

Sleeved CoClustering *

Avraham A. Melkman [†]
Department of Computer Science
Ben Gurion University
Beer Sheva, Israel 84105
melkman@cs.bgu.ac.il

Eran Shaham
Department of Computer Science
Ben Gurion University
Beer Sheva, Israel 84105
shahamer@cs.bgu.ac.il

ABSTRACT

A coCluster of a $m \times n$ matrix X is a submatrix determined by a subset of the rows and a subset of the columns. The problem of finding coClusters with specific properties is of interest, in particular, in the analysis of microarray experiments. In that case the entries of the matrix X are the expression levels of m genes in each of n tissue samples. One goal of the analysis is to extract a subset of the samples and a subset of the genes, such that the expression levels of the chosen genes behave similarly across the subset of the samples, presumably reflecting an underlying regulatory mechanism governing the expression level of the genes.

We propose to base the similarity of the genes in a coCluster on a simple biological model, in which the strength of the regulatory mechanism in sample j is H_j , and the response strength of gene i to the regulatory mechanism is G_i . In other words, every two genes participating in a good coCluster should have expression values in each of the participating samples, whose ratio is a constant depending only on the two genes. Noise in the expression levels of genes is taken into account by allowing a deviation from the model, measured by a *relative error criterion*. The sleeve-width of the coCluster reflects the extent to which entry i, j in the coCluster is allowed to deviate, relatively, from being expressed as the product $G_i H_j$.

We present a polynomial-time Monte-Carlo algorithm which outputs a list of coClusters whose sleeve-widths do not exceed a prespecified value. Moreover, we prove that the list includes, with fixed probability, a coCluster which is near-optimal in its dimensions. Extensive experimentation with synthetic data shows that the algorithm performs well.

Categories and Subject Descriptors: H.2.8 Database Management: Database Applications - Data Mining; I.5.3

*supported in part by the The Lynn and William Frenkel Center for Computer Sciences, and The Paul Ivanier Center for Robotics Research and Production Management

[†]Corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

KDD'04, August 22–25, 2004, Seattle, Washington, USA.
Copyright 2004 ACM 1-58113-888-1/04/0008 ...\$5.00.

Computing Methodologies: Pattern Recognition - Clustering.

General Terms: Algorithms.

Keywords: clustering, coclustering, gene expression data, co-regulation.

1. DESCRIPTION OF THE PROBLEM

A coCluster of a $m \times n$ matrix X is a submatrix determined by a subset I of the rows and a subset J of the columns. Recently many researchers have addressed the problem of finding coClusters with specific properties. One such direction of research originated in mining high dimensional data, in particular projected clustering, [1], [2], [3], [13], [16]. Another direction resulted from the analysis of microarray experiments. In that case the entries of the matrix X are the expression levels of m genes in each of n tissue samples. One goal of the analysis is to extract a subset of the samples and a subset of the genes, such that the expression levels of the chosen genes behave similarly across the subset of the samples, presumably reflecting an underlying co-regulation, [4], [5], [6], [7], [10], [11], [12], [14], [15], [16], [17]. For concreteness, we will in this introductory section describe our approach using terminology from microarray analysis, although its applicability extends beyond this domain, for example to the analysis of term-document matrices, cf. [8].

Call a sub-matrix determined by a set of genes (rows) I and a set of samples (columns) J co-varying if for any two genes i_1 and i_2 in I the ratio of elements in any sample $j \in J$ is a constant independent of j (but possibly dependent on i_1 and i_2): $X_{i_1,j}/X_{i_2,j} = c_{i_1,i_2}$ for all $j \in J$. Equivalently, there is a latent variable, H , whose value for sample j is H_j , such that the expression level of gene i in sample j is proportional to H_j , with the constant of proportionality depending only on the gene: $X_{i,j} \approx G_i H_j$ for all $(i, j) \in (I, J)$.

Presumably, in such a co-varying submatrix, the variation in the expression levels of the genes can be attributed primarily to the influence of one (or more) regulators, represented by H , whereas G_i is indicative of the strength of the regulation of gene i . We want to emphasize that it is not assumed that H_j is proportional to the abundance of some transcription factor in sample j , even if it is the variation in the latter that is responsible, directly or indirectly, for the variation in the expression levels. Suppose, for example, that the functional dependency of the expression level of a gene on the abundance of a transcription factor, x , is of the form $G u(x)$, with G a constant specific to the gene and $u(x)$

a function common to the up-regulated genes. In the model described above, H_j is then $u(x_j)$ (rather than x_j), with x_j the level of the transcription factor in sample j . And the expression levels of two up-regulated genes 1 and 2 in sample j are G_1H_j, G_2H_j .

