Chapter 1

Models and Issues in Consistent Biclustering

O. Erhun Kundakcioglu, Artyom Nahapetyan, Stanislav Busygin, and Panos M. Pardalos

1.1 Introduction

Biclustering is a methodology allowing simultaneous partitioning of a set of samples and their features into classes. Samples and features classified together are supposed to have a high relevance to each other which can be observed by intensity of their expressions. The notion of consistency for biclustering is defined using interrelation between centroids of sample and feature classes. Consistent biclustering also implies separability of the classes by convex cones (see [Busygin *et al.* (2005)]). Previous works on biclustering concentrated on unsupervised learning and did not consider employing a training set, whose classification is given. However, with the introduction of consistent biclustering, significant progress has been made in supervised learning as well.

A dataset (e.g., from microarray experiments) is normally given as a rectangular $m \times n$ matrix A, where each column represents a data sample (e.g., patient) and each row represents a feature (e.g., gene)

$$A = (a_{ij})_{m \times n}$$

where a_{ij} is the expression of i^{th} feature in j^{th} sample.

Biclustering is applied by simultaneous classification of the samples and features (i.e., columns and rows of matrix A, respectively) into k classes. Let $S_1, S_2, \ldots, S_k$ denote the classes of the samples (columns) and $F_1, F_2, \ldots, F_k$ denote the classes of features (rows). Formally biclustering can be defined as follows.

Definition 1.1. A biclustering is a collection of pairs of sample and feature subsets $\mathcal{B} = \{(S_1, F_1), (S_2, F_2), \ldots, (S_k, F_k)\}$ such that

$$S_1, S_2, \ldots, S_k \subseteq \{a^j\}_{j=1,\ldots,n},$$

$$\bigcup_{r=1}^{k} S_r = \{a^j\}_{j=1,\ldots,n},$$

$$S_\zeta \bigcap S_\xi = \emptyset \Leftrightarrow \zeta \neq \xi,$$

$$F_1, F_2, \ldots, F_k \subseteq \{a_i\}_{i=1,\ldots,m},$$

$$\bigcup_{r=1}^{k} F_r = \{a_i\}_{i=1,\ldots,m},$$

$$F_\zeta \bigcap F_\xi = \emptyset \Leftrightarrow \zeta \neq \xi,$$

where $\{a^j\}_{j=1,\ldots,n}$ and $\{a_i\}_{i=1,\ldots,m}$ denote the set of columns and rows of the matrix A, respectively.

By reordering the columns and rows of the matrix according to their classifications, the result of biclustering can be visualized using the Heatmap Builder Software, where a high value of a_{ij} corresponds to a darker grid (see [Heatmap Builder Software (2003)] for more details). The ultimate goal in a biclustering problem is to find a classification for which samples from the same class have *similar* values for that class' characteristic features. The visualization of a reasonable classification should reveal a block-diagonal or "checkerboard" pattern as seen on Figure 1.1.

One of the early algorithms to obtain an appropriate biclustering is proposed by Hartigan, which is known as *block clustering* (see [Hartigan (1972)]). Given a biclustering $\mathcal{B}$, the variability of the data in the block (S_r, F_r) is used to measure the quality of the classification. A lower variability in the resulting problem is preferable. The number of classes should be fixed in order to avoid a trivial, zero variability solution in which each class consists of only one sample. A more sophisticated approach for biclustering was introduced in [Cheng and Church (2000)], where the objective is to minimize the mean squared residual. They prove that the problem is NP-hard and propose a greedy algorithm to find an approximate solution to the problem. A simulated annealing technique for this problem is discussed in [Bryan (2005)].

Dhillon discusses another biclustering method for text mining using a bipartite graph (see [Dhillon (2001)]). In the graph, the nodes represent

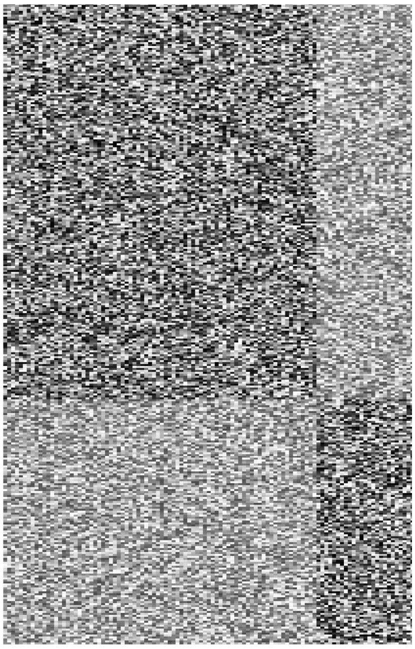

Fig. 1.1 An example of biclustering: “checkerboard” pattern.

features and samples, and each feature i is connected to a sample j with a link (i, j), which has a weight a_{ij}. The total weight of all links connecting features and samples from different classes is used to measure the quality of a biclustering. A lower value corresponds to a better biclustering. A similar method for microarray data is suggested in [Kluger *et al.* (2003)].

The input data is treated as a joint probability distribution between two discrete sets of random variables in [Dhillon *et al.* (2003)]. The goal of the method is to find disjoint classes for both variables. A Bayesian biclustering technique based on the Gibbs sampling can be found in [Sheng *et al.* (2003)].

