Proceedings of ALGORITMY 2005
pp. 250–259

ANALYSIS OF THE SINGULAR VALUE DECOMPOSITION AS A TOOL FOR PROCESSING MICROARRAY EXPRESSION DATA

DESMOND J. HIGHAM*, GABRIELA KALNA†, AND J. KEITH VASS‡

Abstract. We give two informative derivations of a spectral algorithm for clustering and partitioning a bi-partite graph. In the first case we begin with a discrete optimization problem that relaxes into a tractable continuous analogue. In the second case we use the power method to derive an iterative interpretation of the algorithm. Both versions reveal a natural approach for re-scaling the edge weights and help to explain the performance of the algorithm in the presence of outliers. Our motivation for this work is in the analysis of microarray data from bioinformatics, and we give some numerical results for a publicly available acute leukemia data set.

keywords: bioinformatics, clustering, data mining, microarray, power method, singular vaue decomposition.
AMS: 92D10, 92C55, 65F15

1. Introduction. Microarray technology gives information about the expression levels of thousands of genes simultaneously. When microarray data from a number of samples is collected, the natural data structure is a bi-partite graph with non-negatively weighted edges. An important problem is then to partition, or cluster, the graph in an attempt to produce sets of genes and sets of samples such that each set of genes behaves uniformly across each set of samples. Here, we give analytical and experimental support for the use of the singular value decomposition (SVD).

In the next section we motivate the problem. In section 3 we start with a discrete optimization problem and proceed by adding constraints and relaxing to the continuous setting. By breaking the derivation into transparent steps, we are able to gain insights into the potential performance of the algorithm. This analysis generalizes the work in [9] to the bi-partite graph case. In particular, we show that there is a natural, justifiable, way to pre-process the edge weights to account for differently callibrated genes or samples. In section 4 we give an alternative derivation. Here, the solution is expressed as the limit of an iterative procedure that updates the clustering values based on an easily interpreted rule that measures the relative connectedness of the data points.

In section 5 we apply the algorithm to some small scale, artificial test data in order to get a feel for the performance and visualize the output. In section 6 we then apply the algorithm to a microarray data set of acute leukemia published in [3] and [7] (http://www.broad.mit.edu/cgi-bin/cancer/datasets.cgi) and interpret the results biologically.

2. Spectral Bi-Clustering. We are concerned with the case where a number of microarray samples have been generated for a common set of genes [1], [2], [7]. Typically, samples correspond to different pieces of tissue. For each gene in each sample, we suppose that a non-negative weight has been recorded to quantify the activity of the gene in that sample. If there are M genes and N samples, then

*Department of Mathematics, University of Strathclyde, Glasgow G1 1XH, UK. Supported by a Research Fellowship from the Royal Society of Edinburgh/Scottish Executive Education and Lifelong Learning Department and by EPSRC grant GR/S62383/01.
†Department of Mathematics, University of Strathclyde, Glasgow G1 1XH, UK.
‡Beatson Institute for Cancer Research, Glasgow G61 1BD, U.K.

a rectangular matrix $W \in \mathbb{R}^{M\times N}$ stores the weights, with $w_{ij} \geq 0$ representing the activity of the ith gene in the jth sample. This data fits into the general framework of a *non-negatively weighted bi-partite graph*—nodes can be split into two groups (genes and samples) such that a weighted edge exists only for pairs of nodes in distinct groups.

The matrix W is normally long and thin; values such as $M \approx 30,000$ genes and $N \approx 100$ samples are typical. In trying to find easily-summarized structure in the dataset, a reasonable approach is to bi-cluster simultaneously the genes and samples; that is,

> to partition the genes into two distinct groups, A and B, and similarly, partition the samples into two distinct groups, $\widehat{A}$ and $\widehat{B}$, where genes in group A tend to be active in samples $\widehat{A}$ and inactive in samples $\widehat{B}$, and similarly, genes in group B tend to be active in samples $\widehat{B}$ and inactive in samples $\widehat{A}$.

The biological motivation behind a search for this pattern is that genes involved in a common function should be active in a common set samples. Revealing this structure provides information about sets of genes that take part in a common process and about samples in which such a process takes place.

This bi-clustering goal has been considered in [11], where justification is given via existing microarray datasets. Kluger et al. propose the singular value decomposition (SVD) as a means to bi-cluster, and our main aim here is to provide further theoretical support and algorithmic insight into the use of the SVD in this respect.

We will focus throughout on the microarray application and refer to 'genes' and 'samples', but we emphasize that the analysis here is quite general and applies to any bi-partite graph with weighted edges; in particular, very similar problems arise in related areas of bioinformatics and in other data mining applications [4], [8], [12], [14].