To allow for noise in the expression levels of genes, we will look for G_i and H_j such that $X_{i,j}$ is well approximated by G_iH_j . To measure how well $X_{i,j}$ is approximated by G_iH_j it seems natural to adopt the *relative error* criterion,

$$\frac{1}{\eta} \leq \frac{G_iH_j}{X_{i,j}} \leq \eta.$$

Taking logarithms and setting $A_{i,j} = \log X_{i,j}$, $R_i = \log G_i$, $C_j = \log H_j$, and $\varepsilon = \log \eta$, the problem becomes one of finding R_i, C_j such that for all i, j ,

$$-\varepsilon \leq (R_i + C_j) - A_{i,j} \leq \varepsilon.$$

Definition 1. The *sleeve-width* of a sub-matrix of A , defined by a subset I of rows and a subset J of columns, is

$$sw(I, J) = 2 \min_{R, C} \max_{i \in I, j \in J} |A_{i,j} - R_i - C_j|. \quad (1)$$

Definition 2. Let $0 < \sigma, \rho < 1$ be fixed parameters. For $w > 0$, a *coCluster* of A of *sleeve-width* w is a pair (I, J) , with I a subset of the rows and J a subset of the columns, that satisfies the following conditions.

Size: The number of rows is at least a ρ -fraction of all rows, $|I| \geq \rho m \geq 2$, and the number of columns is at least a σ -fraction of all columns, $|J| \geq \sigma n \geq 2$.

Sleeve-width: $sw(I, J) \leq w$, *i.e.*, there are $R_i, i \in I$, and $C_j, j \in J$, such that $|A_{i,j} - R_i - C_j| \leq w/2$ for all $i \in I, j \in J$. $R_i, i \in I$ will be called a column profile of the coCluster and $C_j, j \in J$ will be called a row profile.

Thus, for any gene i in a coCluster there is a shift R_i which places its row $A_{i,j}, j \in J$ of expression values within a sleeve of width w surrounding the row profile of the coCluster.

Remark: Clearly the profiles are not unique, *e.g.*, $R_i + C_j = (R_i + a) + (C_j - a)$ for all i, j and a fixed a . Thus an arbitrary R_i or C_j can be fixed at 0.

Summarizing, we propose to identify subsets of genes that are purportedly co-regulated in a subset of the samples, by log-transforming the expression-data matrix, and then finding in it coClusters of small sleeve-width, w . Observe that, in contrast to a singular value decomposition, the values G_i and H_j found by this method, are guaranteed to be non-negative.

Subsection 2.1 describes a Monte-Carlo algorithm for extracting a list of coClusters all guaranteed to be of sleeve-width w . Then it is proven in subsection 2.2 that the list contains, with fixed probability, a coCluster that is near-optimal in a sense described there.

Section 4 explains how to compute the sleeve-width of a given coCluster. The results of experiments investigating properties of the algorithm and its parameters, performed on synthetic data sets, are presented in section 3.

The approach developed here was stimulated by the work of Cheng and Church [6], who also sought submatrices of A whose entries are approximated well by $R_i + C_j$, but measured the discrepancy using the *mean squared residue score*; they called such matrices biclusters. Further work in this

direction includes that of Yang, Wang, Wang and Yu, [17], and of Cho, Dhillon, Guan and Sra [7]. However, the mean squared residue score has some undesirable features: the score of a bicluster can be smaller than the score of a sub-matrix of this bicluster, the score is insensitive to outliers, and the connection to a regulatory model is unclear. In addition, we expect that our approach will find smaller sets of genes with tighter co-regulation patterns.

In another direction, our results owe much to the work of Procopiu, Jones, Agarwal and Murali [13] on projective clustering, which also formed the basis of the work of Murali and Kasif [12] on gene expression motifs. These authors looked for coClusters in which all the expression values of a row of the coCluster have to fit within a sleeve around the *exact same* row profile, $\arg \min_C \max_{i \in I, j \in J} |A_{i,j} - C_j|$. We, on the other hand, permit shifting each row by an individual fixed amount, so as to place it within a sleeve around the row profile.

In a vein similar to ours, Wang, Wang, Yang, and Yu, [16] proposed an algorithm which finds coClusters with the property that any pair of rows in the coCluster have sleeve-width at most $w/2$. It follows from Theorem 2.1 that the cluster as a whole has then in fact sleeve-width w at most, in our terminology. The running time of their algorithm is, however, exponential.