The concept of *consistent biclustering* is introducted in [Busygin *et al.* (2005)]. Formally speaking, a biclustering $\mathcal{B}$ is consistent if in each sample (feature) from any set S_r (set F_r), the average expression of features (samples) that belong to the same class r is greater than the average expression of features (samples) from other classes. It has been shown that consistent biclustering implies cone separability of samples and features. The model for supervised biclustering involves solution of a special case of fractional 0-1 programming problem whose consistency is achieved by feature selection. Computational results on microarray data mining problems are obtained by reformulating the problem as a linear mixed 0-1 programming problem.

An improved heuristic procedure is proposed in [Nahapetyan *et al.* (2006)], where a linear programming problem with continuous variables is solved at each iteration. Numerical experiments on the same data confirm that the algorithm outperforms the previous result in the quality of solution as well as computation time.

Section 1.2 contains a brief discussion of consistent biclustering. Section 1.3 introduces the application of the technique in the supervised biclustering problem. Complexity results for consistent biclustering are shown in Section 1.4. The heuristic algorithm described in [Nahapetyan *et al.* (2006)] are mentioned in Section 1.5 with numerical results. Finally, Section 1.6 concludes this chapter.

1.2 Consistent Biclustering

Given a classification of the samples, S_r, let $S = (s_{jr})_{n\times k}$ denote a 0-1 matrix where $s_{jr} = 1$ if sample j is classified as a member of the class r (i.e., $a^j \in S_r$), and $s_{jr} = 0$ otherwise. Similarly, given a classification of the features, F_r, let $F = (f_{ir})_{m\times k}$ denote a 0-1 matrix where $f_{ir} = 1$ if feature i belongs to class r (i.e., $a_i \in F_r$), and $f_{ir} = 0$ otherwise. Construct corresponding *centroids* for the samples and features using these matrices as follows

$$C_S = AS(S^T S)^{-1} = (c^S_{i\xi})_{m\times r} \tag{1.1}$$

$$C_F = A^T F(F^T F)^{-1} = (c^F_{j\xi})_{n\times r} \tag{1.2}$$

The elements of the matrices, $c^S_{i\xi}$ and $c^F_{j\xi}$, represent the average expression of the corresponding sample and feature in class ξ, respectively. In particular,

$$c^S_{i\xi} = \frac{\sum_{j=1}^n a_{ij} s_{j\xi}}{\sum_{j=1}^n s_{j\xi}} = \frac{\sum_{j|a^j \in S_\xi} a_{ij}}{|S_\xi|},$$

and

$$c^F_{j\xi} = \frac{\sum_{i=1}^m a_{ij} f_{i\xi}}{\sum_{i=1}^m f_{i\xi}} = \frac{\sum_{i|a_i \in F_\xi} a_{ij}}{|F_\xi|}.$$

Using the elements of matrix C_s, one can assign a feature to a class where it is over-expressed. Therefore feature i is assigned to class $\hat{r}$ if $c^S_{i\hat{r}} = \max_\xi\{c^S_{i\xi}\}$, i.e.,

$$a_i \in \hat{F}_{\hat{r}} \Longrightarrow c^S_{i\hat{r}} > c^S_{i\xi}, \qquad \forall \xi, \xi \neq \hat{r}. \tag{1.3}$$

Note that the constructed classification of the features, $\hat{F}_r$, is not necessarily the same as classification F_r. Similarly, one can use the elements of matrix C_F to classify the samples. Sample j is assigned to class $\hat{r}$ if $c^F_{j\hat{r}} = \max_\xi\{c^F_{j\xi}\}$, i.e.,

$$a^j \in \hat{S}_{\hat{r}} \Longrightarrow c^F_{j\hat{r}} > c^F_{j\xi}, \qquad \forall \xi, \xi \neq \hat{r}. \tag{1.4}$$

As before, obtained classification $\hat{S}_r$ does not necessarily coincide with classification S_r.

Definition 1.2. Biclustering $\mathcal{B}$ is referred to as a *consistent biclustering* if relations (1.3) and (1.4) hold for all elements of the corresponding classes, where matrices C_S and C_F are defined according to (1.1) and (1.2), respectively.

A data set is *biclustering-admitting* if some consistent biclustering for it exists. Furthermore, the data set is called *conditionally biclustering-admitting* with respect to a given (partial) classification of some samples and/or features if there exists a consistent biclustering preserving the given (partial) classification.

Following is the theorem of conic seperability for consistent biclustering.

Theorem 1.1. *Let $\mathcal{B}$ be a consistent biclustering. Then there exist convex cones $P_1, P_2, \ldots, P_k \subseteq \mathbb{R}^m$ such that only samples from S_r belong to the corresponding cone P_r, $r = 1, \ldots, k$. Similarly, there exist convex cones $Q_1, Q_2, \ldots, Q_k \subseteq \mathbb{R}^n$ such that only features from class F_r belong to the corresponding cone Q_r, $r = 1, \ldots, k$.*

Proof. Let $\mathcal{P}_k$ be the conic hull of the samples of class $\mathcal{S}_k$, that is, a vector $x \in \mathcal{P}_k$ if and only if it can be represented as

$$x = \sum_{j \in \mathcal{S}_k} \gamma_j a_{\cdot j},$$

where all $\gamma_j \geq 0$. Note that, $\mathcal{P}_k$ is convex and all samples of class $\mathcal{S}_k$ belong to it. Suppose that, there is a sample $\hat{\jmath} \in \mathcal{S}_\ell$, $\ell \neq k$ that belongs to cone $\mathcal{P}_k$. Then there exists representation

$$a_{\cdot\hat{\jmath}} = \sum_{j \in \mathcal{S}_k} \gamma_j a_{\cdot j},$$

where all $\gamma_j \geq 0$. Next, consistency of the biclustering implies that in the matrix of feature centroids D, the component $d_{\hat{\jmath}\ell} > d_{\hat{\jmath}k}$. This implies

$$\frac{\sum_{i \in \mathcal{F}_\ell} a_{i\hat{\jmath}}}{|\mathcal{F}_\ell|} > \frac{\sum_{i \in \mathcal{F}_k} a_{i\hat{\jmath}}}{|\mathcal{F}_k|}$$