3. Optimization Viewpoint. Let $p_i \in \{-\frac{1}{2}, \frac{1}{2}\}$ be an indicator vector component that determines whether gene i is placed in group A or B. Similarly, let $q_j \in \{-\frac{1}{2}, \frac{1}{2}\}$ be an indicator vector component that determines whether sample j is placed in group $\widehat{A}$ or $\widehat{B}$. Consider first the problem

$$\min \sum_{i=1}^{M}\sum_{j=1}^{N}(p_i - q_j)^2 w_{ij}. \tag{3.1}$$

Here, we seek to minimize the sum of the weights w_{ij} that relate non-matching genes and samples. To avoid the trivial solution where all genes/samples are put into a single group (with the other group remaining empty), we add the constraints

$$\sum_{i=1}^{M} p_i d_{\text{gene}_i} \approx 0 \quad \text{and} \quad \sum_{j=1}^{N} q_j d_{\text{sample}_j} \approx 0, \tag{3.2}$$

where $d_{\text{gene}_i} := \sum_{k=1}^{N} w_{ik}$ is the total expression weight for gene i and $d_{\text{sample}_j} := \sum_{k=1}^{M} w_{kj}$ is the total expression weight for sample j. The constraints (3.2) make sure that overall expression levels are roughly balanced across the two groups of genes and across the two groups of samples.

To reduce this discrete optimization task to a tractable problem, we look for a solution with $p \in \mathbb{R}^M$ and $p \in \mathbb{R}^N$. The idea is that the real-valued solution vectors

p and q will have components that fall into distinct groups, and hence will reveal obvious partitions. This relaxation idea has been successfully used in a number of data mining applications [4]. After relaxing, to avoid the trivial solution $p \equiv 0$ and $q \equiv 0$ we impose the normalization constraints

$$\sum_{i=1}^{M} p_i^2 d_{\text{gene}_i} = 1 \quad \text{and} \quad \sum_{j=1}^{N} q_j^2 d_{\text{sample}_j} = 1. \tag{3.3}$$

Constraints (3.3) damp down the influence of 'promiscuous' genes and samples, that is, genes and samples with large overall connectivity—a large d_{gene_i} or d_{sample_j} value encourages a p_i or q_j value close to zero. Having relaxed to real valued indicator vectors, we may strengthen to exact equality in (3.2), giving

$$p^T d_{\text{gene}} = q^T d_{\text{sample}} = 0. \tag{3.4}$$

Letting $D_{\text{gene}} = \text{diag}(d_{\text{gene}}) \in \mathbb{R}^{M\times M}$ and $D_{\text{sample}} = \text{diag}(d_{\text{sample}}) \in \mathbb{R}^{N\times N}$, the two-sum $\sum_{i=1}^{M}\sum_{j=1}^{N}(p_i - q_j)^2 w_{ij}$ expands to $p^T D_{\text{gene}} p + q^T D_{\text{sample}} q - 2p^T W q$ and hence the problem (3.1) under constraints (3.3) and (3.4) may be written

$$\begin{aligned} \max\{p^T W q \ :\ & p \in \mathbb{R}^M,\ q \in \mathbb{R}^N,\ p^T d_{\text{gene}} = q^T d_{\text{sample}} = 0, \\ & \|D_{\text{gene}}^{\frac{1}{2}} p\|_2 = \|D_{\text{sample}}^{\frac{1}{2}} q\|_2 = 1\}. \end{aligned} \tag{3.5}$$

Theorem 3.4 below solves this problem with the aid of Lemmas 3.1–3.3. First, we recall that any matrix $A \in \mathbb{R}^{M\times N}$ has SVD of the form $A = U\Sigma V^T$, where $U \in \mathbb{R}^{M\times M}$ and $V \in \mathbb{R}^{N\times N}$ are orthogonal, and $\Sigma \in \mathbb{R}^{M\times N}$ is diagonal with entries $\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \cdots \geq 0$ on the diagonal [6]. The columns $u^{(1)}, u^{(2)}, u^{(3)}, \ldots$ of U and the columns $v^{(1)}, v^{(2)}, v^{(3)}, \ldots$ of V are known as the left and right singular vectors of A. The scalars $\sigma_1, \sigma_2, \ldots$ are the corresponding singular values.

LEMMA 3.1. *Let $A \in \mathbb{R}^{M\times N}$ have SVD given by $A = U\Sigma V^T$. Then the problem*

$$\max\{x^T A y \quad :\quad x \in \mathbb{R}^M,\ y \in \mathbb{R}^N,\ x^T u^{(1)} = y^T v^{(1)} = 0,\ \|x\|_2 = \|y\|_2 = 1\} \tag{3.6}$$

is solved by $x = u^{(2)}$ and $y = v^{(2)}$.

Proof. Let $x = Ur$ and $y = Vs$. Then the problem (3.6) becomes

$$\max\{r^T \Sigma s \quad :\quad r \in \mathbb{R}^M, s \in \mathbb{R}^N,\ r_1 = s_1 = 0,\ \|r\|_2 = \|s\|_2 = 1\},$$

for which a solution is clearly given by setting the second components of r and s to 1 and all other components to 0. □

In Lemmas 3.2 and 3.3, we use $\mathbf{1}$ to denote a vector with all entries equal to one, of whatever dimension is appropriate. We also use $\sqrt{x}$ for a vector x to mean the vector whose ith component is $\sqrt{x_i}$. We assume throughout that D_{gene} and D_{sample} are nonsingular.