2. FINDING SLEEVED COCLUSTERS

2.1 The algorithm

Algorithm RSLEEVE

- 1: **loop** ℓ_R times
- 2: choose a set of rows D of size d uniformly at random;
- 3: **loop** ℓ_S times
- 4: choose a column $s, 1 \leq s \leq n$, uniformly at random;
- 5: $J \leftarrow \emptyset$;
- 6: **for** $j \leftarrow 1$ to n **do**
- 7: **if** $sw(D, \{s, j\}) \leq w/2$ **then** add j to J ;
- end for**
- 8: choose a row $r \in D$ uniformly at random;
- 9: $I \leftarrow \emptyset$;
- 10: **for** $i \leftarrow 1$ to m **do**
- 11: **if** $sw(\{r, i\}, J) \leq w/2$ **then** add i to I ;
- end for**
- 12: **if** $|I| < \rho m$ or $|J| < \sigma n$ **then** discard (I, J)
- 13: **else** compute $sw(I, J)$;
- end loop**
- end loop**
- 14: **return** a list of the best (I, J) ;

Figure 1: Algorithm for finding sleeved coClusters

Figure 1 presents a Monte Carlo algorithm, RSLEEVE, which outputs a list of coClusters which are guaranteed to have sleeve-width not exceeding w , cf. Theorem 2.1. Moreover, it will be proven in the next subsection that the algorithm possesses the following near-optimality property: with fixed probability at least one of the coClusters on the list, (I, J) , contains a coCluster, (I^*, J) , of sleeve-width $w/2$ which is optimal in a sense that will be defined formally in the next subsection. Roughly speaking, (I^*, J) is optimal in the balance it strikes between the number of rows and the number of columns.

Implementation notes.

1. The computation of $sw(D, \{s, j\})$ needed in line 7 is easy:

$$sw(D, \{s, j\}) = \frac{1}{2}(\max_{i \in D}(A_{i,j} - A_{i,s}) - \min_{i \in D}(A_{i,j} - A_{i,s})),$$

cf. Theorem 4.1. A similar result holds for $sw(\{r, i\}, J)$ in line 11.

2. Although the coClusters returned by the algorithm are guaranteed to have sleeve-width w at most, they may well have smaller sleeve-width in actual fact. For this reason line 13 computes $sw(I, J)$; the algorithm for doing this is detailed in section 4.

We now prove that all coClusters returned by the algorithm do indeed have sleeve-width w .

THEOREM 2.1. *Let I be a set of row indices, and let $r \in I$. If $sw(\{r, i\}, J) \leq w/2$ for each $i \in I$, then $sw(I, J) \leq w$. Thus each coCluster (I, J) returned by RSLEEVE has sleeve-width at most w .*

PROOF. Consider the definition of sleeve-width, equation (1). Fix $i \in I$. Since $sw(\{r, i\}, J) \leq w/2$, there are R_i and R_r as well as $C_j^i, j \in J$, such that

$$2|A_{i,j} - R_i - C_j^i| \leq \frac{1}{2}w, \quad 2|A_{r,j} - R_r - C_j^i| \leq \frac{1}{2}w, \quad \forall j \in J.$$

Hence $|A_{i,j} - A_{r,j} - R_i + R_r| \leq w/2$ for all $j \in J$ and each fixed $i \in I$. In fact, according to the remark after definition 2, we can always set $R_r = 0$. It follows that

$$sw(I, J) \leq 2 \max_{i \in I, j \in J} |A_{i,j} - R_i - A_{r,j}| \leq w.$$

□

Remark: The bound $sw(I, J) \leq w$ cannot be improved, since it becomes tight as $\max\{m, n\} \rightarrow \infty$. Namely, for each n there exists a $n \times n$ matrix such that every pair of rows has sleeve-width $\frac{n}{n-1}$, whereas the matrix as a whole has sleeve-width 2.

2.2 Near-optimality of the algorithm

Algorithm RSLEEVE can be viewed as a heuristic, and the experimental evidence of section 3 shows it to be efficacious. This subsection proves that there are good theoretical reasons for its efficacy.

Suppose there are several coClusters of the same sleeve-width. Which one is to be preferred? coClusters could be ranked by perimeter, $|I| + |J|$, or area, $|I| \cdot |J|$. But in the context under consideration, the number of rows is several orders of magnitude larger than the number of columns. It makes sense, then, as Procopiuc, Jones, Agarwal and Murali [13] proposed, to specify a trade-off between the number of rows and the number of columns in the coCluster: the inclusion of an additional column in J is worth the exclusion of at most a $(1 - \beta)$ -fraction of the rows in I . We adopt their proposal and rank coClusters by their measure.

Definition 3. The *rating* of a coCluster (I, J) is $\mu(|I|, |J|)$, where $\mu(x, y) = x(1/\beta)^y$, for fixed β .

With this definition our problem can be rephrased as one of finding an optimal coCluster of sleeve-width w , one with the maximum rating.