Plugging in $a_{i\hat{\jmath}} = \sum_{j \in \mathcal{S}_k} \gamma_j a_{ij}$,

$$\frac{\sum_{i \in \mathcal{F}_\ell} \sum_{j \in \mathcal{S}_k} \gamma_j a_{ij}}{|\mathcal{F}_\ell|} > \frac{\sum_{i \in \mathcal{F}_k} \sum_{j \in \mathcal{S}_k} \gamma_j a_{ij}}{|\mathcal{F}_k|}$$

Changing the order of summation,

$$\sum_{j \in \mathcal{S}_k} \gamma_j \left(\frac{\sum_{i \in \mathcal{F}_\ell} a_{ij}}{|\mathcal{F}_\ell|} \right) > \sum_{j \in \mathcal{S}_k} \gamma_j \left(\frac{\sum_{i \in \mathcal{F}_k} a_{ij}}{|\mathcal{F}_k|} \right),$$

or

$$\sum_{j \in \mathcal{S}_k} \gamma_j d_{j\ell} > \sum_{j \in \mathcal{S}_k} \gamma_j d_{jk}$$

On the other hand, for any $j \in \mathcal{S}_k$, the biclustering consistency implies $d_{j\ell} < d_{jk}$, which contradicts the obtained inequality. Hence, sample $\hat{\jmath}$ cannot belong to cone $\mathcal{P}_k$.

Similarly, it can be shown that the stated conic separability holds for the classes of features. □

It also follows from the proved conic separability that convex hulls of classes do not intersect.

By definition, a biclustering is consistent if $F_r = \hat{F}_r$ and $S_r = \hat{S}_r$. Theorem 1.1 proves that a consistent biclustering implies separability by cones. However, a given data set might not have these properties. The features and/or samples in the data set might not clearly belong to any of the classes and hence a consistent biclustering might not be constructed. In such cases, one can remove a set of features and/or samples from the data set so that there is a consistent biclustering for the truncated data. Selection of a representative set of features that satisfies certain properties is a widely used technique in data mining applications. This feature selection process may incorporate various objective functions depending on the desirable properties of the selected features, but one general choice is to select the maximal possible number of features in order to lose minimal amount of information provided by the training set.

A problem with selecting the most representative features is the following. Assume that there is a consistent biclustering for a given data set, and there is a feature, i, such that the difference between the two largest values of c^S_{ir} is negligible, i.e.,

$$\min_{\xi \neq \hat{r}} \{c^S_{i\hat{r}} - c^S_{i\xi}\} \leq \alpha,$$

where α is a small positive number. Although this particular feature is classified as a member of class $\hat{r}$ (i.e., $a_i \in F_{\hat{r}}$), the corresponding relation (1.3) can be violated by adding a slightly different sample to the data set. In other words, if α is a relatively small number, then it is not statistically evident that $a_i \in F_{\hat{r}}$, and feature i cannot be used to classify the samples. The significance in choosing the most representative features and samples comes with the difficulty of problems that require feature tests and large amounts of samples that are expensive and time consuming. Some stronger additive and multiplicative consistent biclusterings can replace the weaker consistent biclustering.

In lieu of (1.3) and (1.4) consider the relations

$$a_i \in F_{\hat{r}} \Longrightarrow c^S_{i\hat{r}} > \alpha^S_i + c^S_{i\xi}, \qquad \forall \xi, \xi \neq \hat{r}, \tag{1.5}$$

and

$$a^j \in S_{\hat{r}} \Longrightarrow c^F_{j\hat{r}} > \alpha^F_j + c^F_{j\xi}, \qquad \forall \xi, \xi \neq \hat{r}, \tag{1.6}$$

respectively, where $\alpha^F_j > 0$ and $\alpha^S_i > 0$. Let α denote the vector of α^F_j and α^S_i.

Definition 1.3. A biclustering $\mathcal{B}$ is called an *additive consistent biclustering with parameter* α or α*-consistent biclustering* if relations (1.5) and (1.6) hold for all elements of the corresponding classes, where matrices C_S and C_F are defined according to (1.1) and (1.2), respectively.

Similarly, instead of (1.3) and (1.4) consider the relations

$$a_i \in F_{\hat{r}} \Longrightarrow c^S_{i\hat{r}} > \beta^S_i c^S_{i\xi}, \qquad \forall \xi, \xi \neq \hat{r}, \tag{1.7}$$

and

$$a^j \in S_{\hat{r}} \Longrightarrow c^F_{j\hat{r}} > \beta^F_j c^F_{j\xi}, \qquad \forall \xi, \xi \neq \hat{r}, \tag{1.8}$$

respectively, where $\beta_j^F > 1$ and $\beta_i^S > 1$. Let β denote the vector of β_j^F and β_i^S.