LEMMA 3.2. *The vectors $u = \sqrt{d_{\text{gene}}}/\|\sqrt{d_{\text{gene}}}\|_2$ and $v = \sqrt{d_{\text{sample}}}/\|\sqrt{d_{\text{sample}}}\|_2$ are left and right singular vectors of $D_{\text{gene}}^{-\frac{1}{2}} W D_{\text{sample}}^{-\frac{1}{2}}$ with corresponding singular value equal to 1.*

Proof. Given a matrix $A \in \mathbb{R}^{M\times N}$, if $u \in \mathbb{R}^M$ and $v \in \mathbb{R}^N$ satisfy $Av = \sigma u$ and $A^T u = \sigma v$ for some $\sigma > 0$, then $u/\|u\|_2$ and $v/\|v\|_2$ are left and right singular vectors of A with singular value σ. Since

$$D_{\text{gene}}^{-\frac{1}{2}} W D_{\text{sample}}^{-\frac{1}{2}} \sqrt{d_{\text{sample}}} = D_{\text{gene}}^{-\frac{1}{2}} W \mathbf{1} = D_{\text{gene}}^{-\frac{1}{2}} d_{\text{gene}} = \sqrt{d_{\text{gene}}},$$

and

$$D_{\text{sample}}^{-\frac{1}{2}} W^T D_{\text{gene}}^{-\frac{1}{2}} \sqrt{d_{\text{gene}}} = D_{\text{sample}}^{-\frac{1}{2}} W^T \mathbf{1} = D_{\text{sample}}^{-\frac{1}{2}} d_{\text{sample}} = \sqrt{d_{\text{sample}}},$$

the result holds. ∎

LEMMA 3.3. *We have* $\|D_{\text{gene}}^{-\frac{1}{2}} W D_{\text{sample}}^{-\frac{1}{2}}\|_2 = 1$.

Proof. The symmetric matrix

$$C = \begin{bmatrix} 0 & W \\ W^T & 0 \end{bmatrix}$$

satisfies $\|C\|_2 = \|A\|_2$. Since $W \to D_{\text{gene}}^{-\frac{1}{2}} W D_{\text{sample}}^{-\frac{1}{2}}$ corresponds to

$$C \to \widetilde{C} = D^{-\frac{1}{2}} C D^{-\frac{1}{2}} \equiv \begin{bmatrix} D_{\text{gene}}^{-\frac{1}{2}} & 0 \\ 0 & D_{\text{sample}}^{-\frac{1}{2}} \end{bmatrix} \begin{bmatrix} 0 & W \\ W^T & 0 \end{bmatrix} \begin{bmatrix} D_{\text{gene}}^{-\frac{1}{2}} & 0 \\ 0 & D_{\text{sample}-\frac{1}{2}} \end{bmatrix}$$

and $D = \text{diag}(C\mathbf{1})$, the problem is now reduced to the symmetric case.

Letting $\rho(\widetilde{C})$ denote the spectral radius of $\widetilde{C}$, we have

$$\rho(\widetilde{C}) = \rho(D^{-\frac{1}{2}} C D^{-\frac{1}{2}}) = \rho(D^{-1} C) \leq \|D^{-1} C\|_\infty = 1.$$

But

$$\widetilde{C} \cdot D^{\frac{1}{2}} \mathbf{1} = D^{-\frac{1}{2}} C D^{-\frac{1}{2}} \cdot D^{\frac{1}{2}} \mathbf{1} = D^{-\frac{1}{2}} C \mathbf{1} = D^{-\frac{1}{2}} D \mathbf{1} = D^{\frac{1}{2}} \mathbf{1},$$

so $\widetilde{C}$ has an eigenvalue 1. Hence, $\rho(\widetilde{C}) = 1$. Since $\widetilde{C}$ is symmetric, we have $\rho(\widetilde{C}) = \|\widetilde{C}\|_2$, completing the proof. ∎

THEOREM 3.4. *The problem (3.5) is solved by taking* $p = D_{\text{gene}}^{-\frac{1}{2}} u^{(2)}$ *and* $q = D_{\text{sample}}^{-\frac{1}{2}} v^{(2)}$, *where* $u^{(2)}$ *and* $v^{(2)}$ *are second left and right singular vectors of the matrix* $D_{\text{gene}}^{-\frac{1}{2}} W D_{\text{sample}}^{-\frac{1}{2}}$.