The main result of this subsection is Theorem 2.3. Paraphrased it states that if the algorithm is run long enough, then the list returned by the algorithm contains with fixed

probability a coCluster (I, J) of sleeve-width w , whose rating is at least as good as that of an *optimal* coCluster of sleeve-width $w/2$.

The central insight of Procopiuc, Jones, Agarwal and Murali [13], in terms of the present context, is that in an optimal coCluster there must be a relatively small subset of rows, of size $\mathcal{O}(\log n)$, that determines which columns participate in the cluster.

Definition 4. Let (I, J) be a coCluster of sleeve-width w , and let $s \in J$. $D \subseteq I$ is a *discriminating set* for (I, J) with respect to s if it satisfies

1. $sw(D, \{s, j\}) \leq w$ for all $j \in J$;
2. $sw(D, \{s, j\}) > w$ for all $j \notin J$.

It is therefore a simple matter to determine J , once s and D are known. The next lemma shows that discriminating sets, with respect to any $s \in J^*$, are both small and plentiful in an optimal coCluster (I^*, J^*) . Consequently in our Algorithm RSLEEVE, both s and D are guessed at, and then J is deduced in line 7.

LEMMA 2.2. *Let (I^*, J^*) be an optimal coCluster of sleeve-width w , with $\beta|I^*| \geq \rho m$, and let s be a column in J^* . Let D be a randomly chosen subset of I^* of size d . Then with probability at least $\frac{1}{2}$, D is a discriminating set for J^* with respect to s , provided $d \geq \log(2n)/\log(1/3\beta)$.*

PROOF. Let $R_i^*, i \in I^*$ be a column profile, and $C_j^*, j \in J^*$ a row profile for (I^*, J^*) . Condition (1) of the definition 4 of a discriminating set is certainly met, since $\{s, j\} \subseteq J^*$, and so $sw(D, \{s, j\}) \leq sw(D, J^*) \leq sw(I^*, J^*) \leq w$.

Therefore D fails to be a discriminating set for J^* with respect to s , only if there is a $j \notin J^*$ such that $sw(D, \{s, j\}) \leq w$. The proof will be completed by showing next that the probability of this happening for a particular j is at most $(3\beta)^d$, so that the probability of it happening for *some* j is bounded by $n(3\beta)^d \leq \frac{1}{2}$.

By definition, $sw(D, \{s, j\}) \leq w$ means that there are C_j, C_s , and $R_i, i \in D$ such that

$$|A_{i,j} - R_i - C_j| \leq \frac{1}{2}w, \quad |A_{i,s} - R_i - C_s| \leq \frac{1}{2}w, \quad \text{for all } i \in D.$$

Hence $|A_{i,j} - A_{i,s} - C| \leq w$ for all $i \in D$, and some $C (= C_j - C_s)$. We want to show that there are no more than $3\beta|I^*|$ rows i that satisfy this inequality, because the coCluster is optimal.

Observe first of all that if $|A_{i,j} - A_{i,s} - C| \leq w$ then

$$(A_{i,s} - R_i^* - C_s^*) - w \leq A_{i,j} - R_i^* - C - C_s^* \leq (A_{i,s} - R_i^* - C_s^*) + w.$$

Since $|A_{i,s} - R_i^* - C_s^*| \leq w/2$ for $i \in I^*$, it follows that $-\frac{3}{2}w \leq A_{i,j} - R_i^* - C - C_s^* \leq \frac{3}{2}w$.

Now the optimality of (I^*, J^*) implies that if there is a subset of rows, $I \subseteq I^*$, and a $j \notin J^*$ such that $|A_{i,j} - R_i^* - c| \leq w$ for some c and all $i \in I$, then $|I| \leq \beta|I^*|$; for otherwise (I, J) , with $J = J^* \cup \{j\}$, is a coCluster of sleeve-width w satisfying $\mu(I, J) > \mu(I^*, J^*)$, contradicting the optimality of (I^*, J^*) .

Therefore, for each $j \notin J^*$ and each of the intervals

$$[-\frac{3}{2}w, -\frac{1}{2}w], \quad [-\frac{1}{2}w, \frac{1}{2}w], \quad [\frac{1}{2}w, \frac{3}{2}w],$$

there are at most $\beta|I^*|$ rows i such that $A_{i,j} - R_i^* - c - C_s^*$ lies in that interval. Thus there are at most $3\beta|I^*|$ rows that satisfy $|A_{i,j} - A_{i,s} - C| \leq w$. □

Definition 5. Let (I^*, J^*) be an optimal coCluster of sleeve-width w . An integer d will be called *acceptable* for (I^*, J^*) if it has the following property. If a subset $D \subseteq I^*$ of size d as well as a column $s \in J^*$ are chosen at random, then D is a discriminating set for (I^*, J^*) with respect to s , with probability $1/2$ at least.