Definition 1.4. A biclustering $\mathcal{B}$ is called a *multiplicative consistent biclustering with parameter* β or β*-consistent biclustering* if relations (1.7) and (1.8) hold for all elements of the corresponding classes, where matrices C_S and C_F are defined according to (1.1) and (1.2), respectively.

An α-consistent biclustering is a consistent biclustering for all values of $c_{i\xi}^S$ and $c_{j\xi}^F$. Also, a β-consistent biclustering is a consistent biclustering if $c_{i\xi}^S \geq 0$ and $c_{j\xi}^F \geq 0$. Note that β-consistent biclustering instances can be found in DNA microarray problems.

The two definitions above can be used to formulate two ways of choosing the most representative subsets of features and samples. In an α-consistent (β-consistent) biclustering problem, the smallest number of features and/or samples is removed from a data set, so that an α-consistent (β-consistent) biclustering exists. Since vectors α and β decrease the number of selected features and/or samples, large values can cause the data set to be restricted. Unfortunately, some important features and/or samples may be left out because of this limitation. In order to overcome this problem, parameters α and β should be chosen based on experimental results.

1.3 Supervised Biclustering

One of the most important problems in real-life data mining applications is supervised classification of test samples. Many real problems already have data sets with known classifications. These data sets are extremely useful in application to the rest of the problem. Supervised classification refers to the capability of a system to learn from these set of examples which is known as the *training set*. The aim in this setup is to classify test samples given the training set and its classification. This is achieved by first processing the training set for feature selection, then classifying the test samples based on these features. In most of the data mining applications, feature selection is crucial since high-dimensionality of data makes complete search computationally infeasible and only a subset of features is expected to be relevant to the classification of interest.

Supervised biclustering uses these accurate data sets to classify features to formulate consistent, α-consistent and β-consistent biclustering problems. Then, the information obtained from these solutions can be used to classify additional samples. This information is also useful for adjusting the values of vectors α and β to produce more characteristic features and decrease the number of misclassifications.

Given a set of training data, construct matrix S and compute the values of $c^S_{i\xi}$ using (1.1). Classify the features according to the following rule: feature i belongs to class $\hat{r}$ (i.e., $a_i \in F_{\hat{r}}$), if $c^S_{i\hat{r}} > c^S_{i\xi}$, $\forall \xi \neq \hat{r}$. Finally, construct matrix F using the obtained classification. Let x_i denote a binary variable, which is one if feature i is included in the computations and zero otherwise. Consistent, α-consistent and β-consistent biclustering problems are formulated as follows.

CB:

$$\max_x \quad \sum_{i=1}^m x_i \tag{1.9a}$$

$$\text{subject to} \quad \frac{\sum_{i=1}^m a_{ij} f_{i\hat{r}} x_i}{\sum_{i=1}^m f_{i\hat{r}} x_i} > \frac{\sum_{i=1}^m a_{ij} f_{i\xi} x_i}{\sum_{i=1}^m f_{i\xi} x_i}, \quad \forall \hat{r}, \xi \in \{1, \ldots, k\}, \hat{r} \neq \xi, j \in S_{\hat{r}} \tag{1.9b}$$

$$x_i \in \{0, 1\}, \quad \forall i \in \{1, \ldots, m\} \tag{1.9c}$$

α-CB:

$$\max_x \quad \sum_{i=1}^m x_i \tag{1.10a}$$

$$\text{subject to} \quad \frac{\sum_{i=1}^m a_{ij} f_{i\hat{r}} x_i}{\sum_{i=1}^m f_{i\hat{r}} x_i} > \alpha_j + \frac{\sum_{i=1}^m a_{ij} f_{i\xi} x_i}{\sum_{i=1}^m f_{i\xi} x_i}, \quad \forall \hat{r}, \xi \in \{1, \ldots, k\}, \hat{r} \neq \xi, j \in S_{\hat{r}} \tag{1.10b}$$

$$x_i \in \{0, 1\}, \quad \forall i \in \{1, \ldots, m\} \tag{1.10c}$$

β-CB:

$$\max_x \quad \sum_{i=1}^m x_i \tag{1.11a}$$

$$\text{subject to} \quad \frac{\sum_{i=1}^m a_{ij} f_{i\hat{r}} x_i}{\sum_{i=1}^m f_{i\hat{r}} x_i} > \beta_j \frac{\sum_{i=1}^m a_{ij} f_{i\xi} x_i}{\sum_{i=1}^m f_{i\xi} x_i}, \quad \forall \hat{r}, \xi \in \{1, \ldots, k\}, \hat{r} \neq \xi, j \in S_{\hat{r}} \tag{1.11b}$$

$$x_i \in \{0, 1\}, \quad \forall i \in \{1, \ldots, m\} \tag{1.11c}$$

The goal in the CB problem is to find the largest set of features that can be used to construct a consistent biclustering. The α-CB and β-CB problems are similar to the original CB problem but the aim is to select features that can be used to construct α-consistent and β-consistent biclusterings, respectively.

1.4 Complexity of Feature Selection

In (1.9), x_i, $i = 1, \dots m$ are the decision variables. $x_i = 1$ if i-th feature is selected, and $x_i = 0$ otherwise. $f_{ik} = 1$ if feature i belongs to class k, and $f_{ik} = 0$ otherwise. The objective is to maximize the number of features selected and (1.9b) ensures that the biclustering is consistent with respect to the selected features.