Proof. Letting $p = D_{\text{gene}}^{-\frac{1}{2}} x$ and $q = D_{\text{sample}}^{-\frac{1}{2}} y$, the problem (3.5) may be written

$$\max\{x^T D_{\text{gene}}^{-\frac{1}{2}} W D_{\text{sample}}^{-\frac{1}{2}} y \ : \ x \in \mathbb{R}^M,\ y \in \mathbb{R}^N,\ x^T \frac{\sqrt{d_{\text{gene}}}}{\|\sqrt{d_{\text{gene}}}\|_2} = y^T \frac{\sqrt{d_{\text{sample}}}}{\|\sqrt{d_{\text{sample}}}\|_2} = 0,\ \|x\|_2 = \|y\|_2 = 1\}.$$

Using Lemmas 3.2 and 3.3, we see that this is of the form (3.6) with $A = D_{\text{gene}}^{-\frac{1}{2}} W D_{\text{sample}}^{-\frac{1}{2}}$. Hence, $x = u^{(2)}$ and $y = v^{(2)}$ solve the problem, which corresponds to $p = D_{\text{gene}}^{-\frac{1}{2}} u^{(2)}$ and $q = D_{\text{sample}}^{-\frac{1}{2}} v^{(2)}$. ∎

4. Power Method Interpretation. In this section we show that the spectral algorithm may be interpreted from an iterative viewpoint. This provides a simple and informative explanation of the algorithm, and also opens up the possibility of modifications that incorporate problem-specific information.

We will consider the use of the singular vector $v^{(2)}$ to categorize samples. An entirely analogous explanation applies to the use of $u^{(2)}$ to categorize genes.

We recall that the power method applied to a general square matrix $B \in \mathbb{R}^{N \times N}$ takes the form [6]

(1) choose $y^{[0]} \in \mathbb{R}^N$, set $k = 0$,
(2) let $y^{[k+1]} = By^{[k]}/\| By^{[k]} \|$,
(3) repeat step (2) until some convergence criterion is satisfied.

Our aim is to interpret such an algorithm as an attempt to assign a location on the real line to each sample, so $q_s^{[k]}$ represents the location of sample number s at iteration number k. The locations are updated in a way that takes account of the "connectedness" of samples, and the aim is that the final result, $q^{[\infty]}$, positions each sample close to samples that are "strongly connected" to it and far from samples that are only "weakly connected" to it. Further, we will assume that only the overall ordering of the samples is important—as in the matrix reordering examples of section 5—so we will simplify by replacing step (2) with $y^{[k+1]} = By^{[k]}$. (In practice, of course, normalization is necessary to avoid underflow and overflow, but our aim here is to develop a new interpretation of the algorithm rather than a practical implementation.)

Making the assumption $\sigma_2 > \sigma_3$, we begin by noting that $v^{(2)}$ is the subdominant eigenvector of the matrix $\left(D_{\text{gene}}^{-\frac{1}{2}} W D_{\text{sample}}^{-\frac{1}{2}}\right)^T \left(D_{\text{gene}}^{-\frac{1}{2}} W D_{\text{sample}}^{-\frac{1}{2}}\right)$; that is, $D_{\text{sample}}^{-\frac{1}{2}} W D_{\text{gene}}^{-1} W D_{\text{sample}}^{-\frac{1}{2}}$. By Lemma 3.2, this matrix has dominant eigenvector $\sqrt{d_{\text{sample}}}$, corresponding to eigenvalue 1. Normalizing to get Euclidean norm of unity, we obtain

$$\frac{\sqrt{d_{\text{sample}}}}{\sqrt{W_{\text{sum}}}}, \qquad \text{where } W_{\text{sum}} := \sum_{i=1}^{M}\sum_{j=1}^{N} w_{ij},$$

as the dominant eigenvector. Hence, the deflated matrix

$$D_{\text{sample}}^{-\frac{1}{2}} W D_{\text{gene}}^{-1} W D_{\text{sample}}^{-\frac{1}{2}} - \frac{\sqrt{d_{\text{sample}}}\sqrt{d_{\text{sample}}}^T}{W_{\text{sum}}},$$

has dominant eigenvector $v^{(2)}$. It follows that $v^{(2)}$ may be computed by applying the power method to this matrix. After some manipulation, the iteration for $q^{[k]} := D_{\text{sample}}^{-\frac{1}{2}} y^{[k]}$ may be written in the form

$$q_r^{[k+1]} = \sum_{s=1}^{N} \left(\alpha_{r,s}^{[k]} - \beta_s\right) q_s^{[k]}, \tag{4.1}$$

where

$$\alpha_{r,s}^{[k]} := \frac{1}{(d_{\text{sample}})_r} \sum_{t=1}^{M} \frac{w_{tr} w_{ts}}{(d_{\text{gene}})_t} \qquad \text{and} \qquad \beta_s := \frac{(d_{\text{sample}})_s}{W_{\text{sum}}}. \tag{4.2}$$

Several remarks are in order.

1 The iteration (4.1) computes a new location $q_r^{[k+1]}$ for the rth sample by forming a weighted sum of all the current locations, $q_s^{[k]}$.