According to the lemma any $d \geq \log(2n)/\log(1/3\beta)$ is acceptable. Our experiments on synthetic data, reported on in section 4, showed this bound to be wildly pessimistic: a random subset of size 5 of the rows of a coCluster, was found to be a discriminating set with probability 1.

THEOREM 2.3. *Assume that A contains an optimal coCluster, (I^*, J^*) , of sleeve-width $w/2$, and let d be acceptable for (I^*, J^*) . Then, with probability at least $1/2$, algorithm RSLEEVE returns a coCluster of sleeve-width w , (I, J) , such that $I \supseteq I^*$ and $J = J^*$, provided $\ell_R \geq (2/\rho)^d \ln 4$, and $\ell_S \geq 2/\sigma$.*

PROOF. The probability that a particular choice of D in the outer loop satisfies $D \subseteq I^*$ is at least ρ^d , since $|D| = d$, and $|I^*| \geq \rho m$. By assumption, given that $D \subseteq I^*$ it is with probability at least $\frac{1}{2}$ a discriminating set for J^* with respect to s . Hence the probability that all ℓ_R iterations of the outer loop fail to find a discriminating set for J^* does not exceed $(1 - \frac{1}{2}\rho^d)^{\ell_R} \leq 1/4$.

Similarly, since $|J^*| \geq \sigma m$, a particular choice of s in the inner loop over columns satisfies $s \in J^*$ with probability at least σ . Therefore the probability that the inner loop fails to find an $s \in J^*$ in all its ℓ_S iterations is at most $(1 - \sigma)^{\ell_S} < 1/4$.

It follows that RSLEEVE chances upon a $s \in J^*$ and a discriminating set $D \subseteq I^*$ with probability at least $3/4 \cdot 3/4 > 1/2$. When it does, it finds $J = J^*$ in lines. The resulting I it then computes, necessarily satisfies $I \supseteq I^*$. Indeed, for any $i \in I^*$ and all $j \in J^*$

$$sw(\{r, i\}, J^*) \leq sw(I^*, J^*) \leq \frac{1}{2}w.$$

Thus $i \in I$. \square

2.3 Running time

The inner for-loops take $\mathcal{O}(mn)$ time. The total number of iterations is upper bounded by Theorem 2.3 as $\ell_S \ell_R = (2/\rho)^d 2/\sigma$. The experiments reported on in subsection 3.3 show that d can be taken as 3, and that the number of iterations of the algorithm is always much less than $16/(\rho^3 \sigma)$.

We note that because of its inherent parallelism the algorithm can easily benefit from special-purpose hardware.

2.4 Extensions

1. Instead of setting w a priori, RSLEEVE could pick a different w for each choice of D . For example, the **for**-loop of line 6 can be replaced by one that first computes $sw(D, \{s, j\})$, and then sets w for the current D according to some density considerations.
2. In line 11 of the algorithm, candidate rows could be tested against all of the discriminating set, rather than just against r . Preliminary testing of this idea indicates that it speeds up the finding of coClusters by some 25% on average.

3. EXPERIMENTS ON SYNTHETIC DATA

We report here on the results of simulations using synthetically generated data, performed on an Intel Xeon CPU 2.40GHz dual processor with 512 KB cache size, running Linux operating system. The purpose of the simulations was to evaluate various aspects of the RSLEEVE algorithm, that can only be assessed with synthetic data. Specific questions we wanted to address were the following.

1. How should the sleeve-width be chosen so that the significant coClusters are found, without introducing artifacts ?
2. What is the size of the discriminating set, d , needed in practice? The bound $d \geq \log(2n)/\log(1/3\beta)$, is of necessity a rough one, and it involves the parameter β , which a user may be loath to specify.
3. How many iterations are needed to find all significant clusters? Theorem 2.3 provides a functional relationship which involves the unknown size of the discriminating set. We wanted also to assess what effect the presence of multiple, possibly overlapping, coClusters has on the running time.

In each simulation run a random matrix of $m = 20,000$ rows and $n = 100$ columns was first generated, by setting the (i, j) -th entry of the matrix to a random integer number in the range $[0, 1000]$.

3.1 Choice of sleeve-width

In general, the sleeve-width approach tends to produce tight coClusters. In our experience, setting w to 5% of the range of values of the matrix provides a good balance between including most of the significant coClusters without introducing spurious ones due to noise. For example, in their analysis of the cell-cycle yeast data, Chen and Church [6] adopted a value of 300 for their parameter δ . This parameter measures the average squared residual of a coCluster; therefore a coCluster of sleeve-width w has a value of $\delta \leq w^2/4$. Their $\delta = 300$ corresponds then to our $w = 30$, which is close to 5% of the yeast data range, $[0, 600]$.

