclusions: First, we see that our BiC(RoBiC) system works extremely well: It produces at least 80% accuracy across all 8 different datasets, and over 90% in 3 of them! It is at least as good as many of the other obvious standard algorithms, and appears better than most, especially on the difficult prognostic datasets.[3] This is reinforced by the permutation tests, which strongly suggest that there is a signal in the bicluster membership values, which can be used in predicting prognostic and diagnostic information.

The comparison with BiC(Plaid, ...) suggest that our RoBiC biclusters, based on "multiplicative relations between genes and samples", can produce better classifiers than biclusters based on "additive" relations; *i.e.*, Equation 2 appears better than Equation 1 for this task.

We found that the actual classifier typically involved several different biclusters. In particular, it is not based just on the first or two biclusters found. This is not surprising: The claim that a set of genes within a subset of patients are correlated does not imply that this correlation is related to any pre-specified condition. So the first bicluster found might relate to some other factor (*e.g.*, gender, or some other disease or condition). Moreover, we may need to "subtract" off the influence of this dominant bicluster to reveal the bicluster(s) that corresponds to the specific condition(s) associated with the diagnosis or prognosis.

5 CONCLUSION

DNA microarray data contain a great deal of information about a sample — often enough to predict a patient's diagnosis and prognosis. Unfortunately, as microarrays are very high-dimensional (with many 10s of thousands of genes) and very noisy, it can be very difficult to find the patterns that correspond to these predictions.

[3] In a simple pre-processing step, our system identifies patients with a large number of missing expression values, and removes them from the training set; here, this meant removing 2 (of the 78) patients from the BreastCancer dataset P_1. To be fair to the original publication, which considered all 78 patients, we could declare that our system misclassified those two patients. This would drop the accuracy of our BiC(RoBiC,SVM) system to $90.79 \times 76/78 = 88.46\%$. This is still significantly better than 73%. All of the other systems listed in Table 1 were based on the set of 76 patients.

Table 2. Empirical Results: Prediction accuracies for the *diagnostic* datasets. (30 biclusters, 5-fold cross validation)

Dataset Info	D_1: Colon (Alon *et al.*, 1999)	D_2: Lung Cancer (Gordon *et al.*, 2002)	D_3: AML-ALL (Golub *et al.*, 1999)	D_4: Prostate Cancer (Singh *et al.*, 2002)
# Samples / # Genes Class distr'n	62 / 2,000 40 / 22 = 64.52%	181 / 12,533 150 / 31 = 82.87%	72 / 7,129 47 / 25 = 65.27%	136 / 12,600 77 / 59 = 56.62%
BIC(ROBIC,SVM)	**88%**	**96.13%**	**84.72%**	**86.77%**

This paper presents an effective way to make accurate predictions from microarray data, by first finding biclusters to reduce the dimensionality of the data, then using them. We present a novel algorithm for identifying biclusters based on the best rank-1 matrix approximation, and a method for using such biclusters to classify samples. Our empirical studies suggest that our overall system works very effectively across a number of microarray datasets; and in particular, that it works better than the other standard microarray classification systems we tested, on difficult prognostic tasks. We also show that our specific "rank-1 approximation" approach is more effective than an alternative "additive" bicluster-er, for this task.

See (Web, 07) for more information about our implementation, and additional experimental results.

REFERENCES

Alon, U., Barkai, N., Notterman, D., Gish, K., Ybarra, S., Mack, D., and Levine, A. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. In *PNAS*, volume 96.

Alpaydin, E. (2004). *Intro. to Machine Learning*. MIT.

Alter, O., Brown, P., and Botstein, D. (2000). Singular value decomposition for genome-wide expression data processing and modeling. *PNAS*, **97**, 10101–10106.

Ding, C. (2003). Unsupervised feature selection via two-way ordering in gene expression analysis. *Bioinformatics*, **19**, 1259–1266.

Golub, G. and Loan, C. V. (1989). *Matrix Computations*. The John Hopkins University Press.