2 The quantity $\alpha_{r,s}^{[k]}$ in (4.2) may be described as the *actual connectivity* for sample s from sample r. It is large if samples r and s have a big percentage of their weights involved in common genes, and small otherwise. Fixing r, we compute $\alpha_{r,s}^{[k]}$ for the sth sample by running over all genes, $t = 1, 2, \ldots, M$, and summing $w_{tr}w_{ts}/(d_{\text{gene}})_t$. The quantity $w_{tr}w_{ts}/(d_{\text{gene}})_t$ measures whether samples r and s are both strongly connected to gene t, relative to the overall promiscuity of gene t. Finally, this sum is divided by $(d_{\text{sample}})_r$ so that the overall promiscuity of sample r is taken into account.

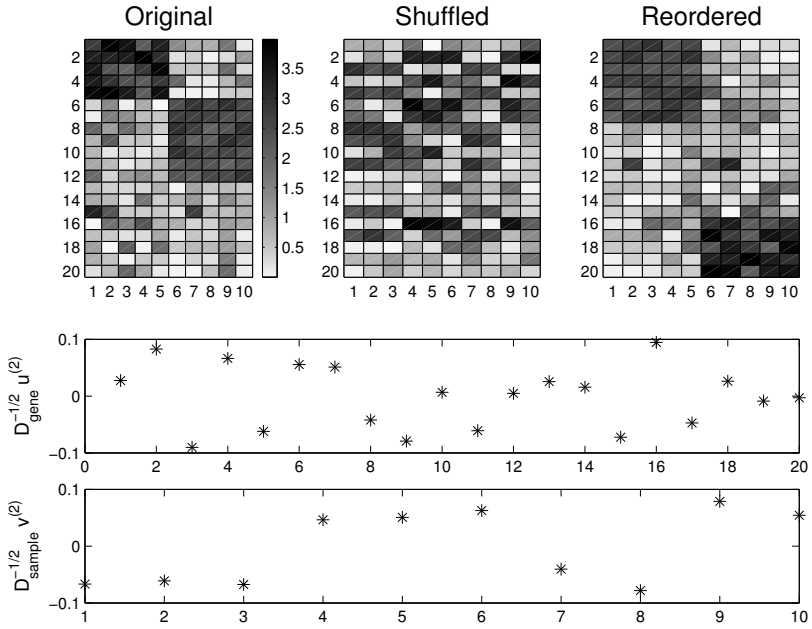

FIG. 5.1. *Upper left: data matrix. Upper middle: shuffled. Upper right: reordered by spectral method. Lower pictures show components of the scaled second singular vectors* $D_{\text{gene}}^{-\frac{1}{2}}u^{(2)}$ *and* $D_{\text{sample}}^{-\frac{1}{2}}v^{(2)}$ *of* $D_{\text{gene}}^{-\frac{1}{2}}WD_{\text{sample}}^{-\frac{1}{2}}$.

3 The quantity β_s in (4.2), which is independent of k, may be described as the *typical connectivity* for sample s. It measures the weight associated with sample s relative to the total amount of weight present.

4 Combining points 2 and 3, we see that in the iteration (4.1), the influence of sample s on the new location of sample r depends on the difference between their actual and typical connectivities. If $\alpha_{r,s}^{[k]} > \beta_s$ then sample r has a strong connection with sample s and hence a positive weight is used in (4.1), which tends to bring the locations of samples r and s closer together at the next iteration. On the other hand if $\alpha_{r,s}^{[k]} < \beta_s$ then sample r has a weak connection with sample s and hence a negative weight is used in (4.1), which tends to force the locations of samples r and s further apart at the next iteration.

Overall, this interpretation, which does not make use of advanced linear algebra concepts such as the SVD, shows that the spectral clustering approach may be viewed as a natural iteration that attempts to shuffle the location of the samples based on their relative connectedness. In addition to giving insight about the nature of the algorithm, this iterative viewpoint offers potential for customized versions, where existing knowledge about the existence of clusters could be built in at each iteration. This idea will be pursued in further work.

5. Experiments. Figure 5.1 illustrates the spectral algorithm. The top left picture shows a matrix $W \in \mathbb{R}^{20\times 10}$ with entries computed from pseudorandom number