Table 1: Distribution of sleeve-widths. low, high and peak values are in percentages of the range. tail refers to a sleeve-width of less than 5%.

rows	cols	low %	high %	peak %	tail
2	4	0.1	97	32.2	2.04%
5	2	0.1	98	38.5	2.52%
5	4	1.7	99	58.2	0.09%
10	4	2.8	99	71.2	0.01%
15	4	3.1	99	76.5	0.01%
10	8	15	100	81.5	0.00%
15	8	31	100	85.5	0.00%

In order to test our intuition, we selected one million submatrices at random, for each combination of coCluster dimensions, and computed for each the sleeve-width. From the accumulated results we report, in Table 1, the low, high and peak sleeve-widths found in terms of percentage of the range. The larger the submatrix the closer to 100% of the range its sleeve-width is expected to be. The last column states the percentage of cases that had a sleeve-width of 5%

or less. It shows that there is an insignificant probability of finding a coCluster with such a sleeve-width in all cases of interest.

3.2 Size of discriminating set

The determination of a discriminating set is a central part of algorithm RSLEEVE. It appears from Lemma 2.2 as if the size of this set is dependent on the parameter β measuring the trade-off of importance between rows and columns. Not only is specifying β not something we want to burden a user with, but also the bound provided by Lemma 2.2 is a very coarse one. Moreover, the notion of a discriminating set would seem not to need a tie-in to this trade-off. To test these ideas we performed the following experiments. After fixing the number of columns at 5, 40, 60 or 80 ($\sigma = 0.05, 0.4, 0.6, 0.8$ respectively), a random coCluster with the given number of columns was generated and planted in the random matrix. Then a subset of size d of the rows of the coCluster was chosen at random 100,000 times, for $d \in \{2, 3, 4, 5, 6, 7\}$.

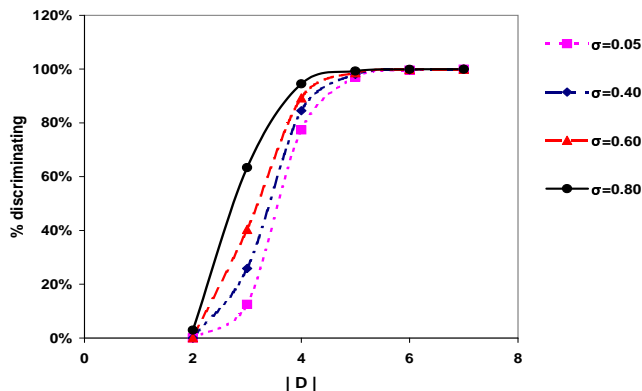


Figure 2: Percentage of discriminating sets among random subsets of a coCluster vs. size of subset, for coClusters with $\sigma = 0.05, 0.4, 0.6, 0.8$.

Figure 2 presents a plot of the percentage of cases in which the chosen subset actually was a discriminating set, as a function of d . The important finding from this experiment is that already a random subset of size 4 is a discriminating set with probability greater than 0.7. Since d appears as an exponent in the estimated running time, we chose $d = 3$ in the following experiments. The plot shows also that coClusters with fewer columns require larger discriminating sets. The reason is that the discriminating set has to filter out more columns not belonging to the coCluster.

3.3 Number of iterations of the algorithm

To test the number of inner iterations of the algorithm, estimated in subsection 2.3 as $((2/\rho)^d 2/\sigma)$, we generated random data in a manner analogous to the one discussed by Procopiuc, Jones, Agarwal and Murali [13]. All data generated had values in the range $[0, 1000]$ and were either cluster points or noise. Each data matrix had $m = 20,000$ rows and $n = 100$ columns, and contained $K = 5$ clusters.

After initializing the $m \times n$ matrix with random data, the coClusters were generated by the following steps.

1. To determine the number of rows m_k of the coCluster k , we first drew random constants of proportion-

ality r_k in the range $[1, 6]$, and computed $m'_k = m \cdot r_k / \sum_{k=1}^K r_i$. Following [13], the imbalance between cluster sizes was increased by setting $m_k = \delta m'_k$, and $m_{k+K/2} = m'_k + (1 - \delta)m'_k$, with δ a parameter. We report here only on results with $\delta = 0.5$, because the differences with the other values we tried, $\delta \in \{0.2, 0.33, 0.5\}$, were not significant.