The optimization problem above is a specific type of *fractional 0-1 programming problem* which is defined as

$$\max \quad \sum_{i=1}^{m} w_i x_i \tag{1.12a}$$

$$\text{subject to} \quad \sum_{j=1}^{n_s} \frac{\alpha_{j0}^s + \sum_{i=1}^m \alpha_{ji}^s x_i}{\beta_{j0}^s + \sum_{i=1}^m \beta_{ji}^s x_i} \geq p_s, \qquad s = 1, \dots, S \tag{1.12b}$$

This problem is NP-hard since linear 0-1 programming is a special class of Problem (1.12) when $\beta_{ji}^s = 0$ and $\beta_{j0}^s = 1$ for $j = 1, \dots, n_s$, $i = 1, \dots m$ and $s = 1 \dots, S$. A typical way to solve a fractional 0-1 programming problem is to reformulate it as a linear mixed 0-1 programming problem, and solve new problem using standard linear programming solvers (see [T.-H.Wu (1997); Tawarmalani *et al.* (2002)]).

In [Busygin *et al.* (2005)], a linearization technique is applied to solve (1.9) due to the NP-hardness of the generalization but whether (1.9) itself is NP-hard or not was an open question.

Theorem 1.2. *Feature selection for consistent biclustering (i.e. formulation (1.9)) is NP-hard.*

Proof. To prove that the problem is NP-hard, a special case of the problem is proved to be NP-hard. In the case considered, there are two classes, and all but one feature belong to the same class. Without loss of generality, assume that m-th feature belongs to one class alone and hence it is selected in the optimal solution unless the problem is infeasible (i.e., $x_m = 1$). Then (1.9b) becomes

$$\frac{\sum_{i=1}^{m-1} a_{i1} x_i}{\sum_{i=1}^{m-1} x_i} > a_{m1} \tag{1.13}$$

$$\frac{\sum_{i=1}^{m-1} a_{i2} x_i}{\sum_{i=1}^{m-1} x_i} < a_{m2} \tag{1.14}$$

It has to be proven that the decision problem is NP-complete in order to prove that the corresponding optimization problem is NP-hard (see [Garey and Johnson (1979)]). The decision version of feature selection for consistent biclustering problem is

D-CB: Is there a set of features that ensures biclustering is consistent, i.e., satisfies (1.13)-(1.14)?

Clearly, D-CB is in NP since the answer can be checked in $O(m)$ time for a given set of features.

Next, the KNAPSACK problem will be reduced to D-CB in polynomial time to complete the proof.

In a knapsack instance, a finite set U_1, a size $s(u) \in Z^+$ and a value $v(u) \in Z^+$ for each $u \in U_1$, a size constraint $B \in Z^+$, and a value goal $K \in Z^+$ are given. The question is

KNAPSACK: Is there a subset $U' \subseteq U_1$ such that $\sum_{u \in U'} s(u) \leq B$ and $\sum_{u \in U'} v(u) \geq K$.

We can modify the knapsack problem as

Π: Is there a subset $U' \subseteq U$ such that

$$\sum_{u \in U'} s(u) \leq 0 \tag{1.15}$$

$$\sum_{u \in U'} v(u) \geq 0 \tag{1.16}$$

$$\tag{1.17}$$

Obviously, Π remains NP-complete, since KNAPSACK can be reduced to its modified variant if we define $U = U_1 \cup t$, $s(t) = -B$, and $v(t) = -K$.

Defining $s'(u) = s(u) + \alpha$, $v'(u) = v(u) + \beta$ for each $u \in U$ and it can easily be seen that

$$\sum_{u \in U'} s(u) \leq 0 \Leftrightarrow \frac{\sum_{u \in U'} s'(u)}{|U'|} \leq \alpha \tag{1.18}$$

$$\sum_{u \in U'} v(u) \geq 0 \Leftrightarrow \frac{\sum_{u \in U'} v'(u)}{|U'|} \geq \beta \tag{1.19}$$

The inequality sign in (1.18)-(1.19) can be changed to strong inequality as follows

$$\frac{\sum_{u\in U'} s'(u)}{|U'|} \le \alpha \Leftrightarrow \frac{\sum_{u\in U'} s'(u)}{|U'|} < \alpha + \epsilon_1 \tag{1.20}$$

$$\frac{\sum_{u\in U'} v'(u)}{|U'|} \ge \beta \Leftrightarrow \frac{\sum_{u\in U'} v'(u)}{|U'|} > \beta - \epsilon_2 \tag{1.21}$$

where $0 < \epsilon_1 < \min_{u,w\in U,s'(u)\neq s'(w)}\{|s'(u) - s'(w)|\}/|U|$ and $0 < \epsilon_2 < \min_{u,w\in U,v'(u)\neq v'(w)}\{|v'(u) - v'(w)|\}/|U|$

As a result the problem is reduced to selecting a subset $U' \subseteq U$ such that

$$\frac{\sum_{u\in U'} s'(u)}{|U'|} < \alpha + \epsilon_1 \tag{1.22}$$

$$\frac{\sum_{u\in U'} v'(u)}{|U'|} > \beta - \epsilon_2 \tag{1.23}$$

$$\tag{1.24}$$

which is in the form of (1.13)-(1.14). The reduction is polynomial and (1.22-1.23) holds true if and only if (1.15-1.16) holds true. Thus D-CB is NP-complete and the proof is complete. □

Corollary 1.1. *Problems (1.10) and (1.11) are NP-hard.*

Proof. Problem (1.9) is a special class of Problem (1.10) when $\alpha_j = 0$ for $j \in S_{\hat{r}}$. Similarly Problem (1.9) is a special class of Problem (1.11) when $\beta_j = 1$ for all $j \in S_{\hat{r}}$. Hence both (1.10) and (1.11) are NP-hard. □