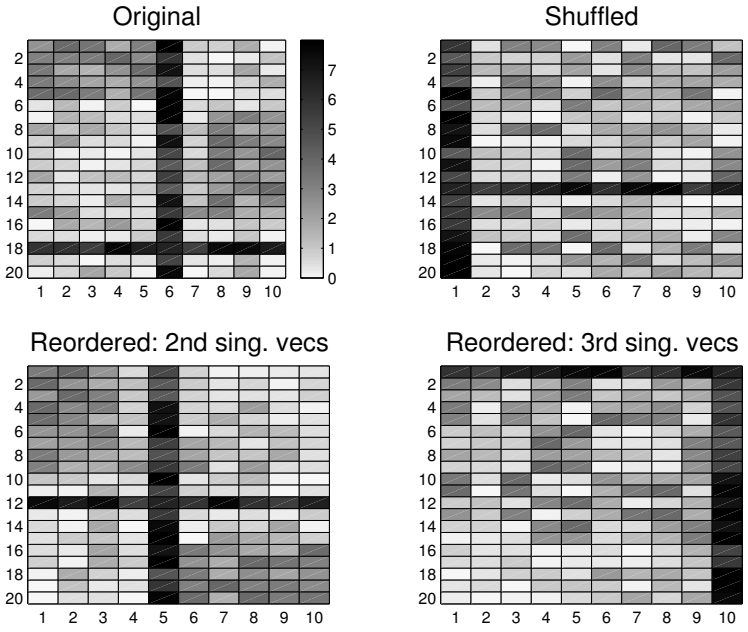

FIG. 5.2. *Reordering with the specral algorithm. Upper left: data matrix. Upper right: shuffled. Lower left: reordered by scaled second singular vectors* $D_{\text{gene}}^{-\frac{1}{2}}u^{(2)}$ *and* $D_{\text{sample}}^{-\frac{1}{2}}v^{(2)}$ *of* $D_{\text{gene}}^{-\frac{1}{2}}WD_{\text{sample}}^{-\frac{1}{2}}$. *Lower right: reordered by third scaled singular vectors* $D_{\text{gene}}^{-\frac{1}{2}}u^{(3)}$ *and* $D_{\text{sample}}^{-\frac{1}{2}}v^{(3)}$ *of* $D_{\text{gene}}^{-\frac{1}{2}}WD_{\text{sample}}^{-\frac{1}{2}}$.

generators according to

$$w_{ij} = \begin{cases} 2 + 2\texttt{rand}, & \text{for } 1 \le i \le 5,\ 1 \le j \le 5, \\ 2 + \texttt{rand}, & \text{for } 6 \le i \le 12,\ 6 \le j \le 10, \\ |\texttt{randn}|, & \text{otherwise.} \end{cases}$$

Here we follow MATLAB notation [10] so `rand` and `randn` denote calls to Uniform $\in (0,1)$ and standard Normal generators, respectively. In summary, W has two blocks of relatively large entries that are clearly visible as darker pixels in the picture. For the purpose of visualization, the upper middle picture shows the same matrix with an arbitrary row and column shuffling. This is the matrix to which the algorithm is applied. (In practice, the algorithm is insensitive to permutations, but the middle picture represents the 'unstructured' format that would be generated in practice.) The middle and lower pictures in Figure 5.1 show the entries in the scaled second singular vectors of $D_{\text{gene}}^{-\frac{1}{2}}WD_{\text{sample}}^{-\frac{1}{2}}$, that is $D_{\text{gene}}^{-\frac{1}{2}}u^{(2)}$ and $D_{\text{sample}}^{-\frac{1}{2}}v^{(2)}$, where W denotes the shuffled matrix. We see that the entries of $D_{\text{gene}}^{-\frac{1}{2}}u^{(2)}$ fall into three bands and the entries of $D_{\text{sample}}^{-\frac{1}{2}}v^{(2)}$ fall into two bands. The component indices for the upper and lower bands in $D_{\text{gene}}^{-\frac{1}{2}}u^{(2)}$ match the row indices for the two large blocks in the matrix. Similarly, the component indices for the two bands in $D_{\text{sample}}^{-\frac{1}{2}}v^{(2)}$ match the column indices for the two large blocks. To see this, the top right picture shows the matrix with rows and columns reordered according to the ordering of components in $D_{\text{gene}}^{-\frac{1}{2}}u^{(2)}$

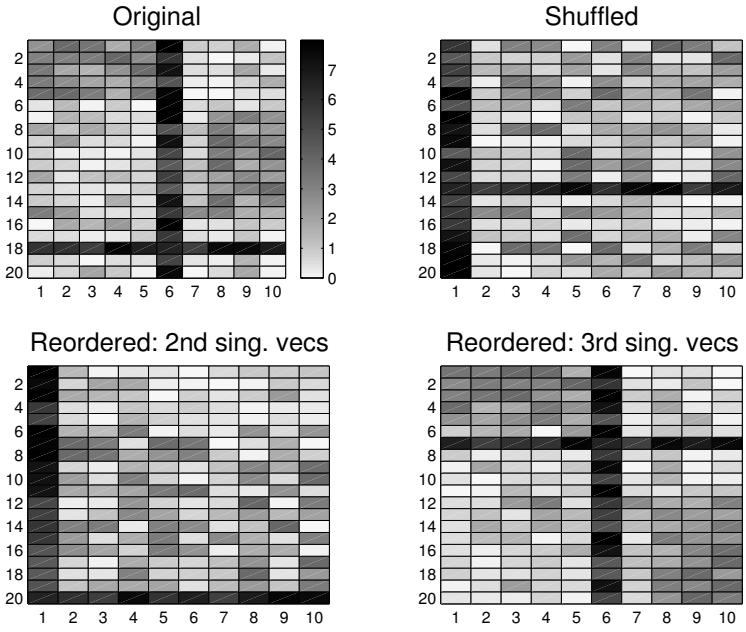

FIG. 5.3. *As for Figure 5.2 except that W is used instead of $D_{\text{gene}}^{-\frac{1}{2}} W D_{\text{sample}}^{-\frac{1}{2}}$. Upper left: data matrix. Upper right: shuffled. Lower left: reordered by second singular vectors $u^{(2)}$ and $v^{(2)}$ of W. Lower right: reordered by third singular vectors $u^{(3)}$ and $v^{(3)}$ of W.*