2. The number of columns of a cluster was set as a random integer in the range $[35, 45]$. In addition, when generating cluster $k + 1$, half of its columns were chosen from among the columns of coCluster k , in attempt to model the sharing of columns between different coClusters.
3. The values of the entries of coCluster k were generated as follows (supplanting the previously generated random values of the matrix). The values of $R_i^{(k)}$, associated with the rows of coCluster k , were generated uniformly at random in $[0, 500]$. The values of $C_j^{(k)}$, associated with its columns, were also generated uniformly at random in $[0, 500]$ (if not already set previously). The value of entry (i, j) of coCluster k was then set to $R_i^{(k)} + C_j^{(k)}$. Finally noise was added, as a uniform random integer in the range $[-25, 25]$, corresponding to a sleeve-width of 5%.

The outer loops were run, with $d = 3$, until all coClusters had been identified. Denote by N the number of iterations needed to retrieve a specific coCluster.

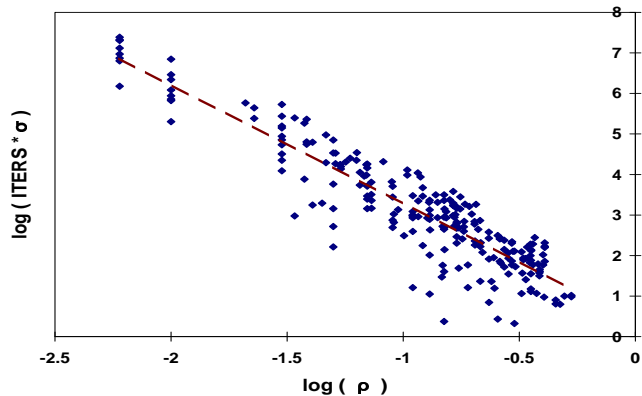


Figure 3: Dependence of the number of iterations of RSLEEVE needed to locate a coCluster on the percentage of rows in a coCluster

Figure 3 is a plot of $\log_{10}(\sigma N)$ as a function of $\log_{10} \rho$. The minimum least squares fit to the data is $\sigma N \approx 4/(\rho^{2.9})$, and always $\sigma N \leq 16/(\rho^{2.9})$.

4. COMPUTING THE SLEEVE-WIDTH

In this section all of the rows and columns of A are taken into account when computing the sleeve-width. By definition,

$$sw(A) = \min_{R,C} \{\|A - B\|_M : B_{i,j} = R_i + C_j, \forall i, j\}, \quad (2)$$

with $\|A\|_M = \max_{i,j} |A_{i,j}|$ the matrix max norm. A similar notion was used in [6], [7], [11], and [17]. However, they

employed the Frobenius norm,

$$\|A\|_F^2 = \sum_{i=1}^m \sum_{j=1}^n A_{i,j}^2.$$

In that case it is easily seen that for the best R^* , C^* ,

$$R_i^* + C_j^* = \frac{1}{n} \sum_{j=1}^n A_{i,j} + \frac{1}{m} \sum_{i=1}^m A_{i,j} - \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n A_{i,j}.$$

For the matrix max norm employed here, there is usually no explicit form for the best R^* , C^* , except in the special case considered in subsection 4.1.

Subsection 4.2 presents an algorithm for computing the sleeve-width of a general matrix; it is a discrete version of the Diliberto-Straus algorithm [9].

4.1 A matrix with two columns

THEOREM 4.1. *Let A be a $m \times 2$ matrix. Then $sw(A) = \frac{1}{2}(\max_i(A_{i,1} - A_{i,2}) - \min_i(A_{i,1} - A_{i,2}))$.*

An explicit solution can also be computed, as follows. First permute the rows of A so that

$$A_{1,1} - A_{1,2} \leq A_{2,1} - A_{2,2} \leq \dots \leq A_{m,1} - A_{m,2}.$$

Set $\varepsilon = sw(A)$, $h = (A_{1,1} - A_{1,2} + A_{m,1} - A_{m,2})/2$, and let ℓ be such that $A_{\ell,1} - A_{\ell,2} \leq h \leq A_{\ell+1,1} - A_{\ell+1,2}$. Then $C^T = \langle 0, -h \rangle$, and

$$R^T = \langle A_{1,1} + \varepsilon, A_{2,1} + \varepsilon, \dots, A_{\ell,1} + \varepsilon, A_{\ell+1,1} - \varepsilon, \dots, A_{m,1} - \varepsilon \rangle.$$

4.2 Iterative algorithm for general matrix

Define the row-midpoint and column-midpoint operators, $P_R(A)$, $P_C(A)$, as follows.