1.5 Application

1.5.1 *Heuristic Algorithm*

Problems (1.9),(1.10), and (1.11) are difficult to solve in the sense that no polynomial time algorithm that finds the optimal solution exists unless $P = NP$. As mentioned earlier, an approach is to reformulate and solve a linearization of the problem. An iterative heuristic procedure has been introduced in [Busygin *et al.* (2005)], which is required to iteratively solve a linear 0-1 problem of smaller size. However, commercial mixed integer programming (MIP) solvers are not able to solve it due to the excessive number of variables and constraints. Another heuristic procedure which iteratively solves *continuous* linear problems is introduced in [Nahapetyan *et al.* (2006)]. The algorithm's versatility and efficiency allows it to be applied to all three problems.

In the problems the expression $\sum_{i=1}^{m} f_{i\xi} x_i$ describes the cardinality of the set of features in the truncated data. In particular, if $x_i = 1$, $\forall i \in \{1, \ldots, m\}$ such that $f_{i\xi} = 1$, then it is equal to the cardinality of F_ξ. Given a vector x, let $F_\xi(x)$ denote the truncated set of features, i.e., $F_\xi(x) \subseteq F_\xi$ such that the features are included in the $F_\xi(x)$ only if $x_i = 1$. If the optimal cardinalities of sets $F_\xi(x)$ are known, they can be fixed at those values, thus the problem turns out to be linear. A series of linear programming problems are iteratively solved, and meanwhile the cardinalities are updated with each available solution.

The first step of Algorithm 1.1 assigns $x_i^0 = 1$, $\forall i \in \{1, \ldots, m\}$, $F_\xi(x^0) = F_\xi$, $\forall \xi \in \{1, \ldots, k\}$, and $p = 0$. In the second step, CB Problem (1.25) is solved, where the cardinalities of the feature sets are fixed at values $|F_\xi(x^p)|$ and integrality of the variables x_i are relaxed.

$$\max_x \quad \sum_{i=1}^{m} x_i \tag{1.25a}$$

$$\text{subject to} \quad \frac{\sum_{i=1}^{m} a_{ij} f_{i\hat{r}} x_i}{|F_{\hat{r}}(x_i^p)|} \geq \frac{\sum_{i=1}^{m} a_{ij} f_{i\xi} x_i}{|F_\xi(x_i^p)|}, \quad \forall \hat{r}, \xi \in \{1, \ldots, k\}, \hat{r} \neq \xi, j \in S_{\hat{r}} \tag{1.25b}$$

$$x_i \in [0, 1], \quad \forall i \in \{1, \ldots, m\} \tag{1.25c}$$

Let $p \leftarrow p+1$ and x^p denote the vector solution of the problem. According to solution x^p, construct sets $F_\xi(x^p)$, where the features are included in the set if $x_i^p = 1$ (i.e., $F_\xi(x^p) \subseteq F_\xi$ such that $x_i^p = 1$). If $\exists \xi \in \{1, \ldots, k\}$ such that $F_\xi(x^p) \neq F_\xi(x^{p-1})$, then go to Step 2 and solve problem (1.25) with updated values of cardinalities. On the other hand, if $F_\xi(x^p) = F_\xi(x^{p-1})$, $\forall \xi \in \{1, \ldots, k\}$, then constraint (1.25b) should be checked for $x_i^* = \lfloor x_i^p \rfloor$. If the constraint is satisfied, the algorithm is stopped, and the value of vector x^* is returned. Otherwise, the variables x_i^p with fractional values cannot take value one and hence those features are permanently removed from the data set.

Observe that solution x^* is feasible to CB problem. In particular, x_i^* takes a value of either one or zero. The sets $F_\xi(x^p)$ include only the features with $x_i^* = 1$. Validity of inequality (1.25b) implies validity of inequality (1.9b). The strict inequality in (1.9b) leads to the following observation.

Algorithm 1.1 Improved heuristic procedure [Nahapetyan *et al.* (2006)]

```
Step 1: Let $x_i^0 = 1$, $\forall i \in \{1, \ldots, m\}$, $F_\xi(x^0) = F_\xi$, $\forall \xi \in \{1, \ldots, k\}$, and $p = 0$.
Step 2: Solve Problem (1.25). Let $p \leftarrow p + 1$ and $x^p$ denote the vector solution of the problem.
Step 3: Construct the set of features $F_\xi(x^p) \subseteq F_\xi$ such that $x_i^p = 1$.
Step 4: If $F_\xi(x^p) \neq F_\xi(x^{p-1})$ then go to Step 2.
Step 5: $x_i^* \leftarrow \lfloor x_i^p \rfloor$. If (1.25b) is satisfied for $x_i^*$ then stop. Otherwise, permanently remove all features with fractional values of $x_i^p$, and proceed with Step 2.
```

The number of features in the truncated data set is maximized with objective (1.25a), and it is beneficial to have the values of variables x_i very close to one. Yet, some variables become fractional at optimality due to constraint (1.25b) and some of the constraints (1.25b) are tight at optimality. If $x^* = \lfloor x^p \rfloor$ satisfies (1.25b), then it is unlikely that the constraints remain tight.