and $D_{\text{sample}}^{-\frac{1}{2}} v^{(2)}$. We see that the algorithm is able to recover the two blocks of large entries.

In Figure 5.2, we repeat this exercise for a matrix $W \in \mathbb{R}^{20\times 10}$ with entries

$$w_{ij} = \begin{cases} 1.5 + 2.5\texttt{rand}, & \text{for } 1 \le i \le 5,\ 1 \le j \le 5, \\ 1.5 + 2.5\texttt{rand}, & \text{for } 7 \le i \le 15,\ 8 \le j \le 10, \\ 4 + 4\texttt{rand}, & \text{for } i = 18,\ 1 \le j \le 10, \\ 4 + 4\texttt{rand}, & \text{for } 1 \le i \le 20,\ j = 6, \\ |\texttt{randn}|, & \text{otherwise.} \end{cases}$$

Overall this matrix has two blocks of slightly larger than average entries, but is dominated by a large row ($i = 18$) and a large column ($j = 6$). This represents an unusually overexpressed gene and an unusually overexpressed sample. The top left picture shows the original matrix and the top right picture shows a row/column shuffled version, to which the algorithm was applied. The bottom left picture shows the reordered matrix, as described for Figure 5.1, and we see that the algorithm has successully located the two blocks, while placing the promiscuous row and column in the middle of the ordering. This agrees with the remarks made in sections 3 and 4. The lower right picture in Figure 5.2 shows the reordering produced by the scaled third singular vectors of $D_{\text{gene}}^{-\frac{1}{2}} W D_{\text{sample}}^{-\frac{1}{2}}$, that is, $D_{\text{gene}}^{-\frac{1}{2}} u^{(3)}$ and $D_{\text{sample}}^{-\frac{1}{2}} v^{(3)}$. Here, the reordering is essentially distinguishing the single large row/column from the remainder. This makes sense—the second singular vectors have picked out the two significant blocks, so the third singular vectors pick out the only remaining structure in the data, namely the 'outlying' gene/sample pair. (For further discussion about the role of higher eigenvectors in the $M = N$ case, see [9].)

In Figure 5.3 we perform the same experiment using the second and third singular vectors from the SVD of W rather than $D_{\text{gene}}^{-\frac{1}{2}} W D_{\text{sample}}^{-\frac{1}{2}}$. Here, we see that reordering with the second singular values $u^{(2)}$ and $v^{(2)}$ serves only to isolate the outliers. Because there is no normalization across genes/samples, the algorithm is heavily influenced by the promiscuous values. However, the third singular vectors do pick out the remaining structure, namely the existence of the two significant blocks.

Overall, the tests here confirm that the SVD-based algorithm can successfully reveal the existence of clusters, and also support the earlier arguments that the normalization $D_{\text{gene}}^{-\frac{1}{2}} W D_{\text{sample}}^{-\frac{1}{2}}$ will tend to mitigate the influence of promiscuous genes/samples.

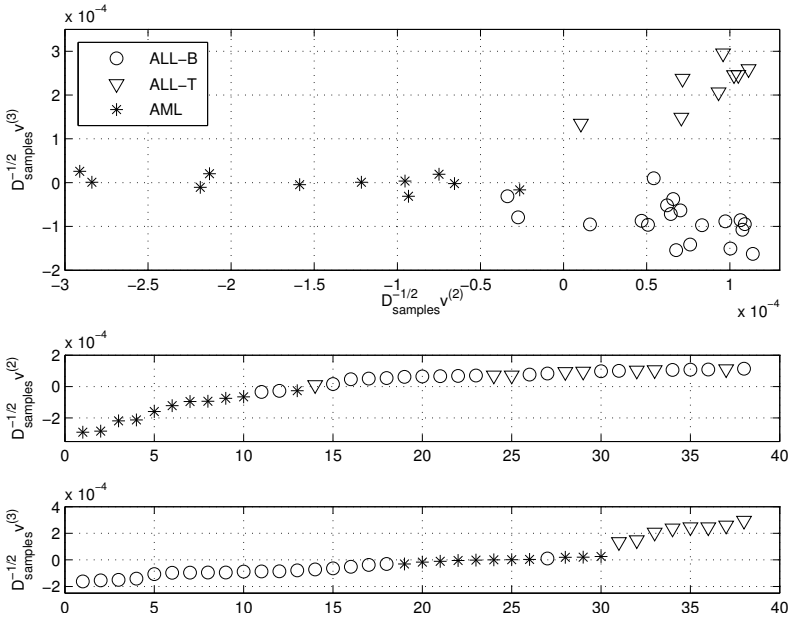

FIG. 6.1. *Leukemia: Upper: Scatter plot of the samples Lower pictures show samples ordered by the second and third singular vectors.*