$$P_R(A)_i = \frac{1}{2}(\max_{\ell} A_{i,\ell} + \min_{\ell} A_{i,\ell}),$$

$$P_C(A)_j = \frac{1}{2}(\max_{\ell} A_{\ell,j} + \min_{\ell} A_{\ell,j}).$$

Starting with initial $R^{(0)}$, $C^{(0)}$, and $E_{i,j}^{(0)} = A_{i,j} - R_i^{(0)} - C_j^{(0)}$, the algorithm computes for $k = 1, 2, \dots$

$$\Delta R^{(k+1)} = P_R(E^{(2k)}), \Delta C^{(k+1)} = P_C(E^{(2k+1)}),$$

$$E_{i,j}^{(2k+1)} = E_{i,j}^{(2k)} - \Delta R_i^{(k+1)},$$

$$E_{i,j}^{(2k+2)} = E_{i,j}^{(2k+1)} - \Delta C_j^{(k+1)}, \quad 1 \leq i \leq m, \quad 1 \leq j \leq n.$$

If desired the algorithm also maintains

$$R^{(k+1)} = R^{(k)} + \Delta R^{(k+1)}, \quad C^{(k+1)} = C^{(k)} + \Delta C^{(k+1)}.$$

THEOREM 4.2. *The DS-algorithm converges for any initial $B^{(0)}$. Moreover, the sequence $\|E^{(k)}\|_M$ decreases monotonically to $\frac{1}{2}sw(A)$, and if for some k*

$$\|E^{(k)}\|_M = \|E^{(k+1)}\|_M = \|E^{(k+2)}\|_M,$$

then $2\|E^{(k)}\|_M = sw(A)$ and $R^{(k)}$, $C^{(k)}$ are optimal column and row profiles.

5. ACKNOWLEDGMENTS

We wish to express our appreciation of a perceptive and helpful reviewer. Avraham Melkman acknowledges with pleasure enlightening conversations with T.M. Murali.

6. REFERENCES

- [1] C. C. Aggarwal, J. L. Wolf, P. S. Yu, C. Procopiuc, and J. S. Park. Fast algorithms for projected clustering. In *Proc. SIGMOD*, pages 61–72, 1999.
- [2] C. C. Aggarwal and P. S. Yu. Finding generalized projected clusters in high dimensional spaces. In *Proc. SIGMOD*, pages 70–81, 2000.
- [3] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *Proc. SIGMOD*, pages 94–105, 1998.
- [4] A. Alizadeh, M. Eisen., and R. D. et al. Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature*, 403:503–511, 2000.
- [5] Y. Chen, Z. Yakhini, A. Ben-Dor, E. Dougherty, J. Trent, and M. Bittner. Analysis of expression patterns: the scope of the problem, the problem of scope. *Disease Markers*, 17:59–65, 2001.
- [6] Y. Cheng and G. Church. Biclustering of expression data. In *Proc. ISMB'00*, pages 93–103, 2000.
- [7] H. Cho, I. S. Dhillon, Y. Guan, and S. Sra. Minimum sum-squared residue co-clustering of gene expression data. In *Proc. SIAM Data Mining Conf.*, pages 114–125, 2004.
- [8] I. S. Dhillon. Coclustering documents and words using bipartite spectral graph partitioning. In *Proc. KDD*, pages 269 – 274, 2001.
- [9] S. P. Diliberto and E. G. Straus. On the approximation of a function of several variables by the sum of a function of fewer variables. *Pacific J. Math.*, 1:195–210, 1951.
- [10] G. Getz, E. Levine, and E. Domany. Coupled two-way clustering analysis of gene microarray data. *Proc. Natl. Acad.Sci. USA*, 97:12079–12084, 2000.
- [11] L. Lazzeroni and A. Owen. Plaid models for gene expression data. *Statistica Sinica*, 12:61–86, 2002.
- [12] T. Murali and S. Kasif. Extracting conserved gene expression motifs from gene expression data. In *Proc. Pac. Symp. Biocomputing*, pages 77–88. 2003.
- [13] C. Procopiuc, M. Jones, P. Agarwal, and T. Murali. A Monte Carlo algorithm for fast projective clustering. In *Proc. SIGMOD*, pages 418 – 427, 2002.
- [14] R. Sharan and R. Shamir. Click: a clustering algorithm with applications to gene expression analysis. In *Proc. ISMB'00*, pages 307–316. 2000.
- [15] P. Tamayo, D. Slonim, and J. M. et al. Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. *Proc. Natl. Acad.Sci. USA*, 96:2907–2912, 1999.
- [16] H. Wang, W. Wang, J. Yang, and P. Yu. Clustering by pattern similarity in large data sets. In *Proc. SIGMOD*, pages 394–405, 2002.
- [17] J. Yang, H. Wang, W. Wang, and P. Yu. Enhanced biclustering on expression data. In *Proc. BIBE*, pages 321–327, 2003.