1.5.2 *Computational Results*

The experiments in this section consider a well known data set, which consists of samples from patients diagnosed with *acute lymphoblastic leukemia (ALL)* or *acute myeloid leukemia (AML)* diseases (see [Golub *et al.* (1999)]). This data set is used in [Nahapetyan *et al.* (2006); Busygin *et al.* (2005); Ben-Dor *et al.* (2000, 2001); Weston *et al.* (2000); Xing and Karp (2001)]. The results we present is obtained by Algorithm 1.1.

The data set is divided into two groups, where the first group is used as a training set, and the second one, *test data set*, is used to verify the quality of the obtained classification. The training set consists of 38 samples from which 27 are ALL and 11 are AML samples. The test data set consist of 20 ALL and 14 AML samples. Each sample consists of 7070 features.

	CB	10-CB	20-CB	30-CB	40-CB	50-CB	60-CB	70-CB	130-CB
# features	7024	7021	7018	7014	7010	6959	6989	6960	4639
# errors	2	2	2	2	1	1	1	1	1

The heuristic algorithm is run to solve CB as well as α-CB and β-CB problems with different values for parameters α and β. Although parameters α_j and β_j can take different values for different features, in these

	CB	1.05-CB	1.1-CB	1.2-CB	1.5-CB	2-CB	3-CB	5-CB	7-CB
# features	7024	7017	7010	6937	6508	5905	5458	5173	5055
# errors	2	2	1	1	1	1	1	2	3

experiments it is assumed that they are all equal. In all cases, a "checkerboard" pattern is obtained that is similar to Figure 1.2.

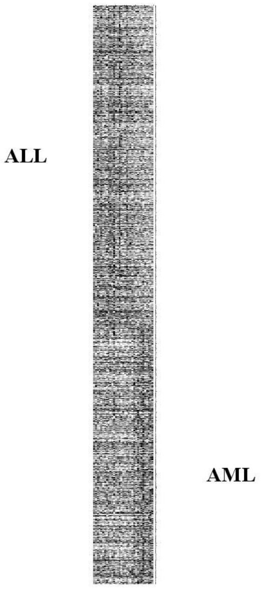

Fig. 1.2 "Checkerboard" pattern for ALL and AML samples.

Table 1.5.2 illustrates the results for the additive consistent biclustering with different values of the vector parameter α. The first row in the table represents the maximum number of features in the truncated data that allow constructing corresponding biclustering. Using the obtained set of features, the samples from the second group of data is classified, and the second row in the table represents the number of misclassifications. It can be noticed that a higher value of the parameter α better classifies the samples. In particular, for α values more than or equal to 40, there is only one error detected in the classification of the test data. In addition, observe that the number of selected features decreases with the increase of the parameter. Based on these observations, we can conclude that as α increases, fewer but more representative features can be used to classify the data. The highest value of the parameter α for which an α-consistent biclustering is obtained is 130. When α is between 40 and 70, each experiment took at most 90 seconds which is reasonable for most practical purposes. On the other hand, it took at most 7 seconds, when α is out of this interval.

Table 1.5.2 reflects the results for β-consistent biclustering. In particular a higher value of β provides a better classification. However, the values of β grater than or equal to 5 turns out to be restrictive and the quality of the classification decreases. The heuristic algorithm proposed in [Busygin *et al.* (2005)] converges after 15 minutes and is able to select 6681 features for $\beta_j = 1.1$, $\forall j \in \{1, \ldots, n\}$. Using the same parameter, Algorithm 1.1 outperforms the previous one by selecting 7010 features within 1.68 seconds of CPU time.

Despite a small number of deleted features[1], the consistent biclustering is crucial to obtain a good classification for the features. If all features are classified using (1.3) and tested using the second group of data, then usually the number of misclassifications is larger. In the case of AML and ALL samples, the number of misclassifications is 19 using this technique. Practically all ALL samples from the test set are classified as AML.

Algorithm 1.1 is also tested on Human Gene Expression (HuGE) Index. The samples are collected from healthy tissues of different parts of human body. The main purpose of the classification is to identify the features that are highly expressed in a particular tissue. Table 1.5.2 illustrates the computational results of the CB and α-CB problems for different values of α. It is interesting to observe that in most of the tissues (e.g., Blood, Brain, and Breast), the number of selected features do not change for different values of α. On the other hand, some tissues (e.g., Ovary) are more sensitive to changes in the parameter. Table 1.5.2 introduces the results for the multiplicative consistent biclustering. Although in these problems the set of sensitive tissues is larger than in the case of α-CB problems, some tissues such as Cervix, Kidney, Placenta, Prostate, Spleen, Stomach preserve the same number of selected features. The last column in the table provides benchmarking data from [Busygin *et al.* (2005)] where a multiplicative consistent biclustering problem with parameter $\beta_j = 1.1$, $\forall j \in \{1, \ldots, n\}$ is

[1] In CB problem, 46 features are deleted.

		CB	α-CB	
Tissue type	# samples		$\alpha = 10$	$\alpha = 70$
Blood	1	472	472	472
Brain	11	615	615	615
Breast	2	903	903	903
Colon	1	367	366	355
Cervix	1	155	155	155
Endometrium	2	226	225	211
Esophagus	1	281	280	272
Kidney	6	159	159	159
Liver	6	440	440	440
Lung	6	102	102	102
Muscle	6	533	533	532
Myometrium	2	162	161	153
Ovary	2	257	255	240
Placenta	2	519	519	519
Prostate	4	281	281	281
Spleen	1	438	438	438
Stomach	1	447	447	447
Testes	1	522	521	515
Vulva	3	187	187	187
Total	59	7066	7059	6996

considered. Observe that for the same value of the parameter, Algorithm 1.1 finds 162 more features.