6. Microarray Data. To ilustrate the performance of the spectral algorithm proposed above we use acute leukemia data as published in [3]. The data set of bone marrow samples from 38 patients contains the expression intensities for 5000 genes. Twenty seven cases were diagnosed as acute lymphoblastic leukemia (ALL) and the other eleven as acute myeloid leukemia (AML). The distinction between ALL and AML, as well as the division of ALL into T and B cell subtypes, is well known. Several methods have been used to rediscover these differences [3], [5], [11], [13] and the leukemia data set has become a benchmark in the cancer classification community.

The two lower pictures in Figure 6.1 show the reordering produced by the scaled second and third singular vectors $D_{\text{sample}}^{-\frac{1}{2}} v^{(2)}$ and $D_{\text{sample}}^{-\frac{1}{2}} v^{(3)}$. We see that the second singular vector has correctly found the essential AML-ALL distinction. Two ALL samples are misclassified into the AML group. This happens for most methods and appears to reflect an inconsistency in the data. The third singular vector picks out the next important structure in the data—distinction between ALL-B and ALL-T. The

upper part of Figure 6.1 shows a plot of the second versus the third scaled singular vectors. Leukemias are here clustered into the three main biological classes.

7. Conclusion. We have given theoretical support for the use of the SVD by deriving a spectral clustering method from two alternative viewpoints. Both derivations give insights into the performance of the method and suggest that it will be insensitive to the presence of outliers. We focused on the application of co-clustering microarray data sets and showed that there is a natural way to normalize expression levels across genes and samples. Results on a publicly available leukemia data set demonstrated how the algorithm can be used to characterize different types of cancer.

Acknowledgement. We thank Nick Higham for suggesting the short proof of Lemma 3.3. This work was funded by EPSRC grant GR/S62383/01. DJH is supported by a Research Fellowship from the Royal Society of Edinburgh/Scottish Executive Education and Lifelong Learning Department.

REFERENCES

[1] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack and A. J. Levine, *Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays*, PNAS, 96 (1999), pp. 67456750.

[2] A. Ben-Dor, L. Bruhn, N. Friedman, I. Nachman and U. Washington *Tissue classification with gene expression profiles*, RECOMB Tokyo Japan, 2000.

[3] J. P. Brunet, P. Tamayo, T. R. Golub and J. P. Mesirov, *Metagenes and molecular pattern discovery using matrix factorization*, PNAS, 101 (2004), pp. 4164–4169.

[4] I. S. Dhillon, *Co-clustering documents and words using bipartite spectral graph partitioning*, Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining(KDD), August 26-29, 2001, San Francisco, California, USA

[5] G. Getz, E. Levine and E. Domany, *Coupled two-way clustering analysis of gene microarray data*, PNAS, 97 (2000), pp. 12079–12084.

[6] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Third ed., The Johns Hopkins University Press, Baltimore, MD, 1996.

[7] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caliguri, C. D. Bloomfield and E. S. Lander, *Molecular classification of cancer: class discovery and class prediction by gene expression monitoring*, Science, 286 (1999), pp. 531–537.

[8] P. Grindrod and M. Kibble, *Review of uses of network and graph theory concepts within proteomics*, Expert Rev. 1 (2004), pp. 229–238

[9] D. J. Higham and M. Kibble, *A unified view of spectral clustering*, University of Strathclyde Mathematics Research Report 02 (2004).

[10] D. J. Higham and N. J. Higham, *MATLAB Guide*, SIAM, 2000.

[11] Y. Kluger, R. Basri, J. T. Chang and M. Gerstein, *Spectral biclustering of microarray data: coclustering genes and conditions*, Genome Research, 13 (2003), pp. 703–716.

[12] S. Rogers, M. Girolami, C. Campbell and R. Breitling, *The latent process decomposition of cDNA microarray datasets*, IEEE/ACM Transactions on Computational Biology and Bioinformatics, to appear.

[13] J. Wang, T. H. Bo, I. Jonassen,O. Myklebost and E. Hovig, *Tumor classification and marker gene prediction by feature selection and fuzzy c-means clustering using microarray data.*, BMC Bioinformatics, 4 (2003), 60.

[14] E. P. Xing and R. M. Karp, *CLIFF: Clustering of High-Dimensional Microarray Data via Iterative Feature Filtering Using Normalized Cuts*, Bioinformatics, 17 (2001), pp. S306–S315.