Golub, T., Slonim, D., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J., Coller, H., Loh, M., Downing, J., Caligiuri, M., Bloomfield, C., and Lander, E. (1999). Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, **286**, 531–537.

Gordon, G., Jensen, R. V., Hsiao, L.-L., Gullans, S. R., Blumenstock, J. E., Ramaswamy, S., Richards, W. G., Sugarbaker, D. J., and Bueno, R. (2002). Translation of microarray data into clinically relevant cancer diagnostic tests using gege expression ratios in lung cancer and mesothelioma. *Cancer Research*, **62**, 4963–4967.

Hastie, T., Tibshirani, R., Eisen, M., Alizadeh, A., Levy, R., Staudt, L., Chan, W., Botstein, D., and Brown, P. (2000). 'Gene shaving' as a method for identifying distinct sets of genes with similar expression patterns. *Genome Biology*, **1**.

Kluger, Y., Basri, R., Chang, J., and Gerstein, M. (2003). Spectral biclustering of microarray data: Coclustering genes and conditions. *Genome Research*, **13**.

Lazzeroni, L. and Owen, A. B. (2002). Plaid models for gene expression data. *Statistica Sinica*, **12**(1), 61–86.

Madeira, S. C. and Oliveira, A. L. (2004). Biclustering algorithms for biological data analysis: A survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, **1**(1), 24–45.

McLachlan, G., Do, K., and Ambroise, C. (2004). *Analyzing Microarray Gene Expression Data*. Wiley.

Mielke, P. W. and Berry, K. J. (2001). *Permutation Methods: A Distance Function Approach*. Springer.

Molla, M., Waddell, M., Page, D., and Shavlik, J. (2004). Using machine learning to design and interpret gene-expression microarrays. *AI Magazine*, **25**, 23–44.

Pomeroy, S., Tamayo, P., Gaasenbeek, M., Sturla, L., Angelo, M., McLaughlin, M., Kim, J., Goumnerova, L., Black, P., Lau, C., Allen, J., Zagzag, D., Olson, J., Curran, T., Wetmore, C., Biegel, J., Poggio, T., Mukherjee, S., Rifkin, R., Califano, A., Stolovitzky, G., Louis, D., Mesirov, J., Lander, E., and Golub, T. (2002). Prediction of central nervous system embryonal tumour outcome based on gene expression. *Letters to Nature*, **415**.

Pranckeviciene, E. and Somorjai, R. (2006). On classification models of gene expression microarrays: The simpler the better. In *Int. J. Conf. Neural Networks*, pages 6878–6885.

Preli, A., Bleuler, S., Zimmermann, P., Wille, A., Bhlmann, P., Gruissem, W., Hennig, L., Thiele, L., and Zitzler, E. (2006). A systematic comparison and evaluation of biclustering methods for gene expression data. *Bioinformatics*, **22**(9).

Singh, D., Febbo, P., Ross, K., Jackson, D., Manola, J., Ladd, C., Tamayo, P., Renshaw, A., D'Amico, A., Richie, J., Lander, E., Loda, M., Kantoff, P., Golub, T., and Sellers, W. (2002). Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell*, **1**.

Somorjai, R., Dolenko, B., and Baumgartner, R. (2003). Class prediction and discovery using gene microarray and proteomics mass spectroscopy data: Curses, caveats, cautions. *Bioinformatics*, **19**.

Tibshirani, R., Hastie, T., Narasimhan, B., and Chu, G. (2002). Diagnosis of multiple cancer types by shrunken centroids of gene expression. In *PNAS*, volume 99.

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

Wall, M., Rechtsteiner, A., and Rocha, L. (2003). *Singular Value Decomposition and Principal Component Analysis*, pages 91–109. Kluwer.

Web (07). `http://www.cs.ualberta.ca/~greiner/RESEARCH/RoBiC`.

Wolpert, D. (1992). Stacked generalization. *Neural Networks*, **5**.