1.6 Closing Remarks

The concept of consistent biclustering and feature selection for consistent biclustering have been discussed. The aim in this setting is to select a subset of features in the original data set such that the obtained subset of data becomes conditionally biclustering-admitting with respect to the given classification of training samples. The additive and multiplicative variations of the problem are introduced to extend the possibilities of choosing the most representative set of features. The NP-hardness of the original problem and the extensions have been proved. A heuristic algorithm proposed in [Nahapetyan *et al.* (2006)] is presented that allows computing the set of

		CB	β-CB		Busygin et al.
Tissue type	# samples		$\beta = 1.1$	$\beta = 2$	$\beta = 1.1$
Blood	1	472	472	467	472
Brain	11	615	615	610	614
Breast	2	903	903	900	902
Colon	1	367	365	348	367
Cervix	1	155	155	155	107
Endometrium	2	226	224	190	225
Esophagus	1	281	278	259	289
Kidney	6	159	159	159	159
Liver	6	440	440	421	440
Lung	6	102	102	101	102
Muscle	6	533	533	515	532
Myometrium	2	162	160	142	163
Ovary	2	257	253	225	272
Placenta	2	519	519	519	514
Prostate	4	281	281	281	174
Spleen	1	438	438	438	417
Stomach	1	447	447	447	442
Testes	1	522	520	506	512
Vulva	3	187	187	182	186
Total	59	7066	7051	6865	6889

truncated data. Unlike the previously presented algorithm where it is required to solve a sequence of integer programming problems, this approach iteratively solves continuous linear problems. Computational results on the same data set conform that this heuristic algorithm outperforms the previous result in the quality of the solution as well as computational time. Although for most values of α and β the heuristic algorithm is likely to converge to a solution, theoretically these parameters might need to be tuned for some instances.

Bibliography

Ben-Dor, A., Bruhn, L., Friedman, N., Nachman, I., Schummer, M. and Yakhini, Z. (2000). Tissue classification with gene expression profiles, in *RECOMB '00: Proceedings of the fourth annual international conference on Computational molecular biology* (ACM Press, New York, NY, USA), ISBN 1-58113-186-0, pp. 54–64.

Ben-Dor, A., Friedman, N. and Yakhini, Z. (2001). Class discovery in gene expression data, in *RECOMB '01: Proceedings of the fifth annual international conference on Computational biology* (ACM Press, New York, NY, USA), ISBN 1-58113-353-7, pp. 31–38.

Bryan, K. (2005). Biclustering of expression data using simulated annealing, in *CBMS '05: Proceedings of the 18th IEEE Symposium on Computer-Based Medical Systems (CBMS'05)* (IEEE Computer Society, Washington, DC, USA), ISBN 0-7695-2355-2, pp. 383–388.

Busygin, S., Prokopyev, O. A. and Pardalos, P. M. (2005). Feature selection for consistent biclustering, *Journal of Combinatorial Optimization* **10**, pp. 7–21.

Cheng, Y. and Church, G. M. (2000). Biclustering of expression data, in *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology* (AAAI Press), ISBN 1-57735-115-0, pp. 93–103.

Dhillon, I. S. (2001). Co-clustering documents and words using bipartite spectral graph partitioning, in *KDD '01: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining* (ACM Press, New York, NY, USA), ISBN 1-58113-391-X, pp. 269–274.

Dhillon, I. S., Mallela, S. and Modha, D. S. (2003). Information-theoretic co-clustering, in *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining* (ACM Press, New York, NY, USA), ISBN 1-58113-737-0, pp. 89–98.

Garey, M. R. and Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness* (W. H. Freeman & Co., New York, NY, USA), ISBN 0716710447.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., Bloomeld,

C. D. and Lander, E. S. (1999). Molecular classication of cancer: Class discovery and class prodiction by gene expression monitoring, *Science* , 286, pp. 531–537.

Hartigan, J. A. (1972). Direct clustering of a data matrix, *Journal of the American Statistical Association* **67**, 337, pp. 123–129.

Heatmap Builder Software (2003). Quertermous Laboratory, Stanford University, http://quertermous.stanford.edu/heatmap.htm.

Kluger, Y., Basri, R., Chang, J. T. and Gerstein, M. (2003). Spectral biclustering of microarray data: coclustering genes and conditions. *Genome Res* **13**, 4, pp. 703–716.

Nahapetyan, A., Busygin, S. and Pardalos, P. M. (2006). An improved heuristic for consistent biclustering problems, .

Sheng, Q., Moreau, Y. and DeMoor, B. (2003). Biclustering microarray data by gibbs sampling, *Bioinformatics* , 19, pp. 196–205.

T.-H.Wu (1997). A note on a global approach for general 0-1 fractional programming, *European Journal Of Operational Research* **16**, pp. 220–223.

Tawarmalani, M., Ahmed, S. and Sahinidis, N. V. (2002). Global optimization of 0-1 hyperbolic programs, *J. of Global Optimization* **24**, 4, pp. 385–416.

Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T. and Vapnik, V. (2000). Feature selection for svms, in *NIPS*, pp. 668–674.

Xing, E. and Karp, R. (2001). Cliff: Clustering of high-dimensional microarray data via iterative feature filtering using normilized cuts, *Bioinformatics Discovery Note* , 1, pp. 1